ICIW 2007
2nd International Conference on i-Warfare and Security

Naval Postgraduate School, Monterey, California, USA
8-9 March 2007

Edited by

Leigh Armistead
Edith Cowan University

Copyright The Authors, 2007. All Rights Reserved.

No reproduction, copy or transmission may be made without written permission from the individual authors.

Papers have been double-blind peer reviewed before final submission to the conference. Initially, paper abstracts were read and selected by the conference panel for submission as possible papers for the conference.

Many thanks to the reviewers who helped ensure the quality of the full papers.

ISBN: 978-1-905305-41-4

Published by Academic Conferences Limited
Reading
UK
44-118-972-4148
info@academic-conferences.org

ICIW 2007

Contents

Paper Title	*Author(s)*	**Proceedings Page**
Preface		iv
Biographies of Conference Chairs, Programme Chair, Keynote Speaker and Mini-track Chairs		vi
Biographies of contributing authors		viii
The Evolution of Information Operations, Contracts across the DoD: Opportunities for Growth in the Future for Academics	*Edwin Leigh Armistead[1] and Thomas Murphy[2]* *[1]Edith Cowan University, Australia* *[2]Northlight Technologies, USA*	1-8
On Manipulability of Algorithms	*Sviatoslav Braynov* *University of Illinois at Springfield* *IL USA*	9-16
Developing a Framework to Improve Information Assurance Battlespace Knowledge	*Adam Bryant and Michael Grimaila* *Air Force Institute of Technology,* *Fairborn, OH, USA*	17-24
Security Risks in USAF Geospatial Information Sharing	*Scott Bryant and Michael Grimaila* *Air Force Institute of Technology* *Fairborn, OH, USA*	25-32
Performance Impact of Connectivity Restrictions and Increased Vulnerability Presence on Automated Attack Graph Generation	*James Cullum, Cynthia Irvine and Tim Levin* *Naval Postgraduate School,* *Monterey, CA, USA*	33-46
Analyzing Security Measures for Mobile Ad Hoc Networks Using Attack and Protection Trees	*Kenneth Edge, Richard Raines, Rusty Baldwin, Michael Grimaila, Christopher Reuter and Robert Bennington* *Wright-Patterson AFB, Fairborn,Oh, USA*	47-56
A Study of DNS Traffic Patterns on a Large Autonomous System	*Sid Faber[1]and Bert Lundy[2]* *[1]Software Engineering Institute, Pittsburgh, PA, USA* *[2]Naval Postgraduate School, Monterey, CA, USA*	57-68
Development of a Defensive Cyber Damage Assessment Framework	*Larry Fortson and Michael Grimaila* *Air Force Institute of Technology,* *Fairborn, OH, USA*	69-76
Anti-Forensics: Techniques, Detection and Countermeasures	*Simson Garfinkel* *Naval Postgraduate School,* *Monterey, CA, USA*	77-84
IPv6: World Update	*Kenneth Geers and Alexander Eisen* *NCIS Washington, DC, USA*	85-94
Leap in the Dark: An Attempt to Theorize Personnel Anomaly Detection for Countering Insider Threats	*Shuyuan Mary Ho* *Syracuse University, NY, USA*	95-100

Paper Title	*Author(s)*	**Proceedings Page**
Conceptual Design of a Microfluidics Suppressor to Protect against Potentially Lethal Printing Devices: A Scenario-based Physical Cyber Security Measure	*Berg Hyacinthe[1] and Yves Anglade[2]* *[1]Florida State University, Tallahassee, FL, USA* *[2]Florida A&M University, Tallahassee, FL, USA*	101-110
'Virtual' Security Teaching Techniques – Teaching IA methods using VMWare Teams	*Derek Isaacs* *Colorado Technical University, and Boecore Inc, Colorado Springs CO, USA*	111-116
Differential Power Analysis Attacks against AES Circuits Implemented on a FPGA	*Keisuke Iwai, Minoru Sasaki and Takakazu Kurokawa* *National Defense Academy, Japan*	117-122
Categorization of Profiles for PSYOPs: Can Technology Help?	*Magdi Kamel, Mark Eramo, and Christopher Sutter* *Naval Postgraduate School, Monterey, CA, USA*	123-132
Implications of Information Flow Priorities for Interorganizational Crisis Management	*Tuija Kuusisto[1], Rauno Kuusisto[2] and Mark Nissen[3]* *[1]Ministry of Defence, Helsinki, Finland* *[2]National Defence University, Helsinki, Finland* *[3]Naval Postgraduate School, Monterey, CA, USA*	133-140
Thinking Strategically About Information Operations	*Irving Lachow and Robert Miller* *National Defense University, Washington, DC, USA*	141-146
Intrusion Detection in open Source Software via Dynamic Aspects	*William Mahoney and William Sousan* *University of Nebraska at Omaha, NE, USA*	147-154
Fine-Grain Security for Military-Civilian-Coalition Operations Though Virtual Private Workspace Technology	*R William Maule and Shelley Gallup* *Naval Postgraduate School, Monterey, CA, USA*	155-162
Creating Hardware-based Primitives that Facilitate the Exposure of State Information Useful for Security Related Monitoring	*Stephen Mott and Paul Williams* *Air Force Institute of Technology, Fairborn, OH, USA*	163-170
Graphical Based user Authentication with Embedded Mouse Stroke Dynamics	*Ken Revett[1], Sergio Tenreiro de Magalhaes[2] and Henrique Santos[2]* *[1]University of Westminster, London,Uk* *[2]University do Minho, Portugal*	171-176
Planning Cost-Effective Deceptive Resource Denial in Defense to Cyber-Attacks	*Neil Rowe* *U.S. Naval Postgraduate School, CA, USA*	177-184
Experiments with a Testbed for Automated Defensive Deception Planning for Cyber-Attacks	*Neil Rowe, Han Goh, Sze Lim, and Binh Duong* *U.S. Naval Postgraduate School, CA, USA*	185-194
Possible Risks Analysis Engine: A Prototype Tool for Managing IT Security Safeguards Acquisition	*Robert Sainsbury and Richard Baskerville* *Georgia State University, Atlanta, GA, USA*	195-204

Modified RSL as a Countermeasure Against Differential Power Analysis	*Minoru Sasaki, Keisuke Iwai and Takakazu Kurokawa* *National Defense Academy, Japan*	205-216
A Framework for Relating Cyberspace Operations to the Cognitive and Physical Domains	*Tiffiny Smith[1], Pamela Woolley[2], Robert Mills[1], and Richard Raines[1]* *[1]Air Force Institute of Technology, Fairborn, OH, USA* *[2] Fairchild AFB WA, USA*	217-224
Similarity Analysis of Malicious Executables	*Anthonius Sulaiman, Sandeep Mandada, Srinivas Mukkamala and Andrew Sung* *New Mexico Tech, Socorro,NM, USA*	225-232
Using Deception for assuring Security	*Stilianos Vidalis, Eric Llewellyn and Chrostopher Tubbs* *University of Wales, Newport ,UK*	233-240
The Design Space of Metamorphic Malware	*Andrew Walenstein[1], Rachit Mathur[2], Mohamed Chouchane[1], and Arun Lakhotia[1]* *[1]University of Louisiana at Lafayette, LA, U.S.A.* *[2]McAfee Avert Labs, Beaverton, OR, U.S.A.*	241-248
Information Terrorism in the New Security Environment	*Ken Webb* *RNSA and Edith Cowan University, Perth, Western Australia*	249-256
Inter-Network Operations Center Dial-By-ASN (INOC-DBA), a Hotline for Critical Internet Infrastructure Management	*Bill Woodcock[1] and Ross Stapleton-Gray[2]* *[1]Packet Clearing House, Berkeley, CA, USA* *[2]Internet Awareness, Inc., Berkeley, CA, USA*	257-262
SME Security in the Digital age	*Don Milne[1], John McCarthy[2],and Bryan Mills[2]* *[1]Buckinghamshire Chilterns University College, High Wycombe, UK* *[2]Service Tec Global Services Ltd, UK*	263-270

Preface

Welcome to the Second International Conference on Information Warfare and Security (ICIW 2007), hosted this year by the Naval Postgraduate School in Monterey, California. The Conference Chair is Dr Dorothy Denning from the Naval Postgraduate School and the Programme Chair is Leigh Armistead from Edith Cowan University in Perth, Australia.

The main aim of this Conference is for individuals working in the area of Information Warfare and Information Security to come together to share knowledge with peers interested in the same area of study.

The opening keynote address this year is given by Tim Thomas, Foreign Military Studies Office, Army Command and Staff College, Fort Leavenworth, USA on the topic of "*Decoding the Virtual Dragon*". On day two of the conference the keynote speaker is John Arquilla from the Naval Postgraduate School, who will talk on "*Information Warfare Today*".

An important benefit of attending this conference is the ability to share ideas and meet the people who hold them. The range of papers will ensure an interesting two days. The topics covered by the papers this year illustrate the depth of the information operations' research area, with the subject matter ranging from the highly technical to the more strategic visions of the use and influence of information. To further enhance the conference experience Leigh Armistead will lead an interactive session encouraging debate and discussion the current status of research in the I-Warfare and Security area.

With an initial submission of 60 abstracts, after the double blind, peer review process there are 33 papers published in these Conference Proceedings. These papers illustrate the true International spread of the conference, coming from Australia, Finland, Japan, United Kingdom, United States.

I wish you a most enjoyable conference.

Leigh Armistead
Edith Cowan University
Programme Chair

Conference Executive:

Edwin Leigh Armistead, Edith Cowan University, Australia
Sviatoslav Braynov University of Illinois at Springfield, IL, USA
Dorothy Denning, Naval Postgraduate School, Monterey, CA, USA
Steve Iatrou, Naval Postgraduate College, Monterey, CA, USA
Andy Jones, Security Research Centre, BT, UK and Edith Cowan University, Australia
Dan Kuehl, National Defense University, Washington DC, UK,
Tara Leweling, Naval Postgraduate School, Monterey, CA, USA
Corey Schou, Idaho State University, USA
Stylianos Vidalis, University of Glamorgan, UK

Conference Committee:

The conference programme committee consists of key people in the information systems, information warfare and information security communities around the world. The following people have confirmed their participation:

Leigh Armistead (Edith Cowan University, Australia); John Arquila (Naval Postgraduate School, USA); Richard Baskerville (Georgia State University, USA); Elisa Bertino (CERIAS, Purdue University, USA); Matt Bishop (University of California, USA); Alexander Bligh (College of Judea and Samaria, Israel); Sviatoslav Braynov (University of Illinois, USA); Rodney Clare (EDS and the Open University, UK); Geoffrey Darnton, (University of Bournemouth, UK); Dipankar Dasgupta (University of Memphis, USA); Dorothy Denning (Navel Postgraduate School, USA); Glenn Dietrich (University of Texas, USA); Xinwen Fu (Dakota State University, USA); Kevin Gleason (KMG Consulting, MA, USA); Sanjay Goel (University at Albany, USA); Drew Hamilton (Auburn University, USA); Dwight Haworth (University of Nebraska at Omaha, USA); Philip Hippensteel (Penn State University, USA); Bill Hutchinson (Edith Cowan University, Australia); Berg P Hyacinthe (Florida State University, USA); Cynthia Irvine (Naval Postgraduate School, USA); Andy Jones (British Telecome, UK); Dan Kuehl (National Defense Forces, USA); Tuija Kuusisto (National Defence College, Finland); Arun Lakhotia (University of Louisiana Lafayette, USA); Michael Lavine (John Hopkins University, USA); Tara Leweling (Naval Postgraduate School, USA); Bin Lu (West Chester University, USA); Bill Mahoney (University of Nebraska, USA); John McCarthy (Buckinghamshire and Chiltern University College, UK); Anne McGee (Industrial College of the Armed Forces, USA); Don Milne (Buckinghamshire and Chiltern University College, UK); Evangelos Moustakas (Middlesex University, UK); Richard Raines (Airforce Institute of Technology, USA); Ken Revett (University of Westminster, UK); Neil Rowe (US Naval Postgraduate School, USA); Julie Ryan (George Washington University, USA); Corey Schou (Idaho State University, USA); Dan Shoemaker (Univesity of Detroit Mercy, USA); Kevin Streff (Dakota State University, USA), Doug Twitchell (Illinois State University, USA); Stylianos Vidalis (University of Glamorgan, UK); Tom Wilsdon (University of South Australia, Australia); William Yurcik (University of Illinois at Urbana-Champaign, USA); Zehai Zhou (Dakota State University, USA).

Biographies of Conference Chairs, Programme Chair and Keynote Speaker

Conference Chair

Dorothy Denning is a Professor in the Department of Defense Analysis and a member of the Center on Terrorism and Irregular Warfare at the Naval Postgraduate School. Her current research and teaching encompasses the areas of conflict and cyberspace, trust and influence, terrorism and crime, and information operations and warfare. She is author of Information Warfare and Security and over 130 articles, and has testified before the U.S. Congress on encryption policy and cyberterrorism.

Programme Chair

Leigh Armistead is the Strategic IO/IA Development Manager for Honeywell Technology Soluntions Inc. He has written two books, the most recent Information Operations: Warfare and The Hard Reality of Soft Power (Brasseys, August 2003) which serves as a textbook for a number of DoD organizations. Formerly a Master Faculty at the Joint Forces Staff College, Leigh is currently enrolled in a PhD program at Edith Cowan University with an emphasis on Information Operations, where he also serves on the Editorial Review Board for European Conference on Information Warfare, is a Co-Editor for the Journal of International Warfare and the Chairman for the IQPC annual conference on Information Operations.

Keynote Speaker

Dr John Arquilla earned his degrees in international relations from Rosary College (BA 1975) and Stanford University (MA 1989, PhD 1991). He is professor of defense analysis at the Naval Postgraduate School, where he has taught in the special operations curriculum since 1993. He also serves as director of the Information Operations Center. His teaching interests revolve around the history of irregular warfare, terrorism, and the implications of the information age for society and security. His books include: Dubious Battles: Aggression Defeat and the International System (1992); From Troy to Entebbe: Special Operations in Ancient & Modern Times (1996), which was a featured alternate of the Military Book Club; In Athena's Camp (1997); Networks and Netwars: The Future of Terror, Crime and Militancy (2001), named a notable book of the year by the American Library Association; and The Reagan Imprint: Ideas in American Foreign Policy from the Collapse of Communism to the War on Terror (2006). His next book will be about military innovation.

Dr. Arquilla is also the author of more than one hundred articles on a wide range of topics in military and security affairs, with his work appearing in both the leading academic journals and in more general publications like *Wired* and *The New Republic.* He is best known for his concept of "netwar" (i.e., the distinct manner in which those organized into networks fight), a term which former defense secretary Donald Rumsfeld used on several occasions to describe the ongoing conflict in Iraq. His vision of "swarm tactics" was selected by *The New York Times* as one of the "big ideas" of 2001. In terms of policy experience, Dr. Arquilla worked as a consultant to General Norman Schwarzkopf during

Operation Desert Storm, as part of a small team of RAND analysts. During the Kosovo War, he assisted deputy secretary of defense John Hamre on a range of issues in international information strategy. Since the onset of the war on terror, Dr. Arquilla's policy contributions have included a brief period of service on the Information Operations Task Force, followed by more extended involvement with the 1st Marine Expeditionary Force, and other units, on practical, information-related "field problems." Currently, he guides an initiative at the request of the vice chairman of the joint chiefs.

Keynote Speaker

Timothy Thomas is an analyst at the Foreign Military Studies Office (FMSO) at Fort Leavenworth, Kansas. He retired from the U.S. Army as a Lieutenant Colonel in the summer of 1993. Mr. Thomas received a B.S. from West Point and an M.A. from the University of Southern California. He was a U.S. Army Foreign Area Officer who specialized in Soviet/Russian studies. His military assignments included serving as the Director of Soviet Studies at the United States Army Russian Institute (USARI) in Garmisch, Germany; as an inspector of Soviet tactical operations under CSCE; and as a Brigade S-2 and company commander in the 82nd Abn Division. Mr. Thomas has done extensive research and publishing in the areas of peacekeeping, information war, psychological operations, low intensity conflict, and political-military affairs. He is the assistant editor of the journal European Security; an adjunct professor at the U.S. Army's Eurasian Institute; an adjunct lecturer at the USAF Special Operations School; and a member of two Russian organizations, the Academy of International Information, and the Academy of Natural Sciences.

Biographies of contributing authors (in alphabetical order)

Edwin Armistead: The Strategic Information Assurance Development Manager for Honeywell Technology Solutions Inc, Leigh was also the editor of Information Operations: Warfare and The Hard Reality of Soft Power (Brasseys, May 2004). A retired US Naval Officer and Master Faculty (IO) at the Joint Forces Staff College, he is currently enrolled in a PhD program at Edith Cowan University (Perth, AU). Leigh has published a number of articles on IO in addition to chairing numerous professional IO conferences around the world, including the International Conference on Information Warfare (2006) and the IQPC IO Conference in the United Kingdom (2002-2005). Selected five years in a row as a research fellow for the International National Security Studies program to conduct IO related research, he also helped to develop an online IO course for the National Security Agency.

Richard Baskerville's research and authored works regard security of information systems, methods of information systems design and development, and the interaction of information systems and organizations. He is a Chartered Engineer, holds a B.S. summa cum laude, from The University of Maryland, and the M.Sc. and Ph.D. degrees from The London School of Economics, University of London.

Sviatoslav (Svet) Braynov (Dr) is an Assistant Professor at the Computer Science Department at the University of Illinois at Springfield. He has published more than 40 papers in refereed conferences and journals. He was also an invited speaker to several conferences, delivered several tutorials, and has served as a co-chair and a program committee member of more than 15 international conferences and workshops. His research interests include computer security, information warfare, electronic commerce, artificial intelligence, and game theory.

Adam Bryant (Captain) completed a B.S. in Social Psychology from Park University in 2001 and was commissioned in 2002. He was promoted to Captain in May 2006. He is a candidate for dual M.S. degrees from the Air Force Institute of Technology in Information Resource Management and Computer Systems, specializing in IA management and database systems

Scott Bryant (Captain) USAF is currently working on his Masters degree in Engineering Management at the Air Force Institute of Technology, Wright-Patterson AFB, OH. He served as the initial GeoBase Integration Officer at Elmendorf AFB, AK and as the War Fighting Headquarters' Chief of Resource, Environment, and GeoBase; 7th Air Force, Osan, AB, Republic of Korea, from 2003-2005.

Mohamed Chouchane is working towards his Ph.D. in Computer Science at the Center for Advanced Computer Studies in University of Louisiana at Lafayette. In his PhD dissertation he is studying the computational, complexity theoretic and practical aspects of the recognition of the output of metamorphic engines. He is a co-author of the paper "Normalizing Metamorphic Malware Using Term Rewriting" that was awarded the Best Paper Award of the IEEE Workshop on Source Code Analysis and Manipulation, 2006

James Cullum holds a Masters of Computer Science, specializing in Information Assurance, from the Naval Postgraduate School. In his thesis work he examined automated attack graph generation software, with Dr. Cynthia E. Irvine. While an intern at the FAA he studied vulnerability analysis.

Kenneth Edge is a Major in the U.S. Air Force. He is currently a PhD candidate at the Air Force Institute of Technology in Dayton, Ohio. His research interests include software protection and algorithms. He received his B.S. degree from the U.S. Air Force Academy in Colorado Springs, Colorado and his M.S. from Wright State University in Dayton, Ohio.

Sid Faber is an analyst at CERT, at the Software Institute, Carnegie Mellon University. He is an experienced network security analyst, and has a BS in Electrical Engineering from Penn State and MS in Information Science from the University of Pittsburgh.

Larry Fortson (Captain) USAF served as a network security operations commander at the AFNOC NSD and was deployed in Baghdad, Iraq. He is currently working on his Master's degree at the Air Force Institute of Technology with a concentration in Information Assurance Management. Capt Fortson's research focuses on operational damage assessment to enable mission capability impact following cyber-information incidents.

Simson Garfinkel is an Associate Professor at the Naval Postgraduate School. Dr. Garfinkel has research interests in computer forensics, the emerging field of usability and security, and personal information management. Garfinkel is the author or co-author of fourteen books on computing.

Kenneth Geers, the Naval Criminal Investigative Service (NCIS) Cyber Division Chief, has worked for many years as a translator, programmer, Web developer, and analyst. He also waited tables in Luxembourg, harvested grapes in the Middle East, climbed Mount Kilimanjaro, was bitten by a deadly spider in Zanzibar and made Trappist beer at 3 AM in the Rochefort monastery. The oddest job he had was working on the John F. Kennedy Assassination Review Board (don't ask). Mr. Geers is the author of Cyber Jihad and the Globalization of Warfare; hacking in a Foreign Language: A Network Security Guide to Russia; Sex, Lies, and Cyberspace: Behind Saudi Arabia's National Firewall; and Greetz from Room 101.

Berg Hyacinthe is completing a PhD in Information at Florida State University. His research interests encompass: Social Informatics, Information Warfare, and Emerging Technologies. His current research on cyber-assisted olfaction technologies will encapsulate (a) users' adoption of these technologies and (b) potential applications of these technologies to global defence and security.

Derek Isaacs-CISSP has 24+ years of experience in the Computer Security / Information Assurance area. He has a strong technical knowledge of a variety of platforms, security administration and protocols. Derek is an IA Engineer for Boecore Inc at the Joint National Integration Center (JNIC). Derek provides Information Assurance support for DITSCAP/DIACAP on a variety of Programs. Tasks include developing detailed System Security Authorization Agreements (SSAA), Trusted Facilities Manuals (TFM), Security Features Users Guides (SFUG), Finite State Machine Documents (FSM) and Certification Test & Evaluation (CT&E) Plans and Procedures. He also performs security assessment and testing (DISA Gold and Platinum disk) as well as other vulnerability tools Retina, ISS, (Nessus/Nmap).

Keisuke Iwai received the B.E. degree in electrical engineering from Waseda University, Tokyo, Japan in 1996 and the M.E. degree in computer science from Keio University, Tokyo, Japan in 1998 respectively. He also received Dr. Eng. degree in Science for open and environmental systems from Keio University, Tokyo, Japan. Presently he is a research associate at the National Defense Academy of Japan. His research interests are

optimizing compilers, multiprocessor architectures, reconfigurable processors and differential power analysis against cryptographic circuits.

Shuyuan Mary Ho researches design, criteria and mechanisms in "Personnel Anomaly Detection (PAD) for countering Insider Threat" and conceptualizes a Security Architecture for management executives, at the School of Information Studies, Syracuse University. Shuyuan is a CISSP by (ISC)2, a CISM by ISACA, and has been actively involved in technologies such as VPN, CA, RBAC, FW, IDS, vulnerability assessment, and policy management

Magdi Kamel (Dr) is an Associate Professor of Information Systems in the Division of Computer and Information Sciences and Operations at the Naval Postgraduate School. His research interests are in the areas of database management and knowledge base systems. He is particularly interested in database models and languages, data modeling, data and text mining, and knowledge discovery in large databases.

Tuija Kuusisto works as a Chief Information Officer for Ministry of Defence in Finland and as an adjunct professor for National Defence College and Helsinki University of Technology in Finland. She has over 20 years experience in the information and knowledge management field and especially in the geographic information management area. She has published over 50 articles in international and national journals, conference proceedings and books.

Irving Lachow is a Senior Research Professor at the National Defense University's Information Resources Management College. He has also worked for Booz Allen Hamilton, the RAND Corporation, and the Department of Defense. Dr. Lachow received his Ph.D. in Engineering & Public Policy from Carnegie Mellon University. He earned a B.A. Political Science and a B.S. in Physics from Stanford University.

Bert Lundy is a faculty member at the Computer Science Dept, Naval Postgraduate School. He has done extensive work in network protocols. He has a PhD from Georgia Tech in computer science.

William Mahoney received his B.A. and B.S. degrees from Southern Illinois University, and his M.A. and Ph.D. degrees from the University of Nebraska. He is a Research Fellow and Graduate Faculty at the University of Nebraska at Omaha Peter Kiewit Institute. His primary research interests include compilers, hardware and instruction set design, and VLSI. Prior to the Kiewit Institute Dr. Mahoney worked for 20+ years in the computer design industry, specifically in the areas of embedded computing and real-time operating systems. During this time he was also on the part time faculty of the University of Nebraska at Omaha. His outside interests include bicycling, photography, and more bicycling.

R. William Maule is Research Associate Professor of Information Science at the Naval Postgraduate School in Monterey, CA

Robert Mills is an Assistant Professor of Electrical Engineering at the Air Force Institute of Technology (AFIT), Wright-Patterson Air Force Base, Ohio. He received his Ph.D. in electrical engineering from the University of Kansas in 1994, his MSEE from AFIT in 1987, and BSEE from Montana State University in 1983. His research interests include communication systems, network management and security, information warfare, and systems engineering.

Stephen Mott graduated the United States Air Force Academy in Colorado Springs, CO as a member of the class of 2005 with a Bachelors of Science in Electrical Engineering. He is currently working towards his Masters of Science in Electrical Engineering at the Air Force Institue of Technology at Wright-Patterson Air Force Base, OH.

Ken Revett is a researcher in biometrics with an emphasis on enhancing C2 level security systems. Although his focus has been on keystroke dyanmics, he is investigating incorporating physiological biometrics within the graphical user authentication. This paper presents some of our preliminary resaerch into this exciting new topic.

Neil Rowe is Professor of Computer Science at the Naval Postgraduate School where he has been since 1983. His primary interest is information security, and he has also done research on data mining, robotics, computer vision, and intelligent tutoring systems. He is the author of 120 research papers and a book.

Minoru Sasaki graduated from National Defense Academy. He joined the Japan Maritime Self Defense Force in 2001. His rank is lieutenant junior grade. He is enrolled in National Defense Academy as a senior graduate student.Moreover, he is interested in reconfigurable architecture as well as computer security.

Srinivas Mukkamala is a senior research scientist with ICASA (Institute for Complex Additive Systems Analysis, a statutory research division of New Mexico Tech performing work on information technology, information assurance, and analysis and protection of critical infrastructures as complex interdependent systems) and Adjunct Faculty of the Computer Science Department of New Mexico Tech. He is a frequent speaker on information assurance in conferences and tutorials. He leads a team of information assurance (IA) “first responders” who are deployed at the request of various government agencies and financial institutions around the state of New Mexico to perform vulnerability analysis, information system security audits, network analysis and forensic incident analysis. He has a patent pending on Intelligent Agents for Distributed Intrusion Detection System and Method of Practicing the Same.

Stilianos Vidalis (Dr) received his PhD in Threat Assessment in July 2004 from the University of Glamorgan. He is currently a lecturer in the University of Wales, Newport, at the department of Computing and Engineering, where he is lecturing in the subjects of information security and computer networks in both the undergraduate and postgraduate level. His research interests are in the areas of information security, Information operations, threat assessment, network security, effective computer defence mechanisms and intrusion detection systems.

Andrew Walenstein is a Research Scientist at the University of Louisiana at Lafayette. He received his Ph.D. in 2002 from Simon Fraser University. His research interests include reverse engineering and malware analysis, program comprehension, visualization and human-computer interaction, and software engineering.

Ken Webb. After graduating from the Royal Military College, Ken mainly served as a qualified commissioned officer with the SAS and other special operations units, such as commanding strategic counter-terrorist, intelligence-gathering and unconventional warfare elements, where he also focused on special operations in the information warfare area. Upon leaving the military he worked globally in the information operations field and is about to complete a doctoral level research project for the Government into enhancing national security from the information operations of terrorist groups. Ken is also the

counter-terrorism research leader for the RNSA, which is a Government initiative aimed at identifying and fostering academic, industry and government research into safeguarding Australia.

Bill Woodcock is research director of Packet Clearing House, a non-profit research institute dedicated to understanding and supporting Internet traffic exchange technology, policy, and economics. Bill has operated national and international Internet service provision and content delivery networks since 1989, and currently spends most of his time building Internet exchanges in developing countries.

The Evolution of Information Operations Contracts across the DoD: Growth Opportunities for Academic Research

Edwin Leigh Armistead[1] and Thomas Murphy[2]
[1]Edith Cowan University, Australia
[2]NorthLight Technologies, USA
earmiste@student.ecu.edu.au
tmurphy@rochester.rr.com

Abstract: The development of Information Operations (IO) as a major military area of concentration in the United States is a relatively new phenomenon, with the first unclassified doctrine only being released within the last ten years (Joint Publication 3-13, *Joint Doctrine for Information Operations,* 9 October 1998). IO is a superset of other military operations such as Psychological Operations (PSYOP), Electronic Warfare (EW), Operations Security (OPSEC) and Information Assurance (IA). Hence IO is often viewed as an overarching integrating strategy that is still maturing, when compared to the older legacy warfare communities mentioned above. This view has in the author's opinion led to an overall lack of support from a fiscal, personnel and policy perspective. The diffuse nature of IO may also lead to confusion or lack of awareness of research opportunities by academics, especially with regard to aligning oneself to Department of Defense (DoD) contracts. Normally a rich source of funding or grants, at first glance it appears that the DoD contract opportunities in the IO and Information Warfare (IW) areas, are more scattered than and not as numerous as the more specific skill sets such as IA and Information Technology (IT). For example, One can easily cite very large DoD and Federal IA contracts such as DISA I-Assure, ANSWER, Millenia, etc., that are all worth billions of dollars, which are available to the DoD to support IT and IA issues. But the same cannot often be readily be said of IO and IW, where the appearance is the contract vehicles that are available tend to be fewer and much smaller. The questions for this paper are then does this dichotomy really exist? Is so, why? If not, why is this assumption so readily accepted? Finally, where are the main DoD contracts in the IO and IW arena that are available to access for academic researchers? Are the majority of these opportunities based on the older legacy areas of EW, PSYOP, IA, etc or on the more broader issue area of IO and IW?. This paper will attempt to answer these questions as it addresses the evolution and development of IO and IW contracts across the DoD, and their potential for academic researchers. It will show why the differential has developed with other areas such as IA, and what are the current opportunities are that exist today. This paper will also show what the future may hold for further growth in this area and how the growth of IO, IW and IA contract vehicles can benefit universities and academics. Finally, this paper will also suggest future areas of research that academics may be interested in exploring, to best optimize their ability to secure grants and contracts over the next few years.

Keywords: Information operations, department of defense, contracts, information warfare, proposals

1. Introduction

The Information Age is an era of manipulated images, both visual and auditory. Nations, groups and individuals are all attempting to manage the message that you see, and they will often conduct their information campaigns in a similar manner – whether they are 'selling' a soft drink or a global terrorist threat. The whole idea is to influence the 'wetware' of the consumer or the public, to get them to believe in a product or cause. The process is to supply information that molds the recipient's knowledge and expectations which then results in behaviour meeting the goals of the information sender. Howard Gardner in *Changing Minds, The Art and Science of Changing Our Own and Other People's Minds* (2005), states his surprise at the high level of pre-publication interest in his book from political, security, corporate, academic and marketing organizations. And yet no one should really be surprised, especially in this world where the first Under-Secretary of State for Public Diplomacy and Public Affairs in the Bush Administration was a former advertising executive. Or the number one private strategic communications firm, conducting official sanctioned information campaigns for the United States, is run by a former campaign manager for Jimmy Carter (Silverstein, 2001). It is all the same in this world, where image is everything, that how a nation, group or even person is perceived is far more important than the actual reality.

While there are many reasons for this, some analysts have also attributed this revolution to the fact that we now live in what we call the Information Age. People often ask what is different about this era with respect to military weapons or other forms of power, and the answer of course is the distribution and management of information. The incredible transformation and flattening of information has spread this element of power across humanity. For whether you are a nation-state, a paramilitary group, a non-governmental organization, a terrorist organization, a cartel, or even as an individual, it is now possible to coordinate, organize and conduct a successful information campaign. And probably the most important concept to remember about information and its management is that it is useful not only as a weapon, but is also a process that provides a way of thinking about relationships. Information is all about invoking perceptions,

and because it is an enabler, or a 'force multiplier', it can be a tool that increases one's ability to shape their operational, financial or cultural environment.

Thus by integrating all of the factors and disciplines involved in the items above, the development of IO has arisen as a central tenet for the United States, both at the federal and cabinet level, in particular for the DoD. This is because IO can support the strategies, campaigns, and processes of traditional military, government as well as business organizations and their growth. And yet while IO presently tends to focus on management and operational issues of combat activities, it is also being modelled to address any large scale, complex organization and its mission. This is because the general requirements for success are often very similar for DoD or for business, instantly giving IO a broader application beyond typical military combat operations.

2. IO and IW as a business growth area

If IO is such a revolutionary area that is going to transform the military, corporate America and academia, then one would think that the business growth opportunities in this issue area would be growing rapidly as well. Yet when one searches the DoD business databases for current and future IO or IW contracts, the results are mixed. The number of IO, IW and IA contracts is growing slowly, while on the other hand, the number of contracts for IT is substantially higher. This of course may be explained that every single organization needs IT support, while these other issue areas are more refined, but the difference in number of opportunities is statistically very significant. While the recent rise in IO, IW and IA contract numbers is encouraging, there is still an apparent dichotomy between these umbrella terms and their more specific warfare components. These facts summarize into the conclusion that while the IO and IW areas are growing, there are still far fewer new opportunities than the more discreet contracts that deal with IT. As shown below, the authors theorize that there are a number of reasons, all of which will be evaluated in this paper:

- IO and IW are still in the early stage of exponential growth.
- IO or IW is more of a operational or combat focus that doesn't apply widely across large sections of the DoD
- Everyone has to conduct IT. These are common tasks to all organizations
- IO and IW are still not well-understood by many within the DoD.
- IO and IW are only thought of as Computer Network Attack (CNA)
- The legacy warfare areas such as EW and PSYOP are well established and do not want to be rolled up into a broader more nebulous environment

3. Market drivers for growth in DoD contracts

There are many reasons for growth (and decline) in DoD contracts. Specifically with regard to IO, IW and IA, the Global War on Terrorism (GWOT) drives many opportunities as do government mandates such as the Federal Information Security Management Act (FISMA) and the connectivity requirements of the Global Information Grid (GiG). All of these factors can be cited as influencers on this overall process, yet interestingly enough, each of these drivers also tends to exclude these areas, specifically IO and IW from growth, because they can be viewed in narrow functional or parochial areas. For example, the GWOT tends to focus on the security issues associated with Iraq and Afghanistan, where EW, PSYOP and OPSEC play a pre-dominate role. Likewise, FISMA requires a very rigorous certification and assessment of all government systems, with more extensive documentation and reporting requirements, and IA is the major issue area that is affected. In addition, because the GIG is such a fundamental shift in management of information, it is an area where IA tends to play a more predominate role, because of the mandates and minimum standards for connectivity to a DoD network. Beyond these three examples, there are many more issues of course, but what these factors demonstrate is that as an umbrella term, IO and IW are not the predominate focus of effort on DoD contracts. Instead components of IO and IW such as IA, EW, PSYOP and others are presently the predominate focus of effort on DoD contracts, instead of the more integrated and managed sets of capabilities inherent in IO and IW. This can be seen in the next few pages as a short analysis of IO, IW and IA contracts was conducted to compare the different opportunities available to academic researchers.

3.1 Analysis of IO, IW and IA contracts

As part of this research, the authors conducted a series of searches on a common DoD business database known as INPUT (http://www.input.com). This tool is useful in that it stores all opportunities – past, present and future in archival form and one can search in both a functional (using multiple keywords) manner as well as an organizational (across the federal government) one. In total, for this paper 49 contracts were reviewed from the IO/IW arena and another 35 from IA (See Appendix A: IO and IW Related Contracts, and Appendix B: Information Assurance Related Contracts). There were also an additional 677 contracts that were listed in the search under IT, but because the authors were most interested in tailoring their research to the broader IO umbrella, of which IT is not commonly considered a part of, these latter vehicles were not included in this study, and only the 84 from the IO, IW and IA arena's were ultimately addressed.

In reviewing the past, current and projected IO, IW and IA contracts, one notices that they tend to come in waves or groups and that there are quite a large and diverge number of different companies that have won new business with the DoD. Specifically for IO and IW vehicles, the researchers searched on the business database using key words such as IO, IW, EW, PSYOP and Strategic Communications. After eliminating the duplicate entries, they found 49 contracts dating back over the last decade, of which 27 had been awarded for a value of $1.735B. Another eight were in the post Request for Proposal (RFP) phase, which they about to be awarded, with 10 more that had been cancelled and a final eight that are forthcoming soon (pre-RFP). The awardees were spread over 29 different companies, since some contracts had split awards, and in this group only four corporations had won more than one vehicle (Sy Coleman – 2; General Dynamics (GD) – 2; Booz, Allen Hamilton (BAH) – 2; Northrop Grumman (NGC)– 3). Based on this preliminary data, there appeared to be no clear defined frontrunner, although the fact that eight contracts are outstanding and about to be awarded in the very near future, may change that perception very quickly.

This is interesting data as compared to the IA contracts of which there have been 35 in the last eight years. Of these 18 have been awarded, six have been cancelled, with two that are about to be awarded (post-RFP) and nine more that are forthcoming (pre-RFP). Like the IO and IW series, there were a series of recompetes and standard vehicles as well as the one offs that also occurred. From these 18 awarded vehicles, there were 76 companies or universities that received funds for a total of $3.823B. Of these corporations only two companies won more than one contract (SAIC – 2; BAH – 3), with a wide diversity of and breadth of options for all of the other companies. As with IO and IA, there is no clear and defined frontrunner, although BAH is doing well, as are small businesses (SBs), universities and accounting firms.

From this small sample of contracts, the author's believe that this data offers the academic researcher several important options. The first is that since there is no clearly defined frontrunner, one could ally themselves with a number of different companies to build a nascient if not significant IO, IW or IA capability. For example, an academic could approach on of the five corporations mentioned above and offer to provide them with a capability to remain a leading entity in this field. Or one could approach one of the other 100+ companies and offer to help them through research to catch up to the leading contenders, in essence to make them a frontrunner. Finally in perhaps a more radical suggestion, an academic or several of their colleagues could start their own company to go after IO, IW and IA contracts. Since there appears to be very low thresholds to enter this market and the field is diverse, this option is very viable as well. Here are some additional ideas to further flesh out the opportunities to conduct academic research in conjunction with DoD contracts.

3.1.1 Allying oneself with the leading contenders

This strategy offers the lowest risk to an academic. Between BAH, GD, NGC, SAIC and Sy Coleman are 14 contracts totaling over $3B in ceiling that one could exploit as an academic. All of these are well-known companies that are always looking for ways to enhance their value to the customer and retain their frontrunner status, so worthy and new areas of research in the IO, IW and IA arenas will certainly draw interest. A problem of course is that they may already have a full research staff that has already investigated your particular concept and they also may not be willing to bring on new teammates that eat away at their profits. But nonetheless, if an academic can ally themselves to one of these corporations, they will likely do well in receiving and having access to grants and funds in this research area.

3.1.2 Developing a frontrunner

The next option is to pick one of the other corporations or universities who have a stake in the competition, i.e. an incumbent who has at least a share of one contract, who is looking to rise above the crowd and one

who wants to distinguish themselves as a frontrunner. There are a lot of companies to choose from (over a 100) and they key is to find a progressive and forward looking group that is willing to stretch their teams to include academics and non – traditional units to work with on DoD contracts. In fact an academic is not limited of course in their search and they can also ally themselves with more than one group to increase their chances of success, as well as collaborating with companies that have not won a contract yet as well. The bottom line to success is to present new and leading edge research that will enable the corporation to present evidence of technology development to the DoD.

3.1.3 Striking out on your own

This is the riskiest and potentially the most rewarding of the options to grow research opportunities in the IO, IW and IA arenas for an academic. As one can see from the database searches in Appendices A and B, there are over a 100 companies and universities that have won at least something from the DoD with regards to these contract areas. Thus the ability for a new company to gain entry and strive for a portion of these contracts is not exceedingly difficult nor impossible and can be done through the correct balance of research, strategic hires, long-term alliances and strategy that focuses on the forthcoming critical needs and mandates. The key of course is that DoD contracting is normally considered a near full time effort, that may not be an option for an academic associated with a university. Other options in this same vein include collaborating with business groups on startup companies or developing new technologies or concepts, which are quickly sold to DoD contractors searching for new areas to exploit.

3.2 Future areas of research

Whatever method an academic chooses to enter the fray of government contracting, they are going to need to add value, real value to join a team. While there are many areas of IO, IW and IA that could be researched, the authors have suggested a few below, that we have seen a need or requirement for from the DoD. Each of these suggested research areas are academic topics that could facilitate a better awareness of IO and make it more understandable to the uniformed services, which may also help secure corporations in the future funding on one of these contract vehicles.

3.2.1 Addressing the dichotomy between PM and CNO

To begin, IO and IW are still difficult to understand. One still needs to understand the factors that tend to affect the funding of IO, IW and IA are mainly centered around their umbrella structure. Some critics have complained that because these terms are so generic, they lose any specific relevance to the war-fighter. Yet in reality, it is exactly due to their vagueness, that these terms have become more popular within the vocabulary of federal officials, and that is where these researchers believe that their potential for the greatest growth may lie. For more often than not, in today's political environment, the definitions of the more discrete warfare areas are too restrictive for the nature of modern combat, where business and operational missions are intermixed and where horizontal integration is key. This horizontal integration merges the previously distinct combat specialities.

In the IO and IW arena, both Perception Management (PM) and Computer Network Operations (CNO) are relatively broad terms that are fairly new in their use throughout the United States government. Perception Management is defined by the DoD as:

> *"Actions to convey and/or deny selected information and indicators to foreign audiences to influence their emotions, motives and objective reasoning; and to intelligence systems and leaders at all levels to influence official estimates, ultimately resulting in foreign behaviour and official actions favorable to the originators objectives"* (Joint Publication 1-02, 1998, p. 340).

PM is comprised of a number of sub-elements including Public Affairs, Public Diplomacy, Psychological Operations, Deception and Covert Action. In reality, it is the ability to shape an image or conduct an influence campaign that is most important. Perception Management has forever been transformed through the use of computers, telecommunications, video, the Internet, e-mail and other technological advances. Instances that are often mentioned were the use of a VHS camera by the Somalia warlord Aideed, the denial of service attacks by the Electronic Disturbance Theatre, and perhaps most influential, the timing of the second explosion at the World Trade Center. All of these events were perception management campaigns designed to manipulate public opinion (Armistead, 2003; Livingston, 1997; Munro, 1999; Nueman, 1996). In each case, the tools used were all different, but the goal was always the same – to produce an effect, a perception in the mind of the target. In fact as some authors suggest it is now the media that often directs foreign policy, not the government.

Likewise CNO is normally seen as a combination of two disparate warfare areas, Computer Network Attack (CNA) and Computer Network Defense (CND). The former is defined by the DoD, as: "Operations to disrupt, deny, degrade or destroy information resident in computers and computer networks, or the computers and networks themselves" (Joint Publication 3-13, 1998, p. GL-5). This is compared to CND which is defined as: "Defensive measures to protect and defend information, computers and networks from disruption, denial, degradation or destruction" (Joint Publication 1-02, 2002, accessed at http://www.dtic.mil/doctrine/jel/doddict/data/c/01169.html). As mentioned earlier, CNO is a relatively new term, which has come into use not because it is restrictive in nature, but instead because it encompasses the offensive and defensive-related computer warfare areas. CNO is very important to the development of an American information security strategy, because so much of the success of any change in policy, education and organization of the United States government, will revolve around the use of information technology. Thus research into the amalgamation of these disparate elements of IO and or perhaps the development of new terms that bring together the different portions of this warfare area, may be a good research topic. There is a definite need to help the DoD and Federal government truly understand the power of these concepts and what they can do in today's Information Age environment.

3.2.2 Applicability of IO and IW to the USG

IO and IW enhance the capability for the United States government to conduct information campaigns around the world. Officials in the United States government have attempted to enable the horizontal integration of activities across the whole interagency and coalition environment. They seek to do this by managing information and using planning tools to synchronize, synergise, and de-conflict information-based activities in an overall plan to affect the adversary. The United States became a world leader in the industrial age because it could mobilize the collective might of its population through mass production, automation, economic incentives and geographic location. Yet that era is over, and the information revolution is now upon us. Like other great empires that have withered and died, the United States could be facing the same fate unless it adjusts to these new realities. In this information age, where perception is everything, the ability to manipulate power, often rests on the outward appearance of a nation to the world. The management of a nation's actions, whether they are overt or not, is not yet an exact science. Neither can the activities be easily measured or calculated. Instead an information strategy is more like an art, one that balances communication skills with knowledge, information, and timing and above all, a political skill that must be honed over years of practice. A deeper understanding and modelling of the combined political, military and information (IO-IW) environments is of immense value in exploring, predicting and optimizing activities of the US Government, the DoD and those who support or attack them. As mentioned earlier, all of these information security strategies of the nation are extremely important, and could mean the difference between success and failure in the conduct of a desired foreign policy initiative. Understanding how to organize, manage and implement the plans and goals effectively and efficiently is a worthy research goal. Therefore any advance or enhancement to IO and IW which improves or enhances a desirable asymmetry while balancing the ability to defend the infrastructure is highly desirable. So a number of important research questions need to be addressed and funded.

- How can we enable the federal bureaucracy to conduct an effective information campaign, while assuring the security of their networks and information systems in this new environment? Likewise,
- How should the United States change its collective interagency structure that has evolved over 200 years, into a networked organization that can master the issues in the information age? Finally,
- How should (or how can) the United States mature into a dominant player during this new era, where industrial capacity is not nearly as important to a nation as its interconnectivity of information nodes?

3.2.3 CNSS as an avenue for growth

In 1986 the DoD security community began to recognize the need for improved education and training, which included academic participation in information assurance. To accomplish this end, they established training standards and a defined set of knowledge skills and attributes (KSA). This early work at the National IA Training and Education Center (NIATEC) to develop a taxonomy, led to several industry professional standards, National Institute of Standards (NIST) publication 800-16 and the Committee for National Security Systems (CNSS) series of publications. These standards, developed by the National Security Agency (NSA), are now widely recognised throughout the DoD and interagency as the de facto baseline of tasks for IA across the federal bureaucracy. In addition, the CNSS series has become widely used in academia, through NSA sponsored IA programs and curriculum. The question of course that arises, is that can the success of

CNSS be translated from the IA arena to IA and IW? Should a similar type organization be stood up? If so, under what government agency?

3.2.4 Developing IO standards

One of the problems with IO and a potential research area that academics may want to pursue in search of grants or funding is the establishment and development of standards for IO and IW. The community needs to create a model of IO analogous to the model of IA used in NSA recognized IA standards. An IO model must account for the scope of IO as an offensive and defensive **set** of information operations, of which IA is only one basic component. Most importantly, the model must include actions and views of both (all) sides of a conflict, the CINC and the adversary, partners etc. The fully symmetry model of IO is captured by the "new adage" – with apologies to Sun Tsu: "Know your enemy and yourself well" AND "influence your enemy to NOT know you or himself well". "Knowing" includes knowing current facts and also expected outcomes based on past and current information and knowledge. Since using language similar to the currently supported IA contracts can be helpful, there is a possible model of IO analogous to IA. This form should be closer to what Dr. Corey Schou derived using a standard process on IA standards work, such as eDACUM, which is an electronic method of involvement and consensus building to determine training needs as identified by skilled workers and professionals (Maconachy, V et. al. 2002). It is a joint NSA and Idaho State University (ISU) venture for producing Training Standards. Starting with the essential IO capabilities and related core competencies desired for IO Planners in the organization, KSAs are needed for competency are developed for IO tasks. The resulting Job/task Analysis is further refined using a taxonomy of specially selected verbs (Blooms taxonomy) related to Knowledge, Performance and attitudes/feelings. This is used as a foundation to writing the IO standards document.

4. Summary

In conclusion, there is a tremendous amount of opportunity available to academics who are interested in conducting research in the IO, IW and IA realms. Billions of dollars are being awarded to defense contractors who need qualified personnel to conduct these tasks, who need proven methodologies to validate their processes, as well as new technologies and skill sets to support the DoD. Universities and academics are continually seen as valuable teammates on many of these contracts, and small businesses that are associated with colleges are also welcome too, especially on these contracts that have high small business goals. The key to success of course, is to develop a reputation in the IO, IW and IW field that the defense contractors can then market as part of their campaign to win more business.The authors hope that this discussion has raised awareness of some opportunities that may be available to researchers in their goal to secure funding and grants. There are opportunities in the IO, IW and IA fields for academics to excel and the authors hope that by selective tailoring of their research, and an alliance with the right companies and corporations, that they can succeed in their quest.

References

Armistead, Edwin ed. (2003) Information Operations: Warfare and the Hard Reality of Soft Power. Dulles, VA: Brassey's, Inc.

Gardner, H (2006) Changing Minds: The Art and Science of Changing Our Own and Other People's Minds, Harvard Business School Press, Boston, pp ix-xv.

INPUT database, "The authority on government business" [online]. INPUT, <http://www.input.com>

Livingston, Steven (1997) Clarifying the CNN Effect: An Examination of Media Effects

According to Type of Military Intervention. R-18, The Joan Shorensten Center on the Press, Politics and Public Policy, John F. Kennedy School of Government, Harvard University.

Maconachy V, Schou C, Frost J. (2002) Introduction to Electronic Develop a Curriculum X, NCISSE Course 610, Idaho State University, April, 2002

Munro, Neil (1999) "Infowar: AK-47s, Lies and Videotape." Association for Computing

Machinery, Communications of the ACM. New York.

Neuman, Johanna (1996) Lights, Camera, War: Is Media Technology Driving International

Politics? New York: St. Martin's Press.

Silverstein, Ken (2001) "Selling the Afgan War," Accessed at http://www.TheNation.com.

U.S. Department of Defense. Joint Chiefs of Staff. Department of Defense Dictionary

of Military and Associated Terms (10 June 1998), Joint Publication 1-02.

U.S. Department of Defense. Joint Chiefs of Staff. Department of Defense Dictionary

of Military and Associated Terms (14 August 2002), Joint Publication 1-02, Accessed at http://www.dtic.mil/doctrine/jel/doddict/data/c/01169.html.

U.S. Department of Defense. Joint Chiefs of Staff. Joint Doctrine for Information Operations (9 October 1998), Joint Publication 3-13.

Appendix A - IO and IW related contracts

INPUT	CONTRACT NAME	STATUS	CONTRACTOR	DATE	AMOUNT
654	IW/IO MGMT TECHNICAL ENGINEERING AND ADMINISTRATIVE SUPPORT	Cancelled		3-Feb-97	
1497	IO MANAGEMENT SUPPORT AND TECHNICAL ASSISTANCE SERVICES	Awarded	BAH, LMC	15-Sep-97	$25M split
1498	IO SYSTEMS ENGINEERING INTEGRATION AND OPERATIONAL SUPPORT	Awarded	EWA, GD, NGC, CA	19-Sep-97	$75M split
1960	INFORMATION OPERATIONS PLANNING SYSTEM (IOPS)	Awarded	GD	4-Sep-98	$115M
2008	IW DEFEND (IW-D) ENGINEERING AND PROGRAM MANAGEMENT SUPPORT	Awarded	BAH	26-May-98	$76.7M
2364	DEFENSIVE INFORMATION WARFARE TECHNOLOGY APPLICATIONS	Awarded	NGC	16-Aug-99	$20M
2709	DEFENSE IO ENTERPRISE MANAGEMENT SYSTEM (DIOEMS)	Cancelled		22-Aug-00	
2764	INFORMATION OPERATIONS (IO)	Cancelled		15-Nov-99	
3401	IO MANAGEMENT SUPPORT AND TECHNICAL ASSISTANCE SERVICES	Cancelled		13-Nov-01	
3403	FULL SPRECTRUM IO SUPPORT TO THE LIWA	Awarded	TASC	24-Apr-02	$314M
3555	SHIPALT ENGINEERING INSTALLATION/DRAFTING SRVS ISO CRYPTOLOGIC IW	Awarded	MILCOM	7-Oct-05	$9.9M
3771	CSCS OMNIBUS (CSCSO)	Awarded	Lincoln Group	26-Sep-06	$6.2M
4829	DATABASES FOR THE 21ST CENTURY II: INFORMATION, INTELLIGENCE, AND IW	Awarded	Litton, PRC	7-Dec-01	$48.9M
5422	ENGINEERING SERVICES FOR INTELLIGENCE IW AND MISSION PLANNING SYS	Awarded	Sverdrup	19-May-06	$4.4M
6705	IW TECHNICAL DEVELOPMENT AND DEMONSTRATION SERVICES (IWTDD)	Cancelled		29-Jan-03	
7823	IW BATTLELAB OPERATIONAL CONCEPT DEMONSTRATIONS	Awarded	Five SBs	27-Sep-04	$1.725M split
7921	SIMULATION AND TRAINING DEVELOPMENT TO ENHANCE TACTICAL KNOWLEDGE AND READINESS OF INFORMATION WARFARE TEAMS	Awarded	Two SBs	2-Oct-03	$700K
8848	ARMY URBAN INFORMATION OPERATIONS CAPABILITIES	Awarded	Harris	4-Feb-04	????
11070	AIR FORCE INFORMATION WARFARE CENTER SOLUTION ANALYSIS	Awarded			
11413	INFORMATION WARFARE EFFECTIVENESS PROGRAM (IWE)	Awarded	SRA	27-Jan-04	$44.7M
12444	INFORMATION WARFARE TECHNOLOGY APPLICATION	Awarded	NGC	14-Dec-05	$24.8M
15419	AIR FORCE INFORMATION WARFARE BATTLELAB	Awarded			
16052	FULL SPECTRUM IO SUPPORT TO THE LIWA	Pre-RFP			
16127	ENHANCE MEDIA CAPABILITIES OF THE JOINT PSYOP SUPPORT ELEMENT	Awarded	SAIC, SyColeman, Lincoln Group	31-Jan-06	$750M
16174	COMMUNICATIONS AND IO AND MANAGEMENT SERVICES	Pre-RFP			
16596	STRATEGIC COMMUNICATIONS PLANNING SUPPORT	Awarded	Sy Coleman	8-Aug-05	$71.5M
16809	FOREIGN MEDIA ANALYSIS STRATEGIC IO SUPPORT	Awarded	SOS International	5-Jul-06	$68M
17034	COMMUNICATION & INFORMATION WARFARE PROGRAM SUPPORT	Awarded			
17108	ARMY SPECIAL TECHNICAL OPERATIONS/SPECIAL IO TRAINING			12-Jul-05	
17122	LOGISTICS STRATEGIC COMMUNICATIONS SUPPORT SERVICES	Cancelled		20-Oct-06	
31751	PROVISION OF A PSYCHOLOGICAL OPERATIONS RADIO NETWORK IN AFGHANISTAN (PSYOPS)	Awarded			
31983	FULL SPECTRUM IO SUPPORT TO THE LIWA	Cancelled		20-Jul-05	
31896	SPECIAL TECHNICAL OPERATIONS AND SPECIAL IO TRAINING (STO, SIO)	Awarded	SRC	12-Oct-05	????
33293	AIR FORCE IW BATTLELAB OPERATIONAL CONCEPT DEMONSTRATIONS (IO IW)	Post-RFP			
34051	INFORMATION WARFARE PROGRAM SUPPORT SERVICES	Awarded	Technosoft	7-Jul-06	$14M
34666	CONTRACTOR SUPPORT SERVICES FOR DARPA OFFICE OF MANAGEMENT OPERATIONS SECURITY AND INTELLIGENCE DIRECTORATE	Awarded	Analex	19-Jul-06	$65M
34383	STRATEGIC COMMUNICATIONS TECHNICAL SUPPORT	Cancelled			
35102	INFORMATION WARFARE EFFECTIVENESS PROGRAM (IWE)	Pre-RFP			
35383	INFORMATION WARFARE OFFENSIVE AND DEFENSIVE COUNTER INFORMATION	Post-RFP			
36462	AIR FORCE IO BATTLELAB OPERATIONAL CONCEPT DEMONSTRATIONS	Post-RFP			
36577	SYSTEMS TECHNOLOGY AND RESEARCH INFORMATION OPERATIONS SUPPORT	Pre-RFP			
36820	TECHNICAL AND PROGRAMMATIC SERVICES IN SUPPORT OF IO AND IA	Post-RFP			
37190	INFORMATION WARFARE PLANNING CAPABILITY SUSTAINMENT	Pre-RFP			
38002	FOREIGN MEDIA ANALYSIS STRATEGIC INFORMATION OPERATIONS SUPPORT	Pre-RFP			
38041	INFORMATION OPERATIONS AND STRATEGIC COMMUNICATIONS	Post-RFP			
38332	TECHNICAL AND PROGRAMMATIC SERVICES IN SUPPORT OF IO AND IA	Cancelled			
38333	TECHNICAL AND PROGRAMMATIC SERVICES IN SUPPORT OF IO AND IA - SB	Post-RFP			
38417	AIR FORCE IO BATTLELAB OPERATIONAL CONCEPT DEMONSTRATIONS	Post-RFP			

Appendix B - Information assurance related contracts

INPUT	CONTRACT NAME	STATUS	CONTRACTOR	DATE	AMOUNT	CHGS
657	INFORMATION ASSURANCE SERVICES	Awarded	LMC, Artel, EDS, CSC, NGC, SAIC, SRA,	27-Jul-00	$1.5B	
2163	TECHNICAL SECURITY AND INFORMATION ASSURANCE SERVICES SUPPORT FOR DSC/BMDO	Awarded	Analex	15-Dec-98	$20M	
2190	NEXT GENERATION INFORMATION INFRASTRUCTURE INFORMATION ASSURANCE	Awarded	SRI, Teknowledge, ZELTEC, U-Penn, Cycorp	29-Jun-99	$15M	
2704	INFORMATION ASSURANCE AND SURVIVABILITY	Awarded	Multiple SB Awardees	20-Jun-01	$132M	
2721	INFORMATION ASSURANCE SCIENCE AND ENGINEERING TOOLS PROGRAM	Awarded	Logicon, UCLA, Shaulik Technologies	13-Jun-01		
2722	AUTONOMIC INFORMATION ASSURANCE	Awarded	Three SBs	19-Sep-00		
2884	INFORMATION ASSURANCE SERVICES	Cancelled		25-Jan-00	$1.5B	
4413	ACCESS CONTROL OPERATIONS FOR THE DIRECTOR FOR SECURITY COUNTER INTELLIGENCE AND IA	Cancelled		16-May-02	$5M	# 6315
4690	RESEARCH & DEVELOPMENT AND IA	Cancelled		21-May-02		
5308	INFORMATION ASSURANCE ENGINEERING SERVICES	Awarded	WCI	30-May-02	$50M	
5311	INFORMATION ASSURANCE ENGINEERING SERVICES	Awarded	Honeywell	30-Sep-02	$48M	
9012	IA AND SOFTWARE ENGINEERING SERVICES	Awarded	SAIC, BAH, L3/Titan	27-Feb-04	$57M	
9841	INFORMATION ASSURANCE PROGRAM	Awarded	BAH	16-Nov-04	$62M	
10393	IA FOR THE US INTELLIGENCE COMMUNITY	Awarded	12 Universities and SBs	13-Jan-05	$17M	
12429	INFORMATION ASSURANCE SUPPORT SERVICES	Cancelled		10-Feb-04		
12891	INFORMATION ASSURANCE VULNERABILITY MANAGEMENT DOD-WIDE ENTERPRISE LICENSE	Awarded	DigitalNet	14-Jul-04		
14111	INFORMATION ASSURANCE AUDIT ASSESSMENTS AND CHIEF FINANCIAL OFFICERS AUDIT SERVICES	Awarded	20 Accounting Firms	1-Feb-05	$977M	
14290	INFORMATION ASSURANCE SERVICES II	Cancelled		28-Nov-06	$1.5B	
13976	INFORMATION ASSURANCE SUPPORT	Awarded	Dynamic Network Enterprises	22-Sep-04	$17M	
15269	INFORMATION ASSURANCE ENGINEERING SUPPORT SERVICES FULL AND OPEN	Pre-RFP		29-Dec-06	$249M	
16271	INFORMATION ASSURANCE ENGINEERING SUPPORT SERVICES SMALL BUSINESS	Pre-RFP		1-Jan-07	$50M	# 5308
31287	IA FOR THE US INTELLIGENCE COMMUNITY	Cancelled		17-Feb-06		# 10393
32698	IA AUDIT ASSESSMENTS AND CFO AUDIT SERVICES	Pre-RFP		1-Aug-09	$977M	
33287	INFORMATION ASSURANCE SUPPORT	Pre-RFP		1-Mar-09		#13976
34916	OPERATION OF INFORMATION ASSURANCE TECHNICAL ANALYSIS CENTER RECOMPETE	Pre-RFP		1-Jun-07	$750M	
35236	IA SUPPORT SERVICES FOR THE OFFICE OF IA AND COMPLIANCE TACTICAL DIRECTORATE	Awarded	Seidcon	14-Jul-06	$5M	
35468	CAPABILITY DEVELOPMENT DOCUMENT STUDY/SPACE MISSION DATA CRYPTOGRAPHIC MOD IA SYSTEM	Awarded	BAH	22-May-06	$10M	
35522	IA AND SOFTWARE ENGINEERING SERVICES	Pre-RFP		1-Jul-07	$57M	# 9012
35513	INFORMATION ASSURANCE	Awarded	Gnostech	31-Jul-06	$11M	
36525	INFORMATION ASSURANCE SYSTEMS ENGINEERING SUPPORT	Post-RFP		3-Aug-06		
36533	IA AND SECURITY CERTIFICATION FOR WARFARE SYSTEMS	Awarded	Southeastern Computer	14-Aug-06	$15M	
35586	ARMY APPROVED INFORMATION ASSURANCE PRODUCTS IN SUPPORT OF COMMUNICATION SECURITY REQUIREMENTS	Pre-RFP		1-Feb-07		
37143	IA ARCHITECTURE AND PORTFOLIO MANAGEMENT	Pre-RFP		1-Dec-06		
37266	INFORMATION ASSURANCE ENGINEERING SERVICES	Pre-RFP		1-Mar-07	$48M	# 5311
38989	INFORMATION ASSURANCE SUPPORT SERVICES	Post-RFP		16-Nov-06	$12M	

On Manipulability of Algorithms

Sviatoslav Braynov
University of Illinois at Springfield, IL, USA
sbray2@uis.edu

Abstract: In adversarial situations, the input data to an algorithm could be manipulated to make the algorithm produce erroneous output or to make a wrong decision. The paper presents a formal definition and a model of algorithm manipulation from a game-theoretic point of view. Algorithm manipulation is viewed as a game between a decision maker and an adversary. The decision maker runs an algorithm to make a decision, whereas the adversary manipulates the input data to his own advantage and to the disadvantage of the decision maker. The paper also proposes a method for decision making based on manipulated input. According to the method, the decision strategy and the manipulation must be in Nash equilibrium. In other words, the decision strategy is the best response to the manipulation and vice versa, the manipulation is the best response to the decision strategy.

Keywords: data manipulation, decision making, algorithm design, adversarial learning, adversarial planning, adversarial plan recognition.

1. Introduction

It has been traditionally assumed that the input of every algorithm is a correct representation of a computational problem to be solved by the algorithm. This assumption is innocuous in situations where the input comes from a trusted source who is interested in solving the computational problem and collaborates with the algorithm. The assumption, however, does not hold in adversarial situations where the entity that provides the input data is different from the entity that uses the output data. In such situations, the adversary might misrepresent, slant, or even falsify the input to his own advantage and to the disadvantage of the party that relies on the algorithm output. Examples of such situations are abundant. For instance, the input for a virus detection algorithm is directly supplied by the virus writer in the form of a binary file. The virus writer has a strong incentive to obfuscate the binary and make it difficult for detection. Similar problems exist in anti-terrorist data mining where the input data is generated by terrorists who modify and disguise their activities in order to avoid detection. Another example is intrusion detection where intruders modify their actions in order to pass as legitimate users. Spam filters also suffer from the same problem because spam is usually disguised as legitimate email. A common feature of all of these situations is the possibility to manipulate algorithm input and defeat the algorithm.

Although the problem has been recently recognized and discussed by research in data mining (Dalvi et al. 2004, Lowd and Meek 2005) and computer security (Barreno et al. 2006), the general understanding of the problem is still limited. In addition, the available solutions are problem specific and tailored to particular algorithms. The contribution of this paper is a general definition of algorithm manipulation that is applicable to a wide range of algorithms and problem situations, including data mining, computer security, and antiterrorist research. Algorithm manipulation is defined as a game between two parties with conflicting interests: a decision maker and an adversary. The decision maker runs an algorithm to make a decision, whereas the adversary manipulates the input data to his own advantage and to the disadvantage of the decision maker.

Another contribution of the paper is a method for decision making based on manipulated input. In many cases, manipulation is unavoidable and decision makers must know how to detect data manipulation and limit its effect on decision making. The method proposed in the paper “reverses” manipulated data to its original state and makes a decision based on the original data.

The paper is organized as follows. Section 2 introduces definitions of algorithm manipulation and discusses their advantages and disadvantages. Section 3 proposes a method for decision making based on manipulated input. The paper concludes with a brief summary of results.

2. Definition of algorithm manipulation

In general, algorithm manipulation involves an algorithm that solves a particular computational problem and two parties with conflicting interests: a decision maker and a manipulator. The decision maker needs to solve an instance of the computational problem in order to make a decision that affects both him and the manipulator. The decision maker feeds the problem instance into the algorithm and makes a decision based on the algorithm output. The interesting part is that the decision maker does not have complete control over the problem instance. Before the problem instance is entered into the algorithm, it could be partially or

completely modified by the manipulator, leaving the decision maker uncertain about the real problem he is trying to solve. For example, the manipulator can modify the problem instance to his advantage and to the disadvantage of the decision maker. Figure 1 represents a schematic view of the main steps of algorithm manipulation.

We assume that the manipulator knows how the algorithm works. This assumption is realistic and complies with Kerckhoffs' principle (Kerckhoffs 1883) used in cryptography. According to Kerckhoffs' principle, the security of a system cannot be based on the secrecy of its internal working because sooner or later the adversary gets to know the system.

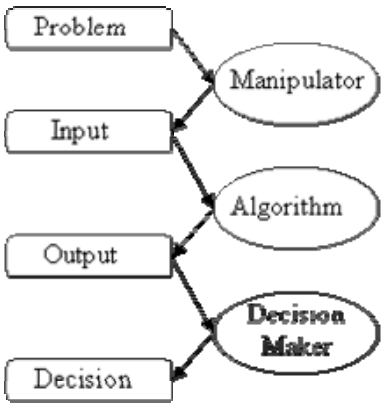


Figure 1: The main steps of algorithm manipulation

As an example of algorithm manipulation, consider a data mining algorithm that takes a trace of suspects' actions as input and detects patterns of potential terrorist activities. If a terrorist knows the internals and the logic of the algorithm, he would be able to find the patterns the algorithm is looking for and modify his behavior in order to avoid detection. In our previous research (Braynov 2006a, Braynov 2006b) we studied different manipulative methods, such as dummy actions, decoy goals, and buffers that could be used by terrorists to obfuscate their traces and make them difficult for analysis. We also proposed a model for detecting manipulations and for identifying cells in malicious networks.

The interaction between the manipulator and the decision maker can be modeled as a two-person non-cooperative game.

Definition 1. The non-cooperative game between the manipulator and the decision maker is defined as a tuple $\langle \mathcal{A}, \mathcal{P}, \mathcal{M}, \mathcal{D}, U_M, U_D \rangle$, where:

- $\mathcal{A}$ is an algorithm that takes a problem instance, $P \in \mathcal{P}$, as input and produces a solution as an output $\mathcal{A}(P)$.
- $\mathcal{P}$ is the set of all problem instances that the algorithm can take as input.
- $\mathcal{M}$ is the set of manipulative actions available to the manipulator. When applied to a problem instance P, $P \in \mathcal{P}$, a manipulative action produces another problem instance P′, $P' \in \mathcal{P}$. In other words, $\mathcal{M} : \mathcal{P} \rightarrow \mathcal{P}$.
- $\mathcal{D}$ is the decision function of the decision maker. The function produces a decision $\mathcal{D}(\mathcal{A}(P))$ on the output $\mathcal{A}(P)$ produced by the algorithm for a problem instance P.
- U_M is the utility function of the manipulator, $U_M: \mathcal{D} \times \mathcal{P} \rightarrow \mathcal{R}$. It maps a pair of a decision and a problem instance into a real number.
- U_D is the utility function of the manipulator, $U_D: \mathcal{D} \times \mathcal{P} \rightarrow \mathcal{R}$. It maps a pair of a decision and a problem instance into a real number.

Definition 2 (strong version) An algorithm A is non-manipulable if for every manipulative action m, $m \in \mathcal{M}$, and every problem instance P, $P \in \mathcal{P}$:

$$\mathcal{A}(P) = \mathcal{A}(m(P))$$

Definition 2 says that for every manipulative action *m* that changes the problem instance from *P* to *m(P)*, the algorithm produces the same output as with the original problem instance *P*. In other words, a non-manipulable algorithm is not affected by manipulations and it always produces correct output, whether it works on a manipulated input or not.

It has to be pointed out that not every manipulation produces a valid problem instance. Some manipulations could corrupt the problem instance and produce illegitimate input. Such manipulations are of no practical interest because they could easily be detected and rejected using proper input validation. Throughout this paper, we assume that manipulations always produce valid problem instances. In other words, they convert one problem instance into another.

Definition 2 requires that the algorithm be somewhat insensitive to variations in problem instances and produce the same output for a large number of different inputs. This is an unrealistically strong assumption that could constrain the applicability of the model. We relax the assumption in the next definition.

Definition 3 (weaker version) *An algorithm A is non-manipulable if for every manipulative action m, $m \in \mathcal{M}$, and every problem instance P, $P \in \mathcal{P}$:*

$$\mathcal{D}(\mathcal{A}(P)) = \mathcal{D}(\mathcal{A}(m(P)))$$

Unlike Definition 2, Definition 3 allows the manipulator to affect the output of the algorithm. For example, by changing the problem instance from *P* to *m(P)*, the attacker could make the algorithm produce output $\mathcal{A}(m(P))$ that is different from $\mathcal{A}(P)$, the output the algorithm would produce if it was given an unmanipulated problem instance:

$$\mathcal{A}(P) \neq \mathcal{A}(m(P))$$

The interesting part about Definition 3 is that the difference in the algorithm output does not affect the decision of the decision maker. The decision remains the same on both unmanipulated, $\mathcal{A}(P)$, and manipulated output, $\mathcal{A}(m(P))$. In other words, it is not the algorithm that is not affected by manipulations, but the decision function $\mathcal{D}$ of the decision maker. The intuition behind Definition 3 is that small changes in the algorithm output might not affect a decision. For example, a terrorist might try to manipulate the output of a data mining algorithm that detects patterns of terrorist activity by camouflaging and hiding some actions. As a result, the confidence level of the detected pattern could fall by 2% which, however, might not be sufficient for the decision maker to consider the pattern inconclusive of evidence. In many cases, the decision maker will allow for some variations in the algorithm output and will not change his decision until the variations exceed some critical level. Therefore, the goal of the manipulator would be to convert the original problem instance *P* to a new instance *m(P)* that will modify the algorithm output sufficiently enough to warrant a change in decision:

$$\mathcal{D}(\mathcal{A}(P)) \neq \mathcal{D}(\mathcal{A}(m(P))) \qquad \textbf{(1)}$$

Definition 3 still does not provide a good quantitative interpretation of manipulability because it neglects two important factors: the manipulation cost that the manipulator incurs and the effect of the manipulation on the decision maker. The manipulation cost is important because some manipulations could be prohibitively expensive for the manipulator and, therefore, practically infeasible. The manipulator will carry out a manipulative action $m \in \mathcal{M}$ on a problem instance $P \in \mathcal{P}$ if the manipulation cost does not overweight the manipulation benefits:

$$U_M(D(A(m(P))), P) > U_M(D(A(P)), P) \qquad \textbf{(2)}$$

On the left side of Equation 2 we have the manipulator's utility from carrying out a manipulative action *m* on a problem instance *P*. The right side of the equation represents his utility without manipulation. Because

different manipulative actions might have different costs, a rational manipulator must always maximize his utility by choosing the most efficient manipulation *m**:

$$m^* = \underset{m \in \mathcal{M}}{\operatorname{argmax}}\ U_M(D(A(m(P))), P)$$

The cost of manipulation draws a demarcation line between theoretical and practical manipulability. Although, a manipulation could exist in theory, it could be prohibitively expensive in practice. A manipulation of a problem instance $P \in \mathcal{P}$ is theoretically possible if there exists a manipulative action $m \in \mathcal{M}$ that satisfies Equation 1. Not every manipulative action, however, will satisfy Equation 2 that guarantees cost efficiency for the manipulator. Because different manipulators have different skills and knowledge, what is only theoretically manipulable for one manipulator could be practically manipulable for another.

Another important factor neglected by Definition 3 is the effect the manipulation has on the decision maker. Theoretically, it could be the case that a manipulation might not always hurt the decision maker. For example, a manipulation could make the decision maker reach a decision that is as good as the decision he would reach without manipulation. To exclude such unnatural cases, we assume that there is a conflict of interest between the manipulator and the decision maker in which the decision maker suffers a loss from every manipulation:

$$U_D(D(A(m(P))), P) < U_D(D(A(P)), P) \qquad \textbf{(3)}$$

The left side of Equation 3 represents the decision maker's utility from a manipulated decision, whereas the right side represents his utility without manipulation. According to the definition, the decision maker regrets making decision, *D(A(m(P)))*, based on the manipulated output of the algorithm. If he knew the actual problem instance *P*, he would act differently and decide *D(A(P))* instead of deciding *D(A(m(P)))*.

The next definition is an improvement of Definition 3 that takes care of the manipulation cost and the effect of the manipulation on the decision maker.

Definition 4 *An algorithm A is non-manipulable if for every problem instance P, $P \in \mathcal{P}$, there does not exist a manipulative action m, $m \in \mathcal{M}$, such that:*

$$U_M(D(A(m(P))), P) > U_M(D(A(P)), P)$$
$$U_D(D(A(m(P))), P) < U_D(D(A(P)), P)$$

According to Definition 4, the manipulator must not have an action, *m,* in his repertoire that allows him to manipulate the problem instance *P* to his advantage and to the disadvantage of the decision maker.

3. Decision making based on manipulated input

In this section we discuss the problem of how to make a correct decision based on manipulated input. To solve the problem we first "reverse" the manipulation and identify the real problem instance. Then, we replace the manipulated input with the real input and make a decision based on the real problem instance.

Suppose that the observed problem instance, *P*, has been manipulated. In this case, it does not make much sense to use the algorithm to solve *P* because the results would be probably erroneous. A better alternative would be to find the real problem instance and use it as a basis for a decision. This could be done by working backwards and computing the set of all possible problem instances from which *P* could be generated by applying manipulative actions:

$$Range(P) = \{P_k \mid \exists\, m \in \mathcal{M},\ m(P_k) = P\}$$

Apparently, the real problem instance belongs to *Range(P)*. The problem, however, is that the real problem instance could be buried under many other possible candidates in *Range(P)*. The larger *Range(P)* is, the more difficult it is to find the real problem instance, and the greater the confusion effect. In the simplest case, *Range(P)* would have only one element, the real problem instance. In most cases, however, the manipulator would try to hide his intentions by choosing *Range(P)* in which all problem instances seem equally likely. This creates multiple possibilities and complicates the analysis.

Finding *Range(P)* could be computationally expensive. In general, an exhaustive search for *Range(P)* has a running time of $|\mathcal{M}||\mathcal{P}|$, where $|\cdot|$ denotes set cardinality. A better approach would be to "reverse" every manipulative action and apply the "reversed" actions on *P* to obtain *Range(P).* This method has a running time of $|\mathcal{M}|$. In other words, for every manipulative action, $m \in \mathcal{M}$, one needs to compute, m^{-1}, such that if $m(P_1)=P_2$, then $P_1 \in m^{-1}(P_2)$. In general, $m^{-1}(P_2)$ could produce more than one problem instance, for example $m^{-1}(P_2)=\{P_1,\ldots, P_k\}$.

The next task after finding *Range(P)* is to narrow down the set of possible problem instances by assuming that the manipulator chooses the most effective manipulation, i.e., the manipulation that maximizes his utility. In other words, if the real problem instance was P°, $P^\circ \in$ *Range(P)*, then the manipulator would choose to manipulate P° and produce P if and only if there is no other manipulation that gives him higher utility:

$$P = \underset{P' \in \mathcal{P}}{\operatorname{argmax}}\, U_M(D(A(P')), P^\circ) \qquad \textbf{(4)}$$

To find the real problem instance from the observed problem instance *P* the decision maker needs to solve Equation (4) for P°. Then, he needs to feed the real problem instance P° into the algorithm and make a decision.

To illustrate the method, consider the following example of a manipulation game in which the role of the manipulator is played by an attacker who wants to reach a target. The decision maker is a defender who must decide how to block the attack. The situation could be conveniently represented by the attack graph shown in Figure 2. The attack graph and the structure of the game, defined in Definition 1, are common knowledge to both players. The notion of attack graph captures the intuition about all possible attacks that can be launched against a set of targets.

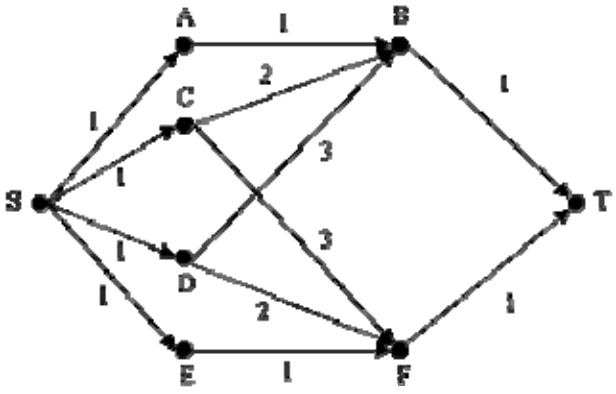


Figure 2: Attack graph

Definition 5. *An **attack graph** is a tuple AG = <$\mathcal{S}$, A, $\mathcal{T}$, s_0, T> where:*

- *$\mathcal{S}$ is a finite set of states,*
- *A(s) is the finite sets of actions available to the attacker in state $s \in \mathcal{S}$,*
- *$\mathcal{T} \in \mathcal{S} \times A \times \mathcal{S}$ is the transition relation,*
- *$s_0 \in \mathcal{S}$ is the initial state,*
- *$T \in \mathcal{S}$ is the set of possible targets.*

Intuitively, the attacker takes actions that move the system from one state to another according to the transition relation $\mathcal{T}$. The attacker starts in the initial state, s_0, and attempts to reach one of the target states. By definition, the attack stops when it reaches a target state, i.e., $A(s) = \varnothing$ for every target state *s*, $s \in T$. Although Definition 5 is similar to the definition of attack graphs used in computer security (Jha et al. 2002, Phillips and Swiler 1998, Swiler et al. 2001) our model can be applied to a wide range of applications, including military operations, surveillance and intelligence gathering, economic applications, etc. For example, the attacker could be trying to penetrate a market that the defender currently dominates.

The attacker starts at the initial position *S*, chooses an attack route from the attack graph, and tries to reach the target *T*. Each arc is labeled with the cost of traversing the arc. The defender's objective is to protect the

target without knowing which particular attack route has been chosen. The situation is complicated by the fact that the defender cannot block all possible routes because of time and resource limitations. In this case, he can block either arc BT or arc FT. Blocking both arcs simultaneously is beyond his means. In addition, blocking arcs takes time and the defender must decide which arc to block right after the attacker has made his first move. We assume that the attacker does not know which arc is blocked and, therefore, cannot switch to another route to bypass the blocked arc. In other words, every blocking action is hidden to the attacker and he needs to decide upfront which attack route to follow.

The game proceeds as follows. The attacker chooses an attack route and follows it until the game ends. The defender observes the first move of the attacker, and based on this observation decides which arc to block: BT or FT. The game ends when the attacker reaches either the target or a blocked arc.

Blocking an arc is equivalent to permanently removing an arc from the attack graph. For example, by blocking arc BT, the attacker simultaneously disables attack routes SABT, SCBT, and SDBT. This, however, is not sufficient to prevent an attack because the attacker could have chosen the attack route SEFT. Similarly, blocking arc FT simultaneously disables attack routes SCFT, SDFT, and SEFT, while keeping routes SABT, SCBT, and SDBT open for an attack.

By observing the attacker's first action the defender needs to predict the route that has been chosen by the attacker. We assume that the defender uses a plan recognition algorithm to predict the attack route. The algorithm takes the attacker's first move and the attack graph as inputs and produces a prediction of an attack route.

This scenario fits well with our model of algorithm manipulation. Here, the problem instance is a pair, *(first move, attack graph)*, the first element of which is generated by the attacker. The problem instance is fed into the plan recognition algorithm that produces a prediction of an attack route. Based on the prediction, the defender decides to block either BT or FT. Because of the conflict of interest between the attacker and the defender, the attacker might try to manipulate the plan recognition algorithm to his advantage. For example, suppose that the plan recognition algorithm is based on the assumption that the attacker follows the shortest path to the target. Suppose further, that the attacker has chosen the shortest attack route SABT. Once he reaches point A, the plan recognition algorithm will unambiguously conclude that he has chosen route SABT, because SABT is the only route that passes through point A. To prevent this, the attacker could first visit point D and then, shift back to the original route by visiting B and T. Such manipulation will hide the attacker‘s intention and fool the plan recognition algorithm into believing that the attack route is ADFT, because DFT is the shortest path that passes through D. Based on this erroneous prediction, the decision maker will decide to block arc FT and, therefore, lose the game. In this case, visiting point D hides the attacker's intention and confuses the plan recognition algorithm. The problem is that there are two routes that pass through point D, SDBT and SDFT, and either route could have been chosen by the attacker.

This example suggests that we cannot further assume that the manipulator chooses the most effective manipulation (described by Equation 4) because it will directly reveal the manipulator's intention. In addition, efficiency might not be the only goal of the manipulator. In many cases, he would hide and obfuscate his intentions in order to increase the probability of success. Therefore, the manipulator might be willing to spend some additional resources for manipulating the algorithm and influencing the decision of the decision maker. Suppose, for example, that the attacker has decided to go through point D and attack the target through arc BT. Although this is the most expensive route he can take, it is worth the extra cost because of the confusion and ambiguity it creates. The attacker could confuse the defender even more by rolling a die at point D and deciding to take route DBT with probability p and route DFT with probability $1-p$. Without loss of generality, we can assume that the attacker rolls a die at the starting point S and randomly takes one attack route. This corresponds to a randomized route:

$$[SABT:p_1,\ SCBT:p_2,\ SCFT:\ p_3, SDBT:p_4, SDFT:p_5,\ SEFT:p_6]$$

where each route is taken with probability p_i, $i = 1,\ldots,6$. It is obvious that $p_1=p_6=0$ because routes SABT and SEFT will be unambiguously recognized by the plan recognition algorithm after the attacker's first move. For example, if the attacker's first move is SA, then $Range(SA)=\{SABT\}$, there is no manipulation, and the intention of the attacker is clear. On the other hand, if the attacker's first move is SC, then $Range(SC)=\{SCBT,SCFT\}$ and both SCBT and SCFT are possible. Apparently, the more elements *Range* has, the more difficult it is to find the attacker's intention.

Because the attacker will not take SABT or SEFT, the randomized route can be simplified to:

$$[SCBT:p_1,\ SCFT:\ p_2,\ SDBT:p_3,SDFT:p_4]$$

where $p_1+p_2+p_3+p_4=0$. Because the attack graph is symmetric, we can search for a symmetric solution in which $p_1=p_4$, and $p_2=p_3$:

$$[SCBT:p/2,\ SCFT:\ (1-p)/2,\ SDBT:(1-p)/2,SDFT:p/2] \qquad \textbf{(5)}$$

The best defense against such a random route is a randomized blocking strategy in which the defender blocks arc BT with probability *q* and arc FT with probability *1-q*. The probability *q* depends on the attacker's first move. For instance, if the defender observes SC as a first move, then he will expect that the attacker will take route CBT with probability *p* and route CFT with probability *1-p*. In this case, the defender must block arc BT with probability *p* and arc FT with probability *1-p* (assuming that blocking BT and blocking FT are equally costly). Similarly, if the defender observes SD as a first move, then he will expect the attacker to take route DFT with probability *p* and route DBT with probability *1-p*. This observation brings us to the following proposition.

Proposition. *If blocking BT is as costly as blocking FT, and if the value of the target for the attacker is G, then the following strategies are in Nash Equilibrium (Nash 1950):*

The defender's strategy: The defender observes the first move of the attacker. If it is SA, then he blocks BT with probability 1. If it is SC, then he blocks BT with probability (G+1)/2G and FT with probability (G-1)/2G. If the first move is SD, the defender blocks BT with probability (G-1)/2G and FT with probability (G+1)/2G. Finally, if the attacker's first move is SE, the defender blocks FT with probability 1.

The attacker's strategy: The attacker chooses the random route from Equation (5), where p=(G+1)/2.

Proof. To make the attacker willing to randomize between SCBT, SCFT, SDBT, and SDFT these routes must have the same utility. The utility of SCBT and SDFT is *G(1-p)-4* and the utility of SCFT and SDBT is *Gp-5*. Therefore,

$$G(1-p) - 4 = Gp - 5$$

□

The proposition shows that instead of choosing the cheapest manipulation, a rational attacker will randomize between several manipulations in order to hide his intentions. This increases the confusion effect and makes the manipulation difficult to understand.

In general, decision making based on manipulated input involves the following steps:

1. If the manipulated problem instance is P, find *Range(P)* that includes all possible unmanipulated problem instances. For example, *Range(SC)={SCBT, SCFT}*.
2. Narrow down *Range(P)* by eliminating those manipulations that do not hide enough the manipulator's intentions. For instance, routes SABT and SEFT will never be taken by the attacker.
3. Find a randomized manipulation strategy by assigning a probability to each problem instance in *Range(P)*:

   Randomized manipulation = $[P_1: p_1, P_2:p_2,\ldots, P_n:p_n]$,

   where $Range(P) = \{P_1,P_2,\ldots,P_n\}$.

   For example:

   Randomized manipulation = *[SCBT:p, SCFT: 1-p]*
4. Find a randomized decision:

   Randomized decision = $[D_1: q_1, D_2:q_2,\ldots, D_n:q_n]$

   where D_i is the decision that would be made if the real problem instance was P_i. In other words, the decision maker makes a randomized decision in which every D_i is made with probability q_i. For example, upon observing that the attacker has moved to point C, the defender decides to block arc BT with probability *q*, and arc FT with probability *1-q*.
5. Solve for p_i and q_i so that the randomized manipulation and the randomized decision are in Nash equilibrium. In our example, *p = q = (G+1)/2G*.

It is very important that the manipulation and the decision strategy are in Nash equilibrium. The equilibrium guarantees that the manipulation is the best response to the decision strategy and vice versa, the decision

strategy is the best response to the manipulation. In other words, even if the manipulator knows the internals of the algorithm and the decision function (from Definition 1), he cannot do anything better than to stick to the manipulation. Therefore, the algorithm cannot be further manipulated and it is safe to expose its internals to the public. This satisfies the Kerckhoffs' principle, while preserving the integrity of the decision making process.

4. Conclusions

The paper introduces the problem of algorithm manipulation in general and provides a definition that views algorithm manipulation as a game of conflict between a manipulator and a decision maker. The definition takes into account several important factors of algorithm manipulation, such as the possibility and the cost of manipulation, the manipulation incentives, and the effect of the manipulation. It is shown that a rational manipulator should avoid cost-efficient manipulations in order to better hide his intentions.

The paper also presents a general model of decision making based on manipulated data. The method "reverses" possible manipulations and tries to restore the input data to its original state. In order to discourage manipulations, the method proposes a randomized decision making strategy.

Because the paper provides a general treatment of the problem of algorithm manipulation, the framework and the definitions could be applied to a wide range of problems, including adversarial data mining, adversarial learning, adversarial planning and plan recognition, antiterrorist research, and information warfare.

References

Barreno, M., Nelson, B., Sears, R., Joseph, A. and Tygar, J. (2006) 'Can machine learning be secure?' *Proceedings of the ACM Symposium on Information, Computer and Communications Security (ASIACCS'06)*, Taipei, Taiwan.

Braynov, S. (2006a) 'Adversarial planning and plan recognition: Two sides of the same coin.' *Secure Knowledge Management Workshop*, New York.

Braynov, S. (2006b) 'Adversarial planning in networks.' *Descartes Conference on Mathematical Models in Counterterrorism*, Rayburn House Office, U.S. Capitol.

Dalvi, N., Mausam, P. and Verma, D. (2004) 'Adversarial classification.' *In the Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 99-108.

Jha, S., Sheyner, O. and Wing J. (2002) 'Two formal analyses of attack graphs.' In C*omputer Security Foundations Workshop (CSFW)*, Nova Scotia, Canada, pp. 49–63.

Kerckhoffs, A. (1883) 'A cryptographie militaire.' *Journal des sciences militaries*, vol. IX. pp. 5-83.

Lowd, D. and Meek, C. (2005) 'Adversarial learning.' *In the Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 641–647.

Nash, J. (1950) 'Equilibrium points in N-person games.' *Proceedings of the National Academy of Sciences*.

Phillips, C. and Swiler, L (1998) 'A graph-based system for network vulnerability analysis.' *In the Proceedings of the 1998 Workshop on New Security Paradigms*, pp. 71–79.

Swiler, L., Phillips, C., Ellis, D., and Chakerian, S. (2001) 'Computer-attack graph generation tool.' *In DARPA Information Survivability Conference and Exposition*, Anaheim, California, pp. 146–161.

Developing a Framework to Improve Information Assurance Battlespace Knowledge

Adam Bryant and Michael Grimaila
Air Force Institute of Technology, Fairborn, OH, USA
adam.bryant@afit.edu
michael.grimaila@afit.edu

Abstract: In any organization that operates in cyberspace, it is necessary to clearly understand the information battlespace. Defensive information battlespace knowledge can be provided by detailed analysis of carefully selected metrics using automated analysis, data mining, and data calls. Gathering data for metrics involves many of the same pitfalls as collecting data in research but also incurs the intra-organizational communication problems inherent in organizations. These factors make it difficult to turn data and metrics into decision-ready knowledge. This paper explores data gathering in an organization as a research program and a knowledge transfer activity.

Keywords: Information assurance, metrics, strategic alignment

1. Information battlespace knowledge

On a traditional battlefield, knowing about the battlespace involves understanding the environment of the battlefield, knowing what assets each opponent has and what their capabilities are, and understanding the players on the battlefield: knowing their strengths, weaknesses, motivations, and what makes them succeed or fail. As a contrast, in cyberspace, every organization and person that operates in cyberspace is a player, and any assets they have that are accessible from cyberspace are also in the information battlespace. Just as on a traditional battlefield, organizations that operate in cyberspace need to know about their own defenses and controls because they are being scrutinized and attacked daily.

This paper examines the process of gaining *defensive* information battlespace knowledge, and addresses some of the pitfalls with popular approaches to solving this problem. It also addresses a new way to frame the problem: as part of a knowledge-development undertaking rather than a data-gathering problem.

The idea of knowing the battlespace is not new, but it is framed nicely by former U.S. Navy Rear Admiral, Bill Owens. In his book, *Lifting the Fog of War* (Owens, 2000), Admiral Owens introduces his "system of systems" model for understanding the battlespace—not just the information battlespace, but also the realm of tanks, airplanes, bombers, battleships, and adversaries with guns. In his model, he presents his concepts as a Venn diagram:

- Dominant Battlespace Knowledge
- Immediate/Complete Battle Assessment
- Near-Perfect Mission Assignment

The idea of these three capabilities is best illustrated with an example: The commander of troops needs a whole picture of the battlefield. He is required to make life or death decisions with only scant information. The quality and timeliness of information used in his decision making determines the quality of his decisions. He needs three things: 1) the ability to see the whole battlespace for situational awareness (dominant battlespace knowledge); 2) the ability to have feedback about his troops' actions to determine what future actions will need to be taken (immediate/complete battle assessment); and 3) the ability to command his troops with as little latency as possible (near-perfect mission assignment). A graphical representation of this model is shown in Figure 1 below.

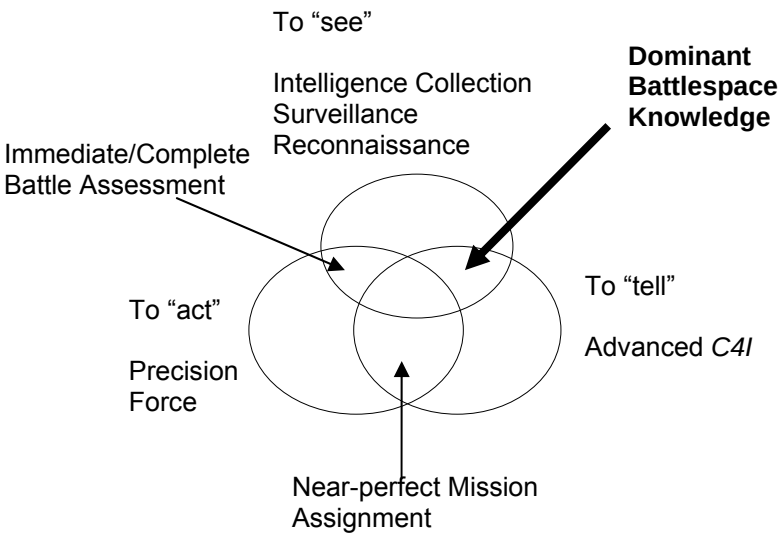


Figure 1: Owens "system of systems" model

This paper addresses the dominant battlespace knowledge element of his model. In Owens' model, battlespace knowledge provides leaders the ability to "see" and "tell," which constitutes what we know as the areas of Intelligence, Surveillance, and Reconnaissance (ISR) and Command, Control, Communications, Computers, and Intelligence (C4I). Although this model can be used for understanding the big picture in any command and control activity, we narrow it to focus strictly on topics related to information warfare.

2. Defensive information battlespace knowledge

In the information assurance arena, knowledge of the information battlespace is a key to having an advantage. Unfortunately, this has proven to be a difficult problem for many organizations. Geer et al. (2003) identified organizations that pressed forward with risk management plans; purchased servers, infrastructure, and network security solutions; and hired a plethora of experts but still could not quantitatively answer a few key questions:

- How secure are we?
- Are we more secure than we were last year?
- Are we more secure than our competitors?

Answers to these questions can be generated in a number of ways. The first is simply to guess what the battlespace looks like based upon implicit knowledge. Although definitely the easiest to implement, this has little value except in producing a quick decision. The second is to rely on "experts" (e.g. consultants, trusted advisors, or staff). This only delegates the problem of understanding the battlespace to a lower level, or in the case of consultants, outsiders. The third method, and the one we will discuss in this paper, is to collect metrics about the organization's information battlespace.

3. Using metrics to gain defensive information battlespace knowledge

There are numerous frameworks which are designed to help organizations understand their cyberspace assets, but it has been difficult to integrate those components into one canonical model for information assurance. Academics and practitioners alike agree that security metrics, or collections of performance measures for Information Assurance (IA), hold the key to better knowledge of an organization's IA capabilities (Geer, 1998).

How does an organization best organize what it knows about itself? One key is by looking at its knowledge of its defenses. From a top-down view, this can be done by asking hard questions like "How secure are our information systems?," "What is our current level of risk?," and "What is our current risk tolerance?" Connection to cyberspace poses many unique risks, and as such, information assurance is a risk

management activity (Geer, 1998; Blakley et al. 2001). When an organization has a better understanding of what the risks are and what the organization's tolerance of risks is, it can be better poised to answer these hard questions. But what is the payoff? When an organization employs a security metrics program, it can make informed decisions about building and maintaining its security capability. Some examples are better allocation of funding, increased performance, fewer unauthorized intrusions or information security incidents.

When it comes to managing risk, there are a few different things an organization needs to know. First, the organization needs to be keenly aware of its threats. Many in the information security community rely upon fear, uncertainty, and doubt as a means to justify resources for information assurance. However, this does little to quantify the likelihood of the risks that the threats impose upon the organization given their vulnerability profile. Second, a valuation of the organizational information assets is necessary to calculate the cost-benefit tradeoffs for various control measures.

Organizations have finite resources, so it is impractical to protect against all possible threats. Finally, the organization must remain vigilant on estimating its threats and discovering vulnerabilities. The dynamic nature of threats and vulnerabilities makes this one of the most difficult tasks. The reality is that despite remaining aware of known threats and vulnerabilities, there is always the possibility of a "zero day exploit" that allows an adversary to exploit a yet undiscovered critical vulnerability. In each of these cases, metrics can be developed to provide an estimation of these values.

4. Three typical methods for gathering metrics to gain defensive information battlespace knowledge

When collecting battlespace knowledge, someone in the organization must decide what to measure and how to measure it. Ideally, this person would be at the strategic level of the organization, but from anecdotal evidence in the authors' experiences, this is not necessarily the case. Many times, the strategic level of the organization has questions that are not easily answered by direct measurement. The organization's strategic leadership often has high level questions about their security capabilities, vulnerability profile, and threat vectors that it cannot answer with any degree of reliability. Further, the level of aggregation desired often masks the underlying temporal trends in the data. The task of gathering data and conducting an analysis can be done in a variety of ways including:

- Using automated analysis, such as an executive information system (EIS) or business analysis portal (BAP);
- Have staff mine available data to see if it can provide insight into the problem or;
- Making a data call for a subordinate organization to answer.

4.1 Automated discovery and analysis

If there is an executive information system in place, the leaders will ideally be able to formulate good questions and make better decisions. An example of this is a leadership IT portal. A chief information officer (CIO) in a large government organization was known to have requested such a portal to be built. He wanted metrics collected and aggregated into a single system which he could view and get a feel of what areas needed his attention. This system also enabled him to drill down to a certain extent and get a more detailed look at the information presented to him.

A similar system, but for wider use was the Navy's Pacific Fleet Homeport. The Department of the Navy CIO (2002) described the Pacific Fleet Homeport as a knowledge portal which linked "over 200 databases through the Internet" and provided a means to reuse knowledge. This does not seem to have been the same as an executive information system, but the primary goal was the same: to aggregate information that existed across the fleet and enable better knowledge-fed decisions.

Another example is offered by Shen-Hsieh and Schindler (2002). They present a case study of a pharmaceutical R&D organization which they worked with to improve how the organization presented its data to its strategic leaders. They combined "visual 'pictures' of data with a deep understanding of individual and organizational decision processes" to enhance the knowledge that strategic leaders had to make decisions.

These kinds of solutions sound like they could be a great benefit, but developing a portal or an EIS, like any large enterprise information systems project, is another source of risk. An EIS is usually a front-end for a decision support system (DSS). Further, it may integrate many such DSS inputs into one system for the

enterprise. Each DSS must draw its inputs from several lower level systems, like transaction processing systems (Dobrzeniecki, 1994). Sometimes, developing such a BAP is not technically feasible because there are no suitable systems at the tactical and operational levels that can provide the necessary inputs.

4.2 Knowledge discovery in databases (data mining)

If the strategic level organization decides to have their staff mine available data, they may be overwhelmed because of the amount and types of data available. Data mining is a step within the practice of Knowledge Discovery in Databases (KDD) which is an attempt to find trends in available data through the use of search algorithms and analysis and make decision-level knowledge out of it (Fayyad, et al, 1996). In general though, the term "data mining" is used to describe starting from the data instead of the questions.

Data mining is the method of metrics generation in many businesses which have the "start with what you have" mentality. One of the reasons this is so popular is likely because it is cheap and intuitive. The data is already available and statistical software packages make it easy to find trends and fit mathematical models that can be used to make inferences about future events. Problems in data mining arise however when models are fit which are not valid. Fayyad (1996) mentions that in data mining, the "space of (possible) patterns (that could emerge) is often infinite." It is very easy to use statistical trends from available data in order to develop cause and effect or event relationships that lack a conceptual base (Leedy and Ormrod, 2005). In other words, it may give you the wrong answers if not used carefully, and may not provide answers to the questions you want.

A notional example of unsophisticated data mining event in an organization is as follows: an organization has no formalized security metrics plan but has data. This data comes in the form of firewall logs, incident reports, and captures from a sniffer trace outside the main firewall or gateway router. The organization's IT leadership wants to know what the biggest threat is to the organization's information assets. The staff aggregates the data from their sources, aggregates the data and joins it in a database on common factors, such as the date an event occurred or the network component that was involved. The staff looks for trends in the data and discovers that several machines infected with viruses all had the same "extra" open source software installed. They can investigate further at this point to prove the open source software is the problem, but in many cases, the staff would take this correlation as evidence enough to create a policy to outlaw that software, and maybe *any* open source software on their network.

A problem with this approach is that although the staff has established a link between the particular software and a virus attack, they have failed to prove causality by eliminating all other possible explanations for the correlation and by proving temporal precedence (i.e. the software was there before the virus). It may be the case that the problem was not the open source software, but the administrator rights that users have on their machines, or that the users of the machines with open source software are from the same department and have a shared login between dozens of people.

4.3 The data call

The final method we discuss is to make a data call to lower levels in the organization. In this case the strategic level or operational level sends a request for data to lower levels which the tactical levels must answer and return. If sent from strategic level to operational level, the operational level can either answer the data call if they have available data, or make a data call of their own to the tactical level organizations within their breadth. This is a common practice in the Department of Defense (DoD).

"Data call" data gathering is time-consuming and extremely personnel and resource intensive. Messages are sent down with e-mail or telephone calls, and responses are sent back up to several weeks later. Many times within military organizations there are also many data calls, only the highest priority data calls will be answered quickly. As the data call filters down, the workload is multiplied out toward every leaf of the organizational tree. This method also promotes filtering of information, which can prevent information overload in senior executives but is done at the cost of richness of the information. Some information may not make it up the chain because of the level of impact, or because it points to failings or "behavioral wrongdoing" in a particular department or manager (Smith, et al, 2001). At some point, due to different levels and types of data, data will have to be mined, sorted, and aggregated by hand.

An example of a data call is when a strategic level organization wants to know the number of users they have not had information assurance awareness training in over a year. The strategic level asks this question to each of their subordinate operational units, who, not having the data available task their subordinate

tactical units with providing them the data. Each of possibly hundreds of tactical units address the task. They gather the number of users they have then transmit it up the chain where the numbers are aggregated and forwarded up the chain to the strategic level IT leadership.

Some problems can emerge though in this process. Suppose in the downward communication of the message, some tactical units count members that are attached to their organization from other organization, but so does the organization those members came from. Also suppose that some units count personnel that haven't completed their training within the current year, instead of those that haven't completed their training within a 1-year period. Also, suppose some units count members that haven't completed at least one of a number of network training tracks, not just their mandatory information assurance training. The end result is that the true numbers will be off in a great number of cases, and the final answer that the strategic level gets may tell a very different story than reality. This metric may even be unusable if leadership wants a precise accounting.

4.4 Top-down or bottom-up approach

There exists a debate among security metrics practitioners about whether the bottom-up or top-down approach is the better way to generate metrics. The bottom-up approach is analogous to data mining, whereas the top-down approach is analogous to research. There are advantages and disadvantages to both strategies. Many organizations lack the energy, resources, and commitment to follow-through with what is required for a top-down strategy (Hong, et al., 2003). In other cases, the top-level objectives are too abstract to be operationalized without changing their meaning. The bottom-up approach lacks ideas for radical change, strategic thinking, and "inertia and business as usual" dominate (Earl, 2003). We'll look at aspects of both top-down and bottom up in our different approaches to measure organizational information assurance.

5. The task of measurement and data gathering

No matter what way the questions are answered, data gathering is one of the most important parts of collecting metrics (Papadapolous, 2004). There are many potential pitfalls, and it is easy to make data say many things, many of them conflicting. Care must be taken to ensure data collected which will be used for organizational purposes is of good quality.

Papadapolous (2004) discusses information analysis and the content supply chain. He outlines the steps of turning data into knowledge. His steps are:

- Data capture or *gathering*
- Data normalization and transformation or *organizing*
- Data analysis or *refining*
- Data reporting or *disseminating*

This paper deals with the first three stages. Tactical data gathering decisions can taint the normalizing phase. Every data collection project relies on numerous decisions. These decisions often have the form of "should this event be counted in category X or category Z?" Two departments with identical data can compile two completely different pictures based on the decisions they make concerning how to organize the raw data. For instance, if an organizational metric measures the ratio of dollars spent per information system attack, one department that includes port scans in their figures as "attacks" will have a very different ratio than the department that classifies port scans in some different way.

The decision to include or not include a certain item in a nominal categorization can be dependent on the motivations of the individual making the decisions. This information is then passed to senior management, with one of two messages: "we are secure...look at all the bad guys we've blocked!" or "we are constantly under attack...we need to increase security's budget." Both of these messages fail to articulate the fact that port scans are incredibly common on any Internet connected device. The only rational decision that this information would justify is port blocking by the firewall technician. But often it is reported to upper management to justify security expenditures, and ends up on performance appraisals to justify promotions. So should the strategic level organization leave it up to each tactical level organization to decide what should and should not be included? It is the authors' opinion that categorization decisions should rest with the strategic leaders in an organization.

5.1 Data gathering as a research activity

Since the questions that come from strategic level leadership drive the data gathering and knowledge formation process, it is not much of a stretch to see metrics generation as a group of research problems for each organization. Each type of question or has its own “best methods” to get at the answer, depending on the type of data relationship that is sought, the type of variables that are being measured, and the control over the environment. Yin (2003) says “each (method) is a different way of collecting and analyzing empirical evidence, following its own logic. And each strategy has its own advantages and disadvantages.”

Different research methods include quantitative and qualitative measures. Qualitative methods include case study, ethnography, and content analysis, which provide a rich understanding of complex interactions in an environment that is heavily context-dependent, such as an organization (Yin, 2003). Quantitative methods include descriptive research, like observation, correlation research, and survey research. Finally, with the most control over the environment, there are experimental methods which offer a great deal of internal validity, but lack generalizability (Leedy and Ormrod, 2005; Scandura and Williams, 2000). For these reasons, research and metrics are very similar. Collecting metrics to gain insight into an organization’s defensive information battlespace knowledge follows the same path as research, and relies on some of the same constraints and pitfalls. The good side is that there is a great deal of available literature and established practice on collecting good, rigorous data, analyzing it, and drawing conclusions.

As mentioned before, the methods should be chosen carefully to match a certain specific problem. The problem should be identified well in advance and independently of the data collection and methodology. The methodology should be decided on next, which is simply the strategy to go from having questions and hypotheses to collecting and analyzing data which answer those questions. Once the proper methodology is determined, the methodology’s procedures are carried out with an eye for attention to detail and a focus on maintaining rigor and repeatability.

In an organization, each type of data that is required to be collected is analogous to a data point in a research problem. What data is collected really does matter. If two pieces of data are put together, they may show a relationship, but it may be moderated or mediated by another relationship which is unseen to the investigator. In this case, any attempts to solve the problem will come up short because the true relationship between the data has not yet been discovered.

5.2 Primer on types of data

The field of research methods defines four different types of variables: nominal, ordinal, interval, and ratio. Nominal data can be categorized as discreet data, or bin-type data, where an item falls into one of several bins. Ordinal data is data which is arranged in a hierarchy, with some items having a greater value than other items. Interval data is like a temperature scale, which is calibrated to provide a measure of difference between different items, and ratio data is data based on an interval scale that has an absolute zero. Ratio data can be ranked, multiplied, and most easily analyzed. Nominal data provides the least information, while ratio data provides the most (McClave, et al, 2005; Leedy and Ormrod, 2005).

Variables exist as independent variables and dependent variables. With a relationship between the variables, a change in the independent variable makes a change in the dependent variable. Qualitative measurement is used to measure nominal or ordinal variables, where quantitative measurement is most appropriate for interval and ratio variables.

Data can also be continuous or discrete. Continuous data implies a measurement where discrete data implies a count or enumeration (McClave, et al., 2005). Statistics can be descriptive, which can tell you about an event, or inferential, which can help predict events (McClave, et al, 2005). These can also be seen as either “leading” and “lagging” indicators which demonstrate cause and effect, respectively (AF Logistics Management Agency, 1999).

The data is analyzed and drawn into conclusions based on either deductive or inductive logic. Finally, deductive logic starts with a premise and expands that to different instances, and inductive logic starts with specific instances, then attempts to draw a conclusion, which those instances would support (Leedy and Ormrod, 2005).

5.3 Up-channeling information

As data calls are requested from lower levels in the organization, the data is gathered at the lowest level and passed back up in a consolidated and filtered format. In the process of going from the lowest tactical level to the highest strategic level, the data is manipulated, aggregated, filtered, compressed, and cut in order to be digestible by a very small but busy audience. This process is sometimes referred to in organizational behavior textbooks as upward filtering (Gibson, et al., 2005).

There will be many more tactical units than operational or strategic ones and the tactical levels may (or may not) have lateral communication channels in place to aggregate or share what they have. In many cases, the operational level departments can be left with the task of defining how to operationalize strategic-level goals, objectives, and questions from the above, and will require the data from the tactical level in order to meet those demands. Personnel who take part in administrative communication processes can be religiously devoted to ensuring the process is satisfied, but they may not have insight into the actual use of the information once it is received at the highest levels of the organization.

Unfortunately, at every level of the organization where upward filtering can occur, there is a chance that the data may be changed in order to meet the needs of those who pass the information. But an important fact to realize about metrics is that they are very powerful. Many have been fired or promoted based upon the metrics of their department. If performance is defined in a certain way, and the manager fails to achieve the performance goals as specified in the collected metrics, then the manager is viewed as an underperformer and may be fired or demoted. With such pressure to do well, one can only expect that data will be manipulated as it travels up through the communication stream of the organization and is converted into information, and finally into actionable knowledge.

6. Conclusion

Information battlespace knowledge is the key to being able to operate and communicate decisions effectively in cyberspace. While knowing about opponents is important, it is also important to know about your own organization's capabilities and limitations. Defensive information battlespace knowledge is the knowledge about your organization's threats, vulnerabilities, and associated risks in operating in cyberspace. Security metrics collection and analysis are the typical forms of knowledge discovery in an organization.

In this paper, we evaluated a simple mental model for collecting and up-channeling security metrics information and showed that an organization can uncover knowledge about itself through automated metrics discovery and analysis, data mining, and through data calls. Data calls are subject to methodological issues in data collection and communication of findings, and we discussed some pitfalls associated with data gathering, including gathering the wrong data, gathering data for the wrong reasons, making data collection decisions at the wrong level, and using the wrong methodology to gather data. We also discussed the different types of data and the different characteristics they have. There are two main approaches in gathering security metrics data, top-down and bottom-up. We showed some of these approaches to using data and metrics to generate organizational knowledge and offered the idea of an organizational research project as an analogy of organizational data gathering. Finally, this paper presents the idea that security metrics are simply another form of organizational knowledge discovery and can be used in concert with other forms of knowledge discovery practiced in the organization.

7. Disclaimer

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the U.S. Government.

References

Air Force Logistics Management Agency (1999). *The Metrics Handbook for Maintenance Leaders*.

Blakley, B., McDermott, E., and Greer, D. (2001), *Information Security is Information Risk Management*. ACM Proceedings of the 2001 Workshop on New Security Paradigms, pp. 97-104.

Dobrzeniecki, A. (1994). Executive information systems. January. Retrieved 10 Nov, 2006, from http://www.cs.ui.ac.id/staf/sjarif/eis.htm

Earl, M. J., (2003). Experiences in strategic IS planning (Chapter 7) in *Strategic Information Management: Challenges and strategies in managing information systems*, 3rd ed. Oxford: Butterworth-Heinnman. Edited by Galliers, L. D. and Leidner, pp. 180-215

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). Knowledge discovery and data mining: towards a unifying framework. *Proceedings of the Second International Conference on Knowldege Discovery and Data Mining (KDD-96)*. Portland, Oregon, August 2-4, 1996.

Geer, D., Soo Hoo, K., and Jaquith, A. (2003). Why the future belongs to the quants. *IEEE Security & Privacy Magazine*. (1) 4. pp. 24-32.

Gibson, J. L., Ivancevich, J. M., Donnelly, J. H., Jr., and Konopaske, R. (2005). *Organizational Behavior*. 12th ed. Boston: McGraw-Hill.

Hong, K.S., Chi, Y.P., Chao, L.R., and Tang, J.H. (2003). An integrated system theory of information security management. Information Management & Computer Security. 11/5 [2003] 243-248.

Kerr, S. (1975). On the folly of rewarding A while hoping for B. *Academy of Management Journal.* (19) 4. pp. 769-783.

Leonard, D. and Swap, W. (2004). Deep smarts. *Harvard Business Review*. September.

McClave, J. T., Benson, P. G., and Sincich, T. (2005). *Statistics for Business and Economics*. 9th ed., Upper Saddle River, NJ: Prentice Hall.

Owens, B (2000). *Lifting the Fog of War*. New York: Farrar, Straus and Giroux

Papadopoullos, A. (2004). Information analysis and the content supply chain. Supplement to KMWorld, November/December.

Scandura, T. A. and Williams, E. A. (2000). Research methodology in management: current practices, trends, and implications for future research. *Academy of Management Journal*. (43) 6. pp. 1248-1264.

Shen-Hsieh, A., and Schindler, M. (2002). Data visualization for strategic decision making. American Institute of Graphic Arts. Experience Design Case Study Archive.

Smith, J. H., Keil, M., and DePledge, G. (2001). *Journal of Management Information Systems*. (18) 2. pp. 189-227.

Tuomi, I. (1999). Data is more than knowledge: implications of the reversed knowledge hierarchy for knowledge management and organizational memory. *Journal of Management Information Systems.* Winter 1999-2000. (16) 3. pp. 103-117.

Womack, J. P. and Jones, D. T (1996). *Lean Thinking*. *Banish Waste and Create Wealth in Your Organization*. New York: Simon & Schuster.

Yin, R. K. (2003). *Case Study Research: Design and Methods*. 3rd ed., Thousand Oaks: Sage Publications.

Security Risks in USAF Geospatial Information Sharing

Scott Bryant and Michael Grimaila
Air Force Institute of Technology, Fairborn, OH, USA
scott.bryant@afit.edu
michael.grimaila@afit.edu

Abstract: Technological advancements such as Geospatial Information Systems (GIS) and the Internet have made it easier and affordable to share information, thus enabling complex and time sensitive decisions to be made with higher confidence. However, the sharing of information also increases the likelihood that an adversary can gain illicit access to the information. Today's military leaders are faced with the challenge of deciding how to make geospatial information collected on military installations and organizations available to authorized communities of interest while simultaneously restricting access to protect operational security. In this paper, we explore the security implications of the US Air Force's applied Geospatial Information System (GeoBase) program. We examine the rapid expansion of the use of GeoBase to communities outside of the civil engineering field; examine the intrinsic and extrinsic security risks of the unconstrained sharing of geospatial information; and explore difficulties encountered when attempting to facilitate sharing geospatial information sharing while minimizing the associated operational risks.

Keywords: Geospatial information security, USAF geoBase, terrorism, targeting, information sharing

1. Introduction

Over the last decade, advancements in information technologies have dramatically reduced the costs involved with storing, managing and disseminating large amounts of data. These advancements have led to the development of Geospatial Information Systems (GIS) within the civil engineering community to share information with larger communities of interest, enabling complex decisions to be made more efficiently, with fewer resources, and at higher confidence levels. Military leaders face the challenge of deciding how to make their geospatial information readily accessible to authorized parties while mitigating the risks associated with information sharing. Unfortunately, many times these decisions are made without consideration of the underlying risks to critical information, infrastructure, and/or resources.

1.1 What is Geospatial information?

Geospatial information can be defined as any information which identifies the geographic location and characteristics of both natural and man-made earth-bound features (Zettler, 2002). The types of geospatial information range from specific latitude/longitude coordinates to a general description of where something is located. This information often takes on the form of maps, overhead images, datasets, websites, addresses (Baker et al., 2004). Geographic location is a key feature of 80-90% of all governmental data (Federal Geographic Data Committee, 2005). The Air Force has begun to use geospatial information to manage their installation infrastructure assets, for example, the locations of power distribution, water, sewer, telecommunications, and roads are stored within a database. Although geospatial information is diverse, it is still information, which is why it is important to understand the laws and policies that form the basis for rule sets used for managing both geospatial and non-geospatial information (Cullis, 2004).

2. Rapid expansion of the use of Geospatial information

As geospatial information systems have been rapidly expanding in the civilian sector, the USAF GeoBase program has witnessed tremendous growth as the need for minimizing fiscal waste and maximizing decision power has grown. This growth stemmed from the Air Force Civil Engineering (CE) community, whose need to provide accuracy, access, and accountability of installation assets drove an investment in the required equipment, skills, and additional data to form the geospatial information infrastructure called GeoBase.

Additional evidence of the expansion of geospatial information can be seen in the formation of the Defense Installation Spatial Data Infrastructure (DISDI) program in late 2004. This program was set up in the fall of 2004 to “organize the broad geospatial data investments found across the business mission area of the DoD’s Global Information Grid” (Cullis, 2005). DISDI’s successes in helping to provide focus to the entire DoD came as a result of seeing the need to reign in individual IT efforts across its massive organization. “DISDI is described today as a DoD mission capability comprised of those people, policies and practices necessary to acquire, steward and share “best available” installation and environmental geospatial data assets across the Global Information Grid—a system to provide users a seamless, secure, and interconnected information environment, for the real-time and near real-time needs of both the warfighter and

the business user" (Cullis, 2005). Through the collected efforts of DISDI and GIS communities, a common direction and enterprise solution was adopted. Today, there is an establishment of not only the Air Force's GeoBase program, but DoD GIS efforts that include the Navy's *GeoReadiness*, the Marines' *GeoFidelis*, whose and the Army's *GIS-R*, which is the Army's Installation Geographic Information & Services (IGI&S) program office. The expansion of program offices and mission related applications just within the last five years shows the need for accurate geospatial information technologies that provide critical information which enables decision makers, supports war fighters, aids planners, and increases overall situational awareness.

3. Identifying the security risks

As the military seeks to make more informed decisions based on information from geospatially related data, there are increasing concerns that this reliance may be exploited. To better understand the risk, the threats and vulnerabilities to the system must be understood. Solomon and Chapple define a vulnerability as "a weakness in a system that may be exploited to degrade or bypass standard security mechanisms" and a threat as "a set of external circumstances that allow a vulnerability to be exploited" (Solomon, Michael G., Chapple, Mike, 2005). When vulnerabilities and threats intersect, risk results as seen in Figure 1.

Figure 1: Identifying risks, (Solomon, Michael G., Chapple, Mike, 2005)

There are a wide variety of common computer threats from viruses, worms, Trojan horses, port scanning, file sharing attacks, Operating System attacks, scams, spamming, phishing, denial of services attacks, password guessing, backdoors, sweepers, sniffers, packet forge spoofing, IP spoofing, to the most obscure such as social engineering (Speed, Ellis, & Korper, 2002). However, concerns regarding the interception of data, theft / release of sensitive or confidential information, unauthorized access to privileged information, theft of other computer hardware or devices, system penetration by an outsider, laptop and hand held computer thefts, computer system and network abuse, and sabotage of sabotage of data or networks are also among the top breaches of security (Australian Institute of Criminology, 2006). Among these real threats, the greatest risk to the security policies is not the physical network, but rather the accountability of the people within the organization. By far, our human nature is our greatest danger. However, we can help mitigate this threat by proper infrastructure design, education, policy, and auditing. The Air Force must also consider the security of the system itself, the management of the database and its integrity, as well as the type and scope of access to the database. Protection of the availability of services and information is important to ensuring that users have access to the information when it is needed. The issue that geospatial information is available publicly is not the only security risk, in 2000 there were a reported 25,000 attempted intrusions into the defense system. Of those attacks, 245 of them were successful. Of this less than 1% of successful attacks, 96% of those were found to be preventable if users had followed established protocols (Onley, 2004). The network security which the GeoBase data relies is heavily monitored and network personnel continue to strive to improve network security by access controls and policies which conform to security best practice. Beyond the security of the network, let us discuss the primary issues that are more closely related to the risks of geospatial information and the capabilities that these systems provide.

Only recently has public access to information become a perceived concern (Tombs, 2005). Following the attacks on 9/11, almost overnight, federal officials became worried that some information is now "too public" and "agencies cut off access to thousands of documents on the Internet, ordered certain information in government libraries to be withheld or even destroyed, and simply stopped providing some information that used to be routinely released to the public" (Matthews, 2002). The concerns over providing a terrorist access to information which would help them develop or use a weapon of mass destruction lies at the heart of our fears. "Thus, digital maps are no longer available online from the National Imagery and Mapping Agency, CDs containing information on the nation's water supplies was ordered destroyed at depository libraries, and

tens of thousands of documents vanished from government Web sites. The information clampdown has touched off a sprawling debate over how much information should be — and legally can be — withheld from the public" (Matthews, 2002). Soon after, the National Imagery and Mapping Agency (NIMA), now the National Geospatial-Intelligence Agency (NGA), asked the RAND Corporation to assist in developing a framework to assess the security implications of publicly available geospatial information. This study remains to be among the few to address these specific types of implications and provides a broad base for future research. Their studies began to put into perspective the scope of federal geospatial information, finding it to be widespread across "465 programs, offices, or major initiatives at 30 different federal agencies and departments that make various types of geospatial information publicly accessible" (Baker et al., 2004). RAND concluded that very few (6% of the 629 datasets studied) appeared to be capable of fulfilling possible terrorist's needs. Even fewer sites (less than 1%) were found to provide critical information, both useful and unique, by their definition to potential terrorists. They also noted that in so many cases, since geospatial information exists in numerous ways, alternate forms of the same information existed readily in the public domain, beyond the control of federal sources" (Baker et al., 2004).

The level of risk that we are willing to take hinges on the values that we place on the following three strategic factors of information: data accuracy, access, and accountability (Schomper & et al., 1996). If we take these demands for geospatial information and evaluate the impacts of not having accurate information, timely access to information when it is needed, or fail in keeping people accountable in their responsibilities to maintain the information.

4. Top challenges

The goal of reducing the security risks and increasing the range of access across communities and knowledge seekers is not sought without challenge. The primary challenges discussed in this section are not unique to only the GeoBase program, but are challenges that in information management face when dealing with information security and information sharing. The first hurdle that must be overcome is in how we define, or don't define, the sensitivity of information. We'll take a look at issues surrounding the classification of information, sharing information, and the inconsistencies of policies and guidance and see how these add to the top challenges of creating a defensible geospatial information strategy.

4.1 Defining the sensitivity of information

Among the top challenges in the balance of information security and information sharing is defining the sensitivity of information. In times of fear, such as during the war on terrorism, new emphasis has been placed "on undefined 'potential' and 'possible' risks to 'sensitive' or 'critical infrastructure' in prohibiting public spatial data access. The concern over how the government chooses to define "sensitive but unclassified" information fuels hesitation to share information. Government watchdogs fear that a new sensitive information category could give agencies a way to hide embarrassing information from public scrutiny (Matthews, 2002). Another problem is that the sensitivity of information changes over time making any static sensitivity assessment dubious. A continuous litmus test must be made which can alter the decisions about access. These decisions affect not only the originating organization, but also the entire chain of users both up and down the information stream (Federal Geographic Data Committee, 2005).The classification of data has been the most effective way to manage the differences in the levels risks that certain information bears. Once the challenge of identifying what information is sensitive, there becomes multiple challenges in applying a designation to information. Overprotecting data severely hinders users that depend on using that information on a day-to-day basis in doing their job and accomplishing the mission. Restricting information has tremendous costs, not only in the added time and maintenance costs that it takes to manage that information, but also the expanded personnel safety risks. For example, electrical or natural gas distribution lines may be considered critical infrastructure to an installation, if classified as "SECRET", that would have tremendous impact on the personnel responsible for maintaining those lines, not to mention the safety hazards for construction crews getting ready to dig in an area where utility lines have not been identified to them. Limiting access to information may have greater risks associated. It is very important to select data protection measures that are commensurate to all the risks; in order to classify or restrict access to data; the risks must outweigh the benefits (United States Air Force, Air Mobility Command, 2005).

Air Force policy towards restricting access to geospatial data is the exception rather than the rule (Dunn, 2005). These restrictions "must be approved by the appropriate Headquarters Air Force (HAF) functional manager and must be based on public law, security classification or other DoD regulatory publication" (Dunn, 2005). Restrictions are to "only be applied to the data identified and not the entire system or collective group of data in which it resides or is produced" (Dunn, 2005). Problems with this type of policy reside in the

burden of management to track and communicate the necessary restrictions across multiple agencies and changing personnel. The challenge to data owners and stewards is to define what "sensitive information" is explicitly. The current guideline which is offered to the geospatial data community for identifying sensitive data, determining their risks, and assessing benefits is based on three factors: 1) risk to security, 2) uniqueness, and 3) net benefit of disseminating the data. These three factors were central to the Federal Geographic Data Committee (FGDC) as they developed a decision tree that is intended to act as a guideline to help organizations decide on what is reasonable access to sensitive data as shown in Figure 3 (Federal Geographic Data Committee, 2005). It remains important to launch discussions within organizations so that they can begin to ask the initial questions to evaluate the content of their information. Guidelines such as these help to provide an evaluation method that offers a hope for consistency.

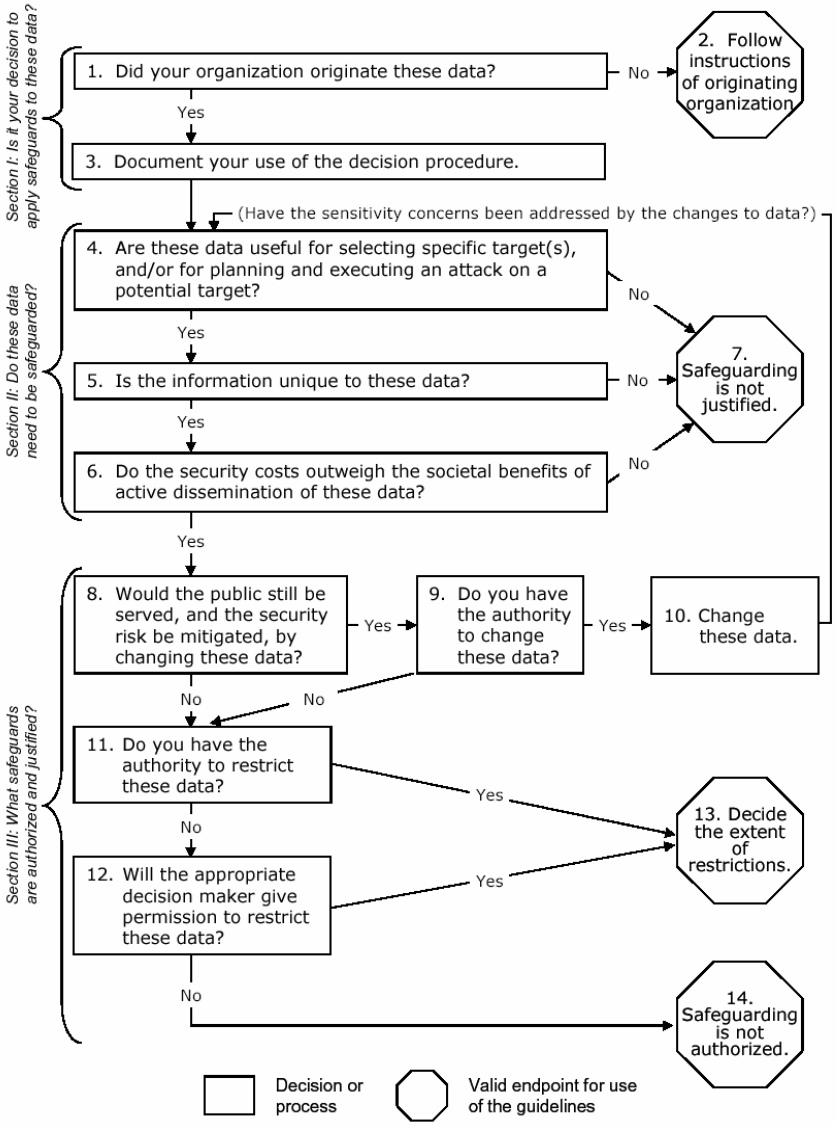


Figure 3: Decision tree for providing appropriate access to Geospatial data in response to security concerns (Federal geographic data committee, 2005).

The sensitivity of derived geospatial data offers additional challenges. Although attempts are being made, there are no established and validated mathematical formulas that can estimate the values of the sensitivity of geospatial data layers when combining or querying multiple datasets. The same thought process in determining the sensitivity of the original work should be applied each time data is extracted or combined to create a new way of looking at the information it provides (Federal Geographic Data Committee, 2005). Concerns over these derived geospatial datasets supplicate the trepidations of aggregating information. At what level of aggregation does information become sensitive? The GeoBase program recognizes that there are some instances where storing and providing access to aggregated data would constitute a vulnerability, but work diligently to provide protection (United States Air Force, Air Mobility Command, 2005). Currently,

each installation is responsible for performing periodic reviews on all datasets and combinations thereof to determine if they come together and constitute an unacceptable risk.

4.2 Information sharing

Although on smaller scale, the GeoBase offices have experienced the same type of challenges the Department of Homeland Security (DHS) is facing in encouraging organizations to voluntarily share information. There is a sense of hesitation and uncertainty among data owners to share information, perhaps over fear liability, embarrassment, or a fear of loosing power or control. Regardless, the importance of sharing is paramount to ensuring consistent, well-informed decisions are being made. Failing to provide information, leaves data users to pursue and use less reliable sources. The U.S. Government Accountability Office (GAO) captures the sentiment of many organizations in their March 2006 report entitled, *Information Sharing: The Federal Government Needs to Establish Policies and Processes for Sharing Terrorism Related and Sensitive but Unclassified Information*. Their report highlights the problems that offices such as the GeoBase office in the Air Force faces as an information broker, both a user and provider of geospatial information.

One of the biggest challenges is in the identification and designation of sensitive information. The GAO study found that over 26 federal agencies surveyed, there where 56 different sensitive but unclassified designations (Powner & Larence, 2006). Typically, GeoBase offices are faced primarily with the following three designations: Classified (SECRET), For Official Use Only (FOUO), and Unclassified. The challenges of managing classified information have been discussed, but we can quickly see how intertwined these challenges are and the need to overcome these hurdles in the quest of sharing information. For example, consider the abilities of an emergency responder or the Survival Recovery Center (SRC) to coordinate a safe cordon around a hazardous chemical spill without informative maps. If information is not shared and available for the people who need it to respond to emergencies or make command decisions, we have failed to secure ourselves by giving the most to the situation we possibly could. Geospatial information provides the security of knowing that the people making decisions have the tools they need to ensure our safety. A challenge of the GeoBase community is getting other organizations to understand the benefits of sharing information and realizing the power of a collective information bank. If information owners had a negative experience or that the collective system did not satisfy their requirements, the willingness to share was gone, thus the potential to leverage their information against other's vanished. Most concerns associated with sharing information were related to the ability to ensure their information would be protected with at least the same level of effort (Powner & Larence, 2006). A post 9/11 GAO report to the Secretary of Homeland Security in August of 2003 on efforts to improve information sharing studied ten barriers that were perceived as a hindrance to the information sharing process. Figure 4 highlights the ten barriers studied in the GAO survey and consolidates the average response of 16 federal agencies, 40 state agencies, 106 large cities, and 122 small cities to give an average percentage of perceived factors that hinder information sharing (Decker & Lepore, 2003).

Although the GAO report identified the lack of integrated database capability as the only significant barrier, it is interesting to note that the top six out of the ten barriers studied all have to do with the security of information (annotated with *) as shown below in Table 1.

Table 1: Top ten perceived barriers to sharing information (Decker & Lepore, 2003)

Rank	Perceived Barrier to Information Sharing
1	Difficulty with provision to secure, maintain, and destroy information*
2	Lack of clearances*
3	Lack of integrated databases*
4	Legal Barriers*
5	Concerns of disclosing sources and methods*
6	Lack of confidence in ability to limit disclosure of information*
7	Concerns about jeopardizing ongoing investigations
8	Authorities lack interest in Information to be provided
9	Culture of "information superiority"
10	Lack of confidence in ability to manage investigations*

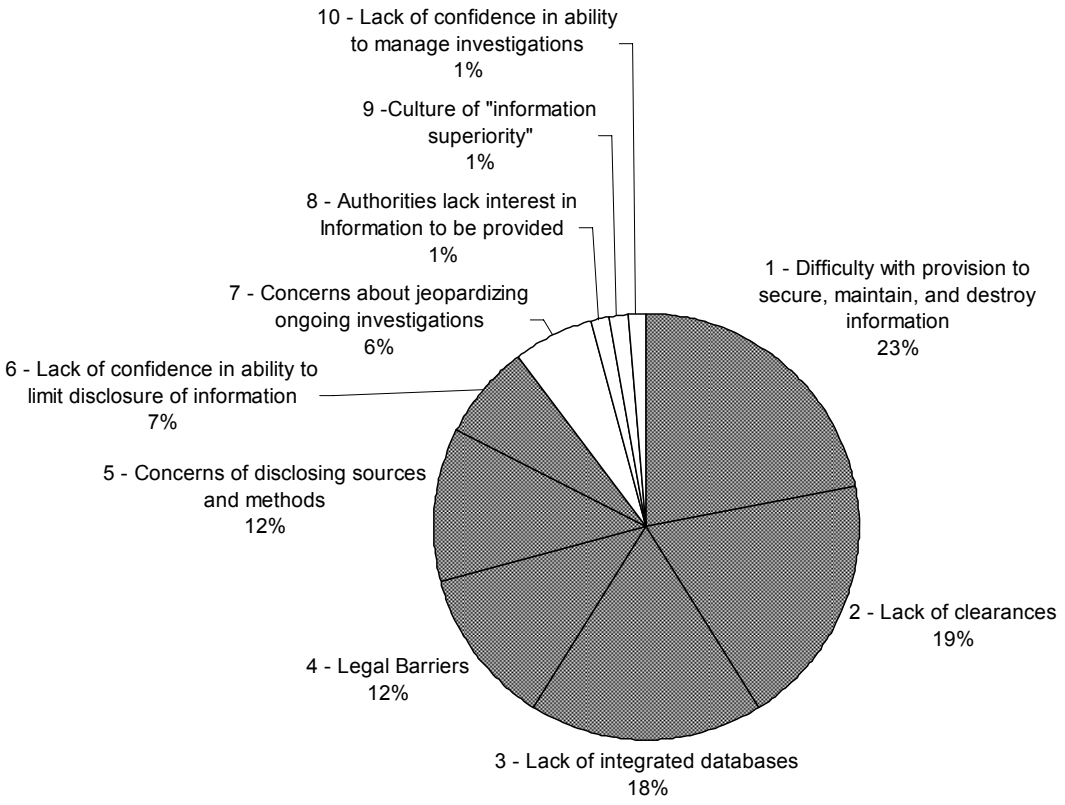


Figure 4: Perceived barriers preventing federal agencies from sharing information (Decker & Lepore, 2003)

These emphasize the problems with inconsistencies and different expectations of information sharing and information security between organizations. Overcoming barriers such as these will continue to test programs such as the USAF GeoBase program which encounters similar challenges when working with internal organizations, joint services, local municipalities, civilian employees, and private contractors. Each of these groups is representative of the mission and need pieces of the information the others have to share. Determining ways to know who it is appropriate to share information with, defining their “need to know” and how the information will be used are among the difficulties in establishing consistent procedures.

The development of multi-level agreements which would help users understand the responsibilities and the organization to understand what information should be given out and to whom. A notable example is when the Air Force partners with commercial entities, such as engineering design and construction firms, which require data for contract execution. Often times, sharing data between services such as the Air Force and the Army, causes its own set of problems, creating duplicate datasets between services is against the “one installation, one map” motto. Serving overseas and working with a host nation has created difficulties in reaching agreements. Not only are there multiple systems which require careful attention to detail on what can and can not be shared. There is some data which both sides need to be able to share and in some situations, this is not happening. Once the data is shared; there are very few controls that remain in place. Some of the major architectural-engineering firms have files and files of critical geospatial information in their project files and reference libraries. Although there are signed agreements, disclaimers, and consents which are given at the time of data conveyance, the reality of the business process is that the government just has to trust that others understand the costs to security.

4.3 Inconsistent policies and guidance

The inconsistency of policies and guidance which drive business process continue to add complication to the information security challenge. Numerous existing studies, policies, instructions, guidance, recommendations, and directives have been issued at nearly every level of the Department of Defense (DoD). For the longest time, the guidance was deferred to each installation commander or relied on existing vague guidance from other communities such as public affairs, operations security, or communications. No one policy or guidance lends assurance to geospatial data security and information sharing, each is interwoven and at times leaves room for interpretation or are contradictory. Even within the Air Force, the

major commands, wings and squadrons differences exist in the way these documents are interpreted. Many installations have developed their own local policies to address their needs. Now, multiply these differences every time organizational leadership boundaries are crossed or as leadership within organizations change.

Further problems exist beyond the initial guidelines. For example, whose authority is it to change or restrict data? Is it different for each data layer? Who is to say that the data is useful for planning and executing an attack? The yes / no decisions are not as simple as the decision tree presents. Until organizations have a mutual level of understanding on how to make the complex qualitative decisions required for safeguarding information, there will continue to be added challenges.

5. Moving beyond the problem

There are significant costs, not only financial, but serious mission degradation and effectiveness are at stake. The first step to moving beyond the problem is establishing and investing in the appropriate business processes to identify the sensitivity of information both on its own and combined with other information. Today's solution of referring to historical documents, policies, and processes and specific requests by the data steward or data layer owner has been a good start, but as the information grows in value and becomes more easily accessible through new technologies, these old decisions must be rethought as they do not completely address the issues of today's information situation. Just as the GeoBase service has helped to overcome the stove-piped mentality on the map between functional areas through the common vision of "One Base, One Map", similar hurdles must still be over come in other areas of the organization. No one particular knowledge set will be able to solve this interwoven problem on its own. We need the expertise of many essential communities such as Information security (Infosec), Operational security (Opsec), Communications security (Commsec), the GeoBase Integration Office (GIO), and the data owners to come together to weave a balanced approach. There is a lot of work ahead, just as constantly as our environment changes, so to must we be willing to adapt. People will continue to find the information they need. If they do not have quick access to the information, then they are bound to either make an uniformed decision or spend money or time to recollect. We must keep in mind is that these efforts are not just about sharing the data itself, but building the awareness that the information exists. The quest for information security must not limit data discoverability and the ability to if not share the information, share that it exists and where it may be found. By actively managing geospatial information and the knowledge it brings we can more effectively identify and build the processes and have the best of all three worlds: accuracy, accountability, and access.

6. Conclusions

Increased use of electronic data sharing denotes a greater chance of information misuse, both inside and outside the Air Force. As the repository of critical information builds, GeoBase information will face a heightened risk of being targeted through cyber terror attacks. The security implications of the USAF GeoBase program are but a subset of a growing national dilemma which plagues academics and practitioners.

We have seen the inevitable swinging of the pendulum from full access to the desire for tightly regulated and restricted information security laws, policies, and procedures. It is imperative that common and explicit guidelines are developed and implemented throughout the USAF and the DoD. There is a need in the Air Force to establish a tacit understanding that security of information is important and that the costs involved with not maintaining security standards are intolerable based on the understanding of the risk. If this type of mindset does not exist, then the entire organization will continue to remain at risk and experience mission degradation, reduced productivity, lost data, revealed military secrets or compromised integrity. Motivating airmen and civilian partners to realize these risks and prepare them to treat geospatial data they handle as if it is their own banking information will go a long way in protecting the missions they serve. Without common standards to dictate minimum security requirements and practices, bases will be left to develop and implement their own security standards. As a whole, the organization is only as strong as the weakest link.

Much research is still needed to understand how to find balance between information security and the need to share data. A greater understanding of the technical side of computer security and the growing threats in cyberspace combined with the knowledge of what data providers know about what they provide in their functional area will help more efficiently and securely align the GeoBase workflows with Air Force Business processes needed for progress. As we become more connected across functional areas and between services, the more important it become to coordinate our actions and collectively protect the valuable information which protects us. Not only is it imperative to develop policies and procedures for sharing data, but it is incumbent upon each community to educate itself on the information that exists today. Information

security cannot limit data discovery; rather, it should encourage one's self to illuminate new data/information while providing the necessary security blanket that the discoveries will remain in the hands of a safe user community.

Disclaimer

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the U.S. Government.

References

Australian Institute of Criminology 2006. (2006). *Australian crime: Facts and figures 2005.* Canberra: Australian Institute of Criminology. Retrieved 15 Nov 06, from http://www.aic.gov.au/stats/crime/cybercrime.html

Baker, J. C., Lachman, B., Frelinger, D., O'Connell, K. M., Hou, A. C., & Tseng, M. S., et al. (2004). *Mapping the risks: Assessing the homeland security implications of publicly available geospatial information.* RAND National Defense Research Institute.

Cullis, B. J. (Jul 26, 2005), Geospatial mandates pave way for DISDI. [Electronic version]. *Military Geospatial Technology, 3*(2)

Cullis, B. J. (2004). *Evolving federal protocols for safeguarding geospatial information*. Paper presented at the *GeoBase Compass Conference,* San Antonio, TX.

Homeland security: Efforts to improve information sharing need to be strengthened. Department of Homeland Security, U.S. Government Accountability Office, (Aug 2003).

Dunn, G. L. (7 Feb 2005). *Installations and logistics (IL) data access policy*. Pentagon, Washington D.C.: HQ USAF/IL-CIO.

Federal Geographic Data Committee. (Jun 2005). *Guidelines for providing appropriate access to geospatial data in response to security concerns.* National Spatial Data Infrastructure (NSDI).

Federal Geographic Data Committee. (27 Jul 2005). *Homeland security and geographic information systems: How GIS and mapping technology can save lives and protect property in post-September 11th America.* National Spatial Data Infrastructure (NSDI).

Matthews, W. (2002, 16 Sep 2002). OMB weighs info classification: Efforts to protect some public data from misuse are generating a mix of reactions. *Federal Computer Week,* 14-15.

Moseley, T. M., & Wayne, M. W. (7 Dec 2005). *SECAF/CSAF letter to airmen: Mission statement*. Washington D.C.: SECAF/CSAF from http://www.af.mil/pressreleases/release.asp?storyID=123013463

Onley, D. S. (25 April 2004). DISA official: Users should be accountable for security. [Electronic version]. *Government Computer News, 1*(1)

Powner, D., & Larence, E. (Mar 2006). *Information sharing: The federal government needs to establish policies and processes for sharing terrorism-related and sensitive but unclassified information.* (GAO Report to Congressional Requesters No. GAO-06-385). Washington D.C.: U.S. Government Accountability Office.

Schomper, C. R., & et al. (1996). *Realizing the potential of information resources: Information, technology, and services. track 2: Policies and standards.* (Reports - Descriptive; Speeches/Meeting Papers No. ED392336). Boulder, CO: CAUSE Exchange Library.

Solomon, Michael G., Chapple, Mike. (2005). *Information security illuminated.* Sudbury, MA: Jones and Bertlett Publishers.

Speed, T., Ellis, J., & Korper, S. (2002). *The personal internet security guidebook: Keeping hackers and crackers out of your home.* Academic Press.

Tombs, B. R. (2005). Policy review: Blocking public geospatial data access is not only a homeland security risk. [Electronic version]. *Urban and Regional Information Systems Association Journal, 16*(No. 2)

United States Air Force, Air Mobility Command. (15 Sep 2005). *Interim AMC GeoBase information security policy.*

Zettler, M. E. (2002). *USAF installation geospatial information security policy [draft policy memorandum].* Washington D.C.

Performance Impact of Connectivity Restrictions and Increased Vulnerability Presence on Automated Attack Graph Generation

James Cullum, Cynthia Irvine and Tim Levin
Naval Postgraduate School, Monterey, CA, USA
jcullum@nps.edu
irvine@nps.edu

Abstract: The current generation of network vulnerability detection software uses databases of known vulnerabilities and scans target networks for these weaknesses. The results can be voluminous and difficult to assess. Thus, the success of this technology has created a need for software to aid in network vulnerability analysis. Although research has shown the effectiveness of automated attack graph generation tools in displaying potential attack paths in a network, research involving the performance of these tools has been limited. The performance impact of connectivity restrictions and the number of vulnerabilities present on a network for these tools is not well understood. Using empirical testing, we have collected quantitative data using CAULDRON, an attack graph generation tool developed at George Mason University, on a collection of simulated networks defined to modulate connectivity at certain points in our networks and represent the number of vulnerabilities present per node. By defining our model to include sets of nodes, which allow connectivity from all nodes to all vulnerable nodes in the set; the number of nodes present in each set, the number of connections between sets; and the number of vulnerabilities per node as our variables, we are able to observe the performance impact on CAULDRON of both connectivity restrictions and the increased presence of vulnerabilities in our networks. The effect of these variables on processing time and memory usage is presented and can be used as a metric to assess the scalability of this tool within various customer environments.

Keywords: Attack graph, network, exploits, vulnerability analysis, performance.

1. Introduction

Automated attack graph generation tools use a database of known vulnerabilities, and input from network vulnerability scanners, to construct *attack graphs* comprising the potential paths through the network from an attacker to a target. The attack graphs are used to better understand the large volume of data produced by network vulnerability detection software. Research involving the performance of automated attack graph generation software has been limited. For example, the performance impact of connectivity restrictions and the number of vulnerabilities present on a network for these tools has not been well understood. We present the results of performance tests of the Combinatorial Analysis Utilizing Logical Dependencies Residing on Networks (CAULDRON) tool (Jajodia 2003), an attack graph generator developed at George Mason University. The tests are based on simulated networks in which the several performance-dependent factors are varied, including: the degree of internal connectivity between subnets (i.e., subsets of network nodes, or *node sets*); the number of vulnerabilities present per node; and the number of nodes in each subnet. The effect of these variables on processing time and memory usage is summarized, which we hope will provide a useful metric to assess the scalability of CAULDRON and similar tools within various customer environments, as well as provide useful feedback to its developers. Our analysis shows that CAULDRON's processing time and memory usage were directly responsive to several of the independent variables. Processing and memory use increased polynomially as the number of nodes in the network increased, as did processing with respect to the number of vulnerabilities per node. Processing time scaled linearly as the number of fully-connected nodes that were reachable from the attacking node increased. Other factors scaled linearly, or had minimal effect. This indicates that CAULDRON has the potential to scale well in large enterprise IT environments.

2. Background and related work

2.1 Cauldron

CAULDRON is an ongoing project developed and supported by George Mason University that uses network vulnerability scan reports to automatically generate attack graphs depicting all known combinations of vulnerabilities that could be systematically exploited by an attacker to reach an attack goal (Jajodia 2003). The version of CAULDRON examined in this paper and used during experimentation is version 2.6.

While there are currently many network vulnerability scanners that can detect the presence of vulnerabilities that exist on individual systems, in many of these tools vulnerabilities are considered in isolation (Ritchey

2002). By automating the task of analysis and by considering the vulnerabilities in combination rather than in isolation, CAULDRON's approach is to depict attack graphs showing how an attacker could systematically compromise a network in order to reach his attack goal (Noel 2002)(Jajodia 2003). CAULDRON also contains additional functionality which allows it to correlate Intrusion Detection System (IDS) events with attack graphs. These correlated results can be used to better distinguish attacks in progress from false alarms, and also provide context to attacks in order to determine where a response should be directed (Noel 2004), although this feature was not analyzed in the current work.

2.2 Cauldron abstract model

This section describes the abstract model underlying production of the attack graph (see Figure 1).

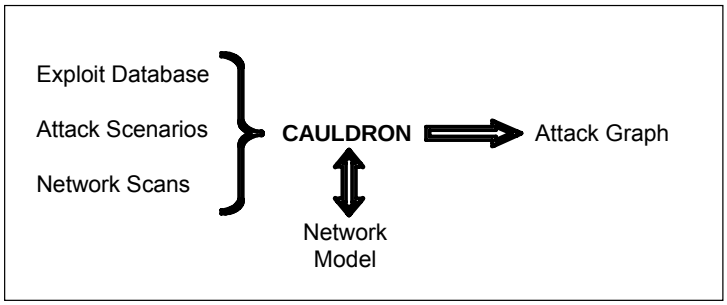


Figure 1: Cauldron engine inputs and output

2.2.1 Exploit Database

As depicted in Figure 1, a database of exploit models is input to CAULDRON. The exploit model is an abstract representation of the pre-conditions necessary for a single vulnerability to be exploited and the post-conditions that result after the exploit has been run. An exploit's pre-conditions define the required vulnerability, connection to that vulnerability, and user privileges that must be present prior to execution of an exploit (Ritchey 2002). Post-conditions reflect the state of the network after the exploit, in which the attacker may be able to obtain more information about the network, and/or obtain elevated user rights (Ritchey 2002). Thus, it is possible to represent how an attacker could chain the post-conditions of one exploit model to the pre-conditions of another exploit model until an attack goal is met. An initial database of exploit models is provided with the CAULDRON tool, which is updated regularly by researchers at GMU, and can also be customized by the end user to include additional knowledge.

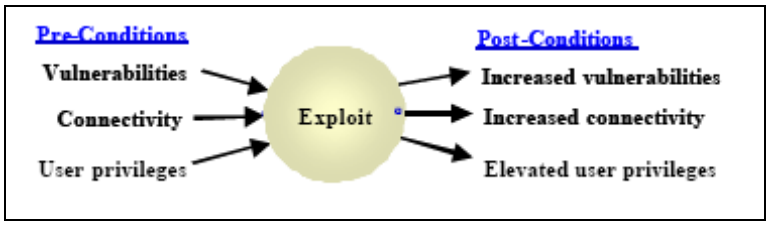


Figure 2: Exploit Model (Ritchey 2002)

2.2.2 Network scans

Another input to CAULDRON is network vulnerability scan data of the target network. If planned correctly, these scans can also show connectivity rules being enforced by filtering routers or firewalls (Ritchey 2002). By conducting a vulnerability scan through the firewall as well as behind the firewall, as depicted in Figure 3 and Figure 4, it is possible to model the attacker's knowledge of the network prior to the initial attack, as well as any knowledge gained after hosts residing behind the firewall have been compromised.

Vulnerability reports, which must be in XML format, are preprocessed by CAULDRON using CAULDRON's GUI to create a *network model* containing vulnerability and connectivity information.of nodes reporting vulnerabilities. Network hardening and other changes to the network model can be made by making modifications to the model's XML code as directed by the CAULDRON customization manual.

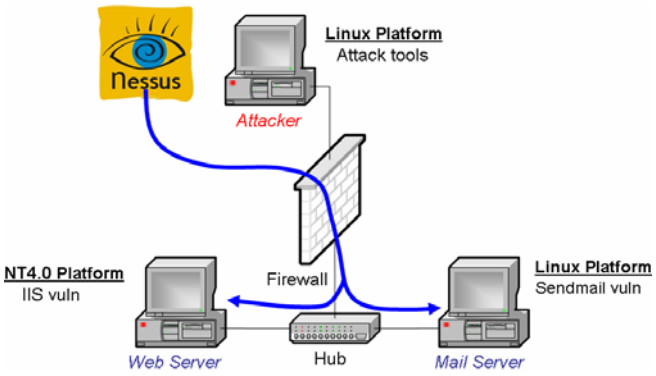


Figure 3: Attacker's initial knowledge of the network

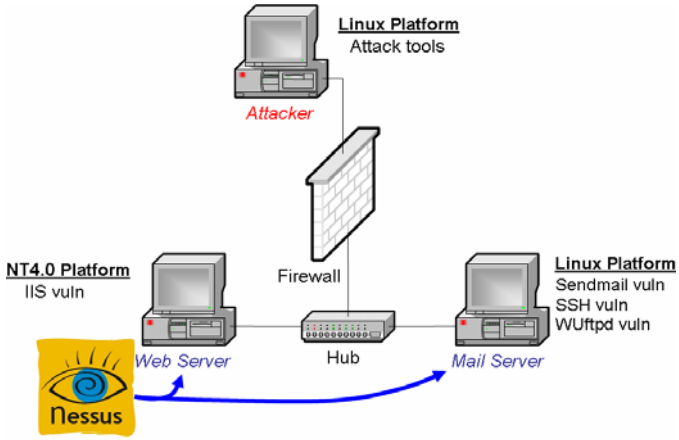


Figure 4: Attacker's knowledge after a host has been compromised

2.2.3 Attack scenario

The attack scenario is provided by the CAULDRON "user" and identifies the attacker and target nodes as well as the initial capabilities of the attacker and desired capabilities of the target. Capabilities of both the attacker and the target are represented in terms of "access level" and user privilege on a given node. The *access level* represents the ability to transfer files and execute code. *User privilege* represents whether or not the user has super user, or root level, access to his machine. The attack scenario is defined manually within the CAULDRON graphical user interface (GUI) using a predefined list of user privileges and access levels to and from nodes included in the network model, and can be edited using the GUI or by making changes directly to the XML code by following instructions provided in the customization manual.

2.2.4 Attack graph

Given the inputs described, CAULDRON generates and displays an attack graph. An attack graph depicts all paths between the attacker and the target (per the attach scenario) that are known to the exploit model. The vertices of the attack graph represent security conditions and exploits and are connected by edges, which represent dependencies (Jajodia 2003). CAULDRON can automatically aggregate certain attack information to produce a simplified attack graph, providing the user with a less complex visual representation of the attack graph. Figure 5 depicts an aggregated attack graph.

2.3 Previous Cauldron Performance Measurements

In research that led to the development of CAULDRON, an initial scalability experiment was conducted by GMU to test the scalability of combinatorial analysis using a model checker as the underlying analysis engine and a fully connected network with an increasing number of nodes on the network as a scaling parameter (Noel 2002). Although combinatorial analysis is still the underlying analysis approach that is used, the analysis engine is now CAULDRON rather than a model checker. Results from the original scalability experiment showed performance scaled in polynomial time, but that results could not be computed for networks containing more than 50 nodes using 512 MB of physical memory.

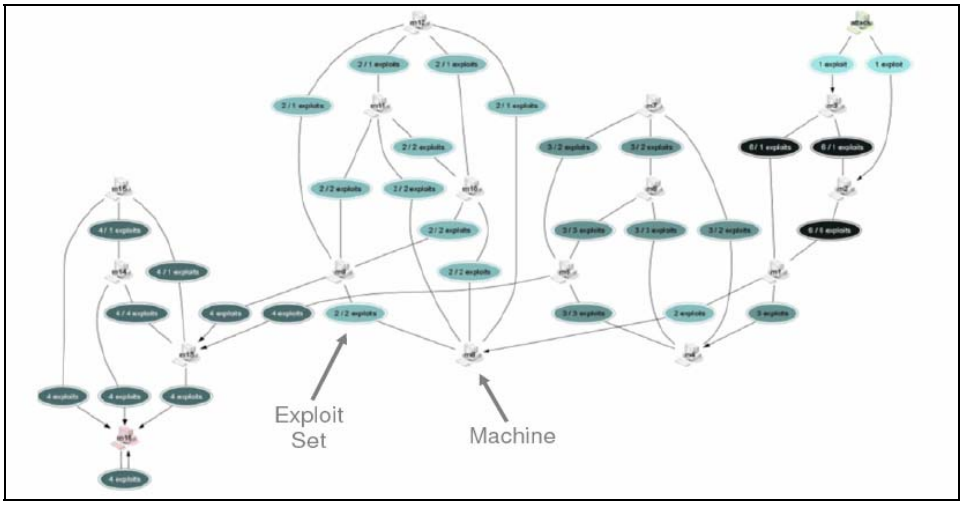


Figure 5: Aggregated attack graph (Jajodia 2004)

2.4 Other automated attack graph generation tools

In addition to ongoing work being done on CAULDRON, there are several other independent efforts underway to develop tools that can automatically generate attack graphs.

2.4.1 CMU - Attack graph toolkit

The *attack graph toolkit* from Carnegie Mellon University uses a *model checker* to generate attack graphs that match an attacker's capabilities with respect to the current state of the network. This system is similar to CAULDRON in that it accepts Nessus vulnerability reports as input in order to gain information about the network's topology and network vulnerabilities, and it also requires the manual creation of exploit models based on vulnerability gathering services. It differs in that it can interface with host based programs such as MITRE Corporation's Outpost and Lockheed Martin's ANGI systems to display vulnerabilities which might not be externally visible from the network, and can (attempt to) identify all possible attack paths to the target node rather than paths from one particular attacker (Sheyner 2004).

2.4.2 Lincoln laboratory – NetSPA

Network Security Planning Architecture (NetSPA) from Lincoln Laboratory generates attack graphs that show all possible paths from a particular attacker to all reachable nodes; in contrast, CAULDRON performs this step and performs further analysis to eliminate all reachable nodes that cannot reach the goal node. NetSPA uses the same general types of inputs as CAULDRON. The developers plan to incorporate the connectivity rules generated from imported firewall and router rules, but this feature has not yet been implemented. NetSPA also avoids the manual entry of an exploit database, such as is required by CAULDRON, by gathering this information from vulnerability descriptions used by Nessus, the Mitre CVE dictionary, and the NIST ICAT database. Lincoln Laboratory, reports favorable performance results (Ingols 2005).

2.4.3 Skybox - Skybox view

Skybox View is a commercially available tool developed by Skybox Security that can automatically generate attack graphs through the use of host-based agents, management interfaces, and an analysis server located on the target network (Skybox 2006). This product is similar to CAULDRON in that it requires a database of exploits and probing prior to analysis to discover which vulnerabilities are present on each host. It differs from CAULDRON because it analyzes live networks. While CAULDRON relies on data collected by network vulnerability scanners, this has the potential security advantage of being air-gapped from the target network. Although no performance details were available from Skybox, an examination of recent patents submitted by Skybox identified the algorithmic complexity of the product as n^4, where n represents the number of nodes present on the network (Lippmann 2005).

3. Experiment

Our hypothesis was that CAULDRON is efficient enough that it could be the basis for a product suitable for use in large-scale industrial (e.g., enterprise) IT environments. In this section, we describe the experiment to measure how changes to the inputs, such as degree of network connectivity and the number of vulnerabilities present per node, effect the time required to produce attack graphs, and the resources required to do so. If these effects are found to be reasonable, e.g., its processing is not on the order of n^p, then we can conclude that CAULDRON is at least a candidate for use in large IT environments.

3.1 Performance measurements

Two primary effects, or *dependent variables*, that could potentially limit the use of CAULDRON are its processing time and memory usage during the analysis of the input data. Processing measurements were recorded for the following phases of CAULDRON's execution:

- Total runtime – reading input files, performing attack graph analysis, and writing the results to the attack graph output file.
- Analysis runtime – running CAULDRON's attack graph analysis algorithm.
- Initialization runtime – reading input files, and determining which exploits were applicable to the network model.
- Forward Chaining Algorithm – running the first half of CAULDRON's attack graph analysis algorithm.
- Backward Chaining Algorithm – running the second half of CAULDRON's attack graph analysis algorithm.

We measured the maximum amount of physical memory in use by CAULDRON per test by measuring the maximum size of CAULDRON's *working set*. The working set is the smallest set of memory referenced by a process for efficient execution (Denning 1968). The test system was fitted with enough memory (2GB) so that performance issues related to exhaustion of memory were not noticed.

We also measured the maximum amount of virtual memory used by CAULDRON per each test.

3.2 High-level experiment description

The experiment was divided into five separate phases, with each phase testing the effects of different characteristics of the inputs, which comprise the experiment's *independent variables*: heap size, number of network nodes, degree of connectivity, vulnerabilities per node, and the size of the exploit database. Additionally, since minor instrumentation of CAULDRON was required, tests were performed to ensure that the modifications did not affect the test results.

3.2.1 Heap size

Since CAULDRON is implemented in Java, all live memory objects are stored in Java's memory *heap*. The Java Virtual Machine (JVM) *initial* and *maximum heap size* are established in CAULDRON's configuration file, and remain constant during runtime. Setting the initial and maximum heap size to the same value is said to increase a process's predictability by freeing heap-size management from the chores of the virtual machine (Sun 2003). In this phase the initial and maximum heap sizes were set equal to each other and gradually incremented from 128MB to 1GB during subsequent tests, leaving other variables constant. Before proceeding to the next phase of the experiment, the results were analyzed, and the heap size that exhibited the most stability was selected for use for the remainder of the experiment.

3.2.2 Number of network nodes

The second phase of the experiment measured how performance of CAULDRON changed when the number of nodes varied, and other variables were constant. Full network connectivity between nodes was configured. Each node contained in the network had a single vulnerability and a corresponding exploit associated with it. The tests included networks that ranged from 2 to 100 nodes and each test was repeated five times in order to verify that results were consistent.

3.2.3 Degree of connectivity

The third phase of the experiment examined how performance changed when the connectivity between nodes varied. In order to modulate connectivity, three variables were utilized: the number of fully connected *node-sets* (i.e., subnets) in the network; the number of nodes per fully connected node-set; and the number

of connections between each pair of fully connected node-sets. Selection of specific connections was randomized.

We define a *fully connected node-set* to be a collection of nodes in which every node has at least one vulnerable service, every node is connected to every other node, and access to vulnerable services is allowed. Multiple fully connected node-sets modelled networks in which connectivity between subnets is restricted, e.g., due to the presence of firewalls or filtering routers. A connection between fully connected node-sets allows a node belonging to one node-set to exploit a vulnerable service on a node belonging to another node-set, if it has access to any node in that node-set. This provided a simple way to define different degrees of connectivity. Although it was possible to model the size of each node-set individually, which would provide a more realistic network model, the effort to analyze the combinations of different sized node-sets was not considered worth the potential information to be gained, so a uniform size was used.

Multiple tests were conducted for each number of connections between node-sets, so that different connection pairs were used each time, lowering the impact the characteristics of specific connection pairs would have on the performance measurements.

3.2.4 Vulnerabilities per node

The fourth phase of the experiment measured how the number of vulnerabilities per node affected performance of CAULDRON. As with the size of node-sets for a given test (above), the number of vulnerabilities per node was uniform for all nodes for a given test. Each test was repeated five times in order to verify the accuracy of results due to expected performance variability.

3.2.5 Size of the exploit database

The final phase of the experiment measured how performance was affected by the size of the exploit database. Exploits were incrementally added that would not appear on the attack graph, but nevertheless would cause the database search space to increase.

3.2.6 CAULDRON baseline

To measure the impact that modifications to CAULDRON had on performance, additional experimentation was conducted using the modified version of CAULDRON and the unmodified version of CAULDRON on an identical network, attack scenario, and exploit database.

3.3 Implementation

The experiment was conducted on a standalone Dell Dimension 5150 3.2 GHz personal computer containing 2GB of memory and running the Microsoft Windows XP SP2 operating system. The experiment utilized simulated network data, rather than data from a real-world production network, or from a laboratory network. By simulating the experimental networks, it was possible to avoid issues that might arise such as revealing a real-world network's vulnerabilities to the public, or spending a lot of time to configure hardware and software, and validate that a laboratory network was setup correctly. Similarly, simulated exploit models were used. An experimental network was created to scale each variable contained in Section 4 independent of one another. By combining the results of each experiment, it is expected that the reader can infer the results of a multiple variable scenario.

Various software tools were required. Cygwin (Cygwin 2006), a Linux-like environment for Windows was used to provide access to string processing tools such as *awk*, *grep*, and *sed*, along and the *bash* script environment. Bash scripts were created to generate the input files, to automate each phase of the experiment, and to collect results. The Windows Performance Monitor was used to measure the peak working set, and the amount of virtual memory allocated exclusively to CAULDRON. To automate the experiment and collect additional performance data, modifications to the CAULDRON source code were made such that the program would accept command line input, terminate after analyzing the attack graph, collect and report the amount of time required to complete the different phases, and record the process ID of CAULDRON.

4. Results and analysis

The nominal test parameters, when not varied for a particular test, were as follows:

- An exploit database containing 1500 entries

- A fully connected network model that consisted of 60 nodes (one node-set containing 60 nodes)
- Each node possessed a single vulnerability that could be exploited by a unique exploit model contained in the exploit database
- The attack scenario was created such that the first node was the attacker and the last node was the victim.
- Each test is repeated five times to average out background noise.

4.1 Heap size

The heap size was changed in 128MB increments from 128MB to 1024MB. A heap size of 768MB was selected for the remainder of the tests since its resource usage was generally most stable: its variability was lowest in memory measurements and was low for time measurements as well.

4.2 Network size

The experiment was conducted using network models constructed with a variable number of nodes ranging from 2-100 nodes.

4.2.1 Observations

The number of nodes present on a network appears to play a significant role in the performance of CAULDRON, and the forward pass of the analysis algorithm appears to be the most significant factor in processing time.

Curve fitting analysis of the plots of the analysis time, total time, and forward pass time measurements showed that each measurement scaled with the number of nodes in approximately the same way. The amount of time to complete the forward pass algorithm is the dominant factor in analysis of the network model. A fourth order polynomial was the closest fit to the data, but the 3^{rd} and 4^{th} order terms are quite small – and the data is close quadratic. Figures A-1, A-2, and A-3 show how the analysis, total, and forward pass time measurements scaled as well as the fit equations that were used.

Curve fitting the backward pass algorithm data revealed that a third order polynomial provided a reasonable fit for the data obtained, but again the higher order terms are very small (see Figure A-4). The actual time required to run the backward pass algorithm was negligible in comparison to the forward pass algorithm. As shown in Figure A-5, the amount of time required to complete the initialization phase appears to scale linearly as the number of nodes are increased. Using the fit equation, which was calculated after removing the one outlier, it was observed that it would require at least 250 nodes to raise the initialization time by one second.

Analysis of the data collected for private memory storage indicates that the number of private bytes used does not scale as nodes are increased. Although a curve fit of the data suggested that private bytes scaled linearly, the slope of the line was shallow enough to cast doubt on whether virtual memory requirements scaled at all. Either way, it appears that the impact of adding nodes to the network is negligible in terms of private memory storage allocated to the CAULDRON process. As the number of nodes increase on the network, a larger working set was requested by CAULDRON (see Figure A-6). The size of the working set requested by CAULDRON appears to scale as a third order polynomial.

4.3 Connectivity

This experiment was designed to simulate the presence of firewalls, filtering routers, or any other network device that imposes connectivity restrictions on a real-world network. This experiment allowed the creation of hierarchical networks, rather than flat, or fully-connected networks, which were used in other experiments. In this test, the number of node-sets and the number of nodes per node-sets were varied. Connections between node-sets were generated at random, and multiple networks were tested in an attempt to identify any individual networks that impacted performance.

4.3.1 Observations

Analysis time is influenced by the number of nodes that are reachable from the attacking node, and the number of connections that are present in the network model. Although connectivity did not affect the amount of memory required, it affects analysis time, mostly during the forward pass algorithm.

The data shows that the size of the node-set, the maximum distance between nodes, and the number of connections have the largest effect on *overall performance*. The size of the node-set has the largest impact on the amount of time required to perform analysis.

Analysis runtime depends on the number of nodes that are reachable from the attacking node, and the number of connections present in the network model. Stratifications in the amount of time required to perform analysis were observed depending primarily on the number of node-sets that were reachable from the node-set containing the attacker. The amount of time required to perform the backward pass algorithm was small for all tests that were run. The time required to perform initialization, the amount of private bytes used, and size of the working set requested by CAULDRON did not appear to be affected by changes to connectivity allowed on the network.

4.4 Vulnerabilities per node

This experiment varied the number of vulnerabilities per node.

4.4.1 Observations

Curve fitting analysis of the data showed that the analysis time, total time, and forward pass time measurements scaled similar to one another just as in previous phases of experimentation. Each measurement increased as a second order polynomial, with the forward pass algorithm requiring the most time during analysis (see Figures A-9, A-10, and A-11). The impact of multiple vulnerabilities per node on memory requirements was small.

The amount of time required to perform initialization increased linearly with a very shallow slope, but was insignificant in relation to that of the forward pass algorithm. Examination revealed that the number of vulnerabilities per node did not affect the total amount of private memory used. The total size of pages belonging to the working set scaled linearly as more vulnerabilities were present per node. Although the size of the working set memory grew to approximately 50MB when all nodes present on the network gained an additional vulnerability, the amount of memory was small compared to other phases of execution.

4.5 Size of exploit database

This experiment varied the number of exploit models present in the exploit database. The smallest exploit database contained 1500 entries and the largest contained 31,500.

4.5.1 Observations

Curve fitting analysis of the data showed that the analysis time, total time, and forward pass time measurements increased linearly as more exploit models were added to the exploit database; however, the slope was very shallow, and minor (see Figure A-12), and does not have a significant effect on the amount of memory needed by CAULDRON to complete these tasks.

Initialization increases linearly as more exploit models are added to the exploit database. The slope appears to be approximately the same as for the forward pass algorithm, analysis and total time measurements (see Figure A-13). The time required to perform the backward pass algorithm, and the size of private bytes did not appear to scale as exploit models were added to the exploit database (see Figure A-14). The size of the working set increased linearly as exploit models are added to the exploit database, which was most accurate as the exploit database grew larger.

4.6 Modifications to CAULDRON

This test was conducted with inputs that would exercise memory and processor usage: a fully connected network model containing 60 nodes, and an exploit database containing 1500 exploit models.

A *Student's t-Test* was performed on analysis time, total time, private bytes, and working set measurements of the two versions of CAULDRON. For analysis time, total time and working set, the student's t-Test rejected the null hypothesis with greater than 99% confidence that there was no difference between the means for each data set. Although the Student's t-Test showed that working set data was influenced by the version of CAULDRON used, it appears that the graph visualization activities present in the unmodified CAULDRON, but not activated in the script-driven test version, account for the difference. In any case, the

data showed that the changes made to CAULDRON resulted in insignificant changes to the time measurements and size of the working set (less than one percent).

5. Conclusion

This performance analysis of CAULDRON provides practitioners with an understanding of CAULDRON's capabilities and limitations and offers developers insight into areas which may need further development. Our analysis shows that CAULDRON's processing time and memory usage were directly responsive to several of the independent variables. Processing and memory use increased polynomially as the number of nodes in the network increased, as did processing with respect to the number of vulnerabilities per node. The number of nodes that were reachable from the attacking node had a pronounced effect on the amount of time required to analyze a network. Other factors scaled linearly, or had minimal effect. Considering that simulated networks contained in our experiments were comprised entirely of vulnerable nodes and assuming that real-world networks would likely not be as saturated with vulnerabilities, we believe that CAULDRON could be used on much larger networks than those used during our experiment. In addition, it was observed that hierarchical networks could be analyzed in far less time than flat networks, suggesting that large hierarchical networks can be processed in a reasonable amount of time. This indicates that CAULDRON has the potential to scale well in large enterprise IT environments.

5.1 Future work

Analysis of performance measurements revealed input characteristics with the most impact on performance; however, additional research questions were also raised:

- In this experiment, all nodes in the network model possessed a unique vulnerability-exploit pair. It is uncertain how performance would have scaled if all nodes possessed the same vulnerability-exploit pair.
- An analysis of the CAULDRON algorithms could confirm and shed light on the results of this experiment.
- CAULDRON is capable of creating exploit models which consist of complex pre-conditions; however, it is unknown what effect the inclusion of more complex exploit models will have on performance.
- Since the experiment indicates that the forward pass algorithm takes the most time, efforts to improve performance should focus on the efficiency of this algorithm.

References

Cygwin (2006) Cygwin Information and Installation, http://www.cygwin.com/.

Denning, P. (1968) The Working Set Model for Program Behavior. Comm. ACM 11, 5, pp 323-333.

Ingols, K., Lippman, R., Scott, C., Piwowarski, K., Kratkiewicz, K., Artz, M., Cunningham, R. (2005) "Evaluating and Strengthening Enterprise Network Security Using Attack Graphs," Lexington, Massachusetts, October.

Jajodia, S., Noel, S., O'Berry, B. (2003) "Topological Analysis of Network Attack Vulnerability," in *Managing Cyber Threats: Issues, Approaches and Challenges*, V. Kumar, J. Srivastava, A. Lazarevic (eds.), Kluwer Academic Publisher.

Jajodia, S., Noel, S. (2004) "Managing Attack Graph Complexity Through Visual Hierarchical Aggregation," in *Proceedings of the ACM CCS Workshop on Visualization and Data Mining for Computer Security*, Fairfax, Virginia, October.

Lippmann R., Ingols, K. (2005) "An Annotated Review of Past Papers on Attack Graphs," Lexington, Massachusetts, March.

Noel, S., O'Berry, B., Hutchinson, C., Jajodia, S., Keuthan, L., Nguyen, A. (2002) "Combinatorial Analysis of Network Security," in *Proceedings of the 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*, Orlando, Florida, April.

Noel, S., Robertson, E., Jajodia, S. (2004) "Correlating Intrusion Events and Building Attack Scenarios through Attack Graph Distances," in *Proceedings of the 20th Annual Computer Security Applications Conference*, Tucson, Arizona, December.

Ritchey, R., O'Berry, B., Noel, S. (2002) "Representing TCP/IP Connectivity for Topological Analysis of Network Security," in *Proceedings of 18th Annual Computer Security Applications Conference*, Las Vegas, Nevada, December.

Sheyner, O., Wing, J. (2004) "Tools for Generating and Analyzing Attack Graphs," in *Proceedings of Formal Methods for Components and Objects*, pp 344-371.

Skybox (2006) Skybox View Suite Brochure, [online], http://www.skyboxsecurity.com/data_sheets/Skybox%20View%20Suite%20Brochure%20July%202006.pdf.

Sun (2006) Tuning Garbage Collection with the 5.0 Java[tm] Virtual Machine, [online], http://java.sun.com/docs/hotspot/gc5.0/gc_tuning_5.html.

Appendix: Performance graphs

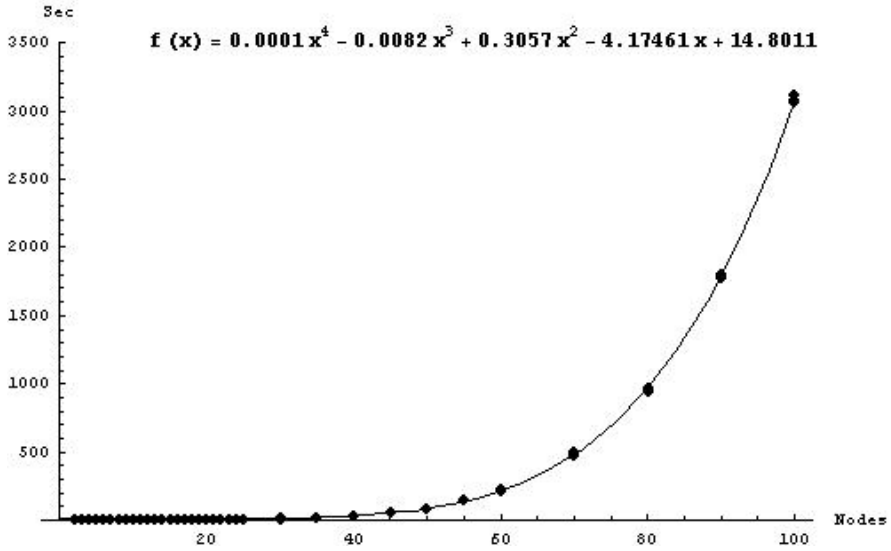


Figure A-1: Time required to complete the Forward Pass Algorithm

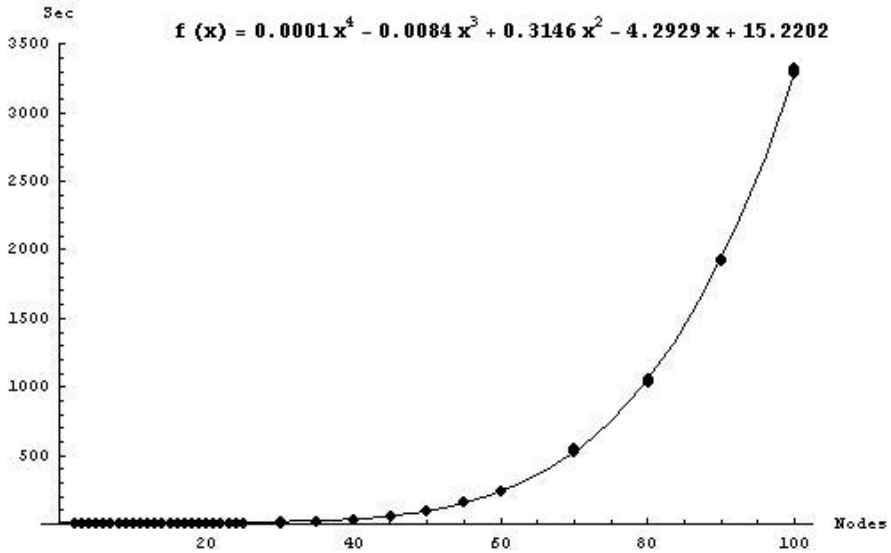


Figure A-2: Time required to complete the Analysis Algorithm

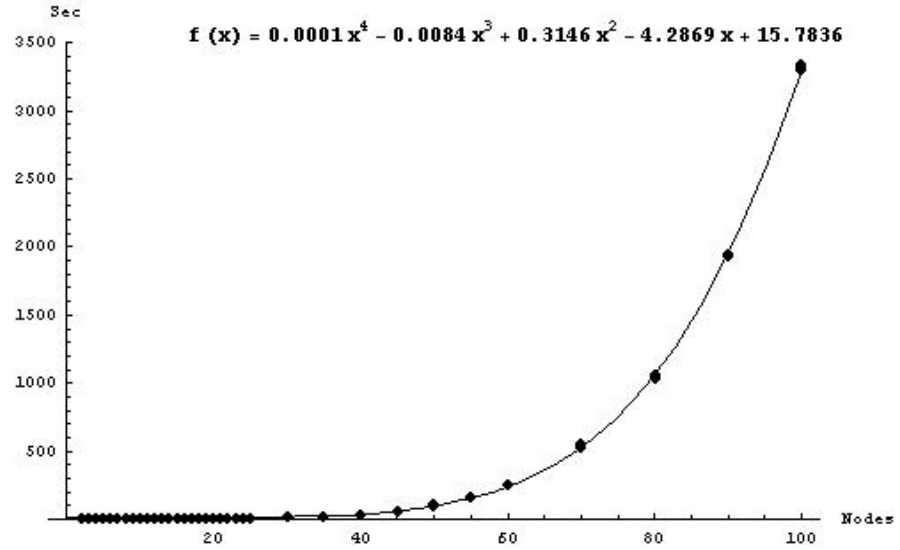


Figure A-3: Total Time required to Analyze

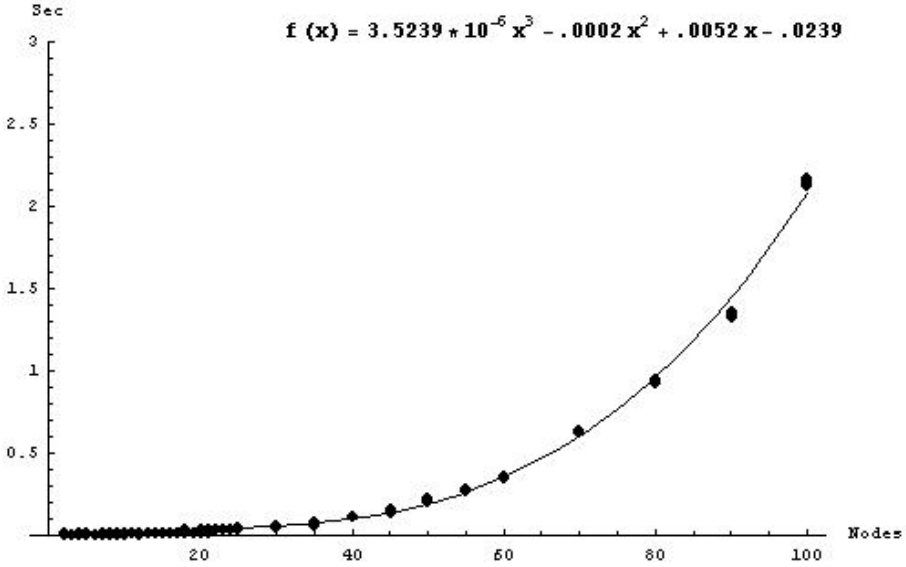


Figure A-4: Time required to complete the Backward Pass Algorithm

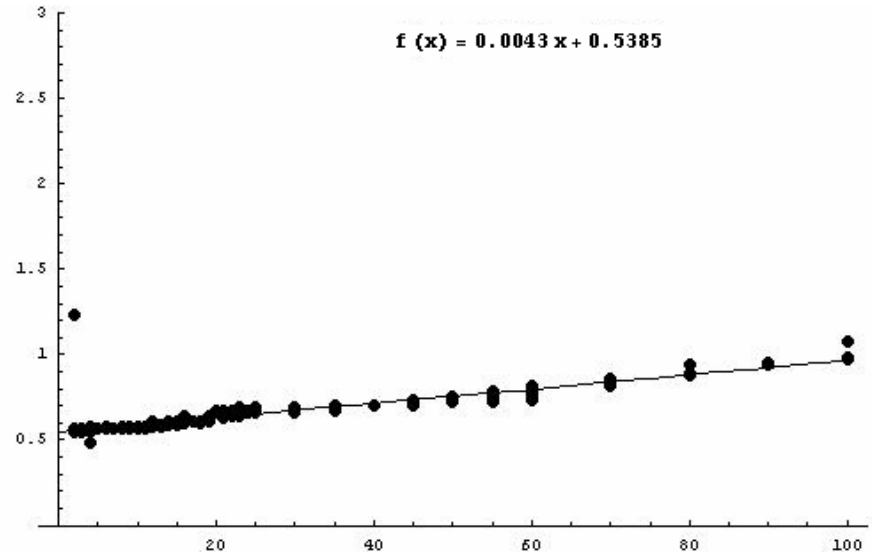


Figure A-5: Time required to complete the Initialization Algorithm

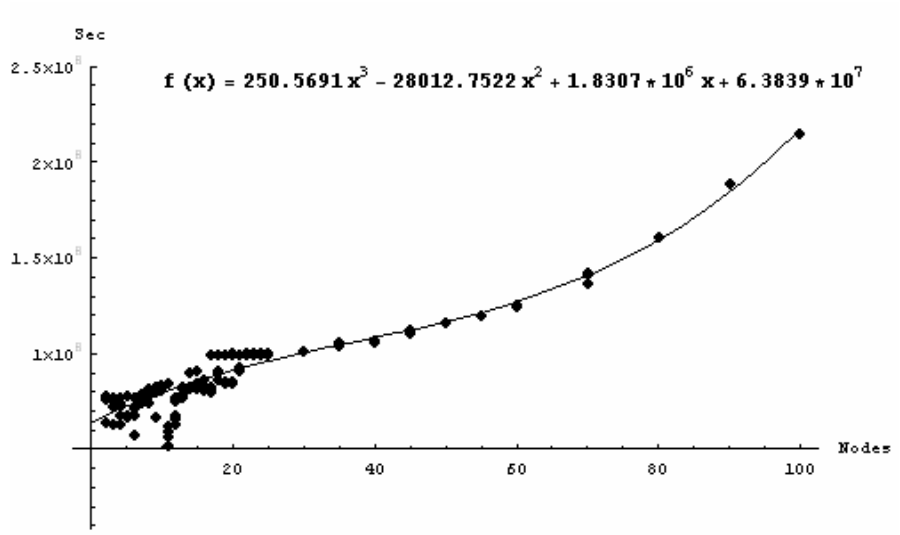


Figure A-6: Working Set Size as Network Size Grows

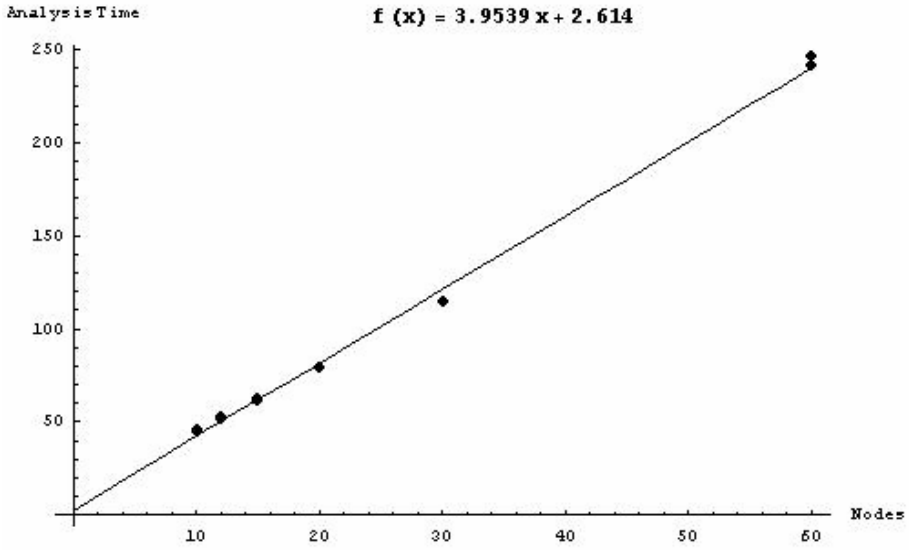


Figure A-7: Time required to perform analysis as node-set size increases

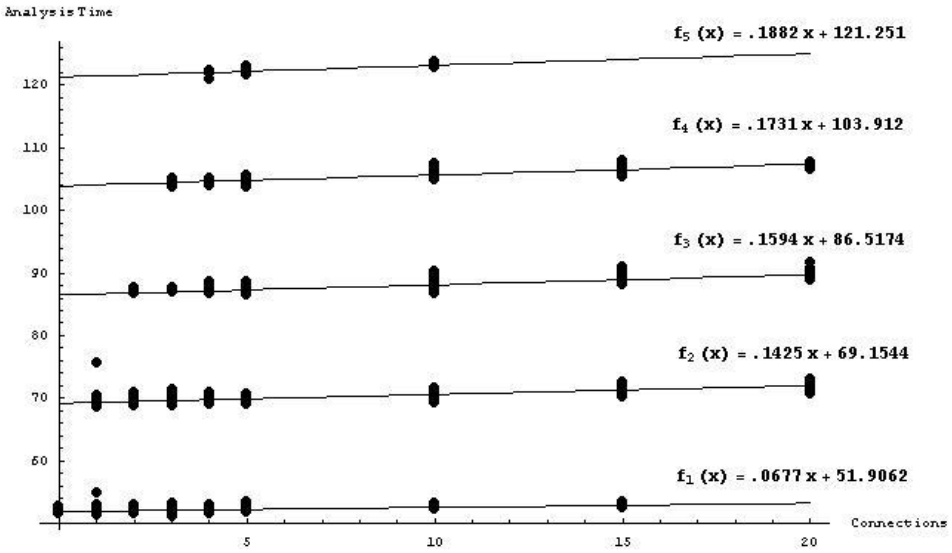


Figure A-8: Stratifications in time required to perform analysis

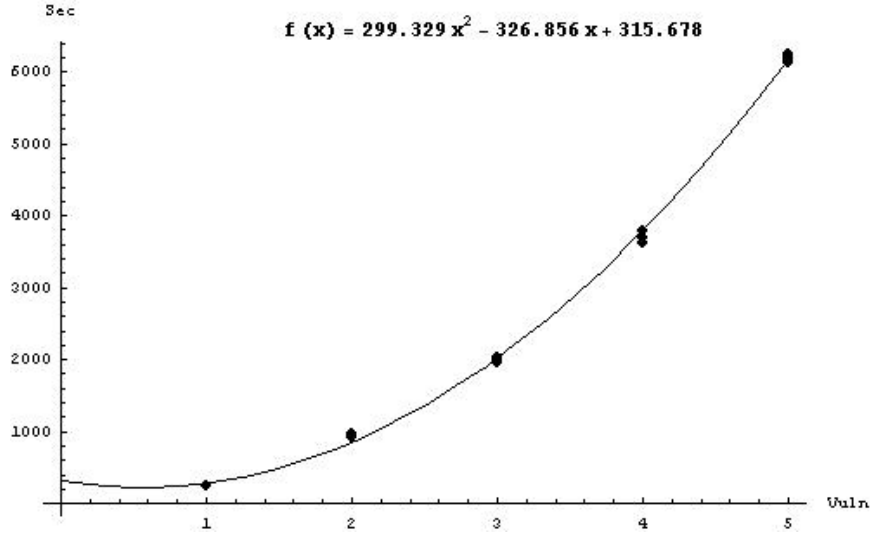


Figure A-9: Analysis time required as the number of vulnerabilities per node scaled

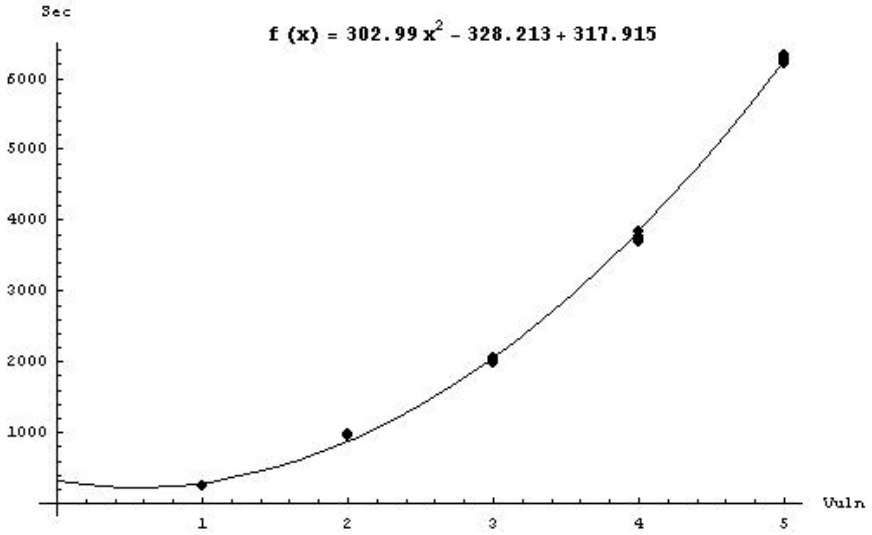


Figure A-10: Total time to complete tests as the number of vulnerabilities per node scaled

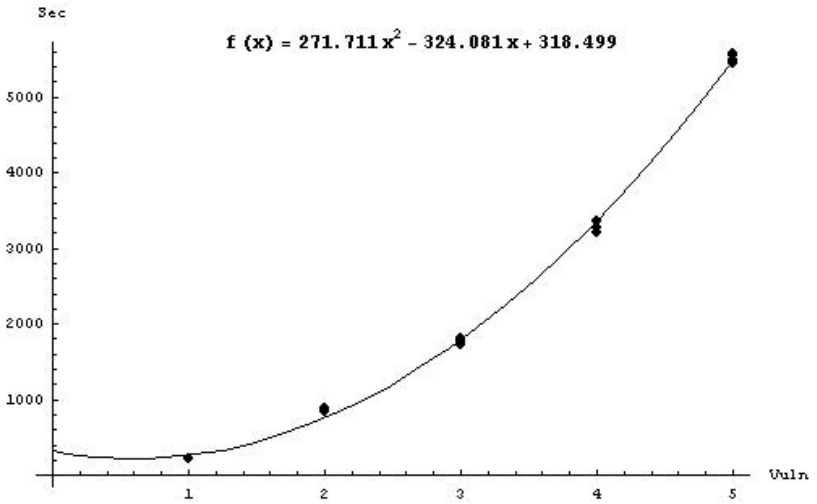


Figure A-11: Time to complete the forward algorithm as the number of vulnerabilities per node scaled

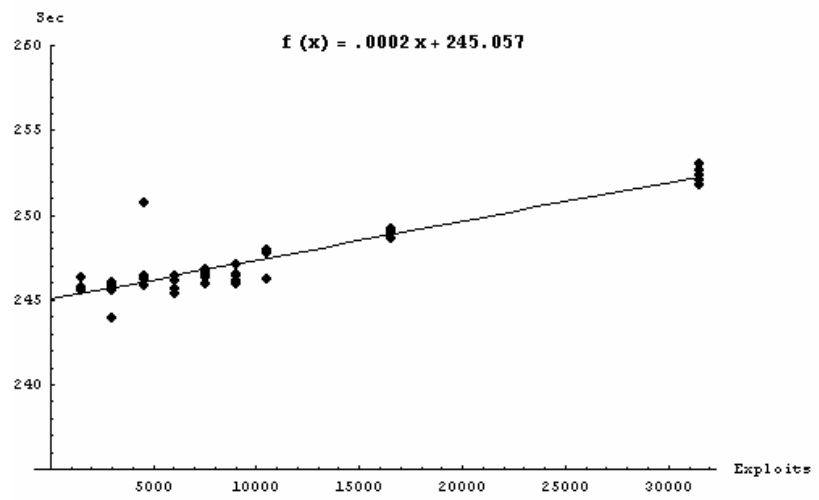


Figure A-12: Analysis time as the exploit models were added to the exploit database

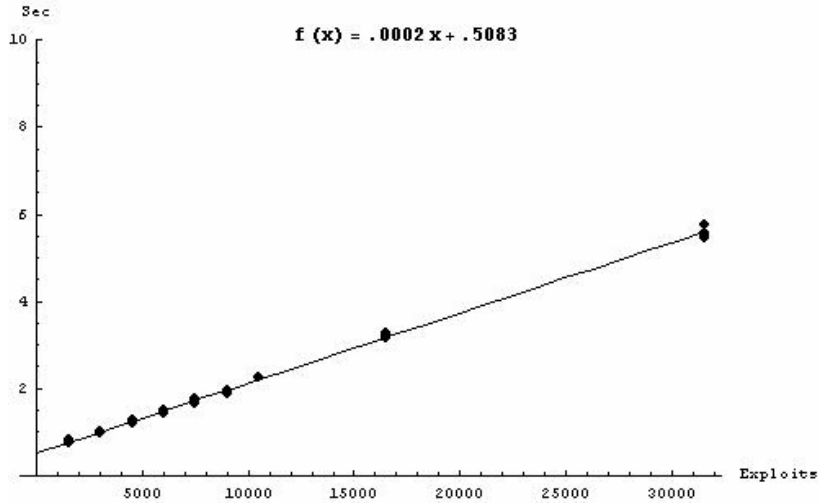


Figure A-13: Time required to perform initialization as size of the exploit database grows

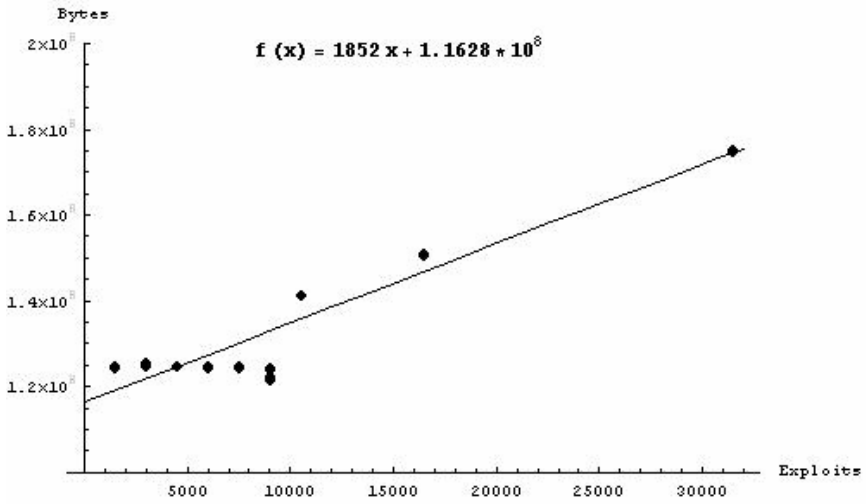


Figure A-14: Working set size as exploits are added to the exploit database

Analyzing Security Measures for Mobile Ad Hoc Networks Using Attack and Protection Trees

Kenneth Edge[1], Richard Raines[1], Rusty Baldwin[1], and Michael Grimaila[1],
[1]Air Force Institute of Technology, Fairborn, OH, USA
Christopher Reuter[2] and Robert Bennington[2]
[2]Air Force Research Laboratory, Fairborn, OH, USA
Kenneth.Edge@afit.edu
Richard.Raines@afit.edu
Rusty.Baldwin@afit.edu
Michael.Grimaila@afit.edu
Robert.Bennington@wpafb.af.mil
Christopher.Reuter@wpafb.af.mil

Abstract: Mobile Ad Hoc Networks (MANETs) consist of multiple individual wireless nodes that together form a mobile computing network without any centralised control. MANETs are desirable for various applications due to their inherent mobility and connectivity. A major disadvantage of MANETs is their complex security issues. This paper explores these security issues using a methodical process that implements attack and protection trees. Attack trees are an established method of conducting a risk analysis on a MANET. This paper builds on this methodology by also incorporating protection trees. Protection trees are derived from attack trees and provide a means to allocate limited resources to defend against specific attacks. Protection trees are produced systematically by first developing an attack tree, computing metrics for each node of an attack, and developing a corresponding protection tree with similar metrics. In this paper, a generic MANET is implemented and attack and protection trees are used to analyze the security of this network. This analysis of the MANET is a multi-objective problem with competing objectives of cost and performance. Methods are developed to use protection trees while determining the most efficient types of protection to implement depending on the application. Using the results of this analysis, a security scheme can be implemented which obtains the best results with the least amount of resources.

Keywords: Protection tree, attack tree, MANET, security, risk analysis, multi objective

1. Introduction

Mobile Ad Hoc Networks (MANETs) are becoming increasingly desirable for many applications due to their flexibility, mobility, and since no fixed infrastructure is required. For these same reasons, security in MANETs is a challenge. To determine vulnerabilities and countermeasures to MANET security, this paper uses attack and protection trees to analyze the security of MANETs. A precise methodology is developed that ensures the most cost effective protections are chosen for implementation. The performance cost of a security scheme in MANETs is often more important than the performance cost of a security scheme in a wired network, especially since nodes in MANETs usually have limited computing power. This paper addresses the financial costs of protection solutions as well as the performance costs and frames the problem as a multi-objective one. Using a sensitivity analysis of the trade-off between financial cost and performance, a thorough examination of the cost factors can aid in establishing a proper balance of competing costs.

1.1 Document outline

First, we discuss and develop the concept of generic attack trees in section 2 followed by protection trees in section 3. Section 4 explains how metrics are used in these types of trees and the rules for propagating the metrics to higher level nodes. After the concepts are developed, section 5 uses attack and protection trees to analyze a notional MANET. Section 6 discusses the results of this analysis, followed by section 7 which discusses the future direction of the research.

1.2 Attack trees

Once a high level threat has been determined, a method is needed to decompose this threat into intermediate objectives. These intermediate objectives can be further decomposed into individual attacker actions. Threat logic trees (TLT) have been used to accomplish this decomposition (Weiss, 1991). The root node in TLTs is the attacker's goal. Child nodes are a decomposition of the parent node and are related to each other by OR or AND relationships. In an OR relationship, if any of the child node tasks are accomplished, the parent node is successful. With an AND relationship, all of the child node tasks must be accomplished for the parent node to be successful. TLTs today are known as attack trees (Schneier, 1999).

Although the name has changed, the structure and operation of attack trees are identical to that of TLTs. To illustrate how attack trees are constructed, a simple example is shown in Figure 1. The attacker's goal is to get a free television from a retail electronics store. The attacker can accomplish the goal by either stealing the television from a customer or by stealing it directly from the store. To steal the television from a customer, the thief must identify and follow a customer who has purchased a television AND actually take the television from the person. This branch is an example of an AND relationship. To demonstrate the OR relationship, the thief has two ways to steal the television from the store. He can either take it from the shelf OR from the delivery truck. This example is deliberately simple to clarify the construction of attack trees. A real attack tree is more complete. While the root node is the attacker's ultimate goal, intermediate nodes represent sub-goals necessary to achieve the root goal. The leaf nodes are attacker actions. An attacker can only influence the system where he can interact with it. These points of influence are represented exclusively by the leaf nodes. All nodes are derived from attacker actions, but an attacker can not enter the attack tree at intermediate nodes.

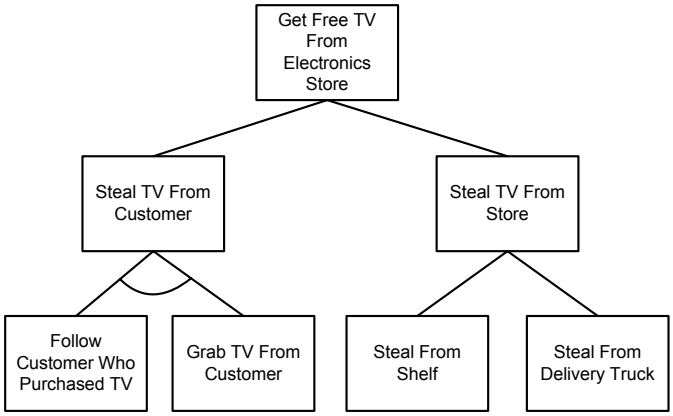


Figure 1: Example attack tree

A benefit of attack trees is their modular structure. Once developed, an attack tree such as the one in Figure 1 can be used again for larger attacks. For instance, the electronics store might want to develop an attack tree for risks to all of its products. This attack tree would consist of several smaller modules such as the one developed here. The modular design also allows analysts to work on separate components of a larger attack tree simultaneously. High level nodes might be developed in a group setting and each branch assigned to specific analysts for further development. The group could later integrate the multiple component attack trees to complete the larger attack tree. This modular design makes automating the process of creating attack trees easier. Once the component attack trees are developed, they can be recalled as needed. An important aspect of such a system is to keep the libraries of attack trees current in the face of new technologies and attacks. Although attack trees are a very useful method of identifying vulnerabilities in a system, analysts must also understand their limitations. First, an attack tree contains only known attacks. An attack tree does not identify attacks an analyst has no knowledge of. They can, however, be used to identify weak points in a system where an exploit might be possible even if the analyst does not know specifically how the system will be exploited. Furthermore, attack trees are only as good as the analyst developing them. Currently, there are no standard methodologies to develop attack trees. Therefore, even if an attack is known, unless the analyst actually building the attack tree is aware of it, it will be omitted. A repository of standard attack libraries of common systems can mitigate this weakness. These libraries could be used by analysts to construct larger attack trees and ensure sub-trees are complete. Generating attack trees for complex systems can be time consuming; automation can help this process. Combined with attack libraries, automation is easily possible with analyst specifying only the novel aspects of a system (Sheyner et al., 2002, Xiang and Cao, 2003, Swiler et al., 2001).

2. Protection trees

Once an attack tree identifies the vulnerabilities of a system, specific protections to mitigate those vulnerabilities must be determined and implemented. This is where protection trees are utilised. Like attack trees, protection trees are tree structures with child nodes related by either an AND or an OR construct. The root node corresponds to the root node of the attack tree; its goal is to protect the same resource that the attacker is trying to compromise. Like attack trees, leaf nodes are actions taken, although in protection trees, the actions taken are to protect the system. Intermediate nodes are determined by the protections introduced in the leaf nodes. A protection tree is constructed by identifying protections that can mitigate an attacker's actions in the leaf nodes of the attack tree. If every leaf node of the attack tree is covered by some

protection, then every identified attack is mitigated to some extent. Although similar to attack trees, the resulting protection tree for an associated system is actually complementary because of the differing goals of the attack and protection trees. For example, if an attack node has children related by an AND construct it means that the attacker must accomplish all actions in the child nodes to realise the parent node. Therefore, the protection system only needs to stop the attacker from accomplishing one action. This results in a protection node that has child nodes related by an OR construct. An attack node that has child nodes related by an OR construct means that the attacker has multiple ways to achieve his goal. To prevent the attacker from succeeding, the protection system must block all paths to the goal. This means the protection node will have children related with an AND construct. To illustrate the differences between attack and protection trees, consider Figure 2. This protection tree was created in conjunction with the attack tree shown in Figure 1.It is easy to see how the AND and OR relationships in Figure 2 are reversed from those of the attack tree in Figure 1. Although, this is usually the case, it does not have to be. If a specific protection can mitigate multiple attack nodes, then a protection node may also have an OR construct even though the attack node did as well. This example also shows that sometimes there are many ways to deal with an attacker's action. In the case of protecting the inventory on the shelves, the protection tree shows three different choices to achieve the same goal. Which one to choose might depend on the effectiveness of the individual protections, the cost to implement them, or other factors. How to choose the protection to implement is further discussed in Section 6.

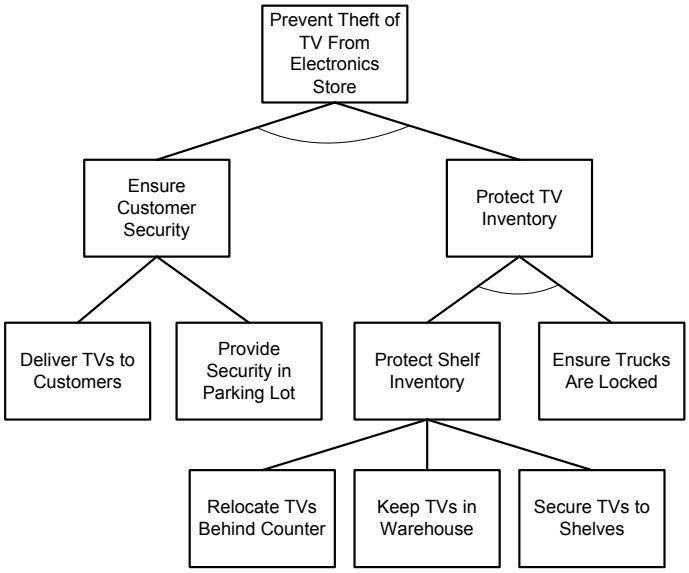


Figure 2: Protection tree for electronics store

3. Metrics

Metrics are needed to make informed security decisions. We need to estimate the chance an attacker will be successful and how much it might cost him. It is also important to know the system impact if the attack is successful at a specific node. With these types of metrics, a risk value can be calculated for every path an attacker may take. This risk value allows a defender to make security decisions on where to best place protections. Protection trees need the probability a protection will stop the attacker and the cost to implement it. In addition to a monetary value, cost can be represented in other terms such as lower throughput, slower speeds, and increased size. Once probability and cost metrics are determined, a cost factor per unit of protection can be determined. If more than one type of cost is used, analysis based on the combination of the attack tree and the protection tree, determine which protections to implement. After the protections are implemented, the attack tree can be recalculated and another iteration of the protection tree is produced. This process continues until the available resources for protecting the system are exhausted or the system is sufficiently protected against known threats. Although the specific metrics for these attack and protection trees are probability, cost, impact, and risk, other systems may have additional metrics. These may include the probability of detection, technical skill required, inconvenience to user, and damage cost to the system. The idea is to capture the characteristics of the specific system being analyzed and consider how it is affected by the attacks and protections. Before metrics can be assigned to nodes, the characteristics of the metrics must be known. Probability is typically determined based on the expert judgment of an analyst. The analyst may use past data or estimation to determine a reasonable probability for a particular node. The cost to implement an attack or a protection can be stated in any unit desired such as dollars, man-hours, or even

a generic cost unit. In the examples herein we use dollars as a monetary cost and a percentage to represent lower system performance for the two cost metrics. To determine the cost, an analyst again may use past data, estimation, or actual cost figures if they are available. The impact to the system is only used in the attack trees and is a number between one and ten representing how the system is impacted if a particular node is successful. To determine an impact number, the analyst can use guidelines such as those listed in Table 1.

Table 1: System impact definitions and numerical ranges

Numerical Range	Impact Definition
1-3	Minor impact to system. May be a nuisance but is easily detected and/or repaired
4-6	Moderate impact to system. Confidentiality, integrity, and/or availability of system affected. Requires non-trivial effort to detect and/or repair.
7-9	Severe impact to system. Significant damage results to system. Considerable effort required to detect and/or repair damage.
10	System completely compromised, inoperable, or destroyed

After basic metrics are assigned to an attack node, a raw value of risk can be calculated using (1). Probability of success is divided by cost to implement and multiplied by impact to system. Risk increases as probability and/or impact increases and risk decreases as the cost of the attack goes up. After working with attack trees in many different situations, we have found that risk values between nodes often differ by several orders of magnitude. When comparing raw risk values between nodes it was apparent that the relative risks were sometimes hard to determine. To compare risk values between nodes, (2) normalises the raw risk value using the minimum risk in the tree and a logarithmic operation. The normalisation assigns the lowest risk node a value of one while the logarithmic operator makes comparing nodes with risk values differing by several orders of magnitude more manageable.

$$risk = (prob / cost) \times impact \quad \textbf{(1)}$$

$$risk_{normalized} = \log(risk / risk_{\min} \times 10) \quad \textbf{(2)}$$

Metrics for attack and protection trees are only assigned to the leaf nodes of the respective trees. To determine values for intermediate nodes and the root node, rules to propagate the metrics have been developed and are shown in Table 2. The probability metrics are calculated using standard probability rules. For AND nodes, an attacker must accomplish all of the nodes so the probability that he succeeds is lower than the highest probability child node. For an OR relationship, the attacker has multiple ways to succeed so the probability is higher than the lowest probability child node.

Table 2: Rule set to propagate metrics up attack tree

	AND	OR
Probability	$\prod_{i=1}^{n} prob_i$	$1 - \prod_{i=1}^{n} (1 - prob_i)$
Cost	$\sum_{i=1}^{n} cost_i$	$\frac{\sum_{i=1}^{n} prob_i \times cost_i}{\sum_{i=1}^{n} prob_i}$
Impact	$\frac{10^n - \prod_{i=1}^{n}(10 - impact)}{10^{(n-1)}}$	$Max_{i=1}^{n} impact_i$

$prob \in (0,1]$, $cost \in [1,\infty)$, $impact \in [1,10]$, $n =$ number of child nodes

The cost metrics for an AND node are additive since the attacker must accomplish all nodes to succeed. For an OR node, it is unknown which way an attacker will exploit the system, so the cost is calculated as a weighted average of the child node costs based on the probability of success. An alternate method of calculation uses the lowest cost child node as this would be the worst case for the defender of the system. The weighted average cost calculation retains more information about the likely cost to an attacker. To propagate the impact measurement of an OR node, the worst case impact is calculated using the maximum

impact of the child nodes. For an AND node, we use a rule similar to the probability rule for an OR node in combination with a normalisation operator. The idea is the impact of a node with an AND relationship must be greater than the impact of each of its child nodes. The attacker has done additional damage to the system with each node that he has succeeded in. Rather than being propagated from lower level node values, risk metrics are calculated at each node using (1), (2), the probability, cost, and impact metrics that propagated from lower level nodes. For protection trees, the rule set to propagate metrics for the initial tree is shown in Table 3. The probability nodes are calculated the same as for attack trees. This assumes all protections are implemented and gives the maximum level of protection available. In the analysis of the trees, the actual levels of protection are determined. The cost calculations assume all protections are implemented. Sometimes, a protection is used multiple times although it has a one-time cost. To account for this, the cost is not counted more than once when multiple parent nodes containing the duplicated child node converge. To facilitate the analysis phase, a cost factor is calculated for each node which determines how much protection per resource unit that node provides. With two cost metrics, three cost factors are computed using (3), (4), and (5) below. They include financial cost factor, performance cost factor, and total cost factor. The total cost factor is determined using a weighting obtained from the decision maker based on a sensitivity analysis that is discussed later. The way the cost factor equations are constructed, a higher cost factor means the system provides more protection per unit of cost.

Table 3: Rule set to propagate metrics up protection tree

	AND	OR
Probability	$\prod_{i=1}^{n} prob_i$	$1-\prod_{i=1}^{n}(1-prob_i)$
Financial Cost	$\sum_{i=1}^{n} cost_i$	$\sum_{i=1}^{n} cost_i$
Performance Cost	$1-\prod_{i=1}^{n}(1-cost_i)$	$1-\prod_{i=1}^{n}(1-cost_i)$

$prob \in (0,1]$, $financial\ cost \in [1,\infty)$, $performance\ cost \in [0,1]$,

$n =$ number of child nodes

$$fcf = p \times (prob / cost) \quad (3)$$

$$pcf = p \times (prob / cost) \quad (4)$$

$$tcf = p \times prob / (fc + (weight \times pc)) \quad (5)$$

where p = number of parent nodes

4. Application

To demonstrate how attack and protection trees are constructed and analyzed, a generic set of trees is described. After explaining how they are used, they are applied to a specific application of analyzing the security of a notional MANET. The attack tree is created by decomposing the attacker's root goal into sub nodes until the level of fidelity is that of the actual attacker actions. The sub nodes and leaf nodes are related by either AND or OR relationships. After all the nodes are defined, metrics are placed on the leaf nodes and propagated up to the root node using the rules discussed in Section 4. Typically, an attack tree is constructed from the root node down to the leaf nodes and the metrics are calculated from the leaf nodes up to the root node. A generic attack tree is shown in Figure 3.

To construct the protection tree, each leaf node in the attack tree is addressed with a possible protection and the next higher node in the attack tree is examined. If the higher node of the attack tree is covered by the existing protection nodes, the next higher level is analyzed until reaching the root node. If the attack is not mitigated, additional protection can be added with either an AND or an OR construct until the attack is mitigated. Once the attack tree is covered up to the root node using the protection tree, metrics are added to the leaf nodes of the protection tree and propagated up to the root node. During the first iteration, it is assumed every possible protection is implemented. This results in the maximum protection state possible. The highest level of protection is available at the highest possible cost. A generic protection tree after the first iteration is shown in Figure 4. In order to calculate the total cost factor for each node, a weighting of $5000 per one percent of performance was used. Later in the paper a sensitivity analysis is presented to determine the importance of this weighting factor.

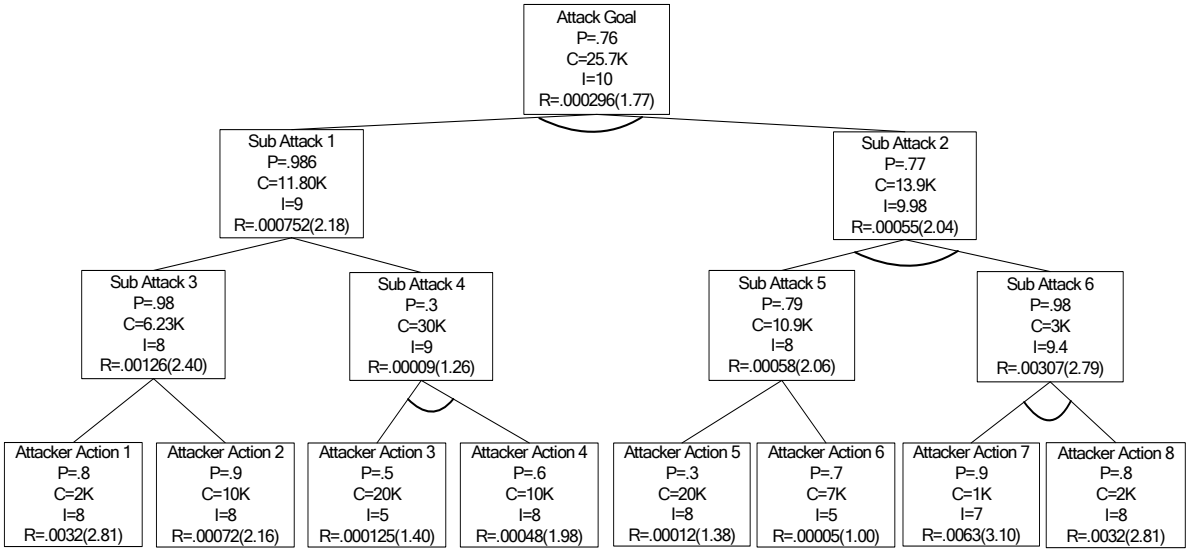


Figure 3: Generic attack tree

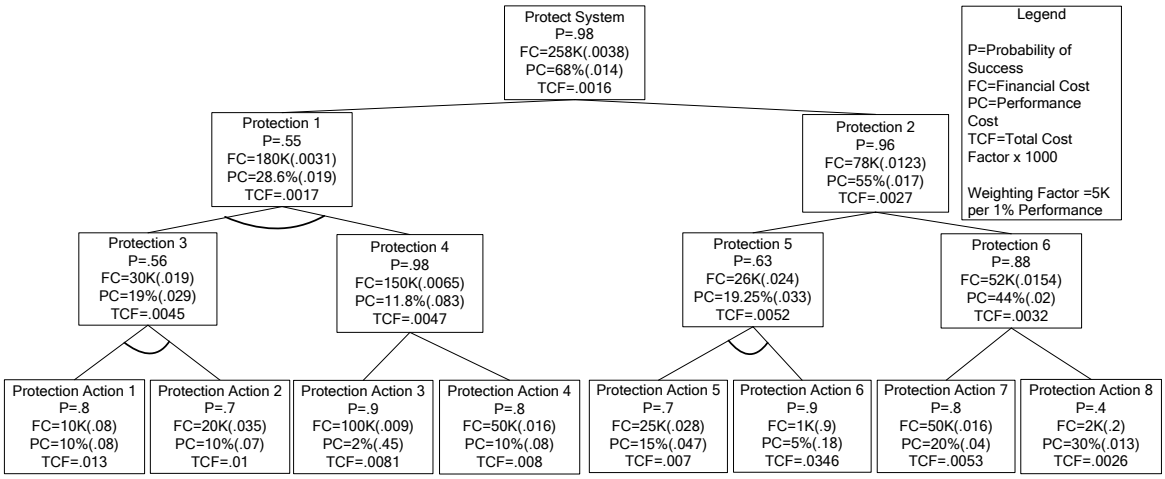


Figure 4: First iteration of generic protection tree

Since protection resources are usually limited, the most efficient protections need to be selected when there is a choice such as in an OR node relationship. If the system is being optimised for either cost or efficiency, the respective cost factor is used to select the protection to implement. If both objectives are important, the total cost factor is used after determining a weighting for the cost factors. Assuming the weighting between cost factors remains the same, the optimised protection tree is shown in Figure 5.

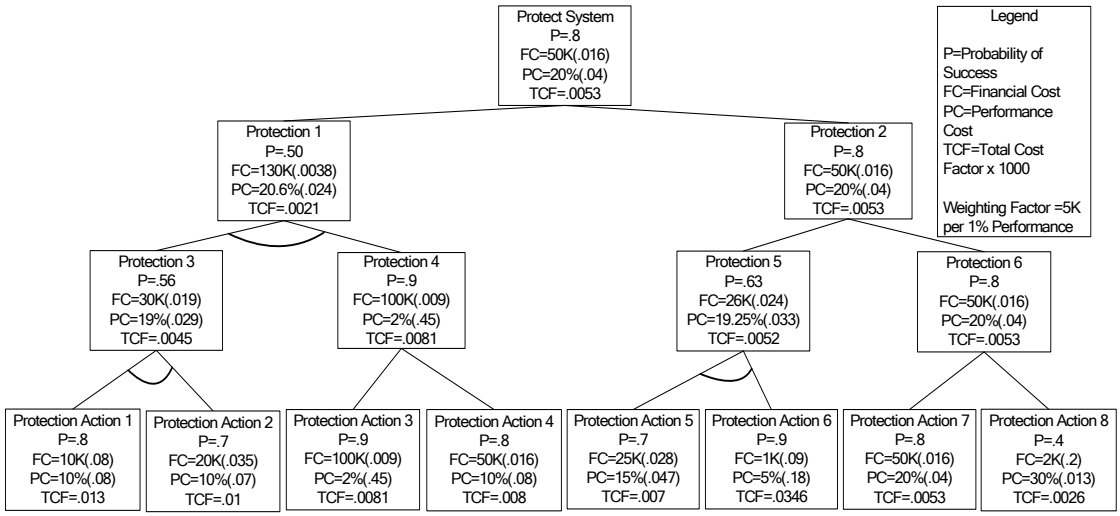


Figure 5: Second iteration of generic protection tree

A sensitivity analysis of the weighting factor must be determined to make an informed selection of the weighting factor. If the weighting factor is varied from $500 to $5000 different protections dominate and are propagated to the root node. A chart showing the trade off between protections is shown in figure 6. This chart it is clear that protection action 8 dominates until the weighting is greater than approximately $816 per one percent of performance. From $816 until $3820 per one percent of performance, the AND node comprising protection actions 5 and 6 dominate. Above $3820, protection action 7 dominates. Based on this analysis, a decision maker might be willing to change the weighting factor. For instance, if $4000 was originally chosen as a weighting factor, the true financial cost of the protection would be $50,000 with a performance cost of 20%. If one could accept a weighting factor of $3800, the true financial cost would be reduced to $26,000 with a performance cost of 19.25% by selecting different protections. Although this seems like a much better choice, the probability of successfully protecting the system drops from 80% to 63%. If the system can function at the lower protection level then the cost savings is significant for a small change in the decision maker's weighting factor.

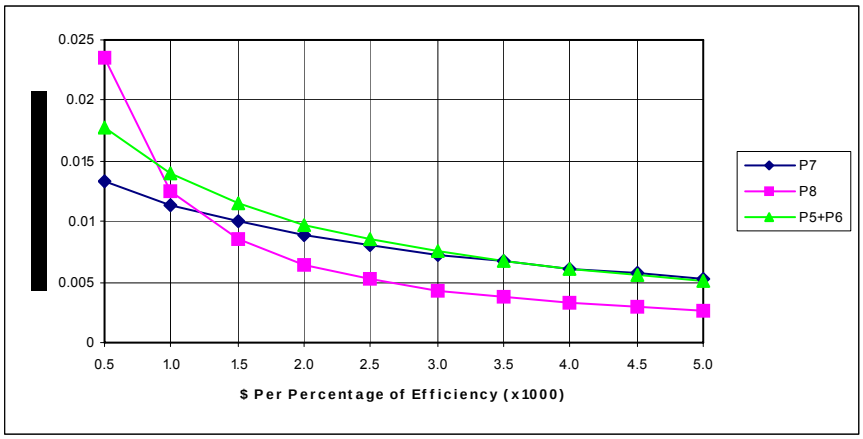


Figure 6: Total cost factor of competing protections

Now that an analysis has been described on a generic attack and protection tree, a notional MANET can be analyzed for security. After establishing the root node as attacking the MANET decomposition can begin. It is determined that there are two fundamental ways to attack a MANET. Either an internal node can be compromised and used for malicious purposes or an external intruder can infiltrate the MANET. Because the attacks attempted by an internal or external node are similar and space is limited, this paper only develops those attacks that originate from an external source. The three types of attacks on security can be decomposed in terms of confidentiality, integrity, and availability. To model these types of attacks, we use the attacker's ability to gain information on the network as a breach of confidentiality, deception as an attack on integrity, and denial of service as an attack on availability. These attacks are further decomposed and the complete attack tree is shown in Figure 7. Notional metrics are assigned to the leaf nodes and propagated up to the root node using the rules from section 4. Note that the numerical values in the attack tree are notional and are used simply to demonstrate the analysis methodology.

After the attack tree is constructed, the associated protection tree is developed as discussed in Section 3. For additional details in constructing attack and protection trees see (Edge et al., 2006). On the first iteration, the maximum protection level is determined. This protection level assumes that every possible protection is implemented regardless of cost. This protection tree is shown in figure 8. The second iteration (not shown) determines the most efficient protections to implement when choosing between protections related by an OR node. This ensures every attack is protected to some level but is usually less then the maximum protection level determined in the first iteration.

At this stage in the methodology, a sensitivity analysis is done on the weighting factor to present to the decision maker. In this example, the most efficient way to prevent an external attack is to prevent access to the network by the adversary. Two methods have been identified to do this; implementing physical access tokens or using strong authentication and encryption. After conducting a sensitivity analysis between these two options and graphing the results in figure 9, it is determined that implementing strong authentication and encryption is a more efficient option if the weighting factor is less then $1050 per unit of performance degradation. Above this level, it is more efficient to use a physical access token.

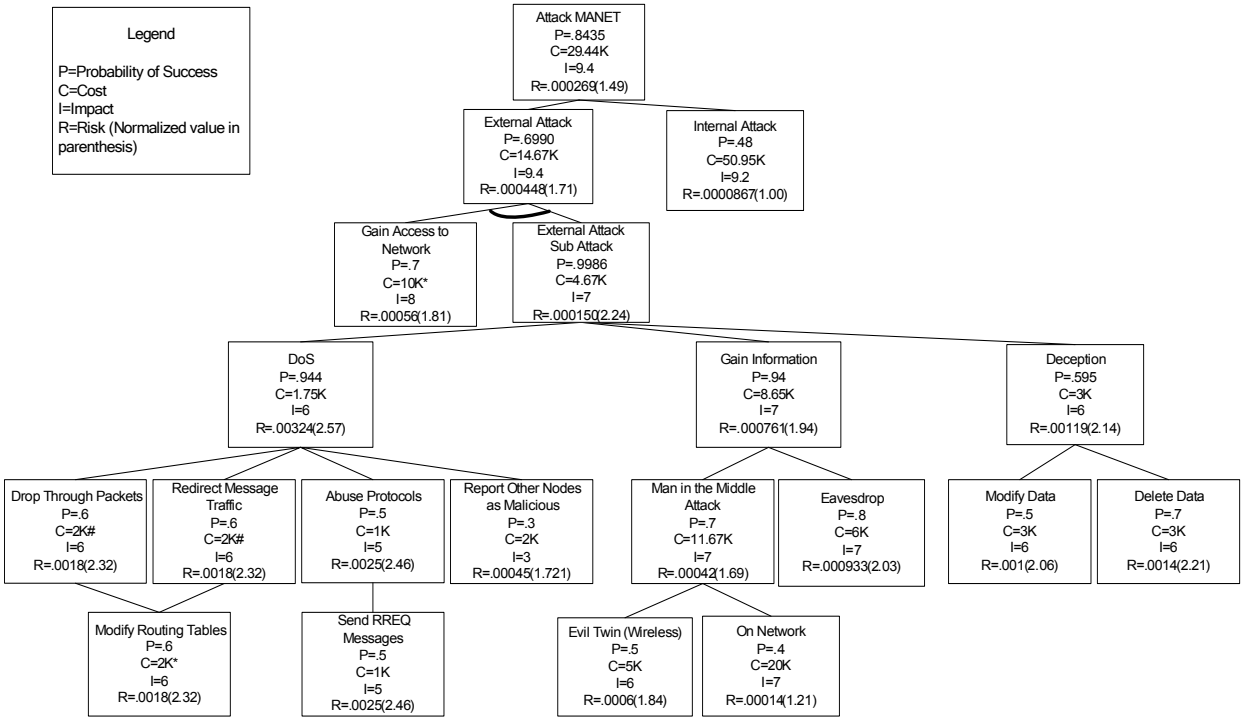


Figure 7: MANET attack tree

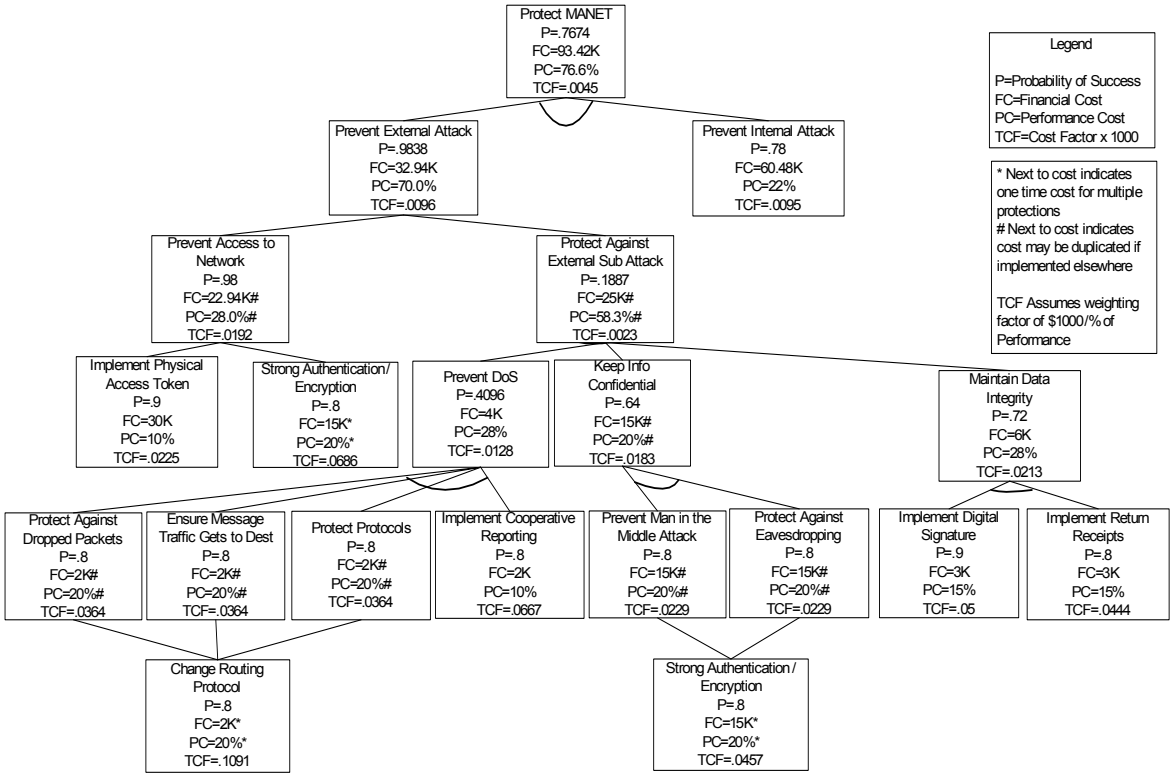


Figure 8: MANET protection tree (first iteration)

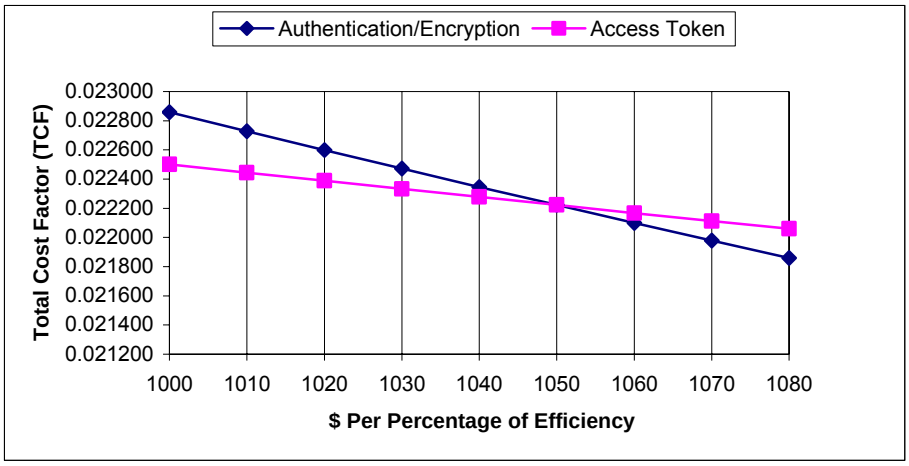


Figure 9 – Sensitivity analysis of weighting factor

5. Results and analysis

From the protection tree analysis, the maximum level of protection has a success rate of about 76% with a financial cost of about $93,000 and a performance cost of about 77%. Clearly, for most applications this performance degradation would be unacceptable. However, this maximum protection level defines a baseline of the highest protection level and what cost would be associated with it. In the next iteration, the performance cost is reduced to 30% and the financial cost is reduced to about $90,000. The level of success is slightly reduced to about 70%. This is a significant performance cost savings while maintaining about the same level of protection. For additional analysis, the internal attack and protection trees would need to be expanded.

6. Conclusions and future work

This paper describes a methodology of decomposing a high level type of attack on a MANET into specific attacker actions and mitigating the attacker actions with the most efficient types of protections. To do this, a protection tree is developed with corresponding metrics and is expanded on from earlier work. Specifically, this paper addresses the problem of analyzing competing cost metrics of the protections. By implementing a weighting factor using a thorough sensitivity analysis, the multi-objective problem is transformed into a single objective problem that can be properly analyzed using a protection tree. This paper presents a broad overview of the concepts and methodologies we have created involving attack and protection trees. Future work includes describing these concepts and methodologies in much more detail. Additional applications that attack and protection trees are being studied for analysis include online banking and software security at the application level. Proofs are also being developed to demonstrate that these methodologies result in secure systems. To do this, the Schematic Protection Model (SPM) is being extended and modified.

Acknowledgements

This research was sponsored by the Anti-Tamper Software Protection Initiative Technology Office, Sensors Directorate, of the U.S. Air Force Research Laboratory.

References

Edge, K. S., Dalton, G. C. I., Raines, R. A. and Mills, R. F. (2006) Using Attack and Protection Trees to Analyze Threats and Defenses to Homeland Security. *The 2006 Military Communications Conference (MILCOM).* Washington, D.C., IEEE Communications Society.

Schneier, B. (1999) Modeling Security Threats. *Dr. Dobbs Journal.*

Sheyner, O., Haines, J., Jha, S., Lippmann, R. and Wing, J. M. (2002) Automated Generation and Analysis of Attack Graphs. *2002 IEEE Symposium on Security and Privacy.*

Swiler, L. P., Phillips, C., Ellis, D. and Chakerian, S. (2001) Computer-Attack Graph Generation Tool. *2nd DARPA Information Survivability Conference and Exposition.*

Weiss, J. D. (1991) A System Security Engineering Process. *14th National Computer Security Conference.* Washington DC.

Xiang, G. and Cao, Y.-D. (2003) Generating IDS Attack Pattern Automatically Based on Attack Tree.

A Study of DNS Traffic Patterns on a Large Autonomous System

Sid Faber[1] and Bert Lundy[2]
[1]Software Engineering Institute, Pittsburgh, PA, USA
[2]Naval Postgraduate School, Monterey, CA, USA
sfaber@cert.org
bertlundy@mac.com

Abstract n this report DNS (Domain Name System) traffic on a major autonomous system is analyzed for unusual or malicious traffic patterns. DNS nodes inside the network are classified according to function. One finding is that DNS security concerns stem primarily from improperly configured DNS clients and servers. Many DNS servers leak private addressing information, some DNS Clients forward all queries to a DNS server that is outside of the autonomous system, and many DNS servers allow public recursion. Although some of these issues have been previously identified, quantitative detection of their existence in a large real-world autonomous system is provided, and methods for mitigating the threat of these misconfigurations are presented.

Keywords: Network security, intrusion detection, traffic anomaly, domain name system, recursion, prisoner.iana.org

1. Introduction and purpose

This report presents a method for categorizing DNS (Domain Name System) traffic on a large network. The purpose of this work is to observe and analyze DNS traffic flows, especially those between nodes inside the network and nodes outside it. In the process, we identify some patterns which present security concerns. Included in this is the categorization of DNS nodes inside the networks, to the extent possible, as servers of different types, and clients or other types of nodes involved in DNS traffic.

Network flow sensors record summary flow information on virtually all the traffic which flows into and out of the network, a large autonomous system. A network sensor is located at each border router. As packets pass through the router, summary information is recorded and written to a disk. This summary flow includes packet header information and data size but does not record the actual data. The authors have access to a software tool set which allows access to this large data base of “flows.” With the tool set, queries can be made to the database, such as “show all the DNS traffic which passed into the network between 3 and 4 PM last Friday.” The tools will then search the data base and respond with all the corresponding "flows" that passed during that time. The information includes the number of packets, source and destination IP addresses, source and destination port numbers and total amount of data. In this way network analysts can often ascertain if some type of unusual behavior is occurring. A large variety of such queries can easily be made. Very often the answer to one query (huge number of packets incoming to a certain IP address) may cause the analyst to generate another (how many packets of the total to IP address X are HTTP traffic?). Answers to such queries can be used, for example, to determine that a certain IP address is home to a web server, or a mail server - or possibly using its host to generate illegal or questionable traffic such as scans, spam, etc. The authors have been examining DNS traffic recently (August-October 2006) in an attempt to baseline DNS traffic and determine if any unusual or possible misconfigurations or illegal use of DNS is occurring.

2. A background on DNS and network flow data

DNS, the domain name system (Mockapetris 1987), is the Internet’s distributed database which maps the names used by users of the web - www.yahoo.com- to the corresponding Internet (IP) address, a 32 bit number. This 32-bit IP address is often expressed in dotted decimal notation, say 234.10.21.122, where each of the four decimal numbers expresses 8 bits as a decimal value. When an Internet user types www.yahoo.com into the window of a web browser, and hits "enter," then the DNS protocol is used to determine the corresponding IP address, which is needed for routing; after getting this address, HTTP, the web page transfer protocol, can communicate with the web server to obtain the desired web page. DNS is used not only by HTTP, but by many other web application programs. It is an essential part of the Internet suite of protocols. The DNS database is distributed over many DNS servers throughout the Internet. In order to learn the IP address corresponding to a web site, say www.foo.com, the web browser causes a query message to be sent to its local or default DNS server. This is sent encapsulated in a transport layer packet, which is sent using IP, the main network layer protocol. If the default server has the corresponding IP

address in its part of the database, it returns a "reply" message with the IP number. If not, further queries must be made to other more distant DNS servers until, finally, the correct IP address is returned.

The "port number" for the DNS protocol is 53. The port number is a part of a TCP or UDP message, in the packet header, used to identify the higher layer protocol the packet is carrying. Any DNS message should have the number 53 as the port number as either the source (sending) or destination (receiving) fields; and in some cases, possibly both. One key point to remember is that the purpose of the DNS protocol is to get information from the DNS database. DNS messages (queries and replies) should always be sent to DNS servers by users, or from one DNS server to another, or from a DNS server to a user. The "reply" message should only be sent by a server in response to a "query" message; that is, the first message sent in any DNS exchange should be a query; and the "query" message should be sent to a server, either from a host or another DNS server. Another key point to remember is that DNS messages are always encapsulated in a transport layer protocol, either UDP (mostly) or TCP (occasionally), which in turn is encapsulated in an IP packet. IP packets, in turn, are encapsulated in a local area network protocol packet, such as CSMA/CD or some other "layer 2" packet for transport over a physical wired or wireless link. The layer 2 packet is then transmitted as a series of bits by a physical layer protocol and medium, such as optical fiber, TV coaxial cable, "wifi" wireless protocol, cellular telephone (microwave frequency), etc. The formats of the UDP, IP and DNS packets are shown in **Figure 1**. The normal size of an IP header is 20 bytes; and a UDP header, 8 bytes. The header size of a DNS packet is 12 bytes. Thus, an IP packet with a UDP-DNS packet should contain at least 20+8+12=40 bytes of header, plus the data fields of the DNS packet. These numbers are critical to some parts of the following discussion.

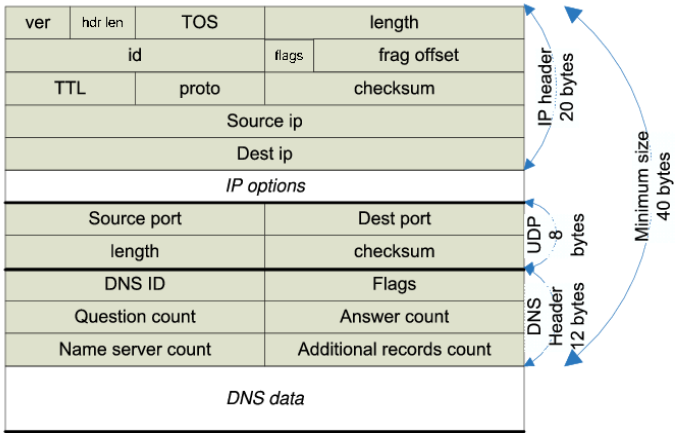


Figure 1: The DNS datagram

DNS traffic may also communicate using TCP. TCP requires a minimum of 20 bytes for header information, and requires at least three packets to establish a communication channel, before sending the required 12 byte DNS header and additional data. Data examination shows that DNS traffic uses the TCP protocol only rarely. Since this research focuses primarily on the most widely used traffic patterns, DNS traffic over TCP will be set aside for future research.The network sensors which record the "flow" info record this on the IP packet and its contents; but not the data link (layer 2) packets. For example, suppose an IP address IP-OUT, outside our network, sends 1 packet to an IP address IP-IN, inside our network. IP-OUT does this using the DNS protocol (query) over UDP. Assume that the DNS query contains 5 bytes of data. Thus, the DNS packet contains 12+5 = 17 bytes total. The IP packet contains 28+17=45 bytes. The network sensor will record the following in the flow database:

Many other types of queries can be sent; a particular IP address can be checked, or all the TCP traffic in a certain time period, for other examples. This is a very brief description of DNS and of network "flows;" for a more thorough explanation, see (Cisco 2006) and (Kurose, Ross 2003).

Table 1: network sensor will record the following in the flow database:

Source IP	IP-OUT
Destination IP	IP-IN
Source Port	1553
Destination Port	53
Protocol	UDP
Number of packets	1
Number of bytes	45

3. Categorization of DNS nodes and traffic

The DNS protocol is fairly well-defined, and from the definitions of the protocol we can expect to see certain types of patterns in the traffic. To a large degree, we can also categorize the nodes sending and receiving the traffic. All DNS traffic should pass to, from or between DNS *servers*. There are several types of servers. These include *root servers*, *domain servers*, *authoritative servers* and *local name servers* (often referred to as *default name servers*). The hierarchical nature of DNS is shown in **Figure 2**.

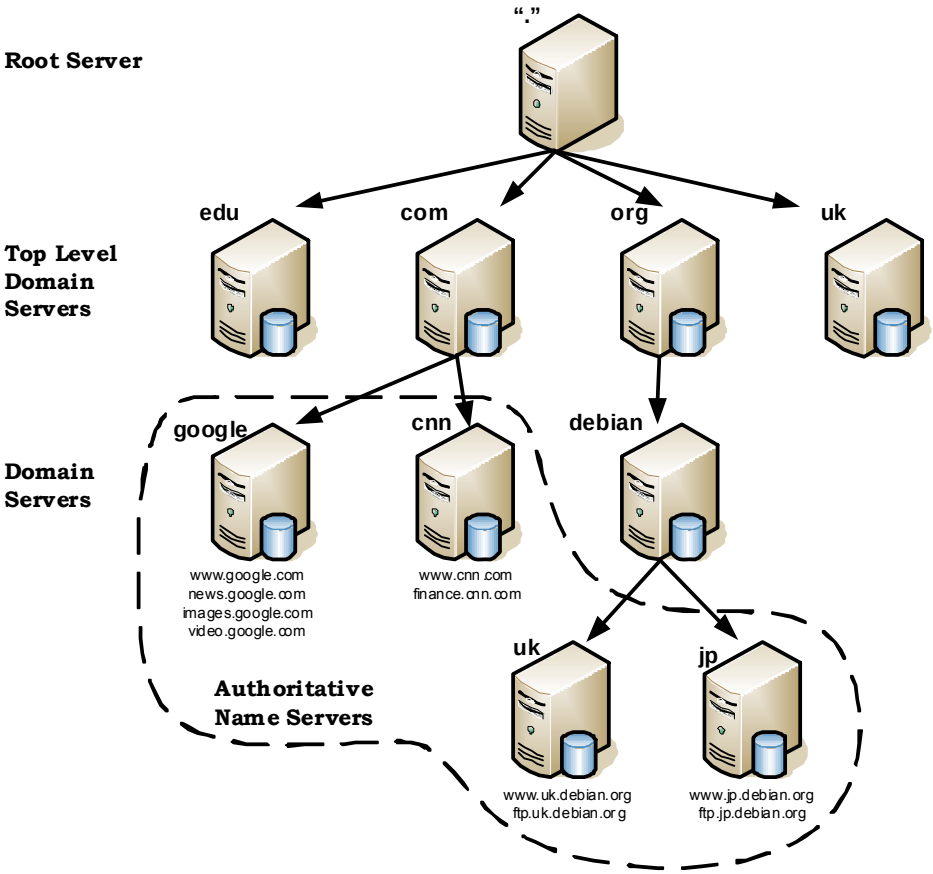


Figure 2: The DNS architecture showing relationship between different types of DNS

Root servers form the top level of the DNS hierarchy. These 13 well known servers maintain the list of top level domain servers and addresses, and will provide the list to other DNS servers upon request. A *domain server* is the DNS server for a domain, such as foo.com. Since the DNS structure is hierarchical, a domain server may be the parent of many sub domains, such as left.foo.com and right.foo.com. A *top level domain* (TLD) server is a specialized domain server, responsible for managing domains such as .com, .org and .gov that form the last part of the Internet domain name (the part following the final "."). There are also two-letter top level domains for each country. An *authoritative name server* is the server who has the IP addresses for the names under its jurisdiction; it is the final authority on these IP domain name - address mappings. For example, the authoritative name server left.foo.com will be able to return the IP address for the DNS name www.left.foo.com. A *local name server* is responsible for querying domain servers on behalf of clients. When a client wants to go to a web page, or for any other reason needs an IP address corresponding to a domain name, it queries its local name server, also know as its *default name server*. The local name server then returns the address from its cache, or it contacts an authoritative name server to find the address. When a host initializes its connection to the Internet, it is configured or provided with the IP address of at least one, and often two or three default name servers.

Additional network devices pass DNS traffic, but are not actually DNS servers. Two categorizations we have observed are called *gateways* and *forwarders*. Gateways are network devices (such as firewalls or network address translators) which pass but do not serve DNS traffic. Many DNS clients connect to the outside network through this firewall, which translates the source address of all requests to its own IP address. Forwarders are similar to local name servers, but they do not contact authoritative name servers directly. Rather, if a DNS entry has not been cached, they will simply forward the request to another local name server, which will then perform the necessary lookups. These two additional types of DNS nodes are

commonly used in network architectures, but limitations on network flow monitoring tend to make categorization of these devices more difficult. One of our objectives is to determine the DNS servers in our network and if possible determine which of these categories they fall into. In particular, the authoritative name servers should be identified. One indication that a node corresponding to a given IP address is a DNS server is that it receives UDP packets with port 53 as the destination port number. If UDP source port 53 replies are seen coming from the same node back to the IP address which sent the query, then it is almost certain that the node is a DNS server. Any node involved in DNS traffic which is not a server is expected to be a client. A client sends a DNS request, or query, with a destination port of 53, and some high-numbered source port number. This characteristic makes client traffic fairly easy to identify. An important characteristic of DNS traffic is that if message exchanges occur between two IP addresses, then the traffic should be balanced, and that queries should occur before responses. When carried in UDP datagrams, the size of the packet headers (which excludes layer 2 headers) should be 20+8+12=40 bytes. Any packet must be larger than this if any DNS queries or responses are included. One other characteristic of DNS traffic is that the sizes of responses are not expected to be a great deal bigger than the queries, especially as compared to other protocols. For many client-server protocols, a query of only a few bytes may return megabytes of data. However, in the case of DNS, the query includes the domain name, and the response also includes the same domain name along with the corresponding IP address, which is 4 bytes long. The domain name itself is usually much larger than the IP address. From this reasoning, we expect that while the replies should be larger than the queries, they are not expected to be much larger, and certainly within the same order of magnitude.

The network sensors which allow us to view network traffic are all located at the boundary of the autonomous system (AS). It is believed that these cover all or almost all of the traffic into or out of the AS; although it is possible that there may be some "back doors" (unmonitored Internet access points). For this study we proceed initially assuming that these back doors are either non existent or insignificant. Internal nodes involved in DNS traffic are assumed to be authoritative servers, local servers, both authoritative and local servers, clients, gateways and forwarders.

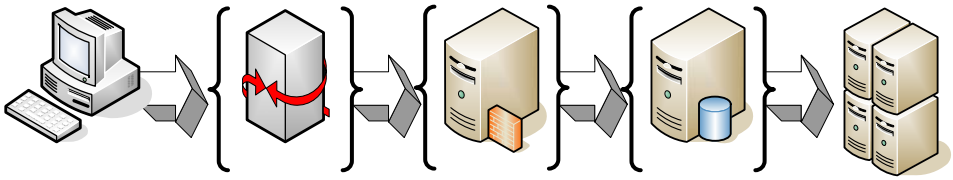


Figure 3: Typical flow for a DNS request; gateways, forwarders and local DNS servers may or may not be used. The DNS response returns through the same nodes as the request.

Figure 3 shows the general flow of data in a DNS request which will be useful for categorizing nodes on our network. Since our sensors only show traffic crossing the boundary of the AS, we will not be able to identify all the nodes involved in the DNS conversation. Rather, we will only catalog the last node within the AS and the first node outside of the AS. For example, if a forwarder within the AS is configured to contact two local DNS servers outside of the AS, we should expect to catalog the forwarder and the servers to which it talks. However, we will not be able to identify the authoritative name servers downstream from the forwarder. Since we will only be observing boundary traffic, it is unlikely that we will identify DNS forwarders. On average, network flow data from traffic passing through a forwarder will appear almost identical to client DNS traffic except in volume.

4. Methodology

The initial query was to simply capture all flow data on UDP port 53 passing through the network boundary over various time periods, typically 1 hour. This query yielded a large amount of flows, numbering well into the hundreds of thousands per hour. Since our flow data is unidirectional, both *inbound* and *outbound* flows were retrieved. Each flow was marked as peer (both internal and external ports 53), client (external port 53) or server (internal port 53). Next, flows were matched to create ***conversations***. A conversation describes the interaction between two nodes. For DNS, we define a new conversation for each distinct source and destination address pair. This can be done because the interaction between two addresses remains the same throughout many flows: the nodes could be peer DNS servers, or one node is a client and the other is a server. For each conversation we note the number of inbound and outbound packets and bytes. Based on

the number of peer, client and server flows between the two nodes, we can characterize the conversation as follows:

4.1 Normal conversations

- Client: Outbound client flows only
- Server: Outbound server flows with no client or peer flows
- Peer: Any outbound peer flows or both outbound client and server flows

4.2 Imbalanced conversations

- Outbound Only: No inbound flows
- Inbound Only: No outbound flows
- Packet Imbalance: Inbound and outbound packet counts differ by more than 2, and more than 10% of the bytes in one direction (inbound or outbound) are unanswered.
- Byte Imbalance: Each byte in one direction (inbound or outbound) results in more than nine bytes in the opposite direction

From normal conversations, we can categorize hosts within the AS. Imbalanced flows are set aside for later analysis of potentially malicious activity. We gather all the conversations for each node exhibiting normal DNS traffic, and apply the distinctions shown in Table 1. For quantitative analysis, observation of normal DNS patterns led to a definition of "few" as three or less, "many" to mean more than a few, and "some" to mean any non-zero amount. In order to validate these results, all nodes involved in DNS conversations were sent a DNS request for the address of www.google.com. All public-facing servers (forwarders, local servers and authoritative servers) should respond to this request, but clients should simply ignore the request by silently dropping the packet. Securely configured servers will return a negative response, which may take different forms. The DNS protocol allows an explicit SRVFAIL or REFUSED response, or an NXDOMAIN response indicating that the server does not manage the requested domain. The server may also simply return a null response, which is a well-formed reply to the query that does not contain any data.

5. Observations

The sample data used included well over 100 gigabytes of DNS data passed between an autonomous system and the Internet in more than one billion packets. Of all the traffic observed, 66% was inbound to the monitored system. Flow records for this data were assessed to define conversations. Combining inbound and outbound flows, the data yielded approximately 16 million conversations. Of these conversations, 88% were normal. 70% of the conversations were from DNS servers on the internal network. One early observation made was in the ratio of byte counts between queries and replies. Among all the DNS traffic which appeared normal, it was noted that the ratio of the sizes (in bytes) of the replies to the queries varied from 1 to 10, with an average of around 1.2. That is, on average, the responses were about 20% larger than the queries. In number of packets, the ratio was observed to be approximately 1 to 1; i.e., 1 response for each query. It is not exactly 1-1, however, as some packets may be lost or sent to servers that are temporarily unable to respond due to a high volume. The querying host may also the resent the query (an application layer resend). Some imbalance can be created in these normal situations.

Over 16 million normal conversations were then used to categorize nodes. Authoritative servers formed the largest fraction of these nodes, accounting for 26% of the total. 12% were clients, and 16% were local servers. Forwarders could not effectively be identified due to the location of instrumentation and the similarity in observed behavioral to other node types. The criteria for categorization as a client require that the node only communicated with a few external DNS servers. With conversation information, it becomes easy to extract which external DNS servers are used by internal clients. In the ideal case, clients within an autonomous system should not actually use external DNS servers; rather, they should communicate with local DNS servers within the AS and we would not observe DNS clients. When an internal client uses an external server for recursion, the external server becomes a critical part of the autonomous system. In addition, the external server has the ability to monitor the client's Internet use patterns through DNS queries it processes.

Table 1: Conversation types expected for various DNS node categorizations

Categorization	Client	Peer	Server	Discussion
Insufficient Data	Less than 100 packets			Not enough data to make a reasonable categorization
Client	Few	None	None	Clients do not host a DNS service, and will only have client conversations with preconfigured local DNS servers on the public network
Gateway	Many	None	None	Gateways proxy many client requests to the public, but are not configured with a DNS service, so they do not see peer or server conversations
Forwarder	Few	Few	None	Forwarders have peer and client conversations with many public authoritative servers,but do not respond to external DNS requests
Authoritative Server	None	Perhaps some	Some	Authoritative servers respond to public queries, which may come from peers
Local Server	Some	Perhaps some	Perhaps some	Local servers query many other servers as peers or as clients, but will never serve DNS requests to the public
Authoritative and Local Server	Some	Perhaps some	Some	If a node acts as both a client and a public DNS server, it must be proxying requests for the local network and authoritatively giving answers to the public

Clients most commonly used external DNS servers at 4.2.2.1 and 4.2.2.2, the primary and secondary Verizon servers. In addition to being the DNS servers for Verizon DSL customers, these public servers are sometimes used because the addresses are easy to remember.

Another common series of DNS servers accessed by DNS clients were prisoner.iana.org, blackhole-1.iana.org and blackhole-2.iana.org. Use of these servers potentially indicates a misconfiguration and security exposure. RFC 1918 (Rekhter 1996) reserves a small number of IP address blocks for private networks. A client on a private network will receive one of these private non-routable addresses such as 192.168.1.105. If another device on the same private network would like to find this client's DNS name, it requests the "PTR" record for 105.1.168.192.IN-ADDR.ARPA. Well-configured networks will have an internal DNS server which answers the request. However, if the request is not intercepted within the private network, the query is passed to the authoritative server for 168.192.IN-ADDR.ARPA. On the Internet, servers at iana.org will return negative responses for PTR requests against these RFC 1918 addresses. DNS servers most commonly engaged in conversations with root servers and top-level-domain servers. However, the iana.org servers also show up as common high-volume destinations. . Similar to client misconfigurations, this will happen when a local server is asked to perform a reverse lookup against a private net block which it does not administer. Rather than returning a negative response, it will forward the request out to the internet, which resolves to an iana.org DNS server.

5.1 Imbalanced conversations

Initial assumptions and a review of related RFCs indicate that most DNS traffic exhibits very predictable characteristics. Each DNS request packet sent to an available DNS server should have a corresponding DNS reply packet, and the reply should be no more than nine times larger than the request. Imbalance ratios

were calculated for all conversations (both normal and imbalanced) with authoritative DNS servers. Imbalance in a conversation is calculated as follows:

$$ByteImbalance = \frac{bytes_{in} - bytes_{out}}{bytes_{in} + bytes_{out}}$$

$$PacketImbalance = \frac{packets_{in} - packets_{out}}{packets_{in} + packets_{out}}$$

As shown in Figure 4, the thresholds for packet and byte imbalance do not exclude normal traffic; in fact, it may be possible to reduce the imbalance measures to improve anomaly detection. Characteristic imbalance measures for DNS clients and local DNS servers are also shown. Interestingly, the client graphs show some conversations in which more data is sent than received. As there are also imbalanced packet counts, this likely represents servers that are congested or temporarily unavailable. Also, a surprisingly large number of conversations have *exactly* the same number of bytes inbound as outbound.

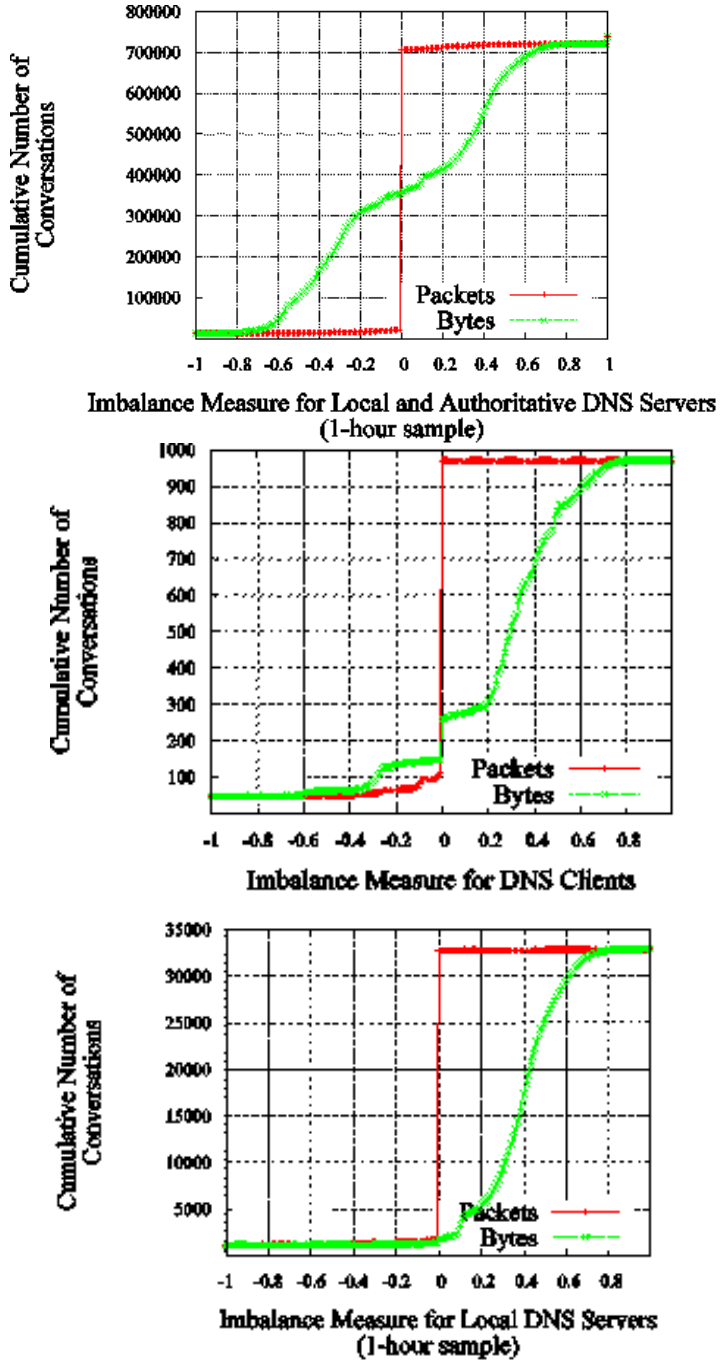


Figure 4: Byte and packet imbalance measures for authoritative DNS servers, DNS clients and local servers.

We know of some well-behaved servers that are widely known and should always exhibit normal behavior.* Table 2 lists 28 well-known servers which we will use as a benchmark for expected good DNS behavior. These servers were widely used within the monitored autonomous system.

5.2 Scanning

For the context of this report, we have defined "scanning" as a DNS packet which does not have a matching request or response. The typical DNS scanner will send a packet to port 53 on a random host; if the host emits a response, the scanner knows the host is running a DNS service. As most firewalls are fairly permissive with allowing DNS traffic, some additional scan types may become evident. For example, an external scanner may send a packet ***from*** UDP port 53 (DNS) ***to*** UDP port 1434. Permissive firewalls would admin the packet, assuming it was a response to a previous DNS query. However, a response from the scan target could actually indicate that it hosts a Microsoft SQL server, which often runs a service on UDP port 1434. The majority of imbalanced conversations were due to inbound scanning. We can expect that external hosts scanning for DNS services inside our AS will attempt to contact a large number of addresses with a small number of packets. Many of these addresses should have no other DNS traffic. In fact, for every address holding a legitimate DNS server, we observed inbound scan attempts to 100 other non-DNS addresses.

When intersecting known good servers with the list of scanners, we found a number of internal hosts apparently being scanned by these known good servers. Often more than one known good server would scan the internal host, and some of the scan conversations consisted of thousands of DNS packets. Most of the internal nodes involved also showed high normal DNS traffic volume. Rather than being actual scanning activity, this is more likely to indicate missing data. When looking at outbound scan activity, we should expect true outbound scanners to have some valid conversations interspersed with a large number of outbound-only scan conversations. On the other hand, DNS clients and local servers will occasionally have outbound-only scan conversations when contacting off line DNS servers. Amongst the number of internal nodes involved in outbound scanning activity, 34% did not have any legitimate DNS conversations. This also seems to indicate missing inbound data.

5.3 Imbalanced conversations

Imbalanced DNS traffic was examined as well. Our initial hypothesis was that highly asymmetrical conversations were likely to be the result of non-DNS traffic traveling on port 53. However, with one exception, all of the nodes exhibiting unbalanced traffic also showed normal traffic. The one exception appears to have been a server updating its root zone file from a root DNS server. Further analysis showed that the most common imbalanced conversations still involve at least one known good DNS server, indicating at least some legitimate DNS traffic. The largest source of imbalanced traffic came from conversations with root and top level domain servers, presumably to get an updated list of top level domain servers. Another large source of imbalance was from DNS servers owned by global content delivery networks. These networks often use DNS tricks to provide content from physically nearby nodes.

5.4 Validation

Results of the categorization using flow data were matched with the active DNS query results. Analysis showed that, as expected, most nodes did not return a response, and are apparently clients. However, of the servers that did respond to the query, a surprisingly large number responded with the IP address for www.google.com. Results were consistent with those previously observed by (The Measurement Factory 2005). As outlined in (US-CERT 2006), all DNS servers should be configured to deny requests against domains for which they are not authoritative.

---

* Since we are concerned with UDP DNS traffic, it may be possible to spoof DNS traffic from a well-known server, which might result in imbalanced conversations. However, this is a relatively sophisticated attack, so we should not expect much spoofed traffic.

Table 2: DNS servers which should exhibit normal traffic

DNS Name	IP Address
a.root-servers.net	198.41.0.4
b.root-servers.net	192.228.79.201
c.root-servers.net	192.33.4.12
d.root-servers.net	128.8.10.90
e.root-servers.net	192.203.230.10
f.root-servers.net	192.5.5.241
g.root-servers.net	192.112.36.4
h.root-servers.net	128.63.2.53
i.root-servers.net	192.36.148.17
j.root-servers.net	192.58.128.30
k.root-servers.net	193.0.14.129
l.root-servers.net	198.32.64.12
m.root-servers.net	202.12.27.33
a.gtld-servers.net	192.5.6.30
b.gtld-servers.net	192.33.14.30
c.gtld-servers.net	192.26.92.30
d.gtld-servers.net	192.31.80.30
e.gtld-servers.net	192.12.94.30
f.gtld-servers.net	192.35.51.30
g.gtld-servers.net	192.42.93.30
h.gtld-servers.net	192.54.112.30
i.gtld-servers.net	192.43.172.30
j.gtld-servers.net	192.48.79.30
k.gtld-servers.net	192.52.178.30
l.gtld-servers.net	192.41.162.30
prisoner.iana.org	192.175.48.1
blackhole-1.iana.org	192.175.48.6
blackhole-2.iana.org	192.175.48.42

Table 3: Summary of responses to DNS queries

Categorization	No Answer	Responded
Gateway	89%	11%
Authoritative and Local Server	5%	95%
Authoritative Server	5%	95%
Client	99%	1%
Local Server	59%	41%
Insufficient Data	91%	9%

Table 3 shows a comparison between active DNS query results and passive categorization from netflow data. Overall, the method appears to work well for separating authoritative servers, local servers and DNS clients. Some difficulties are encountered with gateways, however, which may be improved by examining data over a longer time period. In general, this data corroborates the accuracy of the categorization as well in that servers act like servers, and clients act like clients.

5.5 Weekly statistics

The analysis was repeated to gather daily statistics for seven-day period. Due to the large volume of DNS traffic, conversations for each node could not be processed in parallel. Rather, categorizations for individual nodes were gathered for each day, and daily results were then combined. For example, if a node was classified as a "Gateway" on Monday and a "Local Server" on Tuesday, the overall categorization would be "Local Server". As expected, the passive categorization improves identification of DNS nodes which exhibit only a small amount of traffic. The number of gateways almost doubled, and clients tripled. However, the number of servers (both local and authoritative) only increased by 25%.

We can also examine how categorizations changed for the nodes we cataloged in the initial one-day data snapshot. Table 1 shows how categorizations evolved from one day's worth of traffic data to an entire week. As expected, a number of nodes with insufficient data were properly classified.

Table 4: Change in categorization of DNS nodes when moving from a 1-day to a 7-day view of traffic

Categorization after one day	Recategorization after one week	Pct
Authoritative and Local Server	*stayed the same*	100%
Authoritative Server	*stayed the same*	92%
	Authoritative and Local Server	8%
Client	*stayed the same*	86%
	Gateway	14%
Gateway	*stayed the same*	94%
	Authoritative and Local Server	5%
	Local Server	1%
Local Server	*stayed the same*	85%
	Authoritative and Local Server	15%
Insufficient Data	*stayed the same*	68%
	Authoritative and Local Server	1%
	Authoritative Server	1%
	Client	11%
	Gateway	17%
	Local Server	2%

6. Assessing the threat

Our observations lead to three primary threats against the DNS infrastructure for a typical large autonomous system. First, many DNS servers allow public DNS recursion. Second, many nodes within the autonomous system will pass reverse lookups for private address space out to the Internet. Finally, many nodes depend on external DNS servers for recursion and name resolution. The following sections discuss identification and mitigation of these threats within a large autonomous system using net flow data.

6.1 Public recursion

6.1.1 Threat

Recursive DNS servers present a potential for a denial-of-service amplification attack. In the typical scenario, an attacker will send a DNS request to the recursive server, spoofed to appear as though it came from the target. The recursive server will then respond to the target. When a large number of servers are involved, it becomes possible to saturate the target's internet connection. Recursive servers may also be susceptible to cache poisoning, a much more difficult exploit, but also one with much more serious consequences. For more information, see (Schuba 1993) and subsequent related works.

6.1.2 Detection

Both denial-of-service attacks and cache poisoning attacks depend on sending a large volume of DNS traffic to an individual node. Using the method described for DNS node characterization as described above, these attacks will be identified by an abnormally large number of DNS requests for a short period of time.

6.1.3 Mitigation

Specific recommendations for preventing recursion on Microsoft DNS servers can be found at http://technet2.microsoft.com/WindowsServer/en/library/e1fe9dff-e87b-44ae-ac82-8e76d19d9c371033.mspx?mfr=true. Team Cymru maintains a secure BIND template at http://www.cymru.com/Documents/secure-bind-template.html which disables recursion.

6.2 Reverse lookups for private address spaces

6.2.1 Threat

Reverse lookups against private address spaces expose internal network information to the Internet. These records can be used to fingerprint the originating operating system, identify when systems are online, and map out host names for private networks. Additional details can be found in (Broido 2006). Reverse lookups to the RFC1918 address space are handled within the address block 192.175.48.0/24 (AS112). As this uses anycast addressing, routes to the net block can be freely advertised, allowing misrouting of this private information.

6.2.2 Detection

Reverse lookups can be detected by simply looking for DNS traffic to the 192.175.48.0/24 net block. No legitimate DNS traffic should be observed with these addresses.

6.2.3 Mitigation

A secure configuration will keep reverse lookups on the internal network. For Microsoft clients, the necessary changes are described in (CAIDA 2006) In addition, all local DNS servers should be configured as authoritative for all RFC 1918 reverse lookup zones. Finally, to protect against potentially unknown network configurations, outbound DNS (both UDP and TCP) to 192.175.48.0/24 should be blocked.

6.3 Dependence on external DNS servers

6.3.1 Threat

DNS builds the critical link between names and actual IP addresses for most IP resources. By forwarding all DNS queries to an external server, responsibility is shifted to a server which has no relationship or obligation to the autonomous system. This could present a threat to the confidentiality, integrity and availability of all DNS requests, and subsequently, most IP communications.

6.3.2 Detection

Using the methods described above, we have identified DNS clients; high volume clients should be reviewed to determine if they are actually forwarding queries to external servers.

Any DNS client sending more than a trivial amount of requests to 4.2.2.1 and 4.2.2.2 should be reviewed.

Any local DNS server in which the most commonly contacted external server shows exceedingly high volume conversations to one or two nodes--particularly when other conversations are trivial--should be reviewed.

6.3.3 Mitigation

Strict firewall rules for DNS traffic would mitigate the use of external servers; however, in very large autonomous systems, this may not be possible. Policies must be put in place to ensure the proper configuration of internal DNS servers, and policies must be enforced through detection and correction of misconfigured servers.

7. Conclusion

In this paper a study of DNS traffic into and out of a large autonomous system has been discussed. Observation of DNS traffic shows that it is very predictable. In particular, in almost all cases, every DNS packet sent to a server will result in a single packet in response sent back to the client.

The following three problems and strategies for mitigating their severity were presented:

First, many DNS servers will make recursive requests in response to public queries, even if they are not authoritative for the requested domain. This presents a security concern for distributed denial-of-service attacks and possible cache pollution. Second, the use of external DNS servers within the monitored autonomous system was greater than expected. This extends a key network resource beyond the control of the system's administrators.

Finally, a large number of conversations were observed between the autonomous system and prisoner.iana.org, blackhole-1.iana.org and blackhole-2.iana.org. This traffic indicates private internal address information is unintentionally being exposed to the public Internet.

Some strategies for mitigating these problems were presented. Application of security templates, egress filtering and general adherence to industry best practices will significantly improve security posture.
Possibilities for future work include a thorough investigation of imbalanced traffic. This traffic most often comes from normal but rare conversations with well known servers. Once these well known servers have been identified, and problems with missing data have been corrected, this technique is likely to be useful in identifying non-DNS traffic traveling on UDP port 53. Another possibility for future work would be to extend this research to include DNS traffic on TCP port 53.

References

Abley, J. (2006) "I'm Being Attacked by PRISONER.IANA.ORG!", *Network Working Group Internet-Draft draft-jabley-as112-being-attacked-help-help*.

Abley, J., Maton, W. (2006) "AS112 Nameserver Operations", *Network Working Group Internet-Draft draft-jabley-as112-ops-00.txt*.

Broido, A., Shang, H., Fomenkov, M., Hyun, Y., Claffy, K (2006) "The Windows of Private DNS Updates", *ACM SIGCOMM Computer Communication Review*, July 2006.

CAIDA (2006) "How to Disable Dynamic DNS Updates on Windows ", [online], http://www.caida.org/research/dns/disable_dns_updates.xml

"Cisco IOS NetFlow", [online], Cisco, http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html.

Collins, M., Kompanek, A., Shimeall, T. (2006). "Using SiLK for Network Traffic Analysis", [online], CERT, http://tools.netsa.cert.org/silk/analysis-handbook.pdf

Cranor, C., Gansner, E., Krishnamurthy, B., Spatscheck, O. (2001) "Characterizing Large DNS Traces Using Graphs", *Proceedings of ACM Sigcomm Internet Measurement Workshop 2001*, November 2001.

Kurose, Ross (2003) *Computer Networking: a top-down approach featuring the Internet*, Addison-Wesley.

The Measurement Factory (2005) "DNS Survey: June 2005", [online], http://dns.measurement-factory.com/surveys/200506.html

Mockapetris, P. (1987) "Domain Names - Concepts and Facilities", *RFC-1034*.

Mockapetris, P. (1987) "Domain Names - Implementation and Specification", *RFC-1035*.

Rekhter, Y., Moskowitz, B., Karrenberg, D., deGroot, G., Lear, E. (1996) "Address Allocation for Private Internets", *RFC-1918*.

Schuba, C. (1993) "Addressing Weaknesses in the Domain Name System Protocol", *Master's thesis, Purdue University, Department of Computer Sciences*.

US-CERT (2006). "The Continuing Denial of Service Threat Posed by DNS Recursion (v2.0)", [online], US-CERT, http://www.us-cert.gov/reading_room/DNS-recursion033006.pdf

Development of a Defensive Cyber Damage Assessment Framework

Larry Fortson and Michael Grimaila
Air Force Institute of Technology, Fairborn, OH, USA
larry.fortson@afit.edu
michael.grimaila@afit.edu

Abstract: Information is a vital resource used in all aspects of modern military operations. Information is collected, processed, analyzed, and distributed to support situational awareness, operations planning, intelligence, and command decision making. The dependence upon information creates significant operational risk to the organizational mission when an information incident occurs. In this paper, we discuss the findings of a US Air Force research project focused on improving the defensive cyber damage assessment process. We identify gaps in the risk management process, discuss the existing incident response and damage assessment process, and propose changes to the damage assessment process as well as identify organizational implementation challenges. The results of the research provide a roadmap for the implementation of a real-time situational awareness tool that maps the technical impact of an information incident to an impact on mission capability.

Keywords: Damage assessment, information asset valuation, decision making

1. Introduction

Information has become a critical asset to every modern organization (Denning, 1999). The increased use of automation and networking in core operational processes have created an environment where information compromise, damage, loss can equate to mission failure (Kemmerer, 2003). A cyber attacker's goal is to disrupt the organization's mission by disrupting its information systems (Jajodia et al., 1999); making the need to protect and defend information assets in cyber space a paramount requirement (GAO, 2005). Inevitably, an organization is forced to deal with an information incident, whether by inside or external malicious intent, accident, or natural disaster. When this happens, the organization's decision makers need rapid and accurate visibility of how the incident impacts the organizational mission. Success in military operations depends on rapidly understanding how cyber incidents affect the organization's ability to accomplish the mission. For this reason, this paper tightly focuses on cyber information as an asset, and a framework for its valuation and subsequent devaluation following an information incident.

1.1 The need for a defensive cyber damage assessment framework

From the beginning of organized warfare, commanders have attempted to assess the impact of offensive actions against the enemy's war fighting assets (Diehl and Sloan, 2005). In the cyber realm, we are finding it difficult to perform defensive damage assessment following an information compromise. In 1996, the Department of Defense (DoD) conducted a series of "day after" games to measure their ability to respond to cyber attacks. These exercises revealed that the DoD was not ready to respond effectively to such attacks and cited four critical issues that must be addressed to ensure to improve the DoD response to cyber attacks (Alberts, 1996). Among these was the need for "a 'battle damage' assessment process suitable for IW" (Alberts, 1996). Ten years later, there still exists no standardized operational damage assessment model for information compromises on United States Air Force (USAF) networks (Theim, 2005). Despite the need to understand the organizational impact caused by information incidents, surprising little research has been conducted focused in this area. Existing works tend to focus on the financial impact to for-profit organizations (Rezmierski et al., 2000). Another contributing factor is that economic measurements are more tangible than other forms of impact metrics (Horony, 1999). However in a military operations context, economic metrics do not provide command decision makers with all of the information necessary to make smart and timely decisions when their organization is exposed by an information compromise.

1.2 Foundational terminology

Defining a canonical terminology is essential when communicating ideas to diverse communities of interest. For this reason, we now define the terminology used in the Defensive Cyber Damage Assessment (CDA-D) model and framework proposed in this paper. First, we define the scope and purpose of defensive cyber damage assessment. Joint Publication 1-02 defines military damage assessment as "an appraisal of the effects of an attack on a nation's military forces to determine residual military capability and to support planning for recovery and reconstitution" (JP1-02, 2006). Historically, the focus of damage assessment has been on the effects of offensive actions against the mission capability of enemy forces. Conversely, our work

is focused upon defensive damage assessment which appraises the effects of a cyber-based incident that potentially impacts friendly mission capability. For the purposes of our research, a "mission" describes the overall purpose of the organization. Mission can also be used in a similar context to define the goals and objectives of a specific department, group, or unit within the organization. Thus, the overall mission of an organization is comprised of a hierarchy of subordinate missions. This hierarchy is an "essential component of operational effectiveness" (Alberts and Dorofee, 2005). A mission is supported by one or more operational processes. Operational processes are those processes that enable people or systems to accomplish the mission. In modern organizations, most operational processes are supported by information processes. Information processes are those information flows that support the operational process. An organization's information processes depend on information assets. An information asset is a set of information that holds value to the organization's mission. We recognize a distinct difference between data and information. We define data as the elemental subset of information which has no inherent value but is dependent on external application to obtain value. Information is aggregation of data that are grouped in such a way that it has inherent and independent contextual value (Petrocelli, 2005).

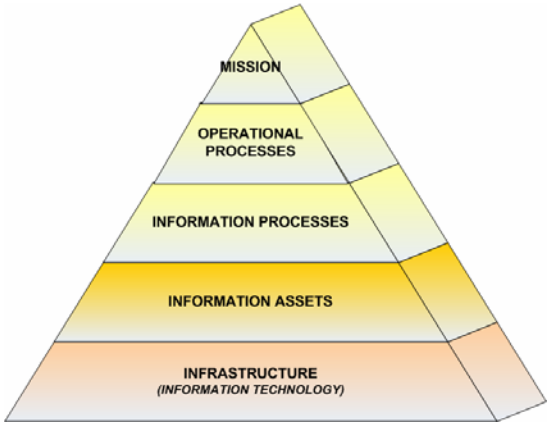


Figure 1: Mission to infrastructure hierarchy

All information assets have producers, owners, consumers, and custodians. The information producer is the creator of the information or the originator of an information flow. The information owner is the entity that bears responsibility for determining the classification, value, and level of protection of the information asset commensurate with its value. The information custodian is the entity responsible for implementing the security measures that protect the information asset. An information consumer is any entity that maintains transitive dependencies on the information. It is important to understand that the information producer, owner, custodian, and consumer are all closely related; and they can be the same entity.

2. Background

2.1 Information taxonomy

The information taxonomy is developed as information progresses from its atomic state, as data, to its ultimate form as tacit knowledge. Data is a subset of information, and is often the focus of traditional cyber security schemes. However, data has no inherent value as it is solely dependent upon its external application to produce value (Petrocelli, 2005). Human utility organizes and aggregates data into usable groupings of contextual relationships that endow the data with "relevance and purpose" (Spiegler, 2000). Thus, data becomes information which, by its nature is inherently associated with meaning (Bourdreau and Couillard, 1999). Information, not data, is the core of this taxonomy, as it is the basic unit that contributes to the development of knowledge for use in all forms of decision making. Knowledge is information further aggregated into a more usable form that can be made actionable to provide value to the organizational mission (Kanter, 1999; Vail, 1999). The catalyst for this metamorphosis is human utility, and the ensuing information taxonomy reflects the increasing value as information is transformed and used within the organization for decision making (Spiegler, 2000). Because information is the core asset of cyberspace, it must be the center of gravity in security planning and risk management if accurate and timely cyber damage assessment is to be possible.

2.2 Information production, consumption and relative ownership

Information production, ownership, and consumption are easily and frequently confused. Information asset ownership is frequently assigned in organizations without regard to who created the information or where it

originated. These issues are often irrelevant as long as the ownership of the information asset is established and the owner has responsibility and authority to perform ownership duties (Stevens, 2005). In the complex domain of military cyber information flow, the roles and responsibilities of each group, by necessity, do not adhere to traditional definition. This is especially true of the relationship between the information producer, information owner, and information consumer, where the clearly defined roles of consumption and ownership become relative to need. Information asset ownership gains a new fluidity, and becomes relative to the contextual value within an organization. In a military context, an organization may receive intelligence information input from multiple external organizations, services, and countries. External information producers classify the information at the point of origin, but the classification only serves as a baseline valuation of the asset. As the organization stores and uses the information, the information also holds a contextual value depending on the asset's support of the organizational mission. At this point, the organization becomes more that just a consumer of the information asset; now the consumer is a *relative owner*. Relative ownership equates to static ownership, local to an organization for the purposes of risk management and damage assessment. Therefore, the information owner is responsible for identification, definition, valuation, and documentation of all information assets owned. Each site where information asset ownership occurs, whether static or relative ownership, asset profiling must be accomplished by the information owner. The owner bears the burden since only the owner maintains a perspective that allows him to understand how the information is used to support the organization's mission, what the value of information asset is to the organization, and where the information asset is stored.

2.3 Information asset value constructs

Determining the value of information is a complex task, due to its innately intangible qualities (Van Alstyne, 1999). Many valuation models rely on economic metrics for information value appraisal. In the military, the intangible value of information often far exceeds its economic value. The DoD possesses a distinct advantage in determining a baseline for the value of its information assets. All information stored on its networks is assigned classification through its uniform system for classifying, safeguarding, and declassifying national security information (EO13292, 2003). However, this only provides a coarse "first cut" for determining the value of information. In military cyber operations, classification of an information asset will be less helpful in determining the impact of an information compromise due to the requirements of classifying a system at the highest level of classification of the information processed or stored on it. In general, the majority of the data will be of equal classification creating a need for a more accurate construct to assess the value of the information asset.

Contextual value is the most important component in information asset valuation. The constructs of the contextual value of information are mission binding, age, and state. Mission binding is a measurement of how closely the information asset is bound to the organization's mission through its supporting information process. An asset that is closely bound to an operational process is assigned a relatively high value, if it critical to the organization's mission the valuation will be even higher. Age is a concept common to all lifecycles (Petrocelli, 2005). As information ages, its mission binding may change. State is the most fluid of an information asset's contextual value constructs. It is comprised of the information asset's level of confidentiality, integrity, and availability. An information incident can impact the state of the information asset causing it to be of less value to the information owner. During initial valuation, it can be assumed that the state constructs are intact unless there is reason to believe otherwise.

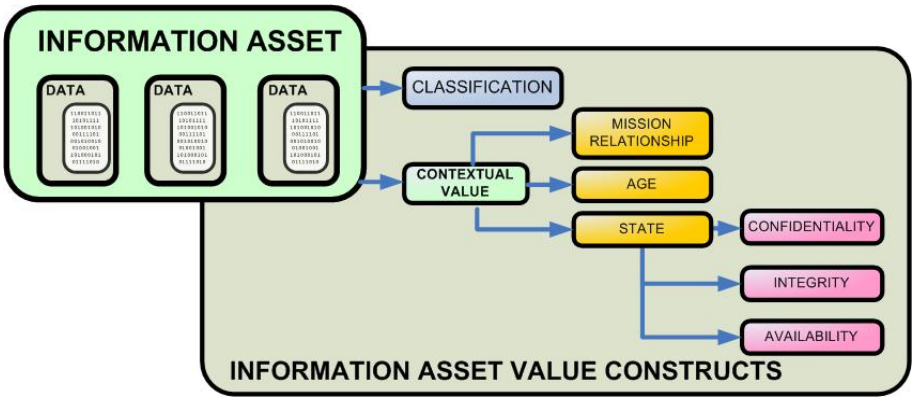


Figure 2: Information asset valuation constructs

2.4 Risk management

Risk management of cyber information assets is essential to building a foundation for an effective security program and making accurate cyber damage assessment (SooHoo, 2000). Risk management is the process of identifying and assessing the risks to the organizations information assets; and applying appropriate mechanisms to reduce, manage and control risks to information assets Many enterprise 'flirt' with the idea of cyber asset risk management, but few actually implement it effectively (Hampton, 2006). Those that do, often use a traditional risk management framework that focuses on technology-based infrastructure, rather than an effective asset-focused cyber risk management such as Carnegie Mellon's OCTAVE (Alberts et al., 2003). Many implementations of the traditional risk assessment produce a narrow focus on technical vulnerabilities and the associated exploit activities (Howard, 1997). Focusing solely on technology overlooks many key factors regarding organization's information assets (Baskerville and Im, 2005, Spears, 2006) . An asset-focused framework, however, enables identification of risks from an asset perspective, facilitating identification, valuation, and documentation of the organization's information assets; laying the foundation for defensive cyber damage assessment.

2.5 Existing damage assessment on Air Force networks

There has been some effort within the Air Force to measure the impact to organizational mission following an information incident. However, the methodologies being employed are ad-hoc and without validation (Theim, 2005). The metrics used to assess damage are economic, in terms of recovery costs, and infrastructure availability. The Air Force Network Operations Center (AFNOC) Network Security Division (NSD) is the agency responsible for leading incident response efforts on Air Force networks and regularly is requested to assess an the impact to an organization's mission capability following a successful compromise. This is an extremely difficult and often impossible task under the current implementation of security management on Air Force networks. This section will briefly examine some of the key areas that confound efforts to perform defensive cyber damage assessment, to include mission capability impact assessment, on Air Force networks.

2.6 Infrastructure-focused approach to network defense

The Air Force implements traditional, infrastructure-focused, network defense architecture. This traditional view of "cyber protection" is focused on the technology-based infrastructure, rather than on the information assets within. This infrastructure-focused view acknowledges data existence, but substitutes the data's value with that of the infrastructure; and implies that by protecting the infrastructure, the data is protected This view has merit, but the assumption that technology is an equitable substitute for the information follows a proven path of failure (Davenport and Prusack, 1998); and introduces great limitations when attempting to perform damage assessment. When an information incident occurs, commanders are unable to get an accurate or timely understanding of how their mission is impacted. One of the central reasons for this is that the traditional view relegates data as an incidental factor without appreciation for the value it provides as information to the organization's operations. As a result, critical information assets are not viewed in a way that requires identification and documentation of the asset as would be done with a physical mission critical asset. When an information incident occurs, the Incident Response Team (IRT) is forced to conduct a mission impact assessment with little or no documentation that shows how the information supports the organizational mission. As a result, the primary assessment is based upon economic factors (remediation and recovery costs) and availability. Subsequently, an effort is made to identify and quantify the impact by contacting a representative within the information owner's organization. Unfortunately, this often leads to a subjective assessment and unreliable assessment of impact.

The following notional example illustrates the problem with the current damage assessment scheme used within the Air Force. Consider a situation where suspicious activity has confirmed the compromise of multiple passwords to military systems. The systems are multifunction, but have access to systems that are critical to the organization's mission. The IRT determined that at least one password allowed administrator access to the system and the mission critical systems were accessed numerous times. All findings, including time of access, information accessed, and other important information, are reported through the standard reporting chain, and the commander at the site of the incident is included. The commander requests more information from the IRT regarding how the incident affects his mission. The IRT is unable to provide this assessment as there is no documentation of how the information provides value to the mission. As a result, the commander may decide that since his system's availability is intact, no action is necessary. The truth, however, may be that the confidentiality of a planned operation was breached which undermines the OPSEC of the mission.

2.7 Lack of effective "Cyber" risk management

Virtually all contemporary security planning methodologies include risk management as the foundation for a success security program (SooHoo, 2000). The Air Force understands the importance and benefits of risk management in operations, but fails to perform effective risk management of its cyber assets. What risk management is accomplished is driven by the infrastructure focused mindset, resulting in documentation of technology rather than information assets. This is analogous to a risk management program documenting only the hangars on a flight line, but ignoring the aircraft within. If the hangar is destroyed, the commander will need to know how many flying and support assets were lost in order to assess his current mission capability. By simply documenting the container and infrastructure, the hangar facilities in this example, there is no way for the commander to rapidly and accurately know how the incident has affected his mission. Likewise, when a piece of cyber infrastructure such as a server has an information incident, only in rare occasions does the information owner understand which assets were impacted. By failing to adequately implement enterprise risk management of its cyber information assets, the Air Force is finding itself blind, at both the unit and enterprise level, of the value of the information it depends upon. As a result, poor command decisions may be made based upon an inaccurate perception of the cyber battle space.

2.8 Lack of information asset documentation

A lack of usable documentation of cyber information assets is a considerable problem that must be addressed in order to enable effective defensive cyber damage assessment. The Air Force relies on its Operations Security (OPSEC) program to identify document critical information resources within an organization. Unfortunately, this program fails to adequately document cyber information resources in a manner that facilitates understanding mission impact following an information compromise. The traditional approach to cyber asset damage assessment attempts to determine the damage caused by an information security incident through assessment of impact to technical and/or infrastructure. Since neither the OPSEC program nor the limited risk assessment activities on Air Force networks produce documentation of the cyber assets used by an organization, the ability to accurately measure mission impact following an information compromise is not possible using the current infrastructure-focused approach to damage assessment.

3. Proposed defensive cyber damage assessment framework

The proposed CDA-D framework is comprised of three main stages: pre-incident activities, response activities, reporting. Information asset identification and valuation occurs in the pre-incident stage, damage assessment occurs in the response stage, and mission impact reporting occurs in the final stage. -The foundation of defensive cyber damage assessment must be laid before the incident occurs. For accurate defensive damage assessment to even be possible, the organization must have developed a portfolio of standardized profiles of its information assets before the incident occurs. When the incident occurs, the information owner and the incident response team work together to determine what information assets were affected by the incident. The information owner may then use the asset profiles of the effected assets to determine the devaluation of these assets from their documented state; and then quickly determine the impact to mission capability. This damage assessment may then consolidated and reported through existing reporting channels to provide fast notification and reaction to the incident. Remediation and recovery occur immediately, but forensic and accountability investigation may continue for an extended time. For this reason, the latter two stages will commonly overlap.

3.1 Pre-incident information asset identification methodology

All assumptions of the CDA-D framework are based on the identification and valuation of the information assets before an incident occurs; therefore the preparatory activities an organization takes before an incident occurs are essential to the successful and accurate cyber damage assessment. Asset identification must be accomplished through effective, asset-focused risk management by the information owner. A validated and asset-focused risk management framework such as OCTAVE can effectively identify and document information assets that are valuable to the organization and focus risk analysis activities on the critical assets identified (Alberts et al., 2003). However, the methodology we advocate is a top-down, operations-oriented, asset-focused approach. We provide a high-level view of the main steps here:

- Identify and document the organizational mission(s).
- Identify, define, prioritize, and document the operational processes that support the mission(s).
- Identify, define, enumerate, prioritize, and documents the information processes that support operational processes.

- Identify, define, document, and valuate the information assets that the information processes depend upon.

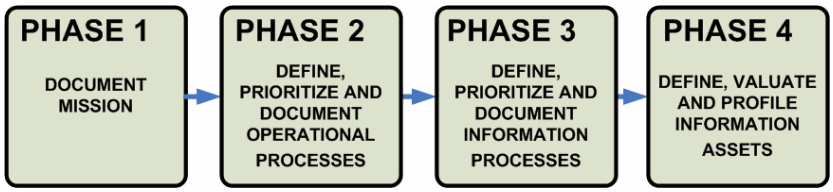


Figure 3: Information asset identification process

In the first step, the information owner must have a clearly defined understanding of the organizational mission. From there, the operational processes that allow the organization to accomplish the mission can be identified. These operational processes must be defined, identified, prioritized according to their respective criticality to the mission, and then documented accordingly. After each operational process is documented, the information processes that support the operational processes are defined, identified, prioritized and documented in a like manner. When this is accomplished, the information owner can identify and profile the information assets that support each or many of the information processes. Information asset profiling is critical to accurate defensive cyber damage assessment when an information incident occurs. Using an operations-oriented approach to information asset identification allows the owner to identify not only the asset's container and value, but the direct mission functions that the asset supports. When an incident occurs, this allows rapid mapping from effected asset to mission impact. The information owner must maintain the information asset profile portfolio, and assign effective security controls to protect against unauthorized access and disclosure.

3.2 Incident response

In the CDA-D framework, there is no noticeable change to the function and activities of the IRT, as currently implemented in the Air Force. The incident response function will remain focused on the traditional activities of response, remediation, and forensic investigation. However, rather than being tasked with determining the impact of an information incident to an organization's mission, the IRT will work closely with the information owner, providing the technical details to allow the information owner to perform damage assessment at the site of the incident. The constructs of an asset-focused pre-incident valuation will allow the IRT and information owner to determine asset devaluation resulting from, not only asset availability, but also confidentiality and/or integrity compromise—an aspect not attainable under the current, infrastructure-focused assessment attempts.

3.3 Reporting activities

Current reporting requires consolidated reporting of information incidents through the AFNOC structure as directed by Air Force Instruction 33-138, Enterprise Network Operations Notification and Tracking (AFI33-138, 2004). Reports consist of an initial incident report to notify parties of interest, an interim report to notify all parties of interest of updates and actions, and a final report at the closure of incident actions. Parties of interest include the NETOPS command and control structure, effected information asset owners and custodians, and all known consumers. Consumers that possess upstream or downstream relative ownership of the effected information asset must assess any mission impact (via asset devaluation assessment) resulting from the incident, and report the results of their local damage assessment.

In most cases, remediation and recovery will begin as early as possible allowing less down time and increased mission continuity. Investigation by the IRT during the incident response stage will facilitate determination of the compromise cause and size. This information must be included in reporting to re-orient security posture and hold responsible parties accountable to prevent repeated security failures.

3.4 Domain of operations

The CDA-D framework can be employed in all three domains of operations: tactical, operations, or strategic. Assessment in the tactical domain deals with real-time information, such as UAV feeds which aid in rapid and accurate battlespace decisions. However, due to the fluid nature of tactical information assets, traditional information response activities are appropriate. The value constructs most likely to be affected by an incident are availability and confidentiality. These must be assessed by the information owner in 'real time' depending

on the pre-determined devaluation model based on predictive statistical distributions of incident effect probability.

Assessment in the operational domain deals with 'standard' scenario where traditional incident response functions are is performed as normal. The IRT works with the information owner to facilitate determination of affected assets. The information owner must determine any asset devaluation of assets identified and documented in the risk assessment process. This devaluation assessment is critical to determining operational impact which is provided to IRT for consolidated reporting and is combined with the technical impact assessment to determine mission impact. Mission impact reporting must be provided to, not only the information owner, but also downstream consumers with transitive dependencies on the affected information assets. Reporting is cyclic, consisting of a minimum initial report, interim report(s), and a conclusive, post-forensic final report.

Strategic assessment is focused accountability to assure that lessons learned from the incident influence future IT and security planning and investment. Traditional economic metrics can be used to determine budgetary impact, which potentially affects such strategic factors as customer and peer confidence. The strategic assessment must consider accountability to prevent future occurrences of an information incident through documenting how and why the incident occurred.

4. Conclusions

In this paper, we reviewed the existing USAF cyber damage assessment process, identified gaps in risk management process, and proposed the implementation of the Defensive Cyber Damage Assessment (CDA-D) model. While the need for effective cyber damage assessment was articulated for more than a decade, little research has been conducted to attain this objective. The explosive growth of military operations and their dependency on cyberspace has awakened commanders to the shortcomings of current damage assessment capabilities.

We have identified that the USAF's distinction between information and information technology is obscured and is negatively impacting its approach to cyber security by placing the focus on information infrastructure rather than information assets. We stressed the importance of deliberate security planning through an operations-oriented, asset-focused risk management process to lay the foundation for cyber damage assessment that maps to mission capability impact. An asset-focused risk assessment facilitates the identification of information assets within the infrastructure to be documented and valuated. This allows the incident response function to work directly with the information owner to assess the mission impact resulting from the incident. We believe that the proposed changes will dramatically improve the quality and timeliness of defensive cyber damage assessment within Air Force and enable the development of a real-time situational awareness tool to provide commanders with a detailed understanding of cyber attacks in terms of their mission capability.

Disclaimer

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the U.S. Government.

Acknowledgements

This work was supported by a research grant from the Air Force Research Laboratory (H98230-Q206-2211).

References

Afi33-138 (2004) air force instruction (afi) 33-138 enterprise network operations notification and tracking. In information, c. A. (ed.) F ed., hq usaf/ilco.

Alberts, c. J. & dorofee, a., j. (2005) mission assurance analysis protocol (maap): assessing risk in complex environments. *Networked systems survivability program.* Carnegie mellon university.

Alberts, c. J., dorofee, a., stevens, j. & wooky, c. (2003) introduction to the octave approach. Pittsburgh, pa, carnegie mellon university.

Alberts, d. S. (1996) *defensive information warfare*, national defense university press.

Baskerville, r. L. & im, g. P. (2005) a longitudinal study of information system threat categories: the enduring problem of human error. *Data base for advances in information systems,* 36**,** 68-79.

Bourdreau, a. & couillard, g. (1999) system integration and knowledge management. *Information systems management,* fall**,** 24-32.

Davenport, t. H. & prusack, l. (1998) *working knowledge: how organizations manage what they know,* boston, harvard business school press.

Denning, d. (1999) *information warfare and security,* upper saddle river, nj, pearson.

Diehl, j. G. & sloan, c. E. (2005) battle damage assessment: the ground truth. *Joint force quarterly***,** pp. 59-64.

Eo13292 (2003) executive order 13292 - further amendment to executive order 12958, as amended, classified national security information.

Gao (2005) weaknesses persist at federal agencies despite progress made in implementing related statutory requirements. *Information security.* Washington, d.c., united states government accountability office.

Hampton, j. J. (2006) emerging risk strategies. *Business insurance,* 40**,** 33, 1pgs.

Horony, m. D. (1999) information system incidents: the development of a damage assessment model. *Gir.* Wright patterson air force base, oh, air force institute of technology.

Howard, j. D. (1997) an analysis of security incidents on the internet 1989-1995. Pittsburgh, pa, carnegie mellon university.

Jajodia, s., ammann, p. & mccollum, c. D. (1999) surviving information warfare attacks. *Computer,* 32**,** pp. 57-63.

Jp1-02 (2006) joint publication (jp) 1-02: department of defense dictionary of military and associated terms. 14 april 2006 ed., united states department of defense.

Kanter, j. (1999) knowledge management, practically speaking. *Information systems management,* fall**,** 7-15.

Kemmerer, r. A. (2003) cybersecurity. *International conference on software engineering (25th).* Portland, or, ieee computer society.

Petrocelli, t. D. (2005) *data protection and information lifecycle management,* upper saddle river, new jersey, pearson education, inc.

Rezmierski, v., carroll, a. & hine, j. (2000) incident cost analysis and modelling project (i-camp ii): a report to usenix association. University of michigan.

Soohoo, k. J. (2000) how much is enough? A risk management approach to computer security. Consortium for research on information security and policy (crisp), stanford university.

Spears, j. L. (2006) the effects of user participation in identifying information security risk in business processes. *Special interest group on computer personnel research annual conference.* Claremont, ca, acm press.

Spiegler, i. (2000) knowledge management: a new idea or a recycled concept? *Communications of the association for information systems,* 3**,** 20.

Stevens, j. F. (2005) information asset profiling. Pittsburgh, pa, carnegie mellon university.

Theim, l. (2005) a study to determine damage assessment methods or models on air force networks. *Department of engineering and management.* Wright patterson air force base, oh, air force institute of technology.

Vail, e. F. (1999) knowledge mapping: getting started with knowledge management. *Information systems management,* fall**,** 16-23.

Van alstyne, m. V. (1999) a proposal for valuing information and instrumental goods. *Proceeding of the 20th international conference on information systems* charlotte, north carolina, united states association for information systems.

Anti-Forensics: Techniques, Detection and Countermeasures

Simson Garfinkel
Naval Postgraduate School, Monterey, CA, USA
simsong@acm.org

Abstract: Computer Forensic Tools (CFTs) allow investigators to recover deleted files, reconstruct an intruder's activities, and gain intelligence about a computer's user. Anti-Forensics (AF) tools and techniques frustrate CFTs by erasing or altering information; creating "chaff" that wastes time and hides information; implicating innocent parties by planting fake evidence; exploiting implementation bugs in known tools; and by leaving "tracer" data that causes CFTs to inadvertently reveal their use to the attacker. Traditional AF tools like disk sanitizers were created to protect the privacy of the user. Anti-debugging techniques were designed to protect the intellectual property of compiled code. Rootkits allow attackers to hide their tools from other programs running on the same computer. But in recent years there has been an emergence of AF that directly target CFTs. This paper categorizes traditional AF techniques such as encrypted file systems and disk sanitization utilities, and presents a survey of recent AF tools including Timestomp and Transmogrify. It discusses approaches for attacking forensic tools by exploiting bugs in those tools, as demonstrated by the "42.zip" compression bomb. Finally, it evaluates the effectiveness of these tools for defeating CFTs, presents strategies for their detection, and discusses countermeasures.

Keywords: 42.zip; anti-computer forensics; cybercrime; EnCase; hacker tools; privacy enhancing tools.

1. Background: Computer forensics and anti-forensics

Computer forensics is a new discipline that uses "scientific knowledge for collecting, analyzing, and presenting evidence to the courts" (USCERT 2005). Computer forensic tools (CFTs) assist forensic examiners by collecting information from a computer system; making a true and permanent copy of that information, so that it can be used in a legal proceeding; and analyzing data to uncover information that may not be immediately obvious.

CFTs fall broadly into two classes. *Persistent data tools* (e.g. The Sleuth Kit (Carrier 2006)) analyze data that is stored and that remains when a computer system is turned off. *Volatile data tools* (e.g. NetIntercept (Sandstorm 2006)) analyze information that is transitory and would be lost if not captured, such as the contents of a computer's memory or the flow of packets as they transverse a network.

"Anti-forensics" (AF) is a growing collection of tools and techniques that frustrate forensic tools, investigations and investigators.

Liu and Brown identify four primary goals for anti-forensics:

- Avoiding detection that some kind of event has taken place.
- Disrupting the collection of information.
- Increasing the time that an examiner needs to spend on a case.
- Casting doubt on a forensic report or testimony (Liu and Brown, 2006).

Other goals might include:

- Forcing the forensic tool to reveal its presence.
- Subverting the forensic tool (e.g., using the forensic tool itself to attack the organization in which it is running).
- Mounting a direct attack against the forensic examiner (e.g., discovering and disconnecting the examiner's network, or bombing the building in which the examiner is working).
- Leaving no evidence that an anti-forensic tool has been run.

There is growing interest in AF tools and techniques (AFTs). AF researchers have developed and distributed tools freely on the Internet. Some developers say their work will force improvements in existing forensic tools and process (Liu and Brown, 2006). AFTs also demonstrate the limitations of today's computer forensic techniques, challenging the so-called "presumption of reliability" that courts have afforded tools such as EnCase (Buskirk and Liu, 2006). Some claim that "the lack of true innovation in the forensics world is because there's no pressure to do so" and that they are "not creating vulnerabilities, just identifying them"

(Foster and Liu, 2005). But an unfortunate side-effect is that these tools may be used to "exonerate a guilty party" or "implicate an innocent party by planting data" (Liu and Brown, 2006).

1.1 Outline of this paper

Section 2 of this paper discusses traditional AF techniques such as overwriting, cryptography and steganography. Section 3 discusses AF techniques that work by minimizing the attacker's footprint; Section 4 discusses techniques that directly attack CFTs. Section 5 discusses techniques designed to unmask the use of CFTs. Section 6 concludes.

2. Traditional anti-forensics

Tools that overwrite information that might be the subject of an investigation are the oldest and most common forms of anti-forensic tools available today. Such tools are easy to write and validate, require little training to run, and are distributed with most operating systems.

2.1 Overwriting data and metadata

There exist many programs for overwrite useful information on a storage device so that it is difficult or impossible to recover. These programs can overwrite data, metadata, or both.

2.1.1 Modes of operation

Overwriting programs typically operate in one of three modes:

- The program can overwrite the entire media.
- The program can attempt to overwrite individual files. This task is complicated by journaling file systems: the file itself may be overwritten, but portions may be left in the journal.
- The program can attempt to overwrite files that were previously "deleted" but left on the drive. Programs typically do this by creating one or more files on the media and then writing to these files until no free space remains, taking special measures to erase small files — for example, files that exist entirely within the Windows Master File Table of an NTFS partition (Garfinkel and Malan, 2005).

Programs employ a variety of techniques to overwrite data. Apple's Disk Utility allows data to be overwritten with a single pass of NULL bytes, with 7 passes of random data, or with 35 passes of data. Microsoft's **cipher.exe**, writes a pass of zeros, a pass of FFs, and a pass of random data, in compliance with DoD standard 5220.22-M. (US DoD, 1995). In 1996 Gutmann asserted that it might be possible to recover overwritten data and proposed a 35-pass approach for assured sanitization (Gutmann 1996). However, a single overwriting pass is now viewed as sufficient for sanitizing data from ATA drives with capacities over 15 GB that were manufactured after 2001 (NIST 2006).

The real challenge that the authors of sanitizing tools face not the act of overwriting, but determining which data to overwrite. Although it is relatively easy to overwrite an entire hard drive (#1 above), it can be extremely difficult to selectively overwrite sensitive information while leaving the rest. Geiger evaluated commercial anti-forensic tools and found that many left sensitive information (Geiger 2005). Many tools also left telltale evidence indicating which tool had been run, potentially exposing the user's actions.[1]

2.1.2 Overwriting metadata

If the examiner knows when an attacker had access to a Windows, Mac or Unix system, it is frequently possible to determine which files the attacker accessed, by examining file "access" times for every file on the system. Some CFTs can prepare"timeline" of the attacker's actions by sorting all of the computer's timestamps in chronological order. Although an attacker could wipe the contents of the media, this action itself might attract attention. Instead, the attacker might hide her tracks by overwriting the access times themselves so that the timeline could not be reliably constructed.

For example, Timestomp will overwrite NTFS "create," "modify," "access," and "change" timestamps (Metasploit 2006). The Defiler's Toolkit can overwrite inode timetamps and deleted directory entries on many

[1] In March 2006, the US Court of Appeals for the 7th Circuit ruled that Jacob Citrin's use of a secure file erase program violated the Computer Fraud and Abuse Act. The Court held that the use of the secure deletion program constituted the willful destruction of information on a computer that Critin was no longer authorized to use (Andersen 2006).

Unix systems; timestamps on allocated files can also be modified using the Unix **touch** command (Grugq 2003).

An alternative to overwriting metadata is for the attacker to access the computer in such a way that metadata is not created. For example, a partition can be mounted read-only or accessed through the raw device to prevent the file access times from being updated. The Windows registry key HKLM\SYSTEM\CurrentControlSet\Control\FileSystem\NtfsDisableLastAccessUpdate can be set to "1" to disable updating of the last-accessed timestamp; this setting is default under Windows Vista (Microsoft 2006).

2.2 Cryptography, steganography, and other data hiding approaches

Cryptography, steganography and other data hiding approaches can be used effectively against most forensic approaches. But while cryptography is very effective at hiding information, encrypted data itself is easy to detect: encrypted data has exceptionally high entropy, and many products embed specific flags, headers or other signatures in their encrypted data. Some forensic tools can decrypt encrypted data if the key is obtained—for example, through spyware or other covert channels. Even if the original plaintext data cannot be recovered, the mere fact that encryption has been used may attract unwanted attention to the attacker.

2.2.1 Cryptographic file systems

Cryptographic file systems transparently encrypt data when it is written to the disk and decrypt data when it is read back, making the data opaque to any attacker (or CFT) that does not have the key. These file systems are now readily available for Windows, Mac OS, and Linux. The key can be protected with a passphrase or stored on an auxiliary device such as a USB token. If there is no copy of the key, intentionally destroying the key makes the data stored on the media inaccessible (Boneh and Lipton, 1996). Even if the cryptographic system lacks an intentional sanitization command or "self-destruct," cryptography can still be a potent barrier to forensic analysis if the cryptographic key is unknown to the examiner.

Cryptography can also be used at the application level. For example, Microsoft Word can be configured to encrypt the contents of a document by specifying that the document has a "password to open." Although older versions of Microsoft Word encrypted documents with a 40-bit key that can be cracked with commercial tools, modern versions can optionally use a 128-bit encryption that is uncrackable if a secure passphrase is used.

2.2.2 Encrypted network protocols

Network traffic can likewise be encrypted to protect its content from forensic analysis. Cryptographic encapsulation protocols such as SSL and SSH only protect the content of the traffic. Protecting against traffic analysis requires the use of intermediaries. Onion Routing (Goldschlag, Reed and Syverson, 1999) combines both approaches with multiple layers of encryption, so that no intermediary knows both ends of the communication and the plaintext content.

2.2.3 Program packers

Packers are commonly used by attackers so that attack tools will not be subject to reverse engineering or detection by scanning. Packers such as PECompact (Bitsum 2006) and Burneye (Vrba 2004) will take a second program, compress and/or encrypt it, and wrap it with a suitable extractor. Packers can also incorporate active protection against debugging or reverse engineering techniques. For example, Shiva will exit if its process is being traced; if the process is not being traced, it will create a second process, and the two processes will then trace each other, since each process on a Unix system may only be traced by one other process. (Mehta and Clowes, 2003)

Packed programs that require a password in order to be run can be as strong as their encryption and password. However, the programs are vulnerable at runtime. Burndump is a loadable kernel module (LKM) that automatically detects when a Burneye-protected file is run, waits for the program to be decrypted, and then writes the raw, unprotected binary to another location (ByteRage 2002). Packed programs are also vulnerable to static analysis if no password is required (Eagle 2003).

2.2.4 Steganography

Steganography can be used to embed encrypted data in a *cover text* to avoid detection. Steghide embeds text in JPEG, MBP, MP3, WAV and AU files (Hetzl 2002). Hydan exploits redundancy in the x86 instruction set; it can encode roughly 1 byte per 110 (El-Khalil 2004). Stegdetect (Provos 2004) can detect some forms of steganography. StegFS hides encrypted data in the unused blocks of a Linux ext2 file system, making the data "look like a partition in which unused blocks have recently been overwritten with random bytes using some disk wiping tool" (McDonald and Kuhn, 2003). TrueCrypt allows a second encrypted file system to be hidden within the first (TrueCrypt 2006). The goal of this filesystem-within-a-filesystem is to allow the TrueCrypt users to have a "decoy" file system with data that is interesting but not overtly sensitive. A person who is arrested or captured with a TrueCrypt-protected laptop could then give up the first file system's password, with the hope that the decoy would be sufficient to satisfy the person's interrogators.

2.2.5 Generic data hiding

Data can also be hidden in unallocated or otherwise unreachable locations that are ignored by the current generation of forensic tools.

Metasploit's Slacker will hide data within the slack space of FAT or NTFS file system. FragFS hides data within the NTFS Master File Table. RuneFS (Grugq 2003) stores data in bad blocks. (Thompson and Monroe, 2006). Waffen FS stores data in the ext3 journal file (Eckstein and Jahnke 2005). KY FS stores data in directories (Grugq 2003). Data Mule FS stores data in inode reserved space (Grugq 2003). It is also possible to store information in the unallocated pages of Microsoft Office files.

Information can be stored in the Host Protected Area (HPA) and the Device Configuration Overlay (DCO) areas of modern ATA hard drives. Data in the HPA and DCO is not visible to the BIOS or operating system, although it can be extracted with special tools. Data can be hidden from network monitoring by encoding it in TCP sequence numbers, window sizes, unused TCP or IP options, or even introducing intentional jitter in packet timing (Cabuk, Brodley and Shields, 2004). With this taxonomy, rootkits can be considered as simply another data hiding technique that hides information about files or active processes from other programs running on the same computer (Brumley 2000). As with the HPA, all of these techniques for data hiding can be detected with tools that understand the typical format of file system or application structures and report when unused or allocated regions are anomalous. For example, The Defiler's Toolkit has provisions to hide data in "bad blocks" of ext2fs file systems. This is effective against The Coroner's Toolkit, which does not look inside blocks that are labeled "bad," but does not work against The Sleuth Kit (Carrier 2006), an updated version of TCT (Wonko-Ga 2004).

3. AF techniques that minimize footprint

Many of the techniques in Section 2 are easy to detect when they are used, provided that the examiner knows where to look. Another approach is to minimize the "footprint," or data that the attacker has left behind. In this way, there is less data for the CFT to analyze.

3.1 Memory injection and Syscall Proxying

Buffer overflow exploits allow an attacker to inject and run code in the address space of a running program, effectively changing the victim program's behavior. Traditionally buffer overflows have just been used as a point-of-entry to a remote system, after which the attacker uploads tools that are then saved on the target machine's hard drive.

The "Userland Execve" technique (Grugq 2003a, Pluf & Ripe 2006) allows programs on the victim computer to be loaded and run without the use of the Unix execve() kernel call, allowing the attacker to overcome kernel-based security systems that may deny access to execve() or log its use to a secure logging service. The ul_exec'ed program can even be read from a network without ever being written to the victim's disk. Instead of uploading an entire exploit program, the attacker can upload a *system call proxy* that accepts remote procedure calls from the attacker's computer, executes the requested system call on the victim's machine, and sends the result back to the attacker. This approach has the advantage that the attacker's tools never need to be uploaded at all to the compromised machine, but its disadvantage of increasing (perhaps dramatically) the amount of network traffic between the compromised machine and the attacker — as well as introducing potentially problematic latency. System call proxying is implemented by Impact, a commercial penetration tool sold by Core Security Technologies.

3.2 Live cds, bootable USB tokens and virtual machines

Live CDs, bootable USB tokens, and virtual machines give attackers a tool for running programs of their choice while simultaneously containing the spread of forensically-useful information on the attacker's system.

- **Live CDs** are an operating system distribution that boots and runs from a read-only device. Live CDs typically have a window system, web browser and SSH client, and run with virtual memory disabled.
- **Bootable USB tokens** are similar to a Live CD except that the operating system is contained within an attachable USB device. These tokens can typically store more information that CDs and allow information to be saved, generally via encryption.
- **Virtual M**are "client" operating systems run inside a virtualization program such as VMWare Player, Parallels, or Microsoft Virtual PC. These systems typically store all of the states associated with the client operating system within a small set of files on the host computer.

Using a Live CD or Bootable USB Token, for example, an attacker could boot a copy of Linux on a PC provided by a library, use that PC to attack a series of computers, then turn off the computer and walk away: no trace of the attack would be left on the computer for later investigative analysis. Virtualization software makes it possible to perpetrate the attack without even rebooting the host computer. Following the attack, the attacker would only need to securely erase the files associated with the virtual machine. Virtual machines can also be used directly by the attacker on the victim's machine as a kind of super-rootkit. CFTs running in the VM would not see the super-rootkit, because it would be running outside the VM (Rutkowska 2004) (King et al 2006).

3.3 Anonymous identities and storage

Attackers have long protected their actual identity by making use of anonymous accounts available at Hotmail, Yahoo and Gmail. Recently the amount of storage associated with these accounts has been dramatically increased. Attackers can utilize this storage to avoid the risk of storing attack tools and captured information on their own computers (Dear 2005).

4. AF techniques that exploit CFT bugs

If an attacker has access to a CFT or knowledge of how that tool works, the attacker can craft data that will manifest bugs within the CFT. Properly triggered, these bugs can accomplish many anti-forensic goals.

4.1 Failure to validate data

As with other programs, CFTs that do not properly validate their input data can potentially be subverted—for example, by a buffer overflow attack. For example, CVE CAN-2001-1279 details a vulnerability in tcpdump's decoding routines for AFS RPC packets which then can be subverted by an attacker and used to run arbitrary code (CERT 2002). Similar problems have been discovered in Snort (Infoworld 2003) and Ethereal (iDefense 2005). This kind of vulnerability in a network forensics analysis tool (NFAT) is particularly easy to exploit, because the NFAT is exposed to a potentially unlimited amount of information from an attacker.

4.2 Denial of service attacks

Any CFT resource (memory, CPU, etc.) whose use is determined by input data is potentially subject to a denial-of-service attack. For example, certain Windows logfile analysis tools will attempt to execute regular expressions that are embedded in log file entries; carefully constructed regular expressions can be used to cause these tools to hang when they are executed (Foster and Liu, 2005). Compression bombs are a special category of DoS attacks on CFTs and other tools that attempt to analyze the content of container files. These bombs are small data files that consumes a tremendous amount of storage when uncompressed. For example, 42.zip is a 43,374 byte file that contains 16 zipped files, each of which contains 16 zipped files, and so on, for a total of 4TB of data (SANS 2001).

4.3 Fragile heuristics

CFTs frequently need to determine the type of file or data object to allow for efficient processing. For example, a forensic examiner may try to save time by omitting the contents of executable files from searches. Many tools determine file type by simply consulting the file's extension and the first few bytes of the file's contents (the "magic number"). Attackers who know the heuristics that a CFT uses for identifying files can exploit them. For example, EnCase will classify a Windows file as executable if it has an extension

of ".exe" and contains the letters "MZ" as the first two characters of the file (Liu and Stach, 2006). The Metasploit Project's Transmogrify program can turn a text file into an "executable" by changing a text file's extension from .txt to .exe and putting the letters "MZ" at the beginning of the file, EnCase will think that the file is a binary and not scan it.

5. AFTs that detect CFTs

AFTs can change their behavior if they can detect that a CFT is in use. For example, a packer might not decrypt its payload if it realizes that it is running on a disk that has been imaged. A worm might refuse to propagate if it discovers that a network is being surveilled.

5.1 Countering forensic analysis with SMART

Self-Monitoring, Analysis and Reporting Technology (SMART) built into most hard drives today report the total number of power cycles (Power_Cycle_Count), the total time that a hard drive has been in use (Power_On_Hours or Power_On_Minutes), a log of high temperatures that the drive has reached, and other manufacturer-determined attributes. These counters can be reliably read by user programs and cannot be reset. Although the SMART specification implements a DISABLE command (SMART 96), experimentation indicates that the few drives that actually implement the DISABLE command continue to keep track of the time-in-use and power cycle count and make this information available after the next power cycle (McLeod 2005). AFTs can read SMART counters to detect attempts at forensic analysis and alter their behavior accordingly. For example, a dramatic increase in Power_On_Minutes might indicate that the computer's hard drive has been imaged (McLeod 2005).

5.2 Detecting network forensics

Network forensics requires some kind of network monitoring. There are two primary techniques for detecting this monitoring: detecting hosts that have their Ethernet interfaces in "promiscuous" mode, and observing when information captured from a monitored network is used.

5.2.1 Detecting hosts in "Promiscuous" mode

Many network forensics systems capture information using an Ethernet interface that in promiscuous mode—that is, the interface captures all packets on the local area network, rather than just those addressed to the host. Properly configured network monitors should be unable to transmit on the network that is being monitored. Monitoring systems are frequently not configured this way. These systems can be detected by the way that they respond to various kinds of malformed IP packets (Sanai 2001).

5.2.2 Detection of monitoring with DNS

Another way to detect network monitoring is for an attacker to send packets across a network that have as their destination an Ethernet and IP address that is on the subnet but not currently in use, and have a source address that comes from a rarely used network. Upon seeing such packets, many network monitoring tools will initiate a reverse DNS request attempting to resolve the hostname of the rarely used network. If the attacker can monitor the DNS server that would handle such requests, the attacker may infer that monitoring of packets or network flows is taking place. This technique can work if there is a flaw in the CFT, or simply if the CFT is used improperly (Rudys 2000).

6. Countermeasures and conclusion

Many of the anti-forensic techniques discussed in this paper can be overcome through improved monitoring systems or by fixing bugs in the current generation of computer forensic tools. Overwriting tools can be frustrated by positioning data so that the attacker does not have the ability to overwrite it—for example, by sending log files to a "log host" or CD-R, assuming that the attacker does not have physical access. Weak file identification heuristics can be replaced with stronger ones. Compression bombs can be defeated with more intelligent decompression libraries. Although there is anecdotal evidence that file encryption and encrypted file systems are beginning to pose a problem for law enforcement, there are also many reports of officers being able to recover cryptographic passwords and keys using spyware, keyboard loggers, and other tactics. The prudent attacker is safer using a sanitization tool than a cryptographic one, because the sanitizer actually destroys information.

Many of the techniques discussed in this paper still appear limited to the research community, although there are occasional reports of specific tools being used by technically sophisticated computer criminals. Because law enforcement resources are limited, it seems reasonable to hypothesize that, other things being equal; attackers employing anti-forensic technology are less likely to be apprehended than those who do not. Because the goal of anti-forensic technology is specifically to confound investigations, it is possible that their use and perhaps even possession will be banned in some organizations. However, the inclusion of high-quality anti-forensic technology in consumer operating systems to promote data privacy goals are sure to make such bans futile. Computer forensics has traditionally relied upon information that was inadvertently left behind by other programs. Organizations soon may need to decide explicitly what information they wish to preserve as part of ordinary operations, and then make arrangements to preserve that information in a forensically sound manner.

Acknowledgements

The author wishes to thank Steven Bauer, Chris Eagle, Kevin Fu, Matthew Geiger, Jonathan Herzog, Beth Rosenberg and Vincent Liu for their valuable comments on a previous draft of this paper.

References

Andersen (2006), "International Air Centers, LLC, v. Jacob Citrin", 05-1522, Argued October 24, 2006, decided March 8, 2006. US Court of Appeals for the 7th Circuit.

Bitsum (2006), "PECompact: For maximum compression and speed," [online] http://www.bitsum.com/pec2.asp

Boneh and Lipton (1996), "A Revocable Backup System," Proceedings of the Sixth USENIX UNIX Security Symposium. [online] http://www.usenix.org/publications/library/proceedings/sec96/boneh.html

Brumley (2002), "Rootkits: How Intruders Hide", 23 Feb. [online] http://www.globalsecurity.org/intell/library/news/2000/02/000223-hack1.htm

Buskrik and Liu (2006), "Digital Evidence: Challenging the Presumption of Reliability", *Journal of Digital Forensic Practice*, 1:19—26, 2006. DOI 10.1080/15567280500541421

ByteRage (2002), "TESO Burneye Unwrapper", July 13. [online] http://www.securiteam.com/tools/5BP0H0U7PQ.html

Cabuk, Brodley and Shields (2004), "IP covert timing channels: design and detection," Proceedings of the 11th ACM conference on Computer and Communications Security, pp. 178-187

Carrier (2006) The Sleuth Kit [online] http://www.sleuthkit.org/

CERT (2002), "Vulnerability Note VU#797201: tcpdump vulnerable to buffer overflow via improper decoding of AFS RPC (Rx) packets," June 7. [online] https://www.kb.cert.org/vuls/id/797201.

Dear (2005), "An Exploration of Future Anti-Forensic Techniques" [online] http://www.assuremind.com/antiForensics.pdf

Eagle, C. (2003), "Strike/Counter-Strike: Reverse Engineering Shiva," Black Hat Federal. http://www.blackhat.com/presentations/bh-federal-03/bh-federal-03-eagle/bh-fed-03-eagle.pdf

Eckstein and Jahnke (2005), "Data Hiding in Journaling File Systems", *DFRWS 2005*. http://www.dfrws.org/2005/proceedings/eckstein_journal.pdf

El-Khalil, R. (2004), "Hydan: Information Hiding in Program Binaries" [online] http://www.crazyboy.com/hydan/

Foster and Liu (2005), "Catch me, if you can," Black Hat Briefings 2005.

Garfinkel and Malan (2005), "One Big File is Not Enough: A Critical Evaluation of the Dominant Free-Space Sanitization Technique,",The 6th Workshop on Privacy Enhancing Technologies, Robinson College, Cambridge, United Kingdom, June 28 - June 30.

Geiger (2005), "Evaluating Commercial Counter-Forensic Tools," *DFRWS 2005.* http://www.dfrws.org/2005/proceedings/geiger_couterforensics.pdf

Goldschlag, Reed and Syverson (1999), "Onion Routing," *Communications of the ACM*, 42(2), pp. 39—41.

Grugq (2002), "Defeating forensic analysis on UNIX," *Phrack Magazine*, 11)6—6, 28 July [online] http://www.totse.com/en/hack/hack_attack/167627.html

Grugq (2003), "To the Art of Defiling," Black Hat Asia 2003 presentation. [online] http://opensores.thebunker.net/pub/mirrors/blackhat/presentations/bh-asia-03/bh-asia-03-grugq/bh-asia-03-grugq.pdf

Grugq (2003a), "The Design and Implementation of ul_exec" Security Focus [online] http://securityfocus.com/archive/1/348638/2003-12-29/2004-01-04/0

Grugq (2005), "The Art of Defiling," Black Hat 2005, [online] http://www.blackhat.com/presentations/bh-usa-05/bh-us-05-grugq.pdf

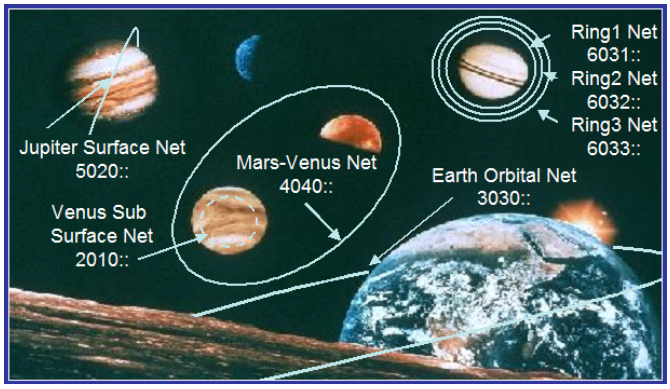
Guttman (1996), “Secure Deletion of Data from Magnetic and Solid-State Memory,” in The Sixth USENIX Security Symposium Proceedings, San Jose, California. July 22-25.
Henry (2006), “Anti-Forensics” (presentation). [online] http://layerone.info/2006/presentations/Anti-Forensics-LayerOne-Paul_Henry.pdf
iDefense (2005), “Ethereal OSPF Protocol Dissector Buffer Overflow Vulnerability,” [online] http://labs.idefense.com/intelligence/vulnerabilities/display.php?id=349
Infoworld (2003), “ISS reports Snort vulnerability,” March 4.
King et al (2006), "SubVirt: Implementing malware with virtual machines", Proceedings of the 2006 IEEE Symposium on Security and Privacy, May. [online] http://www.eecs.umich.edu/virtual/papers/king06.pdf
Li u and Brown (2006), “Bleeding-Edge Anti-Forensics,” *Infosec World Conference & Expo,* MIS Training Institute.
Liu and Stach (2006), “Defeating Forensic Analysis”, CEIC 2006 — Technical Lecture 1. Notes. 4 May. http://www.metasploit.com/projects/antiforensics/CEIC2006-Defeating_Forensic_Analysis.pdf [online]
McDonald, A. and Kuhn, M. (2000), “StegFS: A Steganographic File System for Linux,” in A. Pfitzmann (Ed.), IH’99, LNCS 1768, pp. 463-477. [online] http://www.cl.cam.ac.uk/~mgk25/ih99-stegfs.pdf
McLeod, S. (2005) “Smart Anti-Forensics,” *Forensic Focus*, May 2005. [online] http://www.forensicfocus.com/smart-anti-forensics
Mehta, N. and Clowes, S. (2003), “Shiva: Advances in ELF Binary Encryption,” CanSecWest.
Metasploit (2006), “Timestomp” [online] http://www.metasploit.com/projects/antiforensics/.http://www.metasploit.com/projects/antiforensics/.
Microsoft (2006), “Disabling Last Access Time in Windows Vista to improve NTFS performance,” 7 November. [online] http://blogs.technet.com/filecab/archive/2006/11/07/disabling-last-access-time-in-windows-vista-to-improve-ntfs-performance.aspx.
NIST (2006) “Guidelines for Media Sanitization,” NIST Special Publication 800-88, September. [online] http://csrc.nist.gov/publications/nistpubs/800-88/NISTSP800-88_rev1.pdf
Pluf & Ripe (2006) “Advanced Antiforensics : SELF”, Sept 2, [online] http://www.7a69ezine.org/article/Advanced_Antiforensics_:_SELF
Provos, N. (2004). “Steganography Detection with Stegdetect” [online] http://www.outguess.org/detection.php
Rudys, A. (2000). “Invited Talk: Methods for Detecting Addressable Promiscuous Devices, Mudge, @stake, summarized by Algis Rudys”, *;login:*, Nov. [online] http://www.usenix.org/events/sec00/novconf.pdf
Rutkowska (2004), “Red Pill… or how to detect VMM using (almost) one CPU instruction,” November [online] http://invisiblethings.org/papers/redpill.html
Sandstorm (2006)
Sani, J. (2001), “Promiscuous node detection using ARP packets,” presented at Black Hat 2001 [online] http://www.blackhat.com/presentations/bh-usa-01/DaijiSanai/bh-usa-01-Sanai.ppt.
SANS (2001), Security Alert Consensus 106, Network Computing and the SANS Institute, July 19. [online] http://archives.neohapsis.com/archives/securityexpress/2001/0030.html
SMART (1996) SFF Committee Specification for Self-Monitoring, Analysis and Reporting Technology (SMART), SFF-8035i Revision 2.0, April 1.
Thompson, I., and Monroe, M (2006) “FragFS: An Advanced Data Hiding Technique,” BlackHat Federal, January. [online] http://www.blackhat.com/presentations/bh-federal-06/BH-Fed-06-Thompson/BH-Fed-06-Thompson-up.pdf
TrueCrypt (2006), “Frequently Asked Questions,” TrueCrypt Foundation, October 7. [online] http://www.truecrypt.org/faq.php
US DoD (1995), “Cleaning and Sanitization Matrix,” DOS 5220.22-M, Chapter 8, US Department of Defense. [online] www.dss.mil/isec/nispom_0195.htm
USCERT (2006), “Computer Forensics” [online] http://www.us-cert.gov/reading_room/forensics.pdf
Vrba, Z. (2003), “cryptexec: Next-generation runtime binary encryption using on-demand function extraction,” Phrak 0x0b(0x3f). #0x0d of 0x14. [online] http://www.phrack.org/archives/63/p63-0x0d_Next_Generation_Runtime_Binary_Encryption.txt
Wang (2004), Huaiqing Wang and Shuozhong Wang, “Cyber Warfare: Steganography vs. Steganalysis,” *Communications of the ACM* , October, Vol. 47, No. 10.
Wonko-ga (2004), “Re: The Defiler’s Toolkit”, Google Answers, 13 May. [online] http://answers.google.com/answers/threadview?id=345604

IPv6: World Update

Kenneth Geers and Alexander Eisen
NCIS Washington, DC, USA
kgeers@ncis.navy.mil

Abstract: Governments around the world are setting deadlines for IPv6 compliance. While there is still room for debate regarding specific dates for deployment, there is no question but that your organization must begin to prepare for the Next-Generation Internet, and it should start today. This paper is based on wide-ranging, in-depth research, including interviews with the top thinkers on the most crucial issues surrounding the sleeping giant known as IPv6. It will give you the facts you need in order to plan for what may be difficult times ahead. The tactical, down-in-the-weeds take on IPv6 is examined in detail. This paper includes technical details that will inform the reader of the challenges that await as he or she attempts to keep pace not only with their government mandates, but also with economic competitors from around the world. This paper also explains how hackers can exploit this new technology, and how security administrators can stop black hats from taking advantage of the necessarily long-lasting, heterogeneous environment that will be required during the transition to IPv6. Many nation-states view IPv6 as crucial to their national security plans for the future. This paper will make stops at the White House, Tokyo, Beijing, and Red Square, and cover in detail the most current IPv6 research and deployment events from around the world. It discusses how, if some governments get their way, most of us could well lose our last byte of anonymity on the Internet. Finally, the corporate side of Internet addressing is addressed: what do the Xbox, IPTV, and the number of beers I have left in my fridge at home have in common? Answer: IPv6!

Keywords: IPv6, next-generation internet, protocol, management, national security



1. Introduction

By the beginning of 2007, the publicly accessible Internet will still only have a small number of publicly reachable Internet Protocol version 6 (IPv6) addresses. Even so, this paper will argue that the Next-Generation Internet, which presupposes the widespread use of IPv6, is on the horizon, and that you should begin preparing for the transition today.

Governments worldwide have recognized that the world has become fundamentally network-centric, and this includes even a nation's ability to wage warfare. The possibility and promise of a Next-Generation Internet protocol have encouraged the United States to mandate that its government networks be IPv6 compliant by June 30, 2008. There is also a good chance that, no matter where you are from, your own government has issued guidelines directing you to meet certain IPv6 milestones.

2. Address space: Size does matter

The first thing you should know about IPv6 is that the number of Internet addresses available will rise from about four billion under IPv4 to the astonishing number of 340 undecillion, or about 50 octillion IPs for every human on planet Earth! This means that, thanks to the new 128-bit addressing scheme in IPv6, the current address space crunch is solved, period.

There is little doubt that IPv4 is running out of address space. Network Address Translation (NAT) – the ultimate workaround that has a single IP address doing the work of many – staved off the inevitable for long time, especially outside of America. And NAT even provided the world with "security by obscurity" by having most IP addresses not directly facing the Internet. Further bolstering IPv4's lifespan was that many of the

advanced features and protocol improvements designed for IPv6, such as IPSec, were eventually backported to IPv4 and worked to retard the upgrade.

Coupled with this "galactic scalability" are much more powerful, fixed-length, IP headers that enable software developers to create advanced features to take advantage of the "always on, always connected" devices that will pervade our lives in the years to come. IPv6 advocates call this "telematics", and say that it will lead to more convenient, economical, and entertaining lifestyles from the First World to the Third World.

3. Is utopia around the corner?

There are always two sides to every story. While many have hailed a possible IPv6 era as an Internet Utopia, others are warning that, on the road to the Emerald City, Internet users will remain lost in a Wild West of new and unfamiliar security concerns throughout the transition period. These include a significantly increased "attack surface", and novel challenges in the domains of Internet privacy, anonymity, and accountability.

The questions that IPv6 critics have raised deserve to be answered. If the need for a new protocol is so pressing, why has its adoption been so slow? Is there a better solution for our current shortage of Internet addresses? Are consumers merely the victims of IPv6 marketing hype? Will the trusted Internet model desired by IPv6 visionaries – replacing the existing untrusted model – ever become a reality?

4. Technical background

To understand where we are going, it is important to consider the past. More than twenty years ago, the Internet Protocol (IP) defeated a more feature-rich rival called GOSSIP, precisely because IP was the "dumber" of the two. What it lacked in smarts, it made up for in it elegant simplicity and resiliency. GOSSIP was considered complex to the point of being unnecessarily "bloated". Vendors naturally liked the long support contracts that GOSSIP entailed, but widespread IP adoption made it emerge as the clear winner based on ease of implementation and maintenance. Only time will tell what happens when IPv6 will become a household staple.

One of the fundamental differences between IPv4 and IPv6 lies in the packet structure. IPv6 has extension headers that are much more powerful than their IPv4 counterparts. In the IPv6 era, user and developer-defined customization will rule the day. Packets will be designed for enhanced Quality of Service (QoS). Many aspects of IPv4 will go away, including fragmentation by routers, broadcast addresses (multicast will take this idea in a different direction), IP ID fields, and the IP Record Route Option (a traceroute alternative).

A major selling point for IPv6 salesmen is increased security. But how true are their claims? Is it possible for security to be "built-in" to a protocol? For starters, IPv6 is in fact crypto-friendly. IPSec is not mandatory per se, but it is built into the code stack, and so it is available, just like pay-per-view movies, on demand. However, the devil always lies in the details, and it is important to remember that while IPSec is available, its deployment is optional. One final point to keep in mind is that the improved IP header, described above, is still in the clear.

Old-hands will note that PKI has rarely succeeded in a wide-scale deployment. In fact, PKI is not even widely used except in odd circumstances such as Border Gateway Protocol (BGP) tunnels between routers. The resource-intensive and high-cost nature of PKI is prohibitive to most organizations, and it is possible that the same dynamic, in regard to IPSec implementation, will be hampering the built-in security power of IPv6.

5. Goodbye NAT

One of the most hotly debated aspects of the transition to IPv6 is the move away from Network Address Translation (NAT). Depending on whom you ask, post-NATism could be one of IPv6's greatest benefits, or one of its greatest hazards.

Proud owners of IPv6 addresses can take some joy in the fact that it is almost impossible for an attacker to scan the Internet randomly to find you. The number of IPv6 addresses is so large that finding one randomly would be like shooting a missile into outer space in the vain hope of hitting a particular star. Take your pick; success in either would probably take a few hundred million years to achieve. Now that is a long ping sweep!

Another enormous benefit of this characteristic is that the Next Generation Internet should be largely resistant to the spreading of self-propagating worms. Furthermore, we are still at a stage in IPv6 development that we have an opportunity to improve enormously on the logic and hierarchy of IPv6 address allocation, which will hopefully make Internet navigation and routing much easier in the future.

6. Hello increased attack surface

During the necessarily long transition to IPv6, one well-founded fear is that of an increased attack surface for computer security personnel to defend. At the same time, the level of complexity involved in defending the existing attack surface will also rise precipitously.

System administrators will manage more hosts per enterprise, more network interface cards (NIC) per host, more stacks per NIC, and more multi-homed hosts. Further, they will have native versus tunneled sessions, IPSec complexities, and IPv6 proxy and translation issues to worry about. In this environment, attacks may well become stealthier, and they may potentially slip through the cracks. In fact, unbeknown to your SysAdmin, these issues may already be a problem on your network, given that many modern devices and operating systems are IPv6-enabled by default. And finally, don't forget that your applications, regardless of which protocol they prefer, will still be at risk, and IPv4 legacy challenges will remain.

Another aspect of IPv6 that is filled with both promise and caution is Auto-Configuration. Auto-Configuration is a "welcoming" technology that will ease and increase mobility through enhanced, optimized routing and ad-hoc network associations. If not implemented correctly, it is easy to imagine that Auto-Configuration could make tracking the assets that enter and leave enterprise boundaries much more difficult.

In short, it is hard to determine at this stage whether the IPv6 balance will be tipped in favor of the benefits or the drawbacks. The answer may be that in the long run we will be better off, but that during the transition, there will be increased security risks.

7. Next generation attacks

In the IPv6 environment, attacks should increasingly target Certificate Authorities (CA) and Domain Name Servers (DNS). These will become primary targets in light of the fact that network scanning as we know it may be rendered obsolete in the future. Network discovery and reconnaissance may involve the compromise a DNS server first, and then, armed with the target knowledge gleaned from it, an attacker may hop from Local Area Network (LAN) to LAN. While it is true that IPSec will reduce a target's exposure to spoofing, illicit traffic sniffing and Man-in-the-Middle (MITM) attacks, there is reason to believe that, once the DNS server falls, the IPv6 era should continue the trend toward client-side exploits.

In spite of the security hype that you may have heard, hackers have already demonstrated that most common network-based threats are still possible against the IPv6 protocol: Denial-of-Service (DoS) attacks (Smurf6, Rsmurf6, Redir6, connection flooding, and stealing all available addresses), packet crafting, and MITM attacks (Parasite6, Fake_router6).

In fact, network security monitoring could be significantly more difficult in the near future. Currently, most "information assurance" network monitoring sensors (e.g. intrusion protection systems) are blind to the contents of IPv6 traffic, whether it is native or encapsulated in IPv4 tunnels. Since the traffic is tunneled and/or encrypted – even automatically, unbeknown to the user – traffic analysis is made more complicated by orders of magnitude.

8. People's republic of China: Full speed ahead

In the People's Republic of China (PRC), the rationale for a quick transition to IPv6 is multi-faceted. First, the size of the country's population demands an immediate increase in the number of available network addresses. Second, the government in Beijing believes that the cutting edge nature of IPv6 also gives the Chinese government a good way to development its national Intellectual Property (IP) base.

IPv6 has therefore become a PRC national strategy. Beijing has invested heavily in initiatives such as the China Next Generation Internet (CNGI), and the China Education and Research Network (CERNET), but the dearth of available IPv6 applications has kept the transition relatively sluggish. On the plus side for the PRC, however, is that their significant work on IPv6 will have a definite influence on the composition, standards, and evolution of the protocol.

One aspect of the PRC transition that is worrying human rights groups is Beijing's emphasis on the use of IPv6 to control its population. PRC Internet Society chairwoman Hu Qiheng has stated flatly that static IPv6 IP addresses will help the government to establish a state of "no anonymity" on the Chinese Internet. The PRC already has strict controls on access to the World Wide Web: there are policemen stationed at cyber cafes, and there are "national firewall" filters in place to prevent the free flow of information into and out of the country, including, for example, the passing of illegal videos of Chinese sweatshops to human rights groups abroad. The key question is whether IPv6 could help the PRC squash political dissent within its borders. In late 2006, the answer is still unknown.

To date, the PRC track record on human rights in Cyberspace is not good. Western companies and their technologies, for their part, have already been accused of too much cooperation with the government in Beijing. Google, Yahoo, and Microsoft have all collaborated with the Communist government in prosecutions of a questionable nature. At least one U.S. congressman has termed such cooperation "sickening and evil", and compared it to the work IBM did for the Nazi government during World War II. It is a fair question to ask how much collaboration will actually be possible between major powers like the United States and China on a topic as important as IPv6, but for multi-national corporations, the bottom line usually rules the day.

9. USA: Conflicting signals

In late 2006, the United States is concerned about falling behind in the race to the Next Generation Internet. During the latest State of the Union address, U.S. President George W. Bush presented the American Competitiveness Initiative (ACI), an effort by the White House to encourage U.S. businesses to stay competitive in high technology. Further, the U.S. Executive Branch Office of Management and Budget (OMB) mandated that U.S. government agencies be "IPv6 compliant" by June 30, 2008.

However, the arguments for and against IPv6 are still not clear-cut, and the U.S. government seems conflicted among its agencies. More recently than the OMB mandate, the Department of Commerce (DoC) released a report that stated a premature transition to IPv6 could actually lead to higher overall transition costs and "reduced" security. Not surprisingly, therefore, there have been loud complaints that the American position lacks clarity and thus sufficient direction to be acted upon.

American companies, for their part, feel little direct pressure to switch. For them, there is no shortage of Internet address space within IPv4. What pressure there is comes from complaints at the continued cost of workarounds in support of NAT. Software development time is always expensive, and gaming through NAT also requires rendezvous servers to maintain. The major U.S. investors are Command Information, which has taken the lead in IPv6 technology training, and Hughes, Bechtel, the Carlyle Group, and Microsoft. Microsoft is among NAT's prominent critics, and has made IPv6 the default Internet protocol for its forthcoming operating system, Vista. The Seattle-based corporation has also offered up $100 million in front money to companies that are building applications for Vista.

Finally, determining what the U.S. government means by "compliance" in its mandate has become a matter of debate. Almost certainly, between now and June 2008, the meaning will continue to be watered down until it refers only to a core network's ability to forward native IPv6 traffic on to its intended destination. By 2008, therefore, we may see little real progress on the road to IPv6 compliance.

10. The rest of the world

Elsewhere in the world, the move to IPv6 is gaining momentum, in part due to the need for address space (some African countries own less that 2,000 IP addresses!), and in part due to a desire to move straight to Next-Gen technologies.

India, for example, has the same need for space as China. Indian Web surfers often find themselves behind up to 5 layers of NAT. As New Delhi progresses toward its transition to IPv6, one of its primary concerns has been whether to go it alone or in conjunction with multinational corporations.

European achievements include the development of the first IPv6 router, by Ericsson in 1995. According to the European IPv6 Task Force, the importance of IPv6 "cannot be overestimated". The EU created the IPv6 Wireless Internet Initiative (6WINIT), and has even developed an IPv6 concept car, developed jointly by Cisco and Renault. Still, European companies have complained that tangible rewards – and incentives from Brussels – are needed to keep them moving down the path to IPv6. Recently, the Europeans teamed up

with South Korea, who was previously perceived to be two years ahead of Europe in IPv6 development, on joint projects.

In Japan, the nature of the address space problem is different than in China, but the need for space is the same. Japan does not have the population of China, but it does have many billions of electronic gadgets to address. Unsurprisingly, the Japanese showed early leadership on IPv6. In the year 2000, the Japanese Prime Minister announced "eJapan 2005", of which IPv6 was a major part. Today, not only does Japan already have commercial IPv6 services available, but the University of Tokyo currently has both the IPv4 and IPv6 World Speed records!

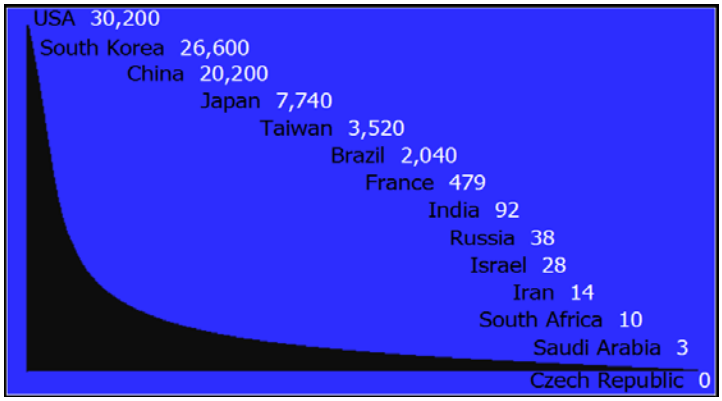


Figure 1: The number of hits available through a Google search on "IPv6" in government domains worldwide, in July 2006.

11. Challenges for internet management

To be truly successful, the worldwide transition to IPv6 will need more structure and logic to it than IPv4 received over the past twenty years. Fortunately, this should be the case. In 1999, the Internet Engineering Task Force (IETF) created the first IPv6 Forum. Today, there are at least 30 IPv6 country fora and task forces worldwide.

One effort to secure an orderly transition is through an award called the IPv6 Ready Logo for software and hardware that meet agreed upon technical standards. Currently, it is possible to obtain the IPv6 Ready Logo for the following:

- Host
- Router
- Operating System
- Protocol Stack
- Special Devices

The politics behind the IPv6 Ready Logo are highly charged. For example, the PRC successfully argued against the direct inclusion of IPSec into the calculus behind the awarding of the Logo. This decision could have enormous implications for individual privacy for many years to come. It is an open question as to whether the U.S. and the EU should have pushed harder during the pertinent negotiations.

The Internet Assigned Numbers Authority (IANA) has published a series of principles that it hopes will guide Internet registries and service providers through the transition:

- Aggregation: distributed in a hierarchical manner, according to the topology of network infrastructure, to permit the aggregation of routing information by ISPs, and to limit the expansion of Internet routing tables
- Uniqueness: every assignment must be unique worldwide
- Registration: in a registry database accessible by the Internet community (within the context of reasonable privacy considerations and applicable laws)
- Seek: "fair" distribution, contiguity of addresses
- Avoid: fragmentation, wasteful practices, stockpiling of unused addresses, overhead

12. Privacy and anonymity

The details of the transition to IPv6 may have serious implications for individual privacy in the years to come. The chief concern is that, in a world where everything is addressable, and where two communicating points on the Internet will almost certainly be connected via end-to-end connectivity, that it will be relatively easy for humans to be directly associated with the IP addresses that they use.

Privacy advocates fear that a widespread use of static IP addresses could more easily trace human beings to their online behavior. For its part, Law Enforcement will of course view the new protocol from the angle of how it will affect prosecutions through better attribution of Internet crimes.

Within this context, there lies the potential for a serious IPSec paradox. Privacy advocates may gain the encryption they need for the confidentiality of their communications. However, the lack of a NATed or dynamic IP address may make basic traffic analysis – seeing who has communicated with whom – easier.

Enter the IPv6 Privacy Address Scheme. This mechanism allows for the randomization of outgoing IP addresses during the initial phase of communications. Again, only time will tell how this will play out, but interested parties will do well to pay close attention to these developments.

Finally, since the transition to IPv6 is a worldwide endeavor, we can expect that there will be significant differences of opinion on these sensitive subjects in different parts of the world.

In Asia, for example, it is generally accepted that there is less privacy in one's daily life, and that the general populace is more comfortable with government intrusiveness than in Europe. European Internet users, highly motivated to protect their online anonymity, are likely to have significantly higher expectations regarding the privacy aspects of the new protocol.

The U.S., in contrast, is thought to be somewhere in the middle. Of course basic banking and financial matters must be secure, but there is a strong desire in the U.S. to give Law Enforcement all of the tools that it needs to prosecute real criminals.

13. Cost

Last, but by no means least, what will the transition to IPv6 cost you? This is a difficult question to answer. Official U.S. government estimates have ranged from $25 to $75 billion for the entire nation-wide effort. There is also a loud chorus of voices in the background claiming that the transition is essentially free, since most software and hardware these days is IPv6 compatible, and that natural "tech refresh" cycles will eventually absorb nearly all of the cost.

Such calculations, however, should not merely rely on supplies that can be bought at the local computer store. Human resources must be trained on IPv6, and there is no doubt that the learning curve is steep. You are probably used to an IP address that looks like this: 207.46.19.60. An IPv6 address looks – and is – quite different: 2001:0db8:0000:0000:0000:0000:1428:57ab, or, for short, 2001:0db8::1428:57ab. Employees will need to gain awareness, understanding, and training on IPv6, all of which are currently lagging far behind where they need to be. As the intelligence and capabilities of our machine counterparts rise, so will that of our human employees. Today, the average new hire has a couple of years experience with TCP/IPv4, and zero with IPv6!

A more appropriate question to ask is what your Return on Investment (ROI) will be. On the plus side, there should be savings in the avoidance of NAT and its associated headaches and workarounds. Further, there should be IPv6-facilitated breakthrough technologies, such as VoIP and improved eCommerce, which may offer both monetary savings and business opportunities. Such opportunities will stem, in part, from a more capable code stack, IPsec integration, and interoperability; these in and of themselves should be preferred every time, if possible, to workarounds and band-aids.

There is a final comparison here worth noting. IPv6 and Y2K share some interesting similarities. Smart resource decisions will have to be made, and you may need to ditch some of your long-cherished programs. Also, niche System Administration skills during the transition will be in high-demand. But who in the world is going to type in those (very long) new IP addresses, and how much are you going to pay them? ;)

14. Our recommendations

- Turn it off! Allow and authorize based on need
- Block Teredo UDP port 3544
- Keep an closer eye on your enterprise boundaries
- Develop an IPv6 policy and training plan
- Begin now, but do not move too fast!
- Budget for expertise, SW, HW concerns
- Get a small number of staff completely up to speed
- Ensure that the transition is controlled
- Calculate your Return on Investment (ROI)
- Identify IP-enabled devices
- Scan and enumerate (remember Y2K?)
- Many enterprises will need outside support
- Make smart resource decisions
- Upgrade through Tech Refresh
- Let IPv6 help solve current problems, not create new ones
- Should I upgrade certain products, or ditch them altogether?
- Watch security developments, minimize disruption

References

Adler, Neil. “Command Information to add hundreds of jobs in N. Va.”, Washington Business Journal, September 11, 2006, http://milwaukee.bizjournals.com/washington/stories/2006/09/11/daily25.html

Barnard, Patrick, “Hexago Successfully Ports Quake III Arena to IPv6”, June 06, 2006, http://news.tmcnet.com/news/2006/06/06/1670879.htm

Barnard, Patrick. “IPv6 Forum Opens Office in Israel”, July 31, 2006, http://ipcommunications.tmcnet.com/hot-topics/security/articles/2044-ipv6-forum-opens-office-israel.htm

“Beijing's 11th Five-Year Plan: Part 3”, 28 March 2006, http://beijinglives.com/Beijing-China-News-and-Views/China-Domestic-News-Agents/Beijing-5-year-plan-2006-3-1/

Bell, Stephen. “Government Shared Network considering IPv6”, Computerworld, 11 April, 2006, http://computerworld.co.nz/news.nsf/0/7BE288C7B08D6374CC257148007C1F71?OpenDocument

“China lead India toward IPv6”, August 18, 2006, http://www.siliconindia.com/shownewsdata.asp?newsno=32844

“China Telecom expands core with Juniper Networks”, August 7, 2006, http://lw.pennnet.com/Articles/Article_Display.cfm?ARTICLE_ID=261775&p=13

“China Telecom Expands Coverage of CN2 Network”, July 06, 2006, http://press.xtvworld.com/article12589.html

Cringely, Robert X. “Through a Metal Detector, Darkly: How NAT traversal could set us free”, August 11, 2006, http://www.pbs.org/cringely/pulpit/pulpit20060811.html

Dang, Amandeep Singh. “The challenge of a unified platform”, May 2, 2006, The Star Online, http://star-techcentral.com/tech/story.asp?file=/2006/5/2/corpit/14077567&sec=corpit

Dickens, David. “New Zealand way behind on IPv6”, May 23, 2006 http://m-net.net.nz/index.php?option=com_content&task=view&id=218&Itemid=95

Essex, David. “Don’t let applications be the missing link in your agency’s successful move to IPv6”, GCN, http://www.gcn.com/print/25_24/41634-1.html

Evers, Joris. “Symantec sees Achilles' heel in Vista”, ZDNet, July 19, 2006, http://www.zdnet.com.au/news/security/soa/Symantec_sees_Achilles_heel_in_Vista/0,2000061744,39263930,00.htm

Evers, Joris. “Will Vista stall Net traffic?”, CNET News, September 7, 2006, http://www.zdnetasia.com/news/software/0,39044164,39433774,00.htm

"Five IPv6 Tips from an Early Adopter", August 14, 2006, http://www.toptechnews.com/news/Five-IPv6-Tips-from-an-Early-Adopter/story.xhtml?story_id=032002XV9B8G
"Framework could aid global information exchange", April 22, 2006, Federal Computer Week, http://www.fcw.com/article92807-04-03-06-Print
Gedda, Rodney. "Coonan to unveil $800K IPv6 portal", Computerworld, February 8, 2006, http://www.arnnet.com.au/index.php/id;1903951545;fp;8;fpid;0
Giannacopolous, Peter. "Nmap 4.01 improves upon past releases", Information Security, May 3, 2006, http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci1186583,00.html
Gilmer, Chris. "Google is not an ISP", July 4, 2006, http://google.weblogsinc.com/2006/07/04/google-is-not-an-isp/
Gralla, Preston. "Windows Vista Collaboration: A Big Step Forward, But Still An Island", Networking Pipeline, May 25, 2006, http://www.informationweek.com/software/showArticle.jhtml?articleID=188500311&subSection=Operating+Systems
Grimes, Brad and Jackson, Joab. "An attempt to define 'IPv6-capable'", May 15, 2006, http://www.gcn.com/blogs/tech/40732.html
Grimes, Brad. "The riddle of IPv6", Washington Technology, August 7, 2006, http://www.washingtontechnology.com/news/21_15/cover-stories/29062-1.html
Hardy, Michael. "AT&T, Global Crossing team to aid IPv6 transition", August 30, 2006, http://www.fcw.com/article95845-08-30-06-Web
Hardy, Michael. "Proposed rule would mandate IPv6", August 28, 2006, http://www.fcw.com/article95816-08-28-06-Web
Hart, Kim. "Center Opens to Train New Web Protocol Users", Washington Post, September 14, 2006, http://www.washingtonpost.com/wp-dyn/content/article/2006/09/13/AR2006091301959.html
Hassell, Jonathan. "What you need to know about IPv6", Computerworld, July 2006, http://www.techworld.com/networking/features/index.cfm?featureID=2684&pagtype=samecatsamechan
Hayes, Simon. "Dissidents stay away from Net, April 04, 2006, http://australianit.news.com.au/articles/0,7204,18693900%5E15343%5E%5Enbv%5E15306-15317,00.html
Hoskins, Robert. "Cisco Systems Joins Pioneering French Technology Initiative "Secure Communicating Solutions" (SCS)", Broadband Wireless Exchange, May 4, 2006, http://www.bbwexchange.com/publications/page1423-133633.asp
"Hughes Network Systems' satellite to feature single-hop data transmission", TMCnet.com, April 12, 2006, http://www.tmcnet.com/usubmit/-hughes-network-systems-satellite-feature-single-hop-data-/2006/04/12/1567112.htm
"ICANN Ratifies Global Policy for Allocation of IPv6 Addresses", AGIP News, September 12, 2006, http://www.ag-ip-news.com/GetArticle.asp?Art_ID=3524&lang=en
"India, China Come Together for IPv6", Techtree, Aug 19, 2006, http://www.techtree.com/techtree/jsp/article.jsp?article_id=75302&cat_id=643
"IPv6 Address Cookies", 14 June 2006, http://www.it-observer.com/news/6466/ipv6_address_cookies/
"IPv6 Forum India, China IPv6 Council to Promote IPv6 Deployment", IT News Online, August 17, 2006, http://www.itnewsonline.com/showstory.php?storyid=5050&scatid=8&contid=1
"IPv6 in India - China to show the way", August 18, 2006, Moneycontrol, http://www.moneycontrol.com/india/news/technology/ipv6networkchina/ipv6indiachinatoshowway/market/stocks/article/235046
"IPv6 India's MoU with China chapter", http://www.expresscomputeronline.com/20060918/market10.shtml
"IPv6: How Painless (or Painful) Will the Transition Be?", PRNewswire, April 27, 2006, http://www.commsdesign.com/press_releases/prnewswire/showPressRelease.jhtml?&CompanyId=1&HeadlineId=X467657
Jackson, William. "IPv6 security: The forgotten element", GCN, http://www.gcn.com/print/25_24/41635-1.html
Jackson, William. "IPv6: The Future is Now", GCN, http://www.gcn.com/print/25_24/41632-1.html
Jaques, Robert. "IEEE looks to the 'New Internet'", VNUNET, May 26, 2006, http://www.computing.co.uk/vnunet/news/2157073/ieee-looks-internet
Keizer, Gregg. "Symantec Finds Flaws In Vista's Network Stack", TechWeb, Jul 19, 2006, http://www.informationweek.com/news/showArticle.jhtml?articleID=190700122&subSection=Breaking+News
Kelley, Jeffrey. "Center to teach IPv6 in Fairfax", Times-Dispatch, September 14, 2006, http://www.dailyprogress.com/servlet/Satellite?pagename=CDP%2FMGArticle%2FRTD_BasicArticle&c=MGArticle&cid=1149190611979&path=!business

Kirkpatrick, David. “China's lead in tech”, Fortune Magazine, May 19, 2006, http://money.cnn.com/2006/05/19/technology/fastforward_fortune/index.htm
“Korea Ranks 9th in IP Address Number”, Yonhap, June 13, 2006, http://times.hankooki.com/lpage/biz/200606/kt2006061320204711910.htm
Krebs, Brian. “Some Highlights from CanSecWest”, The Washington Post, http://blog.washingtonpost.com/securityfix/2006/04/some_highlights_from_cansecwes_1.html
Kreidler, Tom. “Testbeds boost cyber security research”, http://www.enterprisenetworksandservers.com/monthly/art.php?2083
Lawson, Stephen. “Analysts debate need for IPv6 in near future”, Network World, June 26, 2006, http://www.networkworld.com/cgi-bin/mailto/x.cgi
Lemos, Robert. “Covert channel tool hides data in IPv6”, SecurityFocus, August 14, http://www.theregister.co.uk/2006/08/14/ipv6_covert_channel_tool/
Maltagliati, Toni. “Agencies plan their move to a bigger, better Internet”, April 13, 2006, http://federaltimes.com/index.php?S=1673639
Mann, Justin. “IPV4 addresses gone in six years”, TechSpot, May 5, 2006, http://www.techspot.com/news/21502-ipv4-addresses-gone-in-six-years.html
Mark, Roy. “Red Flags Raised About Fed's IPv6 Plans, June 30, 2005, http://www.enterprisenetworkingplanet.com/netsp/article.php/3516816
Marsan, Carolyn Duffy. “Feds lack plans, funding for IPv6, survey says”, Network World, June 26/06, http://www.networkworld.com/cgi-bin/mailto/x.cgi
Milchman, Eli. “Google Denies ISP Rumors”, June 30, 2006, http://www.wired.com/news/technology/internet/0,71293-0.html?tw=wn_index_2
Miller, Jason, “Task forces mark the trail”, GCN, http://www.gcn.com/print/25_24/41633-1.html
Miller, Jason. “Agencies find there’s no single path to IPv6”, GCN, http://www.gcn.com/print/25_7/40248-1.html
Morgenstern, David. “Google and Its Continuing Dark Fiber Mystery”, June 23, 2006, http://www.eweek.com/print_article2/0,1217,a=181772,00.asp
Morgenstern, David. “IPv6 Still Gets No Respect in the United States”, EWeek, June 15, 2006, http://www.eweek.com/article2/0,1895,1977463,00.asp
Naraine, Ryan. “ICANN Adds IPv6 to Root DNS”, July 22, 2004, http://www.enterprisenetworkingplanet.com/netsp/article.php/3384791
“NTT America Launches Consultancy Service to Address Mandatory IPv6 Adoption”, United Business Media, May 16, 2006, http://sev.prnewswire.com/telecommunications/20060516/NYTU04516052006-1.html
Olchondra, Riza T. “Gov’t eyes perks for new telecom system”, Inquirer, August 24, 2006, http://technology.inq7.net/infotech/infotech/view_article.php?article_id=17096
Ou, George. “DNS fear mongering and Windows Vista”, September 6, 2006, http://blogs.zdnet.com/Ou/?p=310
Parthajit. “NTT America announces multi-continent 10 Gigabit ethernet access”, July 21, 2006. http://www.digitalmediaasia.com/default.asp?ArticleID=16891
Patrizio, Andy. “We're Running Out of IP Addresses”, May 12, 2006, http://www.enterprisenetworkingplanet.com/netsp/article.php/3605881
“President urges action on adoption of 'new internet'”, IEEE-USA, May 25, 2006, http://www.physorg.com/news67781150.html
Reimer, Jeremy. “Vista's role in Imminent Net Death overstated”, September 7, 2006, http://arstechnica.com/news.ars/post/20060907-7691.html
Rogers, Garett. “Google's secret IPv6 plans”, June 24, 2006, http://blogs.zdnet.com/Google/index.php?p=241
Rubens, Paul. “Is Your Network Ready for IPv6?”, April 13, 2004, www.enterprisenetworkingplanet.com/netsp/article.php/3339651
Rubens, Paul. “The IPv6 Move: Bigger and Harder Than Y2K”, May 24, 2006, www.enterprisenetworkingplanet.com/netsp/article.php/3608721
Schluting, Charlie. “Networking 101: Subnets Realized (Including IPv6)”, December 9, 2005, http://www.enterprisenetworkingplanet.com/netsp/article.php/3569916
“Should the Net Be Neutral?”, The Wall Street Journal (Online), May 24, 2006, http://online.wsj.com/public/article/SB114839410026160648-M3wlxcUPk357fsGiHNM3BsUiCKY_20060622.html?mod=tff_main_tff_top
Sims, David. “Lucent Shows Off IP in Nice, Kazakh OSS Deal”, TMCnet, May 16, 2006, http://news.tmcnet.com/news/-lucent-ip-kazakh-france-/2006/05/16/1650880.htm
Sullivan, Laurie. “Cyberspace running out of room”, May 8, 2006, http://www.crn.com.au/story.aspx?CIID=37479&src=site-marq

"Time May Be Right for Government Agencies to Adopt Linux", April 14, 2006, Federal Computer Week, http://www.technewsworld.com/story/49944.html
"To the edge of cyberspace", Editorial, San Francisco Chronicle, May 17, 2006, http://www.sfgate.com/cgi-bin/article.cgi?file=/chronicle/archive/2006/05/17/EDGDOIJKNP1.DTL
Tynan, Dan. "IPv6 on the dotted line", GCN, http://www.gcn.com/print/25_24/41637-1.html
Uwaje, Chris. "IT road map: The next great wave 2006-2008 - Part 2", April 12, 2006, http://www.vanguardngr.com/articles/2002/features/technology/tec512042006.html
Villafania, Alexander. "CICT eyes widespread IPv6 deployment in RP", INQ7.net, August 23, 2006, http://newsinfo.inq7.net/breakingnews/infotech/view_article.php?article_id=16849
Weigelt, Matthew. "Agencies slow in adopting IPv6, survey finds", June 26, 2006, http://www.fcw.com/article95023-06-26-06-Web
Weigelt, Matthew. "Councils: Make more IPv6 products available", August 24, 2006, http://www.fcw.com/article95777-08-24-06-Web
Weigelt, Matthew. "GAO: Agencies lag in IPv6 transition", July 31, 2006 http://www.fcw.com/article95491-07-31-06-Web
Weiss, Todd. "GAO: Some federal agencies lagging in move to IPv6 standards", Computerworld, August 1, 2006, http://www.computerworld.com/action/article.do?command=viewArticleBasic&taxonomyName=development&articleId=9002129&taxonomyId=11
Wilson, David. "Privacy, What Privacy?", April 2, 2006, http://english.ohmynews.com/articleview/article_view.asp?article_class=4&no=279541&rel_no=1
Worthen, Ben. "A New Weapon for Control and Intelligence?", July 15, 2006 CIO Magazine, http://www.cio.com/archive/071506/china_sidebar1.html
Worthen, Ben. "China Builds a Better Internet", July, 2006, CIO Magazine, http://www.realtechnews.com/posts/3289
Xtv. "IPv6 to Play Vital Role in Future Disasters", posted May 15, 2006, http://press.xtvworld.com/article11435.html
Yegyazarian, Anush and Spring, Tom. "Should the Internet Play Favorites?", PC World magazine, March 24, 2006, http://www.pcworld.com/resource/printable/article/0,aid,124926,00.asp#

Leap in the Dark: An Attempt to Theorize Personnel Anomaly Detection for Countering Insider Threats

Shuyuan Mary Ho
Syracuse University, NY, USA
smho@syr.edu

Abstract: Personnel Anomaly Detection (PAD) is a study that attempts to uncover the mechanism for detecting malicious personnel behaviour within the context of an organization. Human behaviour is polymorphous. Humans tend to hide their true intentions from their peers. Human behaviour tends to be self-modifying in response to changes in one's environment. This research attempts to propose a theoretical framework for identifying risk indicators in personnel anomalous behaviour. A socio-technical approach is used to derive the framework, which is designed to collect and analyze data through analysis of critical incident technique and natural language processing. This research will contribute to corporate personnel security for countering insider threats.

Keywords: Personnel anomaly detection, insider threat, corporate defence mechanism, personnel security, polymorphous human behaviour.

1. Introduction

Threats to organizations' intellectual property and informational assets have caused increasing social awareness of the need for information security. Countermeasures are mostly designed and developed to prevent threats from outside attacks. Technologies for the purposes of intrusion prevention, intrusion detection and intrusion protection are commonly implemented to actively prohibit unnecessary access to internal corporate resources. However, threats from insiders are subtle and difficult to anticipate. As corporate assets, employees benefit an organization. But their inside knowledge of corporate resources can threaten corporate security due to the possibility of unauthorized disclosure of classified information. Sensitive information that is compromised could even be maliciously used to counter-attack the organization. Identifying and detecting polymorphous personnel behaviour that might harm the organization is critically important. The challenge of Personnel Anomaly Detection (PAD) study lies in the multidisciplinary framework. A socio-technical approach is recommended to detect traces of personnel anomalies as early warning signs for countering insider threat (Ho, 2006).

This paper contains five major sections describing the study of PAD. The significance of PAD is identified as a result of *the Evolution of Intrusion*. Since the emphases of intrusion have been shifted from outside threats to inside threats, a question is pondered and raised in understanding insider threats phenomenon stated in the *Research Question* section. The definition, scope and conceptual framework of PAD are given in the *Conceptualization* section. The multidisciplinary nature of the PAD study is briefly described in the *Challenge* section. The *Conclusion* justifies and summarizes this research in progress.

2. The evolution of intrusion

Threats to protected information and resources exist because of the service value and criticality of corporate assets. No matter how advanced the protection measures are, human behaviour remains the same. Humans have tendency to create and build. Humans also have the tendency to break away from that which is regulated or standardized. "Hacking" into a network to steal services or confidential information started long before the World Wide Web was developed. Soon after AT&T introduced the first dataphone in 1960's, their phone network was hacked. In 1988, the Morris computer worm was created to experiment with the effectiveness of a self-replicating program. This accidentally resulted in more than 6,000 networked computers of ARPANET being infected. The first cyber-espionage case that made international headlines was in 1989, when hackers were arrested for breaking into US Government and corporate computers, then sold the operating system source code to the Soviet KGB.

Intrusion techniques have evolved over time alongside technological advances. In the 1990's, multifarious programs such as computer viruses, worms, and "Trojan horse" programs were developed to infect and block normal services. Computer infection by such multifarious programs progressed from floppy media transmission, to boot sector-executable file infection, and then further to the infection of http, smtp, and ftp services. In 1992, polymorphic viruses were unleashed. These were designed to self-modify and self-transform, thereby avoiding pattern recognition by virus protection programs. The "Melissa" virus - an email propagating virus - started in 1999. It was the first distributed virus over Usenet that shut down Internet mail

systems. In 2000, the "ILoveYou" worm was the first VBScript worm that infected music and multimedia files running on Microsoft Windows operating systems. This worm forced the victim's computer to consume its resources and further obstructed the communication media with a denial of service attack by replicating itself to the email recipients in the infected Outlook address book. In 2001, the "CodeRed" worm was spread to IIS web servers, which were vulnerable to buffer overflow attack. Soon after, the very severe "Nimda" worm was launched and spread out with five ways of infecting vectors. It swiftly infected through email transmissions, network shared access, browsing a compromised web server, exploiting IIS directory traversal vulnerability, and was designed to use the backdoor left open by the "CodeRed II" worm. In 2002, a metamorphic virus, Win32/Simile, was written in Assembly language to attack Microsoft Windows. The virus would reform its code every time it was executed, rebuilding itself with random factors, and replacing its virus code in a virus body encryptor.

As a result of evolving virus attacks and network intrusion from outsiders, the focus of security architects has traditionally been focused on mitigating outside attacks. But this began to change with the case of Robert Hanssen, an FBI agent who worked as a spy for the Soviet Union / Russia for 15 years. His spying started in 1970 and continued until he was arrested in February 1985. In another example, Johnson Pollard had high-level security clearance, and was arrested in 1985 for passing classified U.S. information to the Israelis (Benkoil, 1998; Hayden, 1999; Lamar, 1986). In 1993, the General Accounting Office (GAO) reported that insiders' abuse of the National Crime Information Centre (NCIC) for personal reasons or profit threatened the safety of U.S. citizens (Park & Ho, 2004). It is not just public sectors that have suffered from the possibility of insider threats. Organizations in the private sector have also been subject to the threat of malicious insider activities. For example, the Enron scandal uncovered in 2001 was one of these cases. With 21,000 employees spread throughout 40 countries, Enron's executives lied about its corporate profits and concealed its corporate debts. This is a case where business executives with high social power were able to perpetrate a high-level scam, putting the entire company at risk (BBC Business News Online).

As the focus of intrusion strategies has shifted from outsider threats to insider threats, the emphasis on information security countermeasures has also started to shift; from a system-based approach to a socio-technical approach. The "dark side of man" (Ghiglieri, 1999) can alter the integrity of an employee, who might betray the organization after s/he has obtained high security clearance. By understanding the mechanisms through which potentially destructive behaviour is initiated, it may be possible to detect changes in an individual's personality or conditional state that could eventually lead to a compromise of corporate information security. Through classifying incidents that lead up to an insider's mal-behaviour, these types of behaviours can be codified and identified to protect the integrity of the information systems. While information classification and controllability through encryption, setting security models and standardizing security practices, continues to be important, monitoring authorized personnel access and understanding polymorphous human behaviour will lead to proactive resolutions that can prevent insider threats. While the most malicious virus lies in its polymorphic or metamorphic forms, the threat from malicious insiders is believed to be reflected in polymorphous human behaviour. In order to prevent such an insider threat, it is necessary to study polymorphous human behaviour so that countermeasures can be developed to ward against insider threats.

3. Research question

Polymorphic or metamorphic viruses/worms have tendency to transform their operating codes while infecting information systems. It takes highly sophisticated algorithms within anti-virus programs to recognize coded patterns so as to quarantine an ever-changing virus. In the same way, human behaviour is also polymorphous. Humans tend to hide their true intentions from their peers. In fact, research implies that human behaviour also tends to be self-modifying through changes in one's environment. Pattern recognition from statistical analysis of logged events can only provide a black-box view to reduce the probability of identifying possible malicious behaviour. My research question is then to discover; "**Why is polymorphous human behaviour complex to recognize in the context of organizations**?" More specifically; "What are the random and constant factors in employee's personality and conditional observables that could impact or even threaten the organization? Are there any indicators or threat characteristics that can expose such behaviour in advance?"

4. Conceptualization

The study of Personnel Anomaly Detection (PAD) attempts to derive a theoretical framework that will explain insider threat phenomenon, and make predictions from anomalous behaviour. Weick encourages the process of "theorizing" a study, which theoretical development is at its early stage (Weick, 1995). PAD study

attempts to approximate conditions in personnel anomaly detection and theorize on the insider threat phenomenon by identifying relationships among disparate constructs that can explain, synthesize, generalize and idealize the mechanisms for detecting anomalous behaviour.

4.1 Definition

Personnel Anomaly Detection is defined as *"an approach or a mechanism that deconstructs a disguised act / state from the norm of a person with authorized access to organizational internal and external resources, and flags such a deviational behaviour that comes in observable patterns, in which a person acts in response to a set of conditions or a particular situation / stimulus, and distinguishes or suggests that a possible insider threat could come from specific perpetrator(s), a series of suspicious events or eccentric states."*

4.2 Construct identification

The more critical a job is in an organization, the more likely it is that the individual has access to vital informational resources. Since accessibility and depth of knowledge on corporate resources depends on the level of personnel trustiness, once the integrity and ethical level of an individual is altered, corporate resources are at risk, and vital information can be compromised. In turn, those individuals with the highest security access will present the greatest danger from the perspective of an insider threat. From a study on insider threats done by U.S. Secret Service, it was determined that poor communication and social network anomalies can be identified - along with gaps in the coordination framework of a collaborative teamwork. Embedded in a mesh of coordination mechanisms and communications, a person given high social power but without sufficient trustiness creates a single point of trust failure with the potential for causing insider threats.

To conceptualize on how to detect polymorphic personnel anomalies, my initial findings for this PAD study are framed by six constructs that help project high impact levels of impact when the personnel anomalous behaviour is recognized and detected. The six constructs are described below.

1. *Motivation*: Motivation means an individual's initiation, direction, intensity and persistence of behavior (Geen, 1995). Motivation is extrinsic, temporal and dynamic. It means an individual is having the desire and willingness to do something. In the phenomenon of insider threats, external stimulations may influence individual's emotional stabilty and motivate them to deviate their normal behavior
    - *Financial condition*: Irregular or excessive financial gain might serve as an indicator; however. When a person suffers from an urgent financial crisis, s/he might be tempted to engage in irregular activities for illegal gains. His/her irregular acts might be hidden within regular social activities.
    - *Emotional condition*: Although it is impossible to forecast and predict a person's integrity level, it is possible to monitor a person's emotional state, which includes observable data on his/her intrinsic character, extrinsic behaviour, family status (such as relatives), and circle of friends, etc.
2. *Personality*: Personality is intrinsic. It can be defined as a dynamic and organized set of characteristics possessed by a person that uniquely influences his or her cognitions, motivations, and behaviors in various situations (Ryckman, 2004).
3. *Trustiness*: According to Ghiglieri (1999), "the dark side of man" might alter the integrity level of an employee, who might betray the organization after s/he obtains high security clearance (Ho, et al. 2005). Such a threat is worse when the network is segmented and information is classified. Management might think (with inappropriate self-confidence) that the safeguards have been established. But indeed, the threats lie in the regular and authorized access.
    - *Access Records* (facility, resources, and system logs): Access logs may give a story of irregular access to certain database or physical facilities. However, regular access logs should also be kept as records for forensics purposes since no where it monitors what types of data or information the person accesses.
4. *Social power* (job criticality): Critical positions include both executive-level and system administration. "Rank" is also a problem sometimes in the government and military settings. First, highly-ranked officers might be exempted from the tedious administrative oversight, or might not be required to follow the administrative procedures that the security policy requires. Second, higher authority may often request classified information without proper authorization. Such untold norms practiced in the organization can bypass access control mechanisms for who accesses what resources, and authentication of proper personnel to process classified information. In addition to proper authorization, the increase in

administrative work often causes the application for information obtainment to be opted out from regular access control mechanisms.

- *System Administrative Privileges*: With the domain knowledge, a system administrator tends to be seen as the "God" of the networked environment. Access to intellectual property and resource is virtually unlimited.
- *Business Domain Knowledge*: Domain experts have key knowledge on detailed business intelligence and confidential information. The knowledge of corporate internal resources enables malicious insiders to obtain confidential or classified information easily.

5. *Norm*: The standard, model and patterns in an organization can be observed implicitly and explicitly through various forms.
    - *Policy*: Corporate policy - especially relating to access control - is an explicit sub-construct that details specifically what resources can be access by whom.
    - *Common Sense*: Common sense is an implicit sub-construct that details how untold, preferred practices are normally done.
    - *Coordination*: Coordination, as defined by Malone and Crowston (1994), is managing dependencies between related activities. Good coordination makes any teamwork or organization joint-effort more harmonious, unnoticeable and sometimes even "invisible." In the framework of the Coordination Theory, Malone and Crowston define components that are seen as dependencies between activities that are associated with individuals or entities. Activities imply movement, leading to associations such as task assignments among individuals and entities. Analyzing organizational activities in light of Coordination Theory requires the analysis of natural language processing and cognitive mapping, and includes the linkages between the two (Ho, et al, 2006).
    - *Communication Patterns*: From a mesh of communication patterns, gaps can be easily observed. Communication gaps in threaded discussion can be clues to underlying behaviour, both positive and negative.
6. *Information Behaviour*: It is a social-technical perspective looking at how information is perceived by individuals and how meanings and knowledge are constructed through information in organization. Personnel anomalous behaviour can be examined in the information that is stored and transferred within corporate entities through coordination mechanisms and communication threads.

4.3 Conceptual framework

PAD study is a multi-level analysis, which is applied from the lenses of both organizational and individual levels of analysis, so as to analyze the phenomenon of personnel anomalies in organizations. As illustrated in Figure 1, main constructs are represented in rectangular boxes; sub-constructs are represented in circles. Relationships among constructs are represented by uni- or bi- directional arrows.

Various relationships among constructs are illustrated in the Personnel Anomaly Detection conceptual framework. The individual level of analysis is used to analyze the personnel phenomenon from the social perspective. The constructs such as motivation, personality, trustiness and social power are major factors that would influence personnel behaviour. The organizational level of analysis is seen in the main construct of social norms. Organizational norms are usually influenced by explicit corporate policy and rules, implicit common sense and practices, along with group coordination dependencies, and communication gaps (Figure 1).

Both the individual's emotional stability and the fluctuation of financial gains could have negative effects and motivate individuals to initiate job-related incidents. Trustiness of individuals is often motivated by situations and personality. Social power is a key construct that influence corporate security either positively or negatively, depending on the trustiness of this employee. Social power resides in authoritative positions in organizations that can be influenced and observed through corporate policy, common sense practice, communication patterns and coordination (Figure 1).

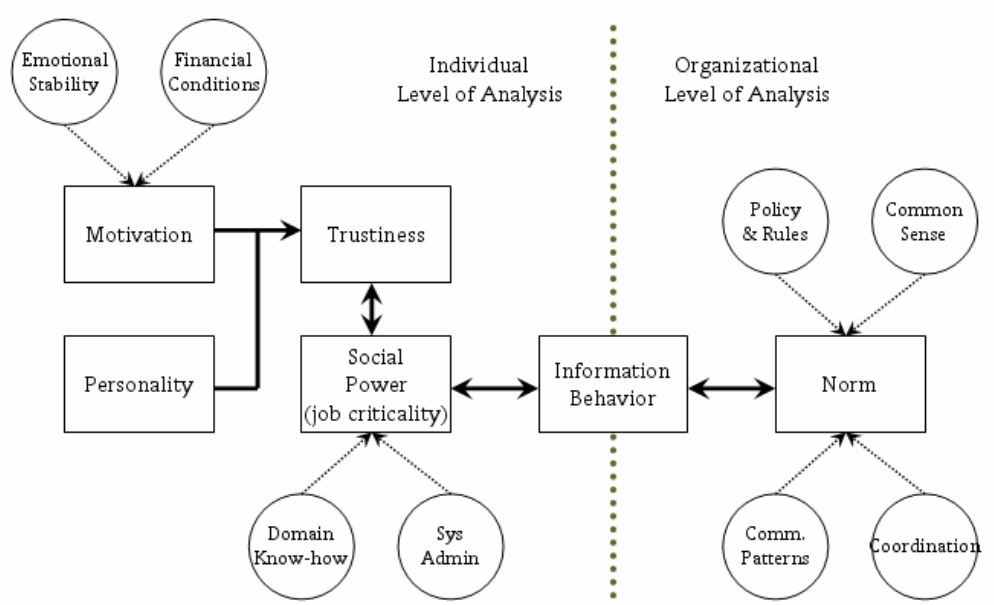


Figure 1: Conceptual framework

In the organizational aspects of constructs, corporate security policy and common-sensed administrative procedures can serve as filters to filter out undesired individual behaviours within the organization. Both the corporate security policy and the common sense procedures can positively influence the coordination and communication patterns among teams and small groups in the organization. User's information behaviour that describes social aspects of how information is transferred between organizations and personnel can bridge the two levels of analysis and identify risk indicators of an individual's uncommon behaviours from a series of events or states.

5. Challenges in multidisciplinary domain and future work

PAD is a multidisciplinary study in nature. In order to understand the socio-technical aspects of mitigating insider threat phenomenon, PAD draws from the disciplines of information systems, information science, organizational science, and social psychology. PAD utilizes classification and categorization principles in the information science discipline. Technical approach is adopted to extract important concepts and discover new knowledge from the data collected. For example, natural language processing (NLP) can be utilized as a method that analyzes information behaviour. Social science discipline is adopted to understand humans and organizations while cognitive psychology can be adopted to understand human behaviour and organizational norms. Critical incident technique, for example, can be used to capture and analyze this switching behaviour that was led to both positive and negative directions (Flanagan, 1954; Butterfield, 2005; Roos, 2002; Edvardsson & Roos, 2001).

The future work of PAD study includes proposition establishment, variable identification, and hypotheses testing of relationships among variables. The dependent variables are considered in the construct of information behaviour. Variable selection and hypotheses testing can affect the validity of framework generalization. This research could be designed to use longitudinal case studies with participant observation, but the generalize-ability of the case study needs to be considered in order for this framework to be theorized. Survey and interviews could fulfil the needs for study generalize-ability. However the participant selection may require significant in-depth understanding of the insider threat phenomenon. When a technical approach is considered (such as in natural language processing or data mining), it can be applied in two dimensions. The first dimension is to use NLP in extracting knowledge and categorizing interviewed data. The second dimension is to use NLP in processing communication data of participants such as the email text or other digital dialogs. This application could be considered a form of participant observation in case studies.

6. Conclusion

In order to effectively quarantine polymorphic virus codes, it has been necessary to study how the codes change. Likewise, by understanding polymorphous human behaviour, we may be able to detect malicious

behaviour and counter insider threats by studying how human behaviour changes through how information is used and transferred within an organization. The PAD indicators provide clues, evidence and early warning signs to investigate unwanted behaviours; however, it is important to note that they by no means provide full assurance to convict crimes. Human intervention will be necessary in this loop, so that a security officer can audit and work with an anomaly detection decision engine to classify and update behaviour patterns and security policies. To conclude, the findings of this exploratory study will contribute to the design, criteria, and mechanism of personnel anomaly detection to defend against insider threats.

Acknowledgement

This research has been supported by 2004-2006 (ISC)2 scholarship. Shuyuan is currently being advised by Jeffrey M. Stanton, Ph.D. Program Director at School of Information Studies, Syracuse University. The author graciously thanks Jeffrey M. Stanton, Elizabeth Liddy, Kevin Crowston, Martha Garcia-Murillo, Barbara Kwasnik, Mike D'Eredita, Robert Heckman for their advice, and Conrad Metcalfe for editing assistance.

References

Benkoil, D. (1998) "An Unrepentant Spy: Jonathan Pollard Serving a Life Sentence", [ABCNEWS.com], October 25, 1998.

Butterfield, L. D., Borgen, W. A., Amundson, N. E., and Maglio, A. T. (2005). Fifty Years of the Critical Incident Technique: 1954-2004 and Beyond. Qualitative Research, Vol. 5, No. 4, pp. 475-497. SAGE Publications, Thousand Oaks, CA. Retrieved from http://qrj.sagepub.com/cgi/content/abstract/5/4/475 on December 17, 2005.

Edvardsson, B. and Roos, I. (2001). Critical Incident Techniques: Towards a Framework for Analyzing The Criticality of Critical Incidents. International Journal of Service Industry Management, Vol. 12, No. 3, 2001, pp. 251-268. MCB University Press.

Flanagan, J. C. (1954) "The Critical Incident Technique", Psychological Bulletin, Vol. 51, No. 4, July 1954, pp. 327-359.

Geen, R. G. (1995), *Human motivation: A social psychological approach*. Belmont, CA: Cole.

Ghiglieri, M. P. (1999) "The Dark Side of Man: Tracing the Origins of Male Violence", Perseus Books, 1st Edition, March 1999.

Haydon, M. V. (1999) "The Insider Threat to U.S. Government Information Systems", National Security Telecommunications and Information Systems Security Committee (NSTISSAM) INFOSEC 1-99, July 1999. http://www.nstissc.gov/Assets/pdf/NSTISSAM_INFOSEC1-99.pdf.

Ho, S. M. (2006) Toward a Deeper Understanding of Personnel Anomaly Detection. Encyclopaedia of Cyber Warfare and Cyber Terrorism, 1st Edition, IDEA Publications.

Ho, S. M. & Eseryel, U. Y. (2006) Monitoring-based Coordinated Defense Through the Lens of the Coordination Theory. Proceedings of International Conference of Information Resource Management Association (IRMA 2006), Hershey, PA.

Ho, S. M. & Li, Q. (2006) Modelling an Effective Coordinated Defense Through the Lens of the Coordination Theory. Proceedings of First International Conference on Information-Warfare and Security, Academic Conference Limited, England.

Ho, S. M. & Caldera, C. (2005) Eight Challenges Faced by a Defence Contractor: An Interview with a Security Chief, (ISC)2 May-June Newsletter 2005, June 2005.

Lamar, Jr. J. V. (1986) "Two Not-So-Perfect Spies; Ronald Pelton is Convicted of Espionage as Jonathan Pollard Pleads Guilty." Time, 16 June 1986.

Park, J. S. and Ho, S. M. (2004) "Composite Role-based Monitoring (CRBM) for Countering Insider Threats", Proceedings of Second Symposium on Intelligence and Security Informatics (ISI), Tucson, Arizona, June 2004.

Roos, I. (2002). Methods of Investigating Critical Incidents. Journal of Service Research, Vol. 4, No. 3, February 2002, pp. 193-204. SAGE Publications.

Rychman, R. M. (2004) Theories of Personality. (8th Ed). Belmont, CA: Wadsworth Publishing. ISBN-10: 0534619835.

Weick, K. (1995) "What Theory is Not, Theorizing Is, *Administrative Science Quarterly,* 40, 3, 385-390.

Conceptual Design of a Microfluidics Suppressor to Protect against Potentially Lethal Printing Devices: A Scenario-based Physical Cyber Security Measure

Berg Hyacinthe[1], Yves Anglade[2]
[1]Florida State University, Tallahassee, FL, USA
[2]Florida A and M University, Tallahassee, FL, USA
bph02@fsu.edu
yves.anglade@famu.edu

Abstract: This paper explores the potential misuse of print cartridges to diffuse lethal airborne substances to a massive population of computer users around the world. This issue touches on engineering design, cyber security, terrorism, and weapons of mass destruction (WMD).The design and implementation of fast and efficient micro biosensors have become a global defence and security priority due particularly to the widespread fear of potential bio-chemical attacks on civilian populations and due to bioterrorism in general. Evidently, a successful research on micro biosensors necessitates a solid background on bioinformatics, molecular and cell biology, and micro-electromechanical systems (MEMS) technology. As civilians, paramilitary forces, and terrorist groups continue to amplify the mutation of common substances and apparati into very lethal weapons, a new computers-as-weapons paradigm has emerged. Hence, in order to anticipate future cyber-related threats, the authors relied on scenario planning (a method that combines detailed analyses with imagination to produce reports as though people might write them in the future). In summary, they conclude that the technologically plausible lethal printer scenario calls for a dual approach: (1) coupling digital forensics with human intelligence gathering to uncover hidden threats before they translate into attacks and (2) designing a microfluidics suppressor as a physical cyber security measure.

Keywords: Lethal weapons, computers-as-weapons, physical cyber security, microfluidics, autonomous systems, scenario planning

1. Background

This paper explores the potential misuse of print cartridges to diffuse lethal airborne substances to a massive population of computer users around the world. On September 11, 2001, several aircrafts were intelligently transformed into missiles to destroy thousands of lives and dismantle several critical infrastructures in the United States. Meanwhile, cars have been morphed into rockets and cell phones modified into remote detonators in the Middle East. More recently (on August 10, 2006 to be exact), British intelligence officials thwarted an alleged terrorist plot, possibly just days away, to blow up U.S.-bound jetliners over the Atlantic in an attempt to kill thousands with liquid explosives concealed in water bottles and other common containers. Hence, it has become clear that civilians, paramilitary forces, and terrorist groups have been tuned to a new paradigm: morphing common substances and apparati into very lethal weapons (Hyacinthe 2006). Although the design of smart weapons originally involved computers as guides and calculators (i.e., intangible weapons), new applications both in conventional military arsenals (quadrennial defence review report 2001) and terrorist tactics reveal a willingness to use digital information technologies as tangible weapons. In order to anticipate similar threats before they translate into attacks, the authors relied on intelligence foresight via scenario planning. To reach their goal, the authors went through four methodical phases of (1) shaping the theoretical framework; (2) adopting an appropriate research methodology; (3) collecting and reviewing relevant data; and (4) interpreting the findings and leaping to the proposed design. In summary, a conceptual microfluidics suppressor is offered as a potential harm-reduction measure.

2. Theoretical framework and literature review

A preliminary review of the literature reveals a consistent use of Rogers' (1989) Technology Acceptance Model (TAM) to examine users' adoption of emergent information technologies. TAM is built on user's perception to anticipate technology adoption. For instance, an extended TAM that includes perceived trust, perceived ease-of-use, and perceived usefulness was used in the survey instrument of Heijden et al. (2003). Their conceptual framework focused on consumer intentions to use online purchasing. The literature also reveals several other instances where trust/credibility had been used to explore user's intention to adopt a digital information technology (Suh and Han 2002; Vijayasarathy 2004). Based on this value put on trust, the present authors hypothesise that computer users are likely to trust a device that comes with a fail-safe mechanism. Even when risks are partially assumed, safety and security are likely to influence a user's decision to adopt or reject a new technology. Therefore, it is principally based on this trust and security paradigm that the authors sought to offer a safer microfluidics diffusion process for next-generation printers

and similar devices. Accordingly, the authors reviewed previous microfluidics models and simulations (Zhang et al. 2002; Fei et al. 2006; Le Gac et al. 2003; Renaudin et al. 2004) in order to exploit various technologies and techniques in this area. The authors' primary data sources include a diligent patent search and a thorough content analysis from of a meticulous compilation of declassified U.S. military reports and theses. Secondary sources include personal correspondences, academic literature, and relevant conferences and symposia. A comparative content analysis of data collected from the cited sources was performed with objectivity and impartiality. Consequently, findings of that analysis guided the conceptualisation and design of the proposed microfluidics suppression protocols (Hyacinthe 2006b). As illustrated in figure 1, preliminary information is collected at the base (e.g. declassified military reports, technology adoption literature, and patent search). At the middle level, the patent search undeniably points to the emerging cyber-olfaction technologies, while the military database confirms the looming biochemical threats on U.S. soil (Figure 1). Accordingly, the authors applied the principles of scenario planning (which takes into account psycho-sociological factors) to address several cyber security issues, particularly, the potential mutation of a printer into a lethal weapon. Several models of user's adoption of new technology also support the latter mutation tendency. In response to this threat, as shown on the top of figure 1, the authors introduced the autonomous microfluidics suppression process.

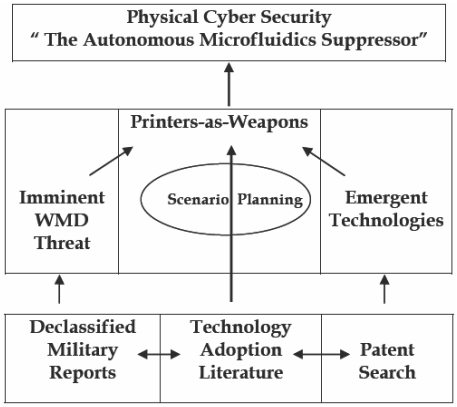


Figure 1: Diagrammatic representation of the conceptual framework

3. Primary research questions

Based on the legitimate computers-as-weapons paradigm invoked in this paper, the authors pose the following questions:

- What if printers were to be morphed to diffuse a lethal substance?
- How would it happen? What countermeasures may be built-into printers in the future?

4. Methodology

Scenario planning combines detailed analyses with imagination to produce reports as though people might write them in the future (Chermack et al. 2001).This technique is widely used to deal with "what if" type of questions. For instance, scenario planning and revealed causal mapping (RCM) allow trained investigators to anticipate social actions and uncover hidden behaviours. RCM is a representation of domain-specific theories that link specific empirical observations with theoretical categories, and then link categories into a network of causal relations (Narayanan, 1990). As the literature further suggests, through link analysis and semantic graphs, hidden information signatures may be detected (Heuer 1999; Popp et al. 2004; Chow 2004). According to many intelligence reports regarding a biochemical attack on U.S. soil, it is a matter of time. Nonetheless, the question yet to be answered is HOW? Although the answer to the latter is too complex to debate in a single instance, national security experts at the Research AND Development (RAND) coalition were among the first to embrace scenario planning as a vital tool to deal with similar complexities (e.g., anticipate future threats and propose solutions). RAND is the original U.S. public policy think tank. Today, RAND emphasises several areas of research that reflect the changing nature of a global society (RAND 2006). As the literature review suggests, scenario planning is a strategic planning technique used to make flexible long-term plans (Armstrong 1985; Schwartz 1991; Ringland 1998). In the context of a dynamic information-driven society, "scenarios" are gaining credibility as effective tools to prepare for an uncertain future, test decisions, alter mental models, and improve performance (Chermack et al. 2001). Accordingly, Chermack and Lynham (2002) presented a complete review of definitions, paradigms, and a capacious historical background of scenario planning. A snapshot is offered below:

> *"Scenario planning first emerged for application to businesses in a company set up for researching new forms of weapons technology in the RAND Corporation. In 1967, Herman Kahn of RAND Corporation pioneered a technique called 'future-now' thinking. The intent of this approach was to combine detailed analyses with imagination and produce reports as though people might write them in the future. Kahn adopted the name 'scenario' when Hollywood determined the term outdated and to the label 'screenplay.' In the mid-1960s, Kahn founded the Hudson Institute, which specialised in writing stories about the future to help people consider the 'unthinkable.' He gained most notoriety around the idea that the best way to prevent nuclear war was to examine the possible consequences" (p.367).*

In many aspects, scenario planning is closely related to foresight techniques (such as game theory, pattern matching, Delphi, and econometrics) frequently used by military planners. Nonetheless, scenario planning is not restricted to military intelligence. For instance, Ahlqvist (2001) used Delphi techniques to identify key information technologies associated with various dynamic sectors (such as academic and corporate settings). Notably, the use of evolving information technologies and theories as visible/invisible weapons to conquer the mind is often associated with Libicki's (1995) principles of cyber and psychological warfare.

5. Data collection and findings

The authors' conceptual design is substantiated by a comparative content analysis (Weber 1985) of a series of technical reports from U.S. military experts addressing biochemical warfare and terrorism, graduate and post-graduate military theses, research articles covering national security and WMD issues (Technical Reports 2004), and emergency rescue operations (Hyacinthe 2004). According to the findings illustrated in Figure 2 below, U.S. military scholars, from the Army War College (e.g., ADA32669) to the Naval War College (e.g., ADA336609) to the Air University (e.g., ADA407098) to the Naval Post Graduate School (e.g., ADA365438) to the U.S. Marine Corps (e.g., ADA407502), have shown great concerns over the imminence of a bio-chemical attack on the U.S soil. Unfortunately, cyber-olfaction technologies are being developed without capabilities that would reduce the dreadful effects of a potential biochemical attack involving today's ubiquitous digital gadgets. It is worth noting that (apparently) neither the reports nor the inventions warned against this potential cyber-WMD.

Bio-Chemical Threats /Declassified Military Reports and Theses	Emerging Cyber-Olfaction Technologies/Patent Search
ADA32669: *Terrorist Use of Weapons of Mass Destruction Within the United States: Asymmetric Warfare Paradigm in the 21st Century*. Carlisle Barracks, PA: Army War College, Strategic Studies Institute, March 1997.	**US5949522A:** *A device is described which can deliver various combinations of scents in rapid succession in a user' nose* in conjunction with videographic images and/or sounds.
ADA407502: *Bugs & Drugs: Chemi-Bio Terrorism & the U.S. Government*. Quantico, VA: Marine Corps Combat Development Command, May 2002.	**US6524537:** This patent disclosed a fragrance emitter used with internet. According to the inventor, *the fragrance emitter is able to emit various kinds of fragrance stored in vessels according to the command sent by a frequency detector* which is used to search for the existence of a specific frequency specially used by a specific website.
ADA236609: *Responding to Chemical Attack*. Newport, RI: Naval War College, Department of Operations, February 1991.	...
...	**US6004516:** The inventors proposed an *apparatus for generating odor upon electronic signal demand using a disk having an aroma-impregnated adsorbent and a substrate*.
ADA346005: *New Enemy: Silent, Lethal, and Invisible*. Carlisle Barracks, PA: Army War College, April 1998.	**US2004024043*:** The inventors described an *information processing apparatus with an interactive scent interface to enrich users' perceptive experiences*. It is important to note that Wang and his associates even proposed cell phone embodiment of their aroma-embedded information system.
ADA407098: *Role of the National Guard in Responding to Weapons of Mass Destruction (WMD) Attacks in the U.S.: Where Do We Stand*. Maxwell Air Force Base, AL: Air University, Air Command and Staff College, April 2001.	** This is a patent publication number.*
ADA365438: *High Resolution Modeling of a Terrorist Chemical Attack in an Urban Area*. Monterey, CA: Naval Postgraduate School, June 1999.	

Figure 2: Excerpts of findings (military reports (left) and patent search (right))

As illustrated in figure 2, the reports warn of "an intent to attack with" biochemical weapons and document on-going "attempts to acquire" such weapons. The patent search also reveals the disturbing introduction of this threat into the bedroom via aroma-embedded cell phones and other portable electronic devices (Wang 2004; Paradies and Hyacinthe 2005; Hyacinthe 2006a). Wary of the fact that certain threats are less evident than others, the authors introduce the lethal printer scenario as a primer to study and learn about a darker

side of digital information technologies. On the other side of the spectrum, the messianic capabilities of bio-MEMS and micro total analysis systems (μTAS) are displayed throughout the proposed microfluidics suppression model. The recent waves of terrorist attacks reiterate the importance of the authors' prescient mutation paradigm (Hyacinthe 2004) and the findings compel them to draw the scenario that follows.

6. Interpretation of findings: The scenario

A terrorist group or enemy state, tuned to this obvious, popular thrill of morphing common devices into lethal weapons, decides to target printers as weapons. Thus, the enemy has successfully managed to transform printers into diffusers of lethal bio-chemical agents by loading print cartridges with a deadly substance. Deadly Innovation Group (DIG) started as a legitimate ink/toner cartridge recharge business two years earlier and strategically focused on small and micro cartridges with the ultimate goal to attack the most critical infrastructures. On October, 10, 2009, aroma-embedded cell phones become popular in Japan and thereafter in the U.S. By August, 12, 2010, DIG becomes the primary distributor of micro fragrance cartridges used in cell phones around the globe (to exchange aromatic information). Through the online supply tracking system proposed and supported by DIG, it was possible for its managers to time delivery and to predict (almost to the hour) when a particular stock will be in-use at any location in the world. What was originally planned as a US assault is about to become a global attack. Suddenly, even handheld printers used in aviation and other mass transport infrastructures (e.g., during mid-flight sales, in restaurants, at car rental locations, and gift shops to print credit card receipts) become at risk. Hence, DIG manages to activate the diffusion of a lethal substance at a large number of computer terminals on September 11, 2011 (symbolically dubbed as "9/11-II").

7. Conceptual design: The proposed solution

Microfluidics and μTAS, which move microscopic and nano-scale drops of fluid, are crossing new frontiers in the realms of real-time enzymatic analysis, DNA analysis, proteomic analysis involving proteins and peptides, and immunoasssays. The distinct advantages that they bestow on applications are pushing lab-on-chip technologies from RandD to commercialisation at a very rapid pace. For instance, biosensors are becoming more sensitive and accurate. They are relatively inexpensive and portable (Frye-Mason et al. 2000; Fair et al. 2004). Microfluidics and μTAS are also making their mark in the defence and public health sectors. The microfluidics suppressor concept emerges as a safeguard against unwanted mixing and/or diffusion of fluid streams in standard and μTAS.

7.1 Description of the proposed model

The proposed model consists of a miniature biosensor that serves as a detection device, a regulator (bio-MEMS) that allows or suppresses diffusion, and a digital controller that connects the biosensor to the valve (nano/MEMS). It functions as an autonomous microfluidics-based filtering system that allows selective suppression and release in μTAS, standard chemical mixing chambers, exhaust systems, and related systems. Presently, the authors' focus is on printers (Figure 3).

7.2 Micro biosensors

Techniques and devices for detecting a wide variety of analytes in fluids such as vapours, gases, and liquids have been around for a long time (Mayzner and Dolan 1978).These devices generally comprise an array of sensors that, in the presence of an analyte, produce a unique output signature (Haseley 2002). The biocatalyst converts the substrate to product. This reaction is determined by the transducer which converts it to an electrical signal. The output from the transducer is amplified, processed and displayed. The transducer is necessary to determine physical changes that accompany the reaction (Collings and Caruso 1997). The methods and technologies vary from calorimetric biosensors (characterised by heat absorption) to potentiometric biosensors (related to the production of an electrical potential through distribution of current charges) to amperometric biosensors (characterised by the movement of electrons in a redox reaction) to optical biosensors (e.g., light absorbance difference between reactants and products) to piezo-electric biosensors (e.g., measuring the effects due to mass of reactants or products). Furthermore, the literature reveals an array of technologies used in the design of various biosensors (Dariva et al. 1999; Preston and Mc Fadden 2001; Srinivasan et al. 2003). Theoretically, as biosensor technologies evolve, the task of retrofitting existing printer designs (with such technologies) will become less sophisticated and more evident.

7.3 Print cartridges

Technology forecasters warned of a paperless society. Instead, a new print culture was born. Apparently, certain printing functions have been shifted from corporate to client, educational institution to students, and from mass production to individual print jobs. As digital cameras continue to gain popularity, portable photo labs allow the user to print high resolution images from very sophisticated print cartridges controlled by very able microprocessors. Generally, a printer consists of a bay (allowing various media to feed through), a toner roller, a photo receptor drum, and a fuser (figure 3).

7.4 Micro print cartridges

Micro print cartridges are omnipresent in commercial, governmental, and private sectors. They can be found in point of sale (POS) terminals around the world. Credit card receipts are often printed from mini terminals equipped with micro print cartridges. For instance, based on its 'smart valve' ink cartridge technology, Epson developed a new cartridge that utilises a system known as 'smart valve control technology' to prevent the delivery of excess ink to the print head. This cartridge is based on micro piezo sensor technology (Epson 2006). As the Epson's technology suggests, the authors' microfluidics suppressor device is near.

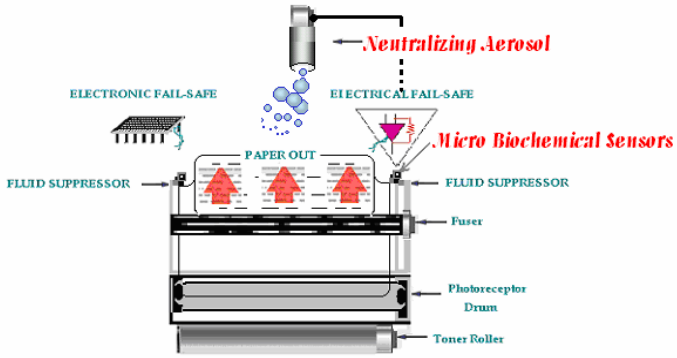


Figure 3: Microfluidics suppression/neutralisation

According to Figure 3, a biochemical warfare agent is detected and an antibody has been diffused to mitigate its harmful effects. The security measure is proposed for the protection of printing devices and similar applications such as aroma-embedded computers or cell phones. figure 3 is an excerpt of provisional patent application number US-60/814320 (Hyacinthe 2006b).

7.5 Fragrance cartridges and cyber-assisted olfaction

Most fragrance cartridges use technologies similar to print cartridges for the diffusion of aromatic plumes. In a broader sense, the expression cyber-assisted olfaction refers to the replication of the human sense of smell by computer systems. As well, it involves the stimulation of an individual's olfactory sensors by computer-emitted bio-chemical or aromatic particles (Hyacinthe 2006a). Although cyber-olfaction technologies are evolving slowly, a recent patent search reveals a renewed interest to take advantage of advances in microfluidics and nano technologies to generate a variety of processes and protocols that would simplify and popularise the design and development of cyber-assisted olfaction technologies and aroma-embedded information systems (Manne 1999; Rasouli et al. 1999; Lee 2003; Wang et al. 2004). In summary, the authors presented only an excerpt of what has been an extended review of a list of recent patent applications filed from Japan to America to Europe (figure 2). the evidence suggests that the Nintendo generation may soon embrace aromatic instant messaging. In every respect, the design and implementation of next-generation aroma-embedded digital information technologies should concern cyber security authorities at all levels.

In Figure 4, the process of microfluidics suppression is illustrated with two fluid streams A and B. The resulting mixture AB is tested inside an electronically monitored chamber. The security diagnostic test is initiated by a biosensor (not shown here). If the mixture AB fails the test, then a signal is sent to suppress fluid circulation; else, no action is taken. The same principles apply to a plurality of fluid streams (A to Z). The authors also envisioned an optional decontaminant diffuser.

7.6 Multi-purposefulness of the proposed model

This proposed apparatus adds a new dimension to digital and cyber security with its novel fail-safe modules. In one embodiment, the proposed model may serve as a fail-safe mechanism to medical diagnostic equipment and drug delivery systems (e.g., insuring that safe fluid streams are delivered to intended targets). In other words, chemicals from various sources (tubes, lines, pipes, and micro channels) can go through a safety control mixing process. In another embodiment, the microfluidics suppressor may be used in cyber-olfaction systems to suppress unwanted fragrances from reaching the end user. For example, scented emails or fragranced e-cards can be scanned for malicious or unintended harmful substances.

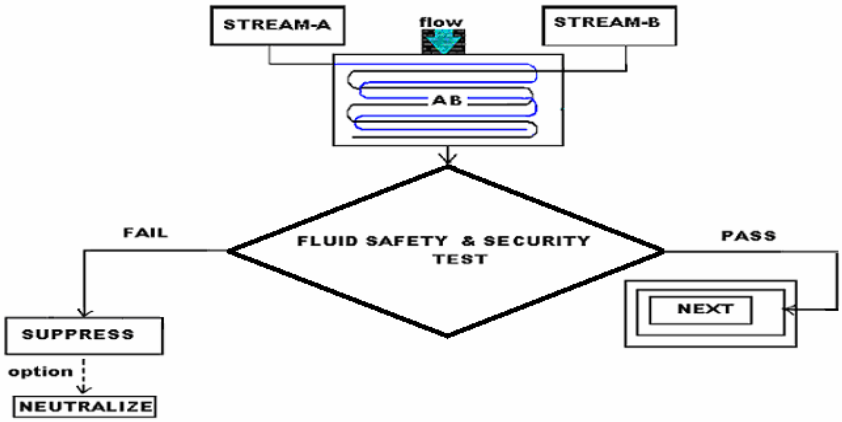


Figure 4: Proposed microfluidics suppression technique

7.7 Fail-safe modules

In the proposed model, the printer is equipped with a micro biosensor which interfaces with an intelligent chip that passes essential information on the status of the ink cartridge to pre-installed programs in the printer. The chip is responsible for passing on data such as whether the cartridge is tampered with or whether it diffuses the desirable fluids. In simple terms, the fail-safe modules control the release/suppress switch of the print cartridges. The authors propose the followings:

- The mechanical (electrical) Fail-safe as a module embedded with devices (such as printers and fax-machines) to trigger an automatic and complete power shutdown, upon detection of an unwanted substance.
- The electronic fail-safe that, contrary to the first option, triggers modular or selective interruption through the electronic manipulation of the microprocessor that controls individual components of the master device. For example, in an All-in-One device (e.g., combination of fax, scanner, copier, and printer), complete power surge might produce unnecessary suspension of other critical tasks.

In sum, the aforesaid fail-safe protocols are particularly important, for example, when dealing with refilled/recharged cartridges from foreign and/or suspicious suppliers. Notably, the mechanical option is less expensive and less sophisticated to implement than the electronic option.

7.8 Decontaminant diffuser

The diffuser will receive signal from a biosensor with biochemical information related to the “type” of threat (e.g., chemical profile of the toxin or bacterium detected). In turn, a pattern matching algorithm will determine which on-board decontaminant is to be activated against the detected biochemical threat (Hyacinthe 2006candd). Once the right “antidote” or decontaminant is identified, a pump will be activated to release the neutralising aerosol in an attempt to reduce the harmful effects of any unwanted or undesirable release. In fact, the storage of a plurality of chemical precursors is likely to increase the efficacy of the neutralisation system.

8. Detecting hidden information signatures

In this section, the authors will address the following question: What can be done to detect hidden information signatures of a group such as DIG before its members move from threat to attack? In this context, the primary task of the intelligence analyst is to anticipate the threat (before DIG), in order to capture its members in an information trap. This is consistent with the information superiority doctrine (National Defence Panel 1997). Ubiquitous digital transactions will create the need for better data mining techniques

and to decipher sophisticated patterns of information signatures for intelligence analysis. For example, with better tools, purchase patterns and other social network activities may be traced through online transactions. In addition, geographical activities can be tracked via GPS technologies as itinerary can be traced through digital information signatures. Nonetheless, digital forensics and strong anti-terror law will be needed to address anticipated challenges. Furthermore, intelligence analysts may use parallel intelligence processing (relying less on historical data and more on anticipatory information) with the indispensable human intelligence component. In effect, regardless of the approach adopted, the utilisation of intelligence foresight to uncover hidden information signatures remains quintessential to any credible defence and security enterprise. As the literature suggests, intelligence analysts may turn to sub-graph isomorphism algorithms, link analyses, and semantic parsing to overcome personal and collective biases (National Defence Panel 1997; Heuer 1999; Morgan 2003; Coffman et al. 2004; Popp et al. 2004).

9. Conclusion

In conclusion, the potential misuse of cybernetics to amplify a biochemical attack in the future is real. A mistake can also trigger lethal diffusions (Hyacinthe 2006c). The proposed autonomous filtering system may, not only measure fluid input and output, but also serve as fail-safe countermeasure against unwanted mixtures mistakenly or maliciously produced. Fortunately, prevention can be accomplished through careful foresight analyses and with the tools proposed herein. Cautiously planned attempts to uncover and disrupt these sophisticated threats are strongly recommended. In essence, a pragmatic way to counter these threats is to develop innovative concepts and to design robust, revolutionary security apparati for the protection of the most critical infrastructures. The foregoing discourse exemplifies the authors' commitment to address the "failure of imagination" cited by the "9/11" commission (9/11 Commission 2004). As argued throughout this paper:

- The "computers-as-weapons" paradigm has taken a new, unforeseen dimension that threatens national security.
- The eventual misuse of printers as weapons can be dreadful.
- Already tuned to this mutation paradigm, paramilitary and terrorist groups, private militia, and enemy states are likely to turn to cyber-assisted olfactory information warfare (e.g., the use of digital information technologies to diffuse lethal airborne substances to a large population) in the near-future.
- In response to this threat, a micro biosensor can be embedded in next-generation printers as a threat detection/harm-reduction measure. Optionally, a decontaminant may be diffused to neutralise particles that might escape between detection and suppression phases. It is also important to note that the microfluidics suppressor can operate as an independent multi-purpose apparatus.
- From a psychological warfare perspective, the proposed microfluidics suppressor can be a powerful deterrent to potential enemies. More importantly, it can be effective and useful even after its public disclosure.

Acknowledgments

The authors hereby acknowledge the academic support of John Gathegi, PhD, JD of Florida State University, the sponsorship of Infosense Technologies and Research, Inc., the support of Electrical Engineer Sophia Hunt, and the influence of Biochemistry teacher Alphonse St. Louis.

References

9/11 Commission (2004) "9/11 Commission Report: Final Report of the National Commission on Terrorism Attacks Upon the United States" [online], http://www.gpoaccess.gov/911/ (12th November, 2006).

Ahlqvist, T. (2005) "From information society to biosociety? On societal waves, developing key technologies, and new professions", Technological Forecasting and Social Change, vol.72, pp. 501-519.

Armstrong, S. (1985) Long-Range Forecasting: From Crystal Ball to Computer, John Wiley: New York.

Chermack, T. and Lynham, S. (2002) "Definitions and outcome variables of scenario planning", Human Resource Development Review, vol. 1, no. 3, pp. 366-383.

Chermack, T., Lynham, S., and Ruona, W. (2001) "A review of scenario planning literature", Futures Res., vol. 17, no. 2, pp. 7-31.

Chow, E. (2004) A graph search heuristic for shortest distance paths, Technical Report UCRL-JRNL-202894, Lawrence Livermore National Laboratory.

Coffman, T., Greenblatt, S., Marcus, S. (2004) "Graph-based Technologies for Intelligence Analysis", CACM, vol. 47, no. 3, pp. 45-47.

Collings, A. and Caruso, F. (1997) "Biosensors: recent advances", Rep. Prog. Phys., vol. 60, pp. 1397-1445.

Dariva, C., Coelho, L., and Oliveira, J. (1999) “A Kinetic Approach for Predicting Diffusivities in Dense Fluid Mixtures, Fluid Phase”, Equilibria, vol.158, pp. 1045-1054.
Epson. (2006) “Micro Piezo technology” [online], http://www.epson.co.jp/e/technology/inkcart.htm (11th November, 2006).
Fair, R., Khlystov, A., Srinivasan, V., Pamula, V., and Weaver, K. (2004). "Integrated chemical/biochemical sample collection, pre-concentration, and analysis on a digital microfluidic lab-on-a-chip platform", Lab-on-a-Chip: Platforms, Devices, and Applications, Conf. 5591, SPIE Optics East, Philadelphia.
Fei, S., Chakrabarty, K., Fair, R. (2006) “Microfluidics-Based Biochips: Technology Issues, Implementation Platforms, and Design-Automation Challenges”, IEEE transactions on computer-aided design of integrated circuits and systems, vol. 25, no.2, pp. 211-223.
Frye-Mason, G., Kottenstette, R., Lewis, P., Heller, E., Manginell, R., Adkins, D., Dulleck, G., Martinez, D., Sasaki, D., Mowry, C., Matzke, C., and Anderson, L. (2000) “Hand-Held Miniature Chemical Analysis System (μChemLab) for Detection of Trace Concentrations of Gas Phase Analytes”, μTAS, vol. 229, p. 3.
Haseley, S. (2002) “Carbohydrate recognition: a nascent technology for the detection of bioanalytes”, Anal. Chem. Acta , vol. 457, pp. 39-45.
Heijden H., Verhagen T., and Creemers M. (2003) “Understanding online purchase intentions: contributions from technology and trust perspectives”, European Journal of Information Systems, vol. 12, pp. 41-48.
Heuer, R. (1999) Psychology of Intelligence Analysis. Washington, DC: Center for the Study of Intelligence, Central Intelligence Agency.
Hyacinthe, B. (2006) “Methods and Apparatus for the Production of Aromatic and Gustatory Information”, JDCL, Chapel Hill, NC. ACM 1-59593-354-9.
Hyacinthe, B. (2004) Emergency Rescue Vehicle. United States Patent and Trade Office, VA. No. US60/634,637.
Hyacinthe, B. (2006) “Hidden Global Security Threats and Emerging Technologies Exposed through Information Warfare Paradigms”, Proc. of the 5th European Conference on Information Warfare and Security, Helsinki, Finland., pp. 101-110.
Hyacinthe, B. (2006a) System and Device for Prevention and Neutralisation of Bioactive Substances and Generating an Aroma-producing Substance. United States Patent and Trade Office, VA. No. US-60/700,700,708.
Hyacinthe, B. (2006b) Apparatus and Methods to Suppress Fluidic Diffusion of Unwanted Substances in Standard Systems, Nano-devices, and Bio-microelectromechanical Systems. United States Patent and Trade Office, VA. No. US-60/814320.
Hyacinthe, B. (2006c) “Autonomous Biochemical Decontaminator (ABCD) against Weapons of Mass Destruction”, SPIE, vol. 6021, pp. 1-16. Orlando, USA.
Le Gac, S., Rolando., C., Arscott, S., Carlier, J., Caron, F., Fourrier, J., Thomy, V., Druon, C., Tabourier, P., and Camart, J. (2003) “Microfluidics for proteomics: discrete liquids”, Japan-European Workshop on Chem Micro Mechatronics Systems and Biochips, ENS Cachan.
Lee, C. (2003) Fragrance emitter for use with Internet. Patent No.US6524537. Washington, DC: U.S. Patent and Trademark Office.
Libicki, M. (1995) What is information warfare? Washington, DC: National Defense University, Institute for National Strategic Studies.
Manne, J. (1999) Multimedia linked scent delivery system. Patent No. US5949522. Washington, DC: U.S. Patent and Trademark Office.
Mayzner, M and Dolan, T. (1978) Minicomputers in Sensory and Information-Processing Research. Lawrence Erlbaum Associates. Hillsdale, New Jersey.
Morgan, P. (2003) “Information Warfare and Domestic Threats to America Security”, Contemporary Security Policy vol. 24, pp. 161-189.
Narayanan, V. and Fayhey, L. (1990) “Evolution of Revealed Causal Maps During Decline: ‘A Case Study of Admiral’”, in Huff (ed). Mapping Strategic Thought, London: John Wiley and Sons, pp. 109 -133.
National Defense Panel. (1997)Transforming Defense: National Security in the 21st Century, Arlington, Va.: National Defense Panel Publications.
Paradies, C. and Hyacinthe, B. (2005) “Patent search on cyber-assisted olfaction technologies”, Law Offices of White, Boggs and Fowler, Tampa, Florida (unpublished).
Popp, R., Armour, T., Senator, T., and Numrych, K. (2004) “Countering terrorism through information technology”, Communications of the ACM., vol. 47, no. 3, pp. 36-43.
Preston, R., and McFadden, P. (2001) “A two-cell biosensor that couples neuronal cells to optically monitored fish chromatophores”, Biosens. Bioelectron., vol. 16, pp. 447-455.
Quadrennial Defense Review Report. (2001) “Creating the U.S. Military of the 21st Century”, Washington, DC.
RAND (2006) “History and Mission” [online], RAND Research, http://www.rand.org/. (16th November, 2006).

Rasouli, F., Arastoopour, H., Oskouie, A. (1999) Apparatus for generating odor upon electronic signal demand. Patent No. US6004516. Washington, DC: U.S. Patent and Trademark Office.
Renaudin, A., Zhang, V., Tabourier, P., Camart, J., and Druon, C. (2004) “Droplet manipulation using SAW actuation for integrated microfluidics", µTAS, Malmö, Sweden.
Ringland, G. (1998). Scenario planning: Managing for the future. New York: John Wiley.
Rogers, E. (1995) Diffusion of Innovations, The Free Press, NY.
Schwartz, P. (1991) The Art of the Long View. Doubleday, New York.
Srinivasan, V., Pamula, V., Pollack., Fair, R. (2003) "A digital microfluidic biosensor for multianalyte detection", Proceedings of the IEEE 16th Annual International Conference on Micro Electro Mechanical Systems, pp. 327-330.
Suh, B. and Han, I. (2002) “Effect of Trust on Customer Acceptance of Internet Banking, Electronic Commerce Research and Applications”, vol. 1, pp. 247-263.
Technical Reports (2004) “Chemical, Biological, and Nuclear Terrorism/Warfare: A Bibliography” [online], U.S. Naval Post Graduate School, http://library.nps.navy.mil/home/bibs/chemtech.htm (12th May, 2004).
Vijayasarathy, R. (2004) ”Predicting consumer intentions to use on-line shopping: the case for an augmented technology acceptance model” Information and Management, vol. 41, pp. 747-762.
Wang, H., Tsai, Y., and Wu, F. (2004) Information apparatus with interactive scent interface. Patent No. US2004024043 (pending). Washington, DC: U.S. Patent and Trademark Office.

Virtual' Security Teaching Techniques – Teaching IA methods using VMWare Teams

Derek Isaacs
Colorado Technical University and Boecore Inc, Colorado Springs CO, USA
Disaacs0514@comcast.net
Derek.Isaacs@CTUOnline.edu

Abstract: Teaching Security techniques and methods today requires that the instructor 'know' the student to a depth of trust and integrity that is often impractical in a University setting. Merely imparting instruction and hands on "how-to" information can sometimes put the instructor and potentially the educational institution at risk – as these tools and techniques for testing security are only a step removed from the methods and means of effecting security breaches or causing incidents. The need for training must therefore be balanced with the responsibility to provide an isolated and restrained environment in which to present material and hone new skills (or to use the vernacular -'skilz'). As a means of solving this, Virtual Machines (VM) and virtual network environments or team architectures offer a solution to these issues. They also provide, from an educational context, a set of 'known' problems and findings with which the student can study and develop the tools, techniques and knowledge to be an effective Information Assurance professional. This paper presents a series of architectures and scenarios proposed to effect such a learning environment that retains the viability of a 'real' network environment while providing an isolated and restricted learning area. This can be achieved through a series of scenarios and VM Team members (scenario players) in a virtual machine based learning environment.

This approach allows a number of benefits:

- Isolation of the learning environment
- Restrict potential issues (DoS / attack signature detection) to a secured space
- Provide limits on tool and technique usage and impact
- Limit and mitigate risk and liability issues for the educational institution

The VM environment also offers a unique opportunity to provide completely known interactions between the student and the learning environment – the software and the underlying virtual hardware. A detailed set of proposed scenarios and a learning environment architecture, including toolsets and targets of evaluation (TOE) systems is proposed. The applicability of the VM 'team' systems approach is discussed through a suite of scenarios designed to illustrate attack causation, monitoring, and detection.

Keywords: Information assurance, security education, virtual systems

1. Introduction

In recent years there have been a variety of training and educational settings involving basic networking, hardware, Operating Systems (OS) and Systems Security. The institutions provided a plethora of equipment – often this equipment was difficult to move across classrooms and due to the expense and effort involved in procurement, setup, and maintenance, they were oft outdated very quickly. Implementing training environments with a software solution like VMware – everything might have fit comfortably in a CD case – with a minimum of additional hardware required for execution.

Education benefits from emulation of real world situations. As has been often quoted:

I hear...and I forget
I see...and I remember
I do...and I understand
Ancient Chinese Proverb

In Information Assurance (IA) training, the emphasis for practical and effective learning points to an increased need for hands-on methods. In a traditional "Brick and Mortar" institution this is often problematic as there is always a shortage of equipment, network capacity, and insulating infrastructure to hinder student misadventure and mischief. Most courses designed for practitioners require basic networking equipment (routers, switches, hubs, etc), Processing hardware (Servers, Workstations, etc), and a variety of Operating Systems (OS), many of which require licensing for each instantiation. Course preparation often involves setting up physical rooms dedicated to a particular set of objectives. In recent developments with virtualized systems the physical hardware requirements have been greatly reduced. This is especially important in regards to the current trend to distance education (sometimes referred to as e-learning). Virtualization tools

such as VMware allow the student to have access to multiple virtual or guest OS's and complete (and again virtual) networks on a single physical system. Disparate OS's such as Linux, Mac's OS-X, and Windows 2003 for example, can be loaded upon and can interoperate on the same system. VMware provides features (such as virtual guest teams) that give the ability to switch from one virtual system to another with ease.

Security methodology is best taught by providing hands-on examples of exploits, vulnerabilities, and security breaches, and then showing a correct way to mitigate the situation or issue encountered. This often takes the form or sharing 'real-world' episodes. Misadventure stories hold the interest of students and provide insight as to the relevance and applicability of the ideas and concepts being presented.

- As an example – Students were assigned to review the SANS Top 20 Vulnerability list, select 3 vulnerability items, and research/write about them
    - One student misinterpreted this assignment - they pointed a Vulnerability scanning tool (Nessus) and then a port scanner (NMap) at the SANS.org site from a public library computer – the student got immediate 'recognition' by the Library IT staff, the Internet Service Provider, and SANS; and the instructor received an interesting set of phone calls from the parties concerned!
- When one provides instruction, especially in Computer Security, one is laying a foundation in the minds of students – what is built upon that foundation, in other words the end use of that knowledge, is OUTSIDE the instructor's (and institution's) control – IA knowledge (like any other form) may be used for good or ill

We can all gain by sharing these, but for that we need a common infrastructure where all these exercises would be valid

2. Background

A virtual machine (VM) is an application software component that runs on the host system OS and emulates the hardware and OS of another machine. The teaching systems exploit a *virtualized* abstraction that emulates the desired hardware. This abstraction emulates the target of evaluation (TOE) hardware closely enough that software running on the underlying physical hardware also runs on the virtual machine. Through virtualization of all the hardware resources, this allows multiple virtual machines to multi-task the resources of the physical machine (Goldberg 1974) Virtualization allows multiple virtual machines, with heterogeneous operating systems to run (relatively speaking) in isolation, side-by-side on the same physical machine.

The individual virtual machine 'sees' its own set of virtual hardware (RAM, CPU, NIC, HDD, etc.) where the guest operating system and applications are loaded and execute. Therefore the OS sees a consistent, operational set of hardware regardless of the physical hardware components.

The operating system running inside a VM is traditionally referred to as the guest OS, and applications running on the guest OS are similarly referred to as guest applications. Various styles of VM's have been a standard part of mainframe computers for 30 years, and recently have been deployed onto commonly available '*commodity*' *x*86 PCs. Hosted VM's such as VMware (VMware 2001/ Sugerman et al 2001), Microsoft's Virtual PC, and recently VIRTUOZZO have emerged that run VM's concurrently with an ordinary host OS such as Windows or Linux. In this setting, the virtual machine performs as just another application running on the host OS.

VMs are also starting making inroads in a number of security related areas such as server replication or consolidation, Honeypot/Honetnet systems, and disaster recovery / Continuity of operations (DR/COOP) settings. VM's are supported over a wide variety of architectures. Recently, as hosted VMs and third-party systems (often referred to as appliances) have appeared on the desktop, they have begun to find other implementations like cross-platform development, application deployment and testing, and patch testing for IAVM management (VMware 2006)

2.1 Benefits of this approach

The deployment of VM's for security training purposes is an attempt to take advantage of recent advances in virtual machine (VM) technology. These advances facilitate solutions for several of the current issues involved in IA educational concerns – namely shortage of physical systems for instructional use, reduction of liability to the host institution, and providing repeatable and re-useable simulations for demonstrating

educational principles in a near real time (and close to real world) environment. There have been numerous proposed applications of this technology – some specifically in the security arena (i.e. Yasinsac, Frazier, and Bogdonav 2002; Hu, Cordel, Meinel – 2004) and even the founder of VMware has predicted the usefulness of this technology:

"In the coming years, virtual machines will move beyond their simple provisioning capabilities and beyond the machine room to provide a fundamental building block for mobility, security and usability on the desktop."
-*Mendel Rosenblum, Chief Scientist, VMware Inc*

3. A virtualized lab environment

The focus of this activity is the creation of a virtual-machine platform that will provide an abstraction of x86 PC hardware so that multiple operating systems can interoperate and run concurrently on a configured PC. For course and lesson plan developers, this means you can run multiple development environments on your desktop without rebooting or repartitioning. In the process, you can isolate and protect operating environments (and the applications and data that are running in them), as well as interoperate among operating systems for networking, device/file sharing, and cut-and-paste. For users, VMware makes it possible to run Windows applications with Linux. VMware Workstation comes in two main forms ('flavors'), depending on the operating system running on the user's Intel based (or compatible) PC -- VMware for Linux, and VMware for Windows 2000/XP/2003. VMware installs like an application, requiring no special hardware support.

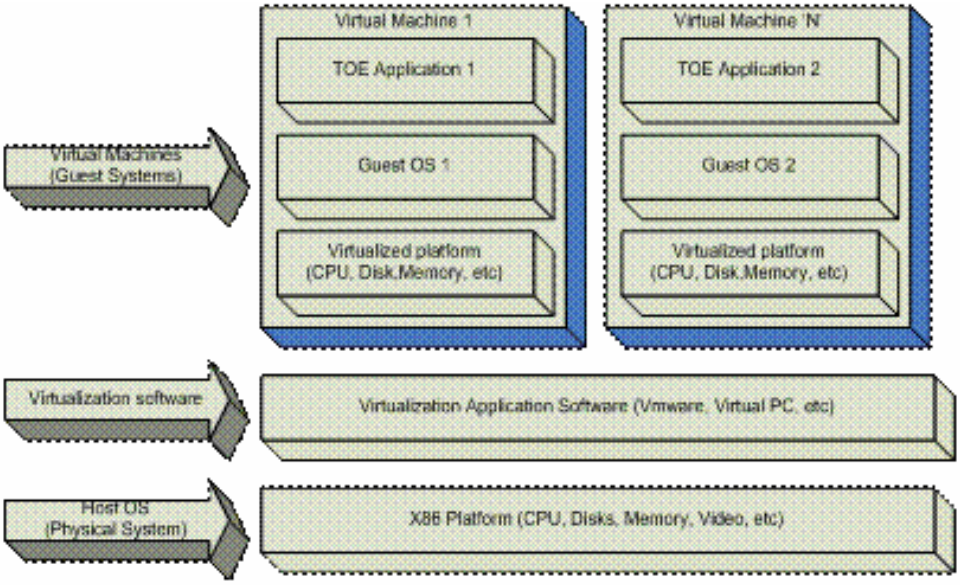


Figure 1: The Physical and virtual platform.

One of the primary reasons to use VM's is the wide range of supported OS and application tools afforded to the student. Rather than having students (or institutions) maintain massive collections of physical systems and being required to install multiple versions of operating system software, application software, and configuring existing machines. Standard configurations of virtual machine appliances can be created and distributed to students – independent of the student's location or network connection situations. An exercise that formerly required several hours of setup and installation (thereby making it difficult to implement in a large or distributed course) can be configured once, saved to an easily replicated form (say CD or DVD) and then distributed. Specific exercises could be tailored for specific virtual machines, allowing for extremely focused and repeatable demonstrations (such as buffer overflow or taking over a server via a DCOM exploit for example), instead of being concerned about the configuration of one exercise conflicting with or otherwise interfering with another exercise. A huge benefit is that even if the entire virtual machine is formatted or destroyed, another environment can be deployed as simply as copying the VMware file to the system. Additionally, if system resources are capable enough, several virtual machines can run simultaneously, connected through a virtual network connection. While not all attacks can be modelled using this environment (especially timing based attacks), the ability to run a demonstration in a normal lab, and indeed even a student's home computer without risk to the computer itself or even the institutional infrastructure makes this an excellent option for security lab development. In the past, this type of switching between

systems and configurations was accomplished using removable hard drives and multiple reboots. While removable hard disks may still be an option for installation, wear and tear on the equipment and the time to reinstall a new operating system and environment are eliminated if VM's are used for these purposes.

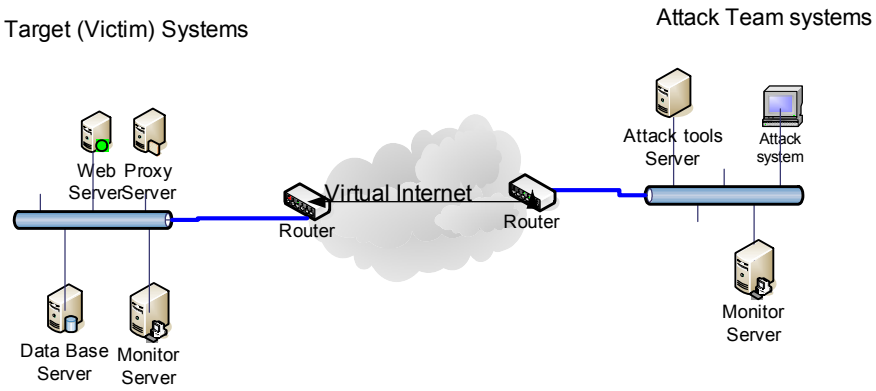


Figure 2: The virtual instruction platform – generalized design.

3.1 Example scenario configurations and objectives

3.1.1 Firewall demonstration scenario.

There are a number of open source and 3rd party appliances available to illustrate firewall operation – but the most straightforward is the available on the Virtual Machine Technical Network (VMTN) website. This is an excellent platform from which to illustrate the operation and activity of a firewall. The VMTN pre-built appliance is downloadable at around 40MB. This appliance, along with a standard workstation image, allows configuration and testing of a protected network via Network Address Translation (NAT), and demonstrates the basics of packet filtering, traffic shaping, port forwarding, and general traffic restriction. By deploying a packet sniffer application (such as Ethereal/Wireshark) both on the Host OS system as well as within the guest OS appliance, one can visually see the effects of the firewall filtering when going to known 'dirty' sites.

The figure below shows the layout of this specific scenario.

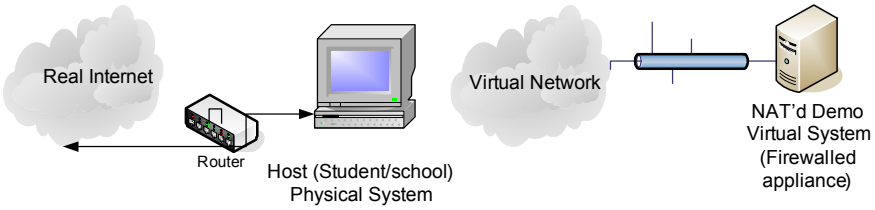


Figure 3: NAT Firewall appliance layout

3.1.2 Server takeover (owning the box) scenario.

This illustration is designed to provide a means of demonstrating to the student how vulnerabilities in the Windows OS can be exploited to provide control of a server – and also provide a 'stepping stone' approach to show how attack systems (such as botnets and zombie nets) can be built up to provide a 'strength in numbers' attack scenario. The layout illustrated provides an excellent platform to show how this activity can be successfully implemented. There are several members of the VMware 'team' in this scenario. Again the VMTN pre-built appliance from above is used – in this case as an attack monitor/eyewitness platform. Two additional guests are employed – an XP workstation (from which the attack is initiated) and a Windows 2003 (member) server configured in an 'out of the box' method (with a minimum number of mitigating choices made during installation). Again deploying a packet sniffer application on a guest OS (monitoring) appliance, one can visually see the watch the process unfold from a network traffic standpoint. This also provides illustration of the vulnerabilities involved in client server and remote terminal connectivity.

The figure below shows the layout of this scenario.

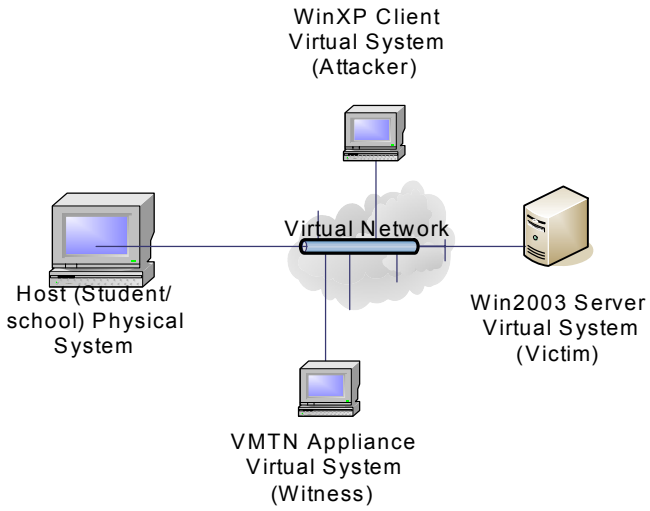


Figure 4: "0wn the box" Attack team layout

4. Liability issues and trust

Liability in security education is becoming an increasing issue in security education at multiple levels. Liability is the legal responsibility of a liable party to compensate an injured party for loss suffered due to the liable party's acts or omissions (Weinstein 1988). In a traditional "Brick and Mortar" environment there is enough contact between instructor and student to make some initial assessment of the ethical conduct and behavior of the student – but as online educational opportunities (sometimes referred to as e-learning) continue to grow, the potential for exposure and liability to multiple parties – the student (for their actions), the educational institution (and in some cases the instructor for the specific knowledge imparted) and in some cases other hosting or sponsoring organizations (as a function of how/where the training is provided and the delivery method).

The legal issues have been covered sufficiently in detail elsewhere – but it is important to reiterate here that the training provider as well as the learner (individual student or corporate sponsor) both may share responsibility (liability) because both are capable of exercising a degree of control over the student receiving the training. Teaching students to defend information infrastructures involves teaching them the methods and strategies involved in attacking those structures from a theory and practice standpoint. It is clear that educators, instructors, and training institutions share in the burden of responsibility to teach students not only technical skills, but ethical and legal behaviour and moral constraints that they can and will apply to their new-found knowledge. There also is an obligation to ensure that the students accept the moral and social obligations inferred or deduced from their learning activities and skills that are not taught directly.

4.1 General responsibilities for students

In attempting to address this issue in a course situation – I have developed the following Ethical contract:

> *In this class I am learning network and computer security principles. This course will impart information to the student that may (if misused) have serious repercussions to the student, as well as damage to the reputation of the student, educational institution, and sponsoring organization. This course requires / expects a foundation in computer systems with added prerequisites of a general understanding of operating systems and computer networking.*

- I assure the instructor, the University, and the community that I am a responsible, and principled person. I will not engage in activity that deprives others in order to benefit from it.
- The techniques and links that I am exposed to are for educational purposes only. As a user of computers and potential network or systems administrator, I must be familiar with the tools that may be used to bring a system or network down. I may engage in a legitimate form of ethical hacking, or as a consultant performing security audits. This is the driving force in learning attack techniques – not misadventure or gain.
- I will not directly provide anyone with the tools to create mischief. Nor shall I pass my knowledge to others without verifying that they also subscribe to similar principles as contained in this statement.

- I will not engage in or condone any form of illegal activity including unauthorized break-ins, cracking, or denial of service attacks.

_____
Name of the student Signature and Date

5. Acknowledgements

I wish to acknowledge the support of my colleagues at Boecore. I also am grateful for the opportunities to develop these ideas within courses I have taught at Colorado Technical University.

6. In conclusion

This paper has presented example scenarios to effect an IA focused learning environment, that retains the viability of a 'real' network environment while providing an isolated and restricted (and hopefully limited liability) learning area. This can be achieved through a series of scenarios and VM Teams (scenario members) in a virtual machine based learning environment.

This approach has tried to show a number of benefits:

- Isolation of the learning environment
- Restricting potential issues (DoS / attack signature detection) to a secured space
- Provide limits on tool and technique usage and impact
- Limits and mitigates risk and liability issues for the educational institution

References

Ethereal – packet capture software http://www.ethereal.com (visited August, 2006)
R. Goldberg. Survey of virtual machine research. *IEEE Computer Magazine*, 7:34–45, June 1974.
Knoppix STD – Security Tools Distribution http://s-t-d.org/ (visited August, 2006)
* Ji Hu, Dirk Cordel, Christoph Meinel. A Virtual Laboratory for IT Security Education.Proceedings of the Conference on Information Systems in E-Business and Egovernment
(EMISA), Luxembourg, 6-8 Oct 2004, pp. 60-71 http://www.informatik.unitrier.de/~meinel/papers/Trier-Emisa04-Hu.pdf
J. Sugerman, G. Venkitachalam, and B. Lim. Virtualizing I/O devices on VMware workstation's hosted virtual machine monitor. In *Proceedings of the 2001 Annual Usenix Technical Conference*, Boston, MA, USA, June 2001
VMware Whitepaper 'Development in Multitier Configurations with Workstation'
(C) 2005 VMware, Inc. 3145 Porter Drive Palo Alto, CA 94304 www.vmware.com
VMware, Inc. VMware virtual machine technology. http://www.vmware.com/ (visited September 2006)
VMware Player. http://www.vmware.com/products/player/ (visited August, 2006)
VMTN Virtual Appliances http://www.vmware.com/vmtn/appliances/ (visited August, 2006)
VMware, Inc. *Oxford University Computing Services Benefits from Using VMware Virtual Infrastructure* (visited September 2006) http://www.vmware.com/customers/stories/oxford.html
* Alec Yasinsac, Jenny Frazier, and Marion Bogdonav, "Developing an Academic Security
Laboratory", 6th National Colloquium for Information Systems Security Education 2002,
June 3-7, 2002, Microsoft Headquarters, Redmond,
Washington.http://www.cisse.info/history/CISSE%20J/2002/yasi.pdf

Differential Power Analysis Attacks against AES Circuits Implemented on a FPGA

Keisuke Iwai, Minoru Sasaki and Takakazu Kurokawa
National Defense Academy, Japan
iwai@nda.ac.jp
g44037@nda.ac.jp
kuro@nda.ac.jp

Abstract: Cryptosystems implemented in embedded systems are more and more increasing thus its tamper resistances are becoming much more important naturally. Differential power analysis (DPA) attacks presented by Kocher et al. are most attractive methods as side channel attacks. This article discusses results of DPA attack against AES cryptosystems implemented in embedded devices which are designed with three different methods. Two of them are designed as hardware circuit using Verilog and Handel-C, and another one is designed as software program on embedded CPU. As results of DPA attacks on them, leaks which include information of secret keys appeared in all circuits but a circuit designed with C-based language shows leaks less than others.

Keywords: DPA, side channel attacks, embedded systems, FPGA, AES

1. Introduction

In recent years, devices such as PDA which implemented using embedding technology become a major product on a market globally. For security purpose, cryptographic circuits are embedded in these devices. Therefore, researches to protect these embedded devices against various attacks are discussed.

Differential power analysis (DPA) attacks presented by Kocher (Kocher, Jaffe and Jun 1999) are most attractive methods as side channel attacks. Modern cryptosystem devices are implemented using semiconductor logic gates in CMOS logic ICs. The power consumption changes when the logic status of a gate switches. These methods exploit the relationship between the switching of circuits and their power consumptions. DPA attacks against cryptosystem devices needs power consumption traces while it executes cryptographic operations. In case of common key cryptosystem, DPA attacks require shorter operation time than other classical cryptanalysis.

On the other hand, progress of the technology of semiconductor process provides a complex circuit design which requires a desperate effort for designers. To cope with these problems, C-based design tools known as Hardware/Software co-design tools are proposed (SPECC, Handel-C, System-C). These tools also help to implement cipher algorithms on hardware such as embedded systems.

Previous reports mainly focused experimental DPA attacks against cryptosystem circuits implemented on hardware designed with HDLs, e.g., Verilog and VHDL, or software implementations on PCs consists of popular processors. This article discusses effectiveness that a difference of design methods gives for DPA against AES. We attempt to experimental attacks on AES circuits implemented on a Field Programmable Gate Array (FPGA) with three different design methods using HDL, C-based language and embedded CPU with software program. Remainder of this article is structured as follows. Section 2 introduces implementation methods of cipher algorithms on embedded systems. Section 3 shortly shows AES and its implementation. Section 4 describes environment of practical examination of DPA attack in detail. In section 5, shows results of DPA attacks against three circuits designed with different method and finally in Section 6 conclusions are drawn.

2. Design methods for embedded cryptosystem circuits

Many cipher algorithms are implemented in various embedded circuits, e.g., PDAs, smartcards and mobile phones. Today also there are varied design methods to implement them on embedded systems. At first, there is a need to decide which way to take realizing target systems as software or as hardware with consideration of hardware costs and developing time carefully when embedded systems is designed. Implementing embedding systems as hardware circuits will bring about not only high performance but also it will force to make up complex HDL codes. C-based design methods which are proposed to be instead of traditional design tools using HDLs provide rapid and easy implementing hardware circuits.

Implementing cipher algorithms on embedded CPUs, such as ARM(ARM), PowerPC(PowerPC) , and SH(SH) (including compatible machines) realizes rapid and easy design because of software including cipher algorithms are able to written in any programming language. Although implementing cipher algorithms with software languages bring about flexibility and quick development, it requires sometime large memory and provides low bandwidth. Many processors and peripherals are provided as IPcores(Intellectual Property cores) which are embedded into user chip design easily today.

This article focuses on AES circuits designed with three different methods shown as below,

- Software implementation on embedded CPUs
- Circuit implementation designed with HDL
- Circuit implementation designed with C-level description

To implement cryptosystems as a hardware circuit, we have two level design flows using HDL and C-base language shown previous sentence. Implementing circuits including cipher algorithms with HDLs (e.g. VHDL and Verilog) is a most popular method. HDLs are able to describe hardware circuits in detail although it is too difficult to implement large circuits because of it requires complex algorithms used modern crypts compile into HDLs.

C-based design, e.g., System-C(SystemC), Spec-C(SpecC) and Handel-C(Handel-C) are becoming major design method today. They provide easier design method for implement algorithms than using HDLs. Most of C-based design tools are in progress to enable more efficient verification and synthesis process.

3. Attacking with DPA against AES and proposed countermeasures

AES (NIST 2001) is the successor of the DES (NIST 1999) which is symmetric block cipher algorithm encrypts 128-bit block plain-text data with a 128, 192 or 256 bit cipher key. The algorithm consists of keyscheduling part and rounds which has 10 rounds for 128 bit length key, 12 rounds for 192 bit and 14 rounds for 256 bit length key. Each round requires round keys which calculated from cipher key in key scheduling function. A round is constructed with `AddRoundkey`、`SubBytes`、`ShiftRows` and `MixColumns`. Also Fig. 1 shows a flow of AES.
All functions except `SubBytes` of each round consist of linear functions. The `SubBytes` transformation is non-leaner transformation which is multiplicative inverse in *GF2^8*.

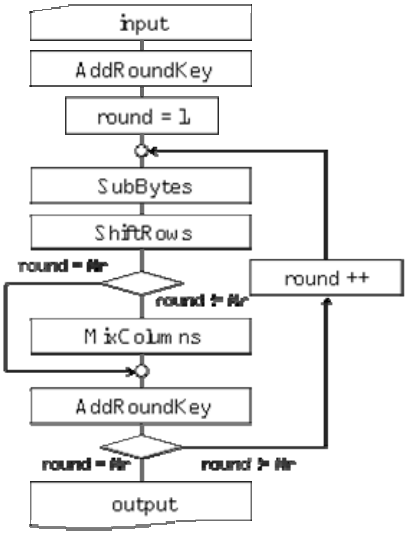


Figure 1: A flow of AES

To attack with DPA, it needs to pay attention where non-liner transformation happens because of DPA detected an electrical leak from a power consumption based on tendency of data bits appears in non-liner transformation. According to the original method to attack AES shown in(Kocher, Jaffe and Jun 1999), as an attacker only needs to attack an 8-bit sub key at once, which leads a key space of *2^8* possible keys, he only has to repeat his attack for the remaining 15 Sboxes to get the full key.

After (Kocher, Jaffe and Jun 1999), many countermeasures have been proposed. Akkar and his coworkers proposed the `SubBytes` transformation with masking countermeasure (Akker and Giraud 2001). It will be more secure AES implementation including software implementation. WDDL(Tiri, Akmal and Verbauwhede 2002), Masked-AND(Trichina 2003), MAND(Shimizu 2004) and Random Switching Logic(Suzuki, Saeki and Ichikawa 2004) are more secure solutions based on implementing leak-less custom gates although they requires special hardware. Also results of testing these countermeasures against DPA attacks are reported. Most of these experiments are implemented in ASICs, FPGAs or considered simulations with contemporary methods such as HDL based design or implemented software running PCs.

4. FPGA implementations of symmetric-key cryptographic techniques

4.1 Implementation platform

The purpose of this article is to clear the effectiveness given from a difference of implementation methods hence it needs a platform which is able to emulate various embedded cryptosystems. For this experiment, we arrange a platform consists of FPGA named SCAPE (Side Channel Attack Platform for Evaluation) board developed by Mitsubishi Electric Corporation. SCAPE consists of FPGA named Virtex2 Pro(XC2VP70), I/O including RS232C and some outlet posts to connect probe of oscilloscopes. FPGA is a reconfigurable device which makes hardware circuit such as ASICs and a system on a chip consists of CPU and memories.

4.2 Design environment

`The SubBytes` transformation is implemented with three kind of design method two of hardware and one of software. We use Verilog HDL to implement it as traditional design method and Handel-C as modern C-based design. As a software implementation we use a processor named Microblaze which is embedded on FPGA provided by Xilinx. Several design tools are shown in Table 1, Table 2 and Table 3.

Table 1: Design environment of Verilog.

Language	Verilog-HDL
Synthesis	Xilinx ISE v7.1i
Place and route	Xilinx ISE v7.1i

Table 1: Design environment of Handel-C

Language	Handel-C
Synthesis	Handel-C
Place and route	Xilinx ISE v7.1i

Table 2: Design environment of Microblaze

Language	C
Synthesis	Xilinx XSP v7.1i
Place and route	Xilinx ISE v7.1i

4.3 A procedure of experimental attack

In this examination, we set a target to find out an 8-bit subkey *Klast* which used in final round of AES. To observe a leak in power consumption we need to make a selection function *D(Ki,C*). (*Ki* is a subkey of round *i* key, *C* is cipher text.) Purpose of this examination is found out a subkey of final round which is *Klast*. *D(Klast,C)* is defined as follows,

- Expect Klast
- Make Exclusive-OR *Klast* by *C*. This is a part of `AddRoundKey`
- Calculate inverse `ShiftRows` with a result of last step.
- Calculate inverse `SubBytes` with a result of last step.
- A bit of result of last step is outputted as answer.

Using this function *D(Klast,C), C* is classified to 2 groups of sets named *D0* and *D1*. Each inputted plain texts in final round are classified into *D0* and *D1* too. We record multiple power consumption traces of this round and they are divided *D0* and *D1* with inputted plain texts. If expected key was correct, we could observe a significant wave (called leak) which is a result of difference from set of power consumption curves of *D0* and *D1*.

A procedure of this experiment is shown in Fig. 2 and explains below step by step.

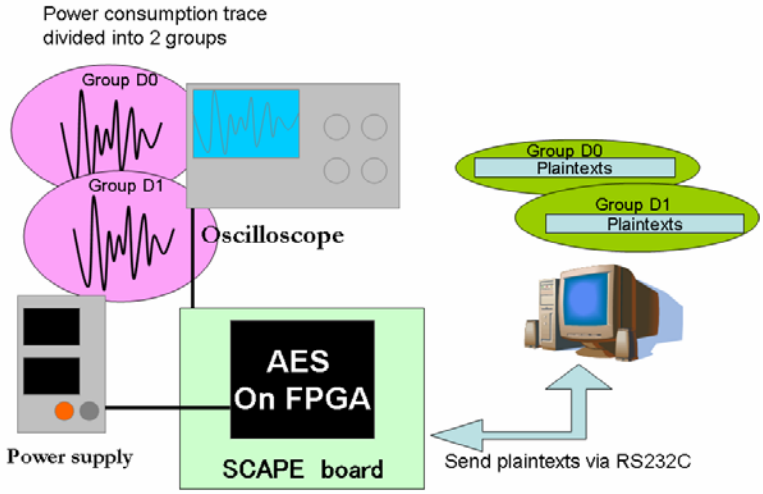


Figure 2: Examination environment

- A plaintext is randomly generated.
- It is encrypted through final round of AES and get a trace of power consumption simultaneously.
- A trace is classified into two groups *D1* or *D2* using select function *D(Klast,c)* with expected 8-bit subkey.(*Klast* is expected subkey)
- Repeat 1-4 *N* times.
- Each set of the average is computed and difference between the two averages is calculated.

If the expected subkey was correct, it would be appeared a significant wave in trace of difference of two averages.

5. Results

This section shows results of DPA on a round of AES explained previous section. Fig. 3, Fig. 4 and Fig. 5 shows results of DPA attacks on AES circuits which implemented with different way of design. Fig. 3 is a result of DPA attack against AES which designed as hardware circuit described with Verilog. Fig. 4 is a result of same one round algorithm designed as software on Microblaze. Also Fig. 5 shows a result of attack against AES circuit which designed with Handel-C. All results shows two traces, one is a difference of two power consumption averages divided by select function *D(K,C)* with correct subkey, another trace shows a difference of two power consumption averages divided by wrong subkey. These results are lead by 40000 plaintexts (*N*=40000).

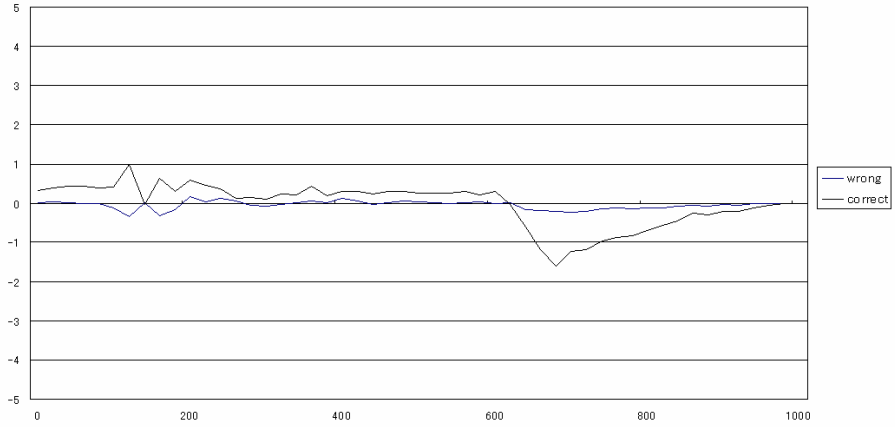


Figure 3: Attack result: Verilog (*N*=40000)

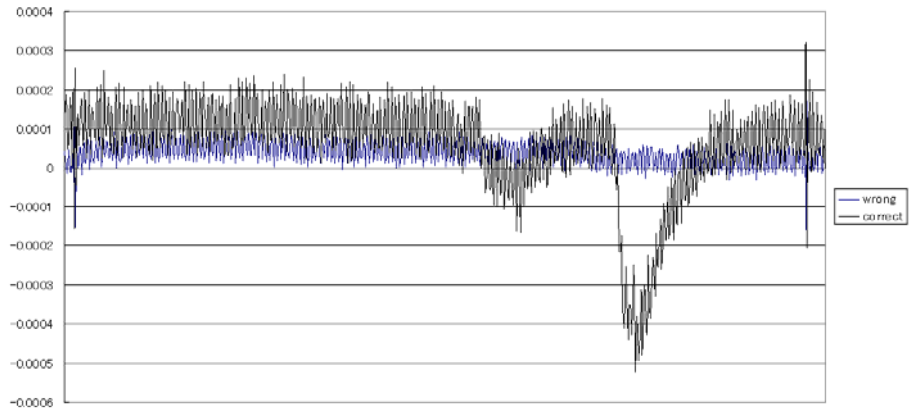


Figure 4: Attack result: Microblaze (*N*=40000)

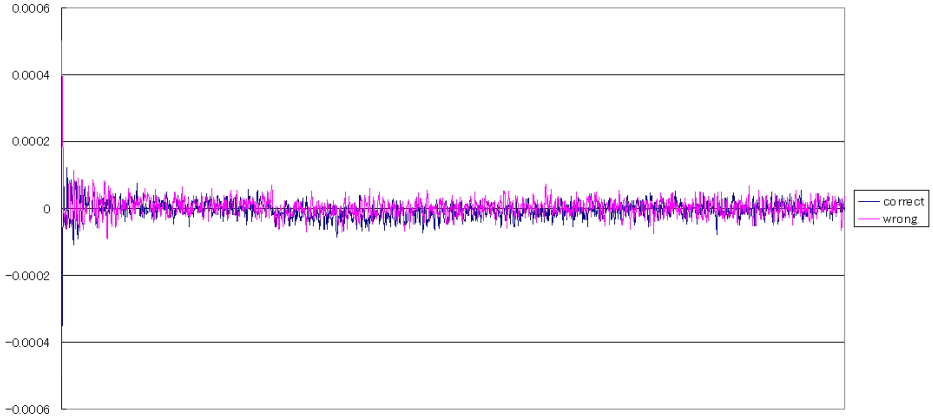


Figure 5: Attack result: Handel-C(*N*=40000)

Fig. 3 and Fig. 4 shows significant wave when expected subkey is correct. It means both of circuits have a possibility DPA attacking. Fig. 5 shows no significant wave. It means difficult to attack on circuit implemented with Handel-C circuit with DPA. However we attempted to attack same circuit with more plaintexts. A result shown in Fig 6 is appeared a leak clearly.

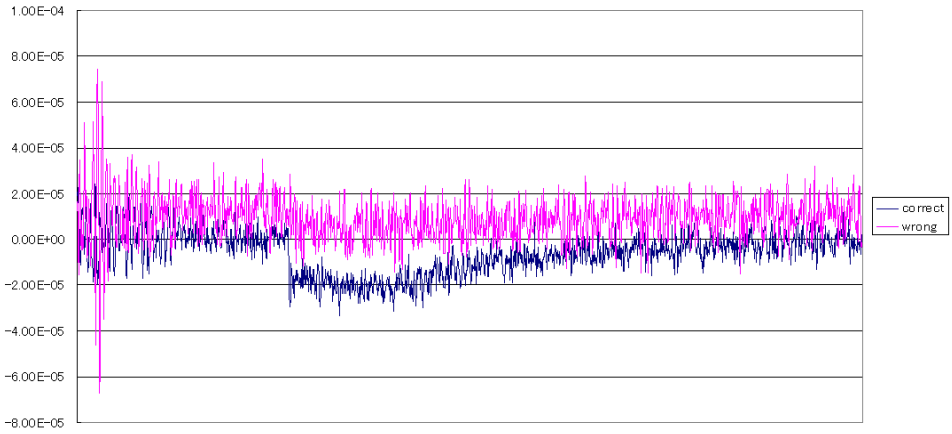


Figure 6: Attack result: Handel-C (*N*=960000)

According to these results, there is hiding leaks of key information in power consumption traces because of implemented with Handel-C circuit consists of more complex structure than HDL implementation. On the contrary, it shows distinctly significant waveform in case of implemented with embedded CPU. As a result, in a successful attack, the power consumption trace of these three circuits with the correct subkey displays a significantly higher correlation.

6. Conclusion

We implemented circuits of a round of AES cryptosystem including SBoxes with three different methods and so they are observed leaks of key information in power consumption trace of all circuits. Consequently it means there is possibilities of DPA attacks are succeed in these three circuits. In case of AES implementation with software algorithm on CPU have a pronounced tendency to leak information clearly although to observe leaks from circuit which implemented with Handel-C is difficult. According to a power consumption model proposed by Messerges (Messerges 2000), processor such as MPUs leaks information the Hamming weight of the data being processed, so that the selection function *D(Klast,c)* which used this article detects a relationship of secret key and plaintexts because of of distinguished power consumption traces by selection function correlate with the Hamming weight in case of AES implementation of CPU. On the other hand, in case of hardware circuit implementation, the selection function used this article did not detect enough correlate with power consumption and secret key because of there is a need to divide traces by tendency of switching of the circuit according to power consumption model for circuits implemented in CMOS technology leaks depends on their switch transition (Suzuki, Saeki and Ichikawa 2005). Discussions about more efficient selection functions are future work.

Other side, as a result of this experimentation C-based design method provides rapid implementation of hardware circuit as expected. We ware able to implement it easily with Handel-C actually. Though most of C-based design tools which have a possibility to hide leaks as side effect are in progress, it could generate large and complex circuits automatically with short steps. Future work, we will implement some countermeasures against DPA with three design method same as shown this article to evaluate their vulnerability.

References

Akkar, M. L and Giraud, C (2001)“An Implementation of DES and AES Secure against Some Attacks”, Cryptographic Hardware and Embedded Systems - CHES 2001, LNCS2162, pp309-318 ARM [online] http://www.arm.com

Celoxica [online] http://www.celoxica.com

IBM PowerPC microprocessors. [online] http://www-3.ibm.com/chips/power/powerpc/

Kocher, P., Jaffe, J. and Jun, B. (1999) “Differential power analysis, In Advances in Cryptology”, CRYPTO’99, LNCS 1666, pp388-397

Messerges, T. (2000) “Using Second-Order Power Analysis to Attack DPA Resistant Software”, Cryptographic Hardware and Embedded Systems - CHES 2000, LNCS1965, pp238-251.

NIST (1999) “Data Encryption Standard (DES)”, [online], Federal Information Processing Standard FIPS Publication 46-3 http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

NIST (2001) Advanced Encryption Standard (AES), [online] Federal Information Processing Standard FIPS Publication 197 http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf Publication 197

Renesas technology.[online] http://www.renesas.com

Shimizu, H. (2004) “Countermeasure against Side Channel Attack using Mask Logic Elements”, ISEC2004, Vol.104, No.315, pp15-19 in Japanese.

SpecC Technology Open Consortium [online] http://www.specc.gr.jp/

Suzuki, D., Saeki, M. and Ichikawa,.T. (2004) “Random Switching Logic: A Countermeasure against DPA based on Transition Probability”, Cryptology ePrint Archive (http://eprint.iacr.org/), Report 2004.

Suzuki, D., Saeki, M. and Ichikawa, T. (2005) “DPA Leakage Models for CMOS Logic Circuits”, Proceeding of Workshop on Cryptographic Hardware and Embedded Systems CHES2005, LNCS 3659, pp354-365.

The Open SystemC Initiative.[online] http://www.systemc.org

Trichina, E. (2003) “Combinational logic design for AES subbyte transformation on masked data”, [online], Cryptology ePrint Archive, http://eprint.iacr.org/2003/236.pdf

Tiri, K., Akmal, M. and Verbauwhede, I. (2002) “A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on SmartCards”, Proc. of 28th European Solid-State Circuits Conference, pp403-406.

Categorisation of Profiles for PSYOPs: Can Technology Help?

Magdi Kamel, Mark Eramo and Christopher Sutter
Naval Postgraduate School, Monterey, CA, USA
mnkamel@nps.edu

Abstract: Influencing one's adversary has always been an objective in warfare. To date the majority of psychological influence operations have been geared toward the masses. A tailored approach of individual targeting is preferred, but this approach requires unattainable resources. This paper investigates whether state-of-the-art data and text mining tools can be used to automate the categorisation/segmentation of individual profiles for psychological operations. Five data and text mining software applications were tested and their results compared with those of a social psychologist. Using statistical analysis, it was concluded that current data and text mining tools are not mature enough to produce results comparable with those produced by psychologists.

Keywords: Information operations, psychological operations, data mining, text mining, text categorisation.

1. Introduction

The informational instrument of statecraft, dubbed Information Operations by the United States military and includes: Psychological Warfare, Deception, Operational Security, Electronic Warfare, and Computer Network Operations, is arguably a very important instrument of national power. It is not however widely accepted as such because the operations are not widely known and the benefits are not easily measured. Most people tend to focus on the other three instruments of statecraft, political, economic, and military, because they are easily identifiable (AUSA 1962). However the informational instrument, specifically the psychological approach, known as PSYOP, can have tremendous effects on the other three instruments by changing an adversary's perception. For this reason, the informational arm of statecraft must be brought to the forefront regarding future operations and PSYOP must be given greater attention (Radvanyi 1990). To date in both military and non-military actions, the majority of influence operations have targeted the mass audiences with little attention for the individual differences among the population. Examples of such operations include radio broadcasts, television airings, leaflet droppings, etc. Although somewhat effective, this approach does not differentiate between the different subsets of the population. It is intuitive that high value targets must receive more personalised attention in order to conduct precision influence. For the remainder of the target population, it is not feasible to provide individual personalisation to every target since the population could be quite large.

When attempting to influence a large number of targets which can not be addressed individually, a different strategy must be formulated to tackle the problem. One strategy is to 'get the best result possible', by sending one message to the masses while precision targeting a very small number of high value targets. This type of strategy does not however address the variability of specific segments of the population but rather identifies a small number of individuals to target and ignores the idiosyncrasies of the remaining masses. A second approach breaks down the target population into smaller groups based on common attributes and tailors the message to each group accordingly. This multi-segment approach, used extensively in marketing, targets the identified segments, conscious of the segment differences. High value targets will still require individual human analysis, but the remaining masses would be grouped into predefined categories based on the categorisation developed for high value targets. In essence, the knowledge inherent in the output of the psychologists could be used repeatedly on the identified categories. Different messages are crafted that provides the best results among the segments. Each segment is treated separately and identified as unique/discriminate customer. The downside of this approach is that sifting through and categorising thousands of profiles would require significant time and manpower with time usually being the critical driver since it would not be practical to postpone a PSYOP campaign. PSYOP messages must be delivered when they are most likely to be beneficial. In order to precision target more individuals, the ability of computers to categorise profiles based on limited personal data will be tested. Specifically, this paper investigates the use of data and text mining tools to automate the process of categorising people based on their personal profiles for psychological targeting purposes. A key assumption is that profiles containing descriptive attributes exist for each target in the desired population, but because it is a time intensive process that far exceeds available manual resources, the task of categorising all the personnel cannot be accomplished in a timely manner. Through the process of text mining and natural language processing, computers can extract the attributes of a person from their profile and classify them into predefined categories for follow-on targeting.

The paper is organised as follows. Section 2 overviews the field of data and text mining, and discusses data and text mining tasks, methods, and algorithms. Section 3 presents the candidate data and text mining tools used in this study. Section 4 presents the research hypothesis, while section 5 describes the data collection procedures. Section 6 provides the data analysis, and sections 7 and 8 present and compare the results. Finally section 9 concludes the paper with a summary, provides useful insights, and identifies research issues based on the findings of the study.

2. Data and text mining

The explosive growth of government, business, and scientific databases has overwhelmed the traditional, manual approaches to data analysis and created a need for a new generation of techniques and tools for intelligent and automated analysis of data. The field of data and text mining is an emerging and rapidly evolving field that draws from other established disciplines such as databases, applied statistics, visualisation, artificial intelligence and pattern recognition that specifically focus on fulfilling this need. The goal of data and text mining is to develop techniques for identifying novel and potentially useful patterns and relationships in large data sets. These identified patterns typically are used to explain existing data, make predictions about new data, summarise existing data to facilitate decision making, and to visualise complex data sets. Whereas most efforts have been on mining data stored in structured databases, there has been an increasing interest on the analysis of very large document collections and the extraction of useful knowledge from text. More than 80% of stored information is contained in textual format, which cannot be handled solely by existing data mining approaches.

2.1 Data mining tasks

A data mining task refers to the kind of patterns to be found by the data mining effort. The five main data mining tasks include: Association analysis, sequential analysis, classification analysis, cluster analysis, and anomaly detection (Tan, Steinbach and Kumar 2006). Association analysis is used to discover patterns that describe items that appear together in an event such as a purchase transaction. Sequential analysis is also used to discover the association of items but that are related over time. Classification analysis is used to identify the characteristics that indicate the group to which each case belongs. Classification analysis can be used both to understand the existing data and to predict how new instances will behave. Cluster analysis seeks to find groups that are very different from each other, and whose members are very similar to each other. Unlike the classes of a classification analysis, the clusters and which attributes the data will be clustered accordingly are unknown at the beginning of the analysis. Finally, Anomaly Detection is the task of identifying observations whose characteristics are significantly different from the rest of the data.

2.2 Text mining tasks

Text mining tasks include keyword-based association analysis, document classification analysis, and document clustering analysis (Han and Kamber 2006). Keyword-based association analysis collects sets of keywords or terms that occur frequently together and then finds the association or correlation relationships among them. Document classification analysis automatically organises documents into classes to facilitate document retrieval and subsequent analysis/action. This task is the focus of this paper. The general procedure for document classification is as follows. First, a set of preclassified documents is used as the training set. The training set is then analyzed in order to derive the classification scheme. The derived classification scheme can then be used for classifying new documents. Finally, document clustering analysis collects documents similar to each other into different clusters in an unsupervised manner.

2.3 Data and text mining models and algorithms

While data and text mining tasks refer to the type of problems to be solved, data and text mining models and algorithms are the methods used to solve a particular mining task. Several combinations of data and text mining tasks and data and text mining algorithms are possible. This means that a data/text mining task can be accomplished using several data and text mining algorithms, and a data and text mining algorithm could be used to accomplish several data and text mining tasks. Models and algorithms used in data and text mining include neural networks, rule induction, decision trees, k-nearest neighbour, bayesian networks, and genetic algorithms (Han and Kamber 2006). It is important to note that no one model or algorithm can or should be used exclusively. There is no ‘best’ model or algorithm. Usually, best results are generated by a combination of models and algorithms. The type of problem will affect the choice of models and algorithms. A variety of techniques and approaches need to be employed to find the best possible model or combination of models used sequentially or in parallel. For example, one common technique is to use a clustering method

to discover natural groupings in the data, and then to use these clusters as a predictor, along with other variables, as input to a predictive technique, such as decision trees or regression models. In many cases, the final data mining model is a composite of several, separate models. Thus, predictions can be made with both a neural network and a decision tree model. When both models agree, they are likely to be more accurate. And when they disagree, the prediction from the model that is more often accurate is selected. In the following sections we discuss some of the best known and widely used models and algorithms.

3. Candidate data and text mining tools

An extensive review of data and text mining tools was undertaken to identify best-of-breed tools based on suitability for the task, models supported, popularity, and users feedback. Five Data and Text mining tools were selected for testing text categorisation on sample data. These tools include Reel Two from Reel Two Inc., SERglobalBrain from SER Solutions, Inc., PolyAnalyst 4.6 from Megaputer Intelligence, LexiQuest Categorise from SPSS, and Enterprise Miner with Text Miner from SAS. Each of these tools incorporates multiple methodologies using multiple algorithms. Table 1 lists the data and text mining tools selected for the study and the algorithms used to perform classification.

Table 1: Data and text mining tools selected for the study

Product	Company	Classification Methods Used
Reel Two	Reel Two Inc.	Naïve Bayesian Network Weighted learning; skewed data
SERglobalBrain	SER Solutions	Neural networks K-nearest neighbour
PolyAnalyst	Megaputer	Decision Trees Fuzzy Logic Memory based reasoning
Lexiquest Categorisation System	SPSS	Undisclosed multiple approaches
Enterprise Miner	SAS	Undisclosed multiple approaches

4. Research hypothesis

The main research hypothesis of this paper is that based on a personal profile of an individual, the time intensive task of identifying which psychological category an individual falls into can be automated through data and mining software tools at a level comparable to that of a social psychologist. In the following sections we describe an experiment designed to accept or reject the research hypothesis.

5. Data collection procedures

5.1 Data sets

The data set consisted of 349 leadership profiles from a population of unknown size. Although approximately 375 profiles were randomly selected from the population, only 349 profiles consisted of usable data. Only those profiles that a psychologist could categorise were extracted from the population. Therefore, there was an injected bias as to what profiles could be considered for further testing. This type of discrimination was consistent with the goal of the research since it was assumed that usable data existed for any individual that would be psychologically targeted. Each profile consisted of a one to three page description of a single person. Each contained various facts about an individual to include chronologies, career information, personality traits, social interaction, and examples of how that person responded in various situations.

5.2 Expert analysis

A social psychologist, from the University of California at Santa Cruz, specialising in influence analysis was provided with the 349 profiles. His task was to determine if the profiles could be categorised into specific categories based on their content. After reviewing approximately 20 to 30 profiles, he determined that four categories were adequate to describe the major personality types in the data set. These categories were consistent with prior research in social influence (Greenwald and Breckler 1985; Greenwald and Praktanis 1984). The four categories determined were: power, personal standards, social approval, and other. The following is a brief description of each of the four categories (Pratkanis (in press)).

- Power: The power hungry type is interested in promoting his or her power and will most likely be calculating the benefits and cost of any course of action in terms of reaching that goal. In general, these are people who would betray others for personal gain and would not be considered trustworthy.
- Personal (private) standards and achievement orientation: An individual with this type has strong personal principles; sees political activity as a way to achieve those principles; internal locus of control; honours commitments; gains self-esteem from living consistent with principles and making achievements that he or she finds important.
- Social approval / public orientation: This social type gains self-esteem by approval form other in general; motivated by being approved of by others in general.
- Other: The other category includes individuals that exhibited varying personality traits. Often these individuals were a conglomerate of the other categories; thus, these individuals were too hard to categorise into one group.

After roughly 60 hours of analysing profiles, the social psychologist categorised 256 documents into the four predefined categories. Figure 1 shows the overall distribution of the categorised 256 profiles.

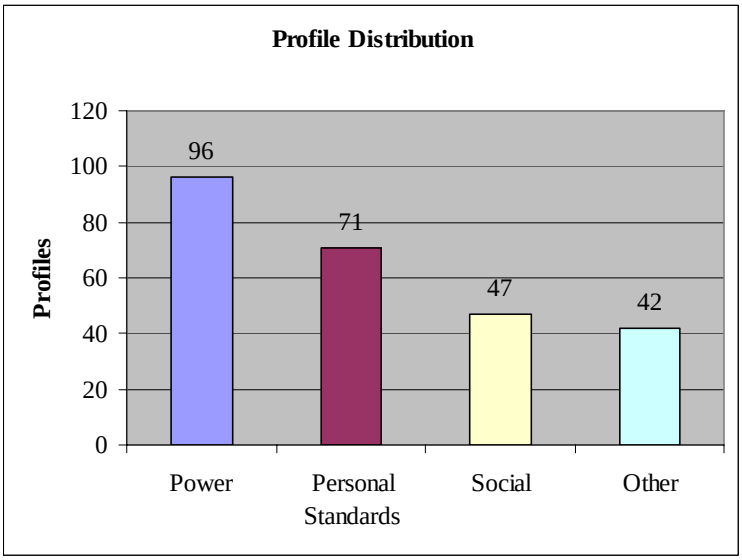


Figure 1: Profile distribution

Table 2 shows the percentage of each category within the data set. Since the sample is large enough, it is expected that the distribution of categories within the population under evaluation is the same.

Table 2: Profile distribution percentages

Power	Personal Standards	Social	Other
37.5%	27.7%	18.4%	16.4%

Table 2 also shows how likely a profile, randomly placed into one of the four categories, was categorised correctly. For example, if a population of 100 profiles was randomly sorted into the four categories (each category receiving 25 documents), the distribution percentages in Table 2 illustrates how many profiles would likely have been placed in the correct category. Consider the power category; it is expected that 37.5% of the 25 documents would be correctly categorised since 37.5% of the population is known to be the power type; therefore, about 9 of the 25 profiles randomly placed into the power category would have been categorised correctly. By adding up the number of documents that are likely to be correctly categorised for each category (~9 for power, ~7 for personal standard, ~5 for social, ~4 for social), the percentage of documents likely to be categorised correctly via guessing would be 25. Therefore, the expected accuracy of randomly placing the documents into one of the four categories is 25%. In order for a software tool to show a better performance than guessing, it must have an overall accuracy that is statistically more significant than 25% within a 95% confidence interval.

5.3 Training and evaluation

From the 256 categorised documents, twenty profiles per category were needed to train the software packages. Sixty profiles (20 each for power, personal standards, and social) were randomly selected to

represent the predefined categories. Each software package used the same 60 profiles as training documents in their supervised learning systems. Because the "other" category had no definitive characteristics, it was not trained as a specific category. This technique was in-line with capabilities of the software tools that allow for an additional category for documents that did not exhibit the attributes within the predefined categories established by the user. The remaining 196 manually categorised documents were then randomly segregated into two evaluation sets. The first set consisted of 97 profiles and the second set consisted of 99 profiles. Once a software package had been trained using the 60 training profiles, an evaluation set was then entered into the tool. Each evaluation data set was categorised by the each tool independently of one another. Figure 2 shows the distribution of how the 256 manually categorised profiles were divided into training documents and evaluation sets.

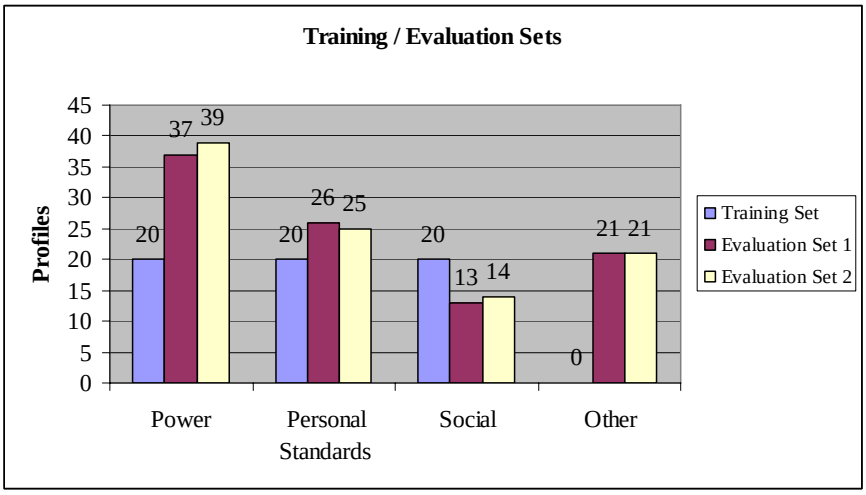


Figure 2: Training / evaluation sets

After the categorisation was complete, the evaluation documents were counted to determine how many documents were categorised into each group, by the software tool, and whether or not each document was categorised correctly. The accuracies of categorisation of the tools relative to that of the social physiologist were then determined through statistical analysis.

6. Data analysis procedures

In order to evaluate each software tool, an assessment method was developed that would examine the results from two separate perspectives with two separate trials. The first perspective evaluated the goodness of fit between the actual results generated by the social psychologist and the specific software package. The second perspective evaluated the overall accuracy of the results obtained by the software tools versus the expert results obtained from the social psychologist; this was considered the second level of confidence. The null hypothesis for the level one confidence analysis was that the distribution produced by a human would not be different from the distribution produced by the software tools. The alternate hypothesis was that the distribution and the method that produced the distribution (human or software system) were not independent. A chi-square test that compares an observed value to an expected value was used to accept or reject the null hypothesis (Moore and McCabe 2003).

$$\chi^2 = \sum \frac{(\text{observed count} - \text{expected count})^2}{\text{expected count}}$$

The second level of confidence derived the accuracy of each software tool evaluated by comparing the actual categorisation of a profile by the human expert with that of the software tool. For each software tool used, accuracy was computed both by category and overall. Accuracy for a category was calculated by dividing the number of correct placements within the category by the total number of documents placed into that category. Similarly, overall accuracy was calculated by dividing the total correct placements in all categories by the total number of documents in the evaluation set. To determine the most accurate software tool, each tool was compared against each other to determine if there was a statistically significant difference between them. Thus, for each of these comparisons a separate statistical test was performed. In order to compare the two populations (software system vs. the social psychologist) the difference in the two populations was used.

$$D = \hat{\rho}_1 - \hat{\rho}_2$$

Because the sample size for the evaluation were sufficiently large (>30) it could be assumed that the sampling distribution of the difference between the two distributions was approximately normal. In order to prove that the differences were or were not statistically significant, confidence intervals were created. If the confidence interval included zero (because the distributions could be assumed to be normal with a mean of zero and a standard deviation of one) then it was proven that the difference was not statistically significant indicating the two systems that were compared performed approximately equally. The confidence intervals were generated with the following formula:

$$\left(\hat{\rho}_1 - \hat{\rho}_2\right) \pm 2 \times SE_D$$

Where:

$$SE_D = \sqrt{\frac{\hat{\rho}_1\left(1-\hat{\rho}_1\right)}{n_1} + \frac{\hat{\rho}_2\left(1-\hat{\rho}_2\right)}{n_2}}$$

In addition to comparing the two proportions by providing confidence intervals, it was useful to test the null hypothesis that the two population proportions were equal. The alternate hypothesis was that the two proportions were not equal.

$$H_0 : \rho_1 = \rho_2$$

$$H_a : \rho_1 \neq \rho_2$$

In order to test this hypothesis the pooled estimate, pooled standard error and z statistic were computed (Moore and McCabe 2003).

$$\hat{\rho} = \frac{\text{number of successes in both samples}}{\text{number of observations in both samples}} = \frac{X_1 + X_2}{n_1 + n_2}$$

$$SE_{D_p} = \sqrt{\hat{\rho}\left(1-\hat{\rho}\right)\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}$$

$$z = \frac{\hat{\rho}_1 - \hat{\rho}_2}{SE_{D_p}}$$

Once the z statistic was computed it was compared to the critical z (derived from the desired level of confidence) and if it was above or below the positive and negative critical z the null hypothesis was rejected. Otherwise the null hypothesis could not be rejected. This test showed whether or not the software tools provided approximately the same level of accuracy or that there was a statistically significant difference between them. Once each tool was compared against each other the same statistical tests were used to compare each tool to the overall accuracy of random guessing. The overall proportion of correct placements with guessing was computed through probability and reinforced through simulation. From the manual categorisation, the overall proportions of profiles were: .375 power, .277 personal standards, .184 social, and .164 other (Table 1). The probability of a profile being randomly placed into any one category had a probability of .25. By using these probabilities through simulation, it was determined that the overall number of successes in one hundred trials would be approximately twenty-five. Therefore, the expected accuracy of guessing based on one hundred profiles and four categories would be 25%. The comparison between each software tool and guessing provided the needed reinforcement that each application was either correctly categorising documents or simply providing a random guess as to where profiles should be placed. It should be noted that although Enterprise Miner (EM) was one of the five software applications that was to be evaluated, it was unable to produce measurable results that could be compared to the other applications since it only provides unsupervised learning and concept extraction which is short of the goal of automated categorisation based on training sets. Enterprise Miner was not therefore considered further. It should also be noted that PolyAnalyst was not capable of performing supervising learning without user defined taxonomy. This required supplying a list of descriptors to the software that specified what to look for and match for each category. Descriptors include individual words, phrases, synonyms, and root forms of words.

7. Experiment results

The results of Level I for each tool are summarised in tables 3 and 4. For both trials, all the software tools failed the Level I confidence evaluation using the chi-square test. The failure of level one confidence indicates that software tools were not performing the task of categorisation in the same manner as the social psychologist.

Table 3: Trial I results

Software Package	Version/Release	Settings	Results	
			Level I Confidence	Level II Confidence (Proportion Successful)
ReelTwo	Classification System 2.4.6	Taxonomy depth: full Exclude empty directory: No Subsumption: No Check for duplicates: No Split Documents: No	Reject null hypothesis	.3299
SER Globalbrain	Personal Edition 1.7.0	Minimum pattern size: 3 Maximum dictionary size: 100,000 Include numbers: Yes Absolute relevance value: 51 Relative distance value: 1	Reject null hypothesis	.2990
Lexiquest Categorise	Lexiquest Taxonomy Manage for Categorisation 1.5	Minimum number of categories per document: 1 Maximum number of categories/document: 3 Minimum confidence level for single response: 1	Reject null hypothesis	.2990
PolyAnalyst 4.6	4.6.498	Key sense type: All Find collocations: No Taxonomy	Reject null hypothesis	.3608

Figure 3 explores the actual distributions for the categories across the software tools in order to determine which category (or categories) caused the null hypothesis to be rejected. Figure 3 shows that the categories of power and personal standards are not significantly different than what would be expected. However, the categories of social and other are significantly different than what would be expected. This provides evidence that the classification of the category of social caused the null hypothesis to be rejected by all the software tools.

Figure 3: Trial I level i confidence

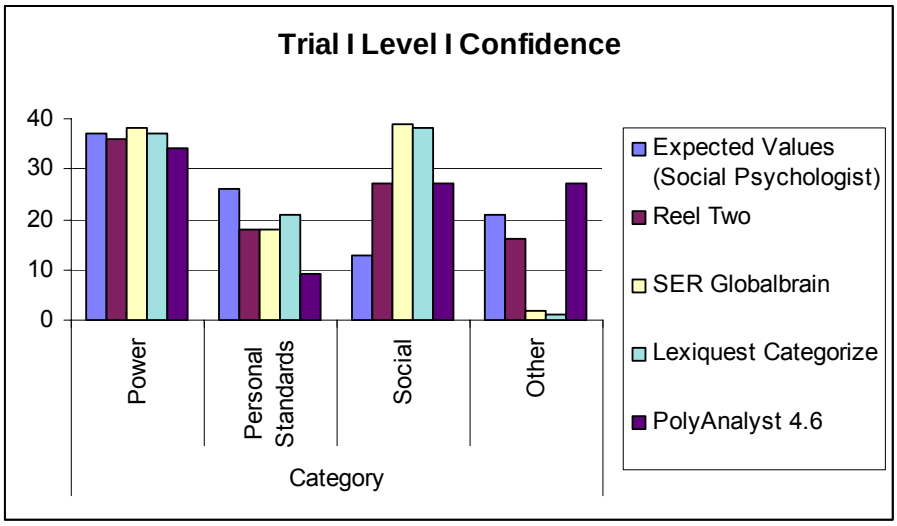
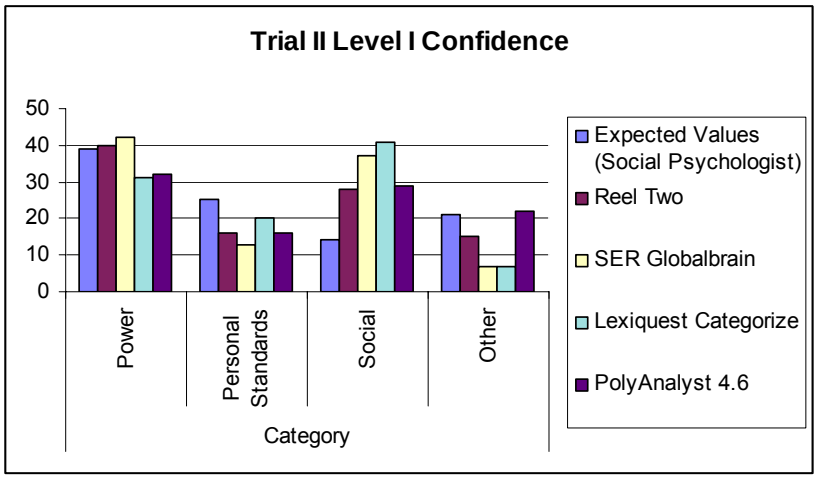


Table 4: Trial II results

Software Package	Version/Release	Settings	Results	
			Level I Confidence	Level II Confidence (Proportion Successful)
ReelTwo	Classification System 2.4.6	Same as trial I	Reject null hypothesis	.3535
SER Globalbrain	Personal Edition 1.7.0	Same as trial I	Reject null hypothesis	.3030
Lexiquest Categorise	Lexiquest Taxonomy Manager for Categorisation 1.5	Same as trial I	Reject null hypothesis	.3232
PolyAnalyst 4.6	4.6.498	Same as trial I	Reject null hypothesis	.3232

For the second trial the null hypothesis was rejected for all tools again. Figure 4 provides the detailed breakdown of the distribution for each application. Again the classification of the social category caused the rejection of the null hypothesis.

Figure 4: Trial II level I confidence



The results of Level II confidence that measure the accuracy of the software tools compared to each other and to random guessing confirmed the results of Level I testing. For both trials of Level II confidence, the null hypothesis, indicating no difference between the accuracy of tools as well as between the accuracy of the tools and random guessing, was accepted. The only exception was the results of PolyAnalyst in Trial one that rejected the null hypothesis indicating a statistically significant difference between the tool classification and random guessing. At the completion of two independent trials, that both rejected the null hypothesis for Level I confidence and accepted the null hypothesis of Level II, further exploration of the data was conducted. The social category was creating the most difficulty for the software tools. A new hypothesis to explain why the social category was problematic pointed towards the concepts in the social profiles. The concepts in a social profile are similar to those of other categories when simply comparing frequencies of occurrence. It is believed that each application was extracting concepts independent of their context or overall document sentiment. A social approval profile could possess many words that are similar to the keywords in other categories. For example, the words power or aggressive could be used in many profiles that span the categories, but it is the context in which they are used that determines the sentiment of the profile and category in which it belongs.

8. Comparison of results

In order to determine which of the software tools performed the best, individual comparisons were conducted between each tool and random guessing. The results of the comparison are graphically represented in Figures 5 and 6.

Figure 5: Trial I accuracy comparison

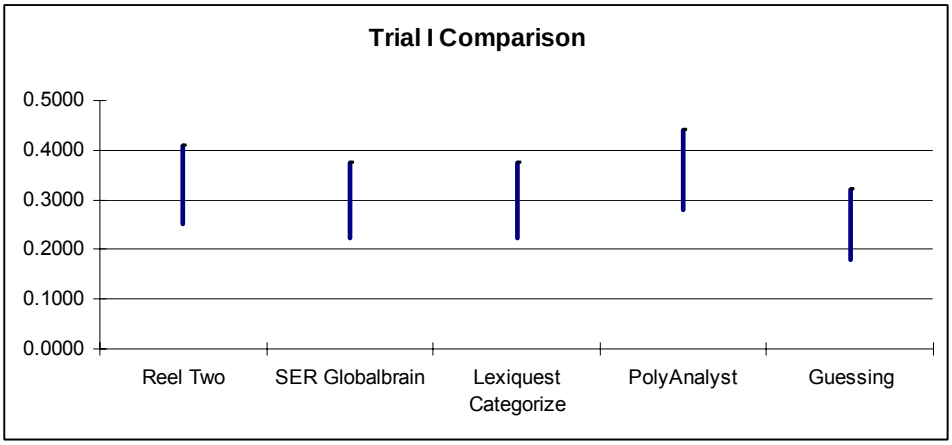
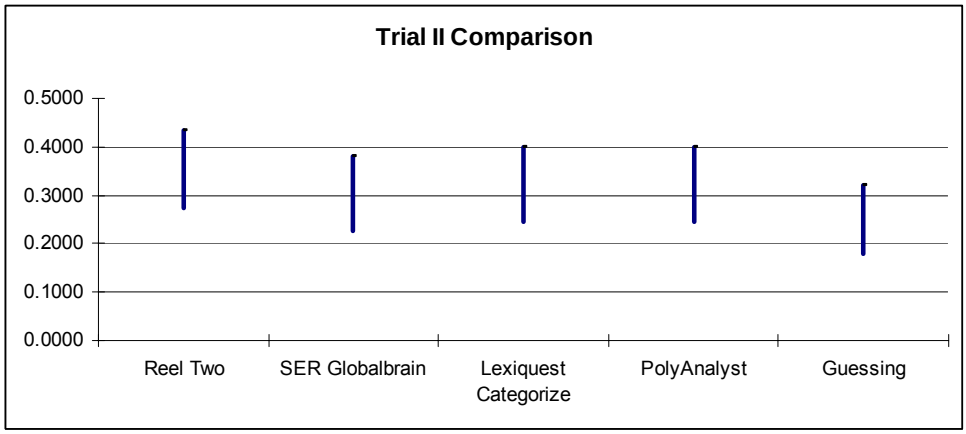


Figure 5 represents the successful classification percentage of each software application along with its 95% confidence interval. When each tool was evaluated against each other, it was determined that none of the software tools were significantly different than any other. Additionally, there was no statistical significance between random guessing and the results any of the software tools with the exception of PolyAnalyst. On the second trial, all the software applications performed approximately the same with ReelTwo performing marginally better. Similar to Trial one, there was no statistically significant difference between the results of each tool with each other, or between the results of each tool and random guessing. Figure 6 depicts the results of the second trial.

Figure 6: Trial II comparison



9. Conclusions

The ability to categorise individuals of interest in a timely manner for the purpose of psychological targeting is highly desirable. Unfortunately, this task is one that requires significant time from skilled psychologists, and is therefore not feasible to accomplish even with a moderate number of individual profiles. This paper attempted to answer the question of whether the new crop of state-of-the art data and text mining tools can automate the process of classifying psychological profiles into defined categories at the same level of proficiency as that of a psychologist. Based on statistical analysis, the results of the research indicate that current tools are not robust enough to perform automated categorisation at a level comparable to that of an expert psychologist. The null hypothesis for the Level I confidence, indicating no difference between the classification by an expert and that of a software tool, was rejected while the null hypothesis for Level II confidence, indicating no difference between the accuracy of tools and the accuracy between the tools and random guessing, was accepted.

While the research did not conduct a detailed analysis to determine the reasons for the rather poor performance of the tools, some useful insights were gleaned from examining the results of each tool. We hypothesise that although the tools could extract concepts from profiles quite well, they were unable to use the extracted concepts in context and develop an accurate sentiment for the overall profile for the purposes of classification, as an expert psychologist would do. Although the results of the current research indicate the inadequacy of the current tools to perform automated categorisation, it provides useful insights on what approaches to use to improve the efficiency of the task. As indicated one of the tools, PolyAnalyst, could not perform automated classification under a supervised environment based on training documents, but was capable of performing automated classification based on user defined rule sets. By using rule sets, list of descriptors that specify what to look for and match for each category, the software was able to match words, phrases, and concepts that were defined by the user rather than relying on extracted concepts from training document. In this regard, the system was given less responsibility for recognising which category a profile fits into and was able to instead match words, phrases, and concepts that it determined were similar. This type of approach produced the only test results that were statistically more significant than guessing.

This result suggests that rather than having psychologists analyze hundreds of profiles and manually categorise them, their efforts should focus on the creation of expertly crafted rule sets that contain key phrases, words, and concepts that would enable software tools to conduct the classification of the uncategorised documents. These rule sets could then be used by multiple tools to automate the classification of profiles quickly and inexpensively. Further research is needed to validate this approach. Since the tools used in this study are proprietary in nature, little detailed information is available on the algorithms they use. Further research and collaboration with vendors is required to better understand the approaches used, identify those which hold the greatest promise, and develop tailored new approaches and tools specifically for the automatic categorisation of profiles.

References

Association of the United States Army (1962). *Army*, vol. 13, no. 5.

Greenwald, A.G. and Breckler, S. (1985). To whom is the self presented?. In Schlenker, B. (ed), *The self and social life*, McGraw-Hill, New York.

Greenwald, A.G. and Pratkanis, A.R. (1984). The self. In Wyer, R.S. and Srull, T. (eds), *The Handbook of Social Cognition*, Lawrence Erlbaum, Hillsdale, NJ.

Han, J. and Kamber M. (2006). *Data Mining: Concepts and Technique*, 2nd ed, Morgan Kaufman, San Francisco.

Moore, D.S. and McCabe, G.P. (2003). *Introduction to the Practice of Statistics*, 3rd ed, W.H. Freeman and Company, New York.

Pratkanis, Anthony, R. (in press), Social influence analysis: An index of tactics. In. Pratkanis, A.R (ed), *The Science of Social Influence: Advances and Future Progress*, Psychology Press, Philadelphia.

Radvanyi, J. (1990). *Psychological Operations and Political Warfare in Long-term Strategic Planning*, Praeger Publishers, New York, NY.

Tan, P., Steinbach, M., and Kumar V. (2006). *Introduction to Data Mining*, Addison Wesley, Boston.

Implications of Information Flow Priorities for Interorganizational Crisis Management

Tuija Kuusisto[1], Rauno Kuusisto[2] and Mark Nissen[3]
[1]Ministry of Defence, Helsinki, Finland
[2]National Defence University, Helsinki, Finland
[3]Naval Postgraduate School, Monterey, CA, USA
tuija.kuusisto@defmin.fi
rauno.kuusisto@tse.fi
MNissen@nps.edu

Abstract: The aim of this paper is to increase understanding about the implications of information flow priorities for interorganizational crisis management. The paper is based on the main results of a prior study that addressed the information requirements of high-level decision-making activities during a sudden crisis situation. The study identified priorities of retrieved and delivered information flows from the perspectives of information content and information update frequency needs. The results about the information priorities based on update frequency needs provide novel insights into information flow priorities for decision-making in crisis management. These results show the most effective information categories to prioritize for decision-making in the context of interorganizational crisis management.

Keywords: Information flows, decision-making, homeland security, crisis management

1. Introduction

Typical crisis management activities involve networks of governmental organizations and units as well as private companies and special interest groups. The activities both exploit and create information associated with these networks of actors. Historically in crisis management situations, all the information needed for management, decision-making and action is not available across the entire network of organizations and actors (Figure 1). Further, it has been difficult to anticipate which specific types of information will be required across such interorganizational network, as well as the frequency with which such information types will need to be updated. Information has to flow smoothly for crisis management from one actor to another, both internally in an organization and collaboratively between organizations, to enable effective crisis management activities. Actors across the various organizations must achieve and share situation awareness and understanding. This requires an integration of technological systems with human analysis and organization.

This paper reports the results of a study to assess the implications of information flow profiles to support interorganizational crisis management. The paper is based on the results of a prior empirical study that addressed the information requirements of high-level decision-making activities during a sudden crisis situation. The target of the study was the Finnish national administration. In the balance of this paper, first, the theoretical framework of the study is summarized. Then, discussion of changes in information profiles when moving from normal to crisis conditions is presented. This includes outlining priorities of information flows from both information content and information temporality perspectives. Finally, proposals for developing the structure of information delivering interorganizational coalitions are presented based on the priorities of information flows. Crisis management involves activities performed across several organizational layers (Figure 2). Information is delivered and retrieved on each layer and between the layers. The layers include a political-strategic layer that guides and is reported by operational-theatre level. Across such layers, crisis management activities are implemented and managed by a variety of actors, who participate across a diversity of organizations. History with crisis management responses suggests that the classic, hierarchical command and control (C2) approach is inadequate to organize the associated decisions and actions effectively. Crisis situations change too quickly, and hierarchical information flows too slowly, for this C2 approach to be effective. This represents a fundamental limitation of hierarchical C2 in the crisis management context.

Alternatively, a newly conceptualized C2 approach involves dispensing with most hierarchical information flows, and replacing them with horizontal flows within and between operational level organizations. Such horizontal information flows, where sufficiently rich and prioritized, can enable diverse organizational actors to self-organize their roles, and to self-synchronize their activities. In the military environment synchronization is defined as the arrangement of military actions in time, space, and purpose to produce maximum relative combat power at a decisive place and time (DoD 2005). Self-synchronization involves the accomplishment of

such arrangement without hierarchical direction. A critical requirement for self-synchronization is high quality information and shared situation awareness (Alberts and Hayes 2003). This includes shared understanding about the situation, capability and plausible futures.

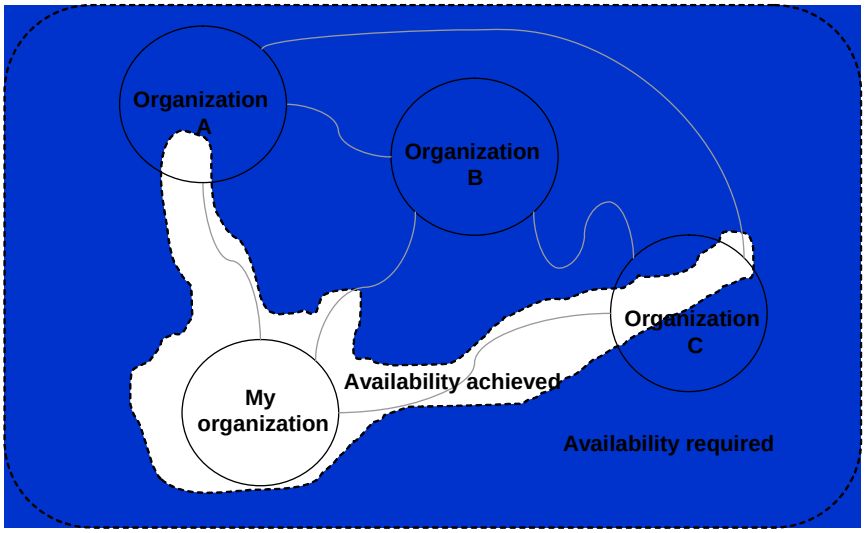


Figure 1: Information availability challenge in crisis management

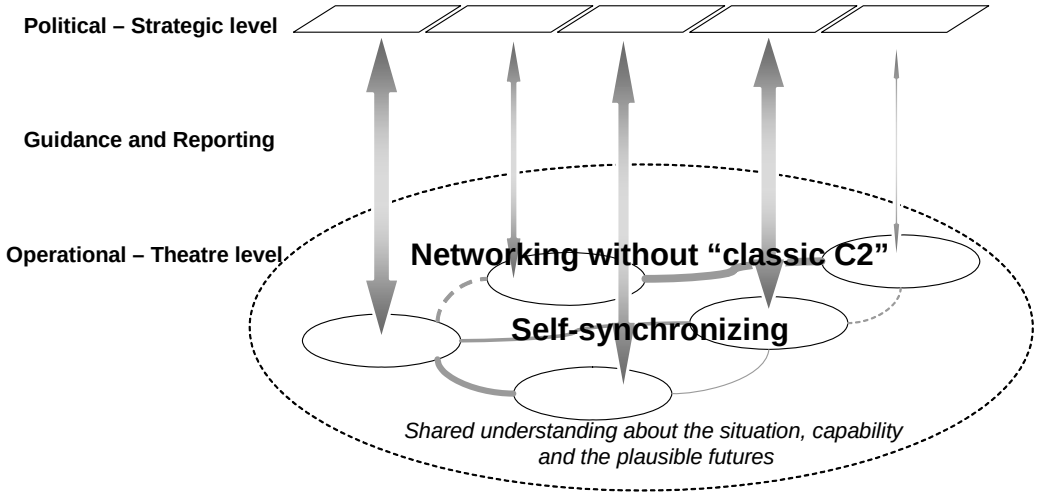


Figure 2: Organizational levels in crisis management

The organizational structures associated with crisis management are typically not designed to promote effective collaboration and information delivery across diverse organizations. Indeed, such structures are seldom designed at all. Rather, they tend to carry over from one generation to the next, with a primary focus on their internal operations, and little if any thought pertaining to interorganizational collaboration. The development, planning and implementation of crisis management activities require an *inter*organizational structure that enables a variety of diverse organizational actors to retrieve, share, understand and prioritize information quickly and effectively.

2. A general model of planning and decision-making

Kuusisto (2004) studied information and knowledge categories and processes in decision-making. He presented a general system model about planning and decision-making (Figure 3). The model contains three main categories of information—input facts, norms and conclusions—as well as information refining steps to develop a process model (Kuusisto 2006).

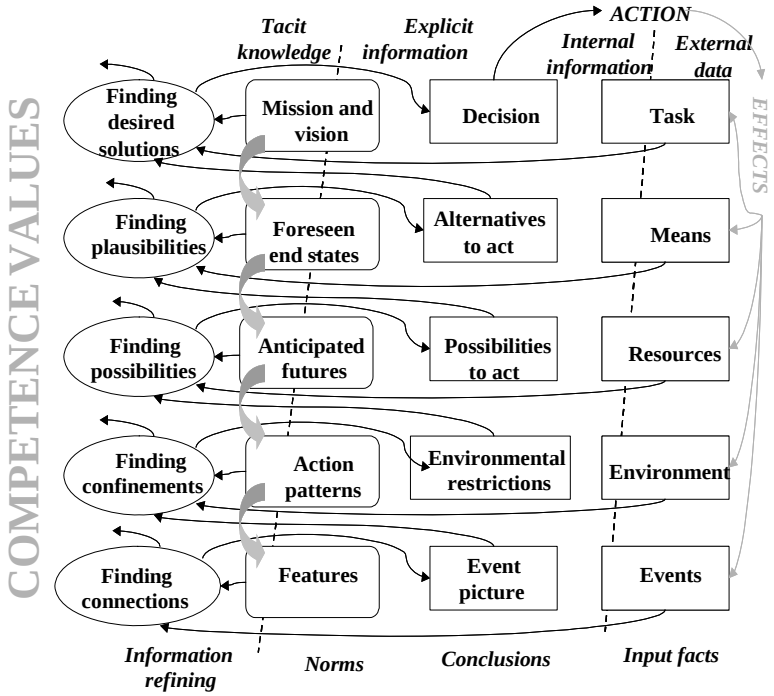


Figure 3: A general model of planning and decision-making (Kuusisto & Kuusisto 2006), based on (Kuusisto 2004, Kuusisto 2006)

As Kuusisto (2006) describes, two kinds of information flows can be detected in Figure 3. First, from right to left, incoming facts are refined through explicitly expressed information to enrich explicit and finally tacit knowledge. The second flow goes bottom-up through all five layers. Every layer has a specialized task in the overall process. The bottom layer deals with event information, and produces a continually updated picture of events to help decision-makers to find connections between events. On the next layer, the constraints are sorted out. This means the restrictions and possibilities that the environment and the action capabilities of actors have. Explicitly expressed information is the event picture and the information about environmental circumstances. Conclusions are abstracted analyses about restrictions and possibilities for an action. Mostly tacit input information is derived to develop understanding about the possibilities of the development of the overall situation. The next two layers contain information about resources and means as input facts. These input facts as well as information about events and environment and knowledge about the composition and the development of the situation and possible end-states are used as the basis for finding possibilities and plausibilities. The chain of deduction can be continued until the top, decision-making layer is reached. There, all output information from the lower layers shall be available in explicitly expressed form. Conclusions of a neighboring layer are relatively more meaningful than information on lower layers. The decision-maker must be able to know the action patterns, anticipate the change of the situation, foresee the end-state of the action, and deeply understand the meaning of the mission as a part of the bigger continuum of action.

The rough contents of the information categories are defined as follows for reference:

- Events: Describes time-limited events caused by actors. For example, meetings, accidents, hostile activity.
- Event picture: A description that enables the outlining of the pattern of a situation. For example, reports, documents, analyzed conclusions such as quality reports, statistics, pictures and maps.
- Features: Describes properties of objects such as the properties of an organization or equipment. Are stored in databases or is tacit knowledge, e.g., infrastructure descriptions, properties of equipments and competencies of people.
- Environment: Describes an area or a space that affects an actor. For example, activities of media, market trends, national trends, global trends and higher-level decisions.
- Environmental restrictions: Things that have to be concerned before planning the use of resources and means. For example, restrictions placed on activities and conditions of information acquisition.

- Action patterns: Describes how an actor can behave. Are stored on databases or is tacit knowledge, e.g., process descriptions, manuals, instructions and action plans.
- Resources: Available material and human resources such as people, financial resources, material and office space and time.
- Possibilities to act: Describes a thing, event or development that can be taught or is expected. Possibilities to act are derived from strategies.
- Anticipated futures: Describes possible paths to the goal that the actor can choose and that provide something new to the actor. For example, strategy alternatives.
- Means: Activities or methods to reach an aim or fulfill a purpose.
- Alternatives to act: Description of possibilities or proposals to act.
- Foreseen and states: Future situations most certainly reached when activities are finished.
- Task: Given activities or work to be performed. For example, activities originated by upper-level management or by the development of a situation.
- Decision: A solution based on thinking and assessment.
- Mission and vision: An end-state of the actor.

These descriptions have been used for empirical data gathering about priorities of decision-making information flows (see Kuusisto 2005, Kuusisto et al. 2005, Kuusisto 2006, Kuusisto et al. 2006a-b). Next the results of an empirical study about decision-making information profiles in time critical activities are presented.

3. Decision-making information profiles in time-critical activities

An empirical study about situation awareness in crisis management was conducted in governmental organizations in Finland in 2005. The aim of the study was to collect information for improving interorganizational collaboration services and processes of crisis management. The study focused on the changes in situation awareness when moving from normal situations to those involving disruptive events and exceptional conditions (e.g., as associated often with crisis situations). The method of the study was semi-structured interviews. 11 people representing governmental authorities were interviewed. The interviewees were active actors in the area of domestic and international security or tightly related to these actors. The theoretical basis of the study was the general model of planning and decision-making outlined above. The interviewees (Kuusisto 2005) were asked to assess changes in the priorities of information flows when moving from normal to disruptive situations with exceptional conditions. Specifically, interviewees were asked to identify both increases and decreases in the relative priorities of various information flows in crisis situations. To aid and provide consistency in identification of increases and decreases in priorities, such information flows were organized and presented according to the general model presented in Figure 3 above. Changes in priorities are reported in terms of percentage changes relative to those corresponding to normal conditions, and are summarized in Table 1. These percentage changes represent averages across the entire pool of interviewees. The results presented in Table 1 describe decision-makers' point of view on decision-making. They show that information flows that increase the most in terms of relative priority in crisis situations include the priority of events, alternatives to act and decision information. In addition, the priority of resources and deriving of possibilities to act increase slightly as well. Input facts are triggers that start decision-making activities in crisis situations. An actor's internal assumptions about futures are shown in decisions. These findings are consistent generally with the decision-making literature.

Table 1: Increase in the need to receive more accurate and reliable information by information flows when moving from normal to disruptive situations and exceptional conditions (Kuusisto 2005)

Norms		Conclusions		Input facts	
category	share	category	share	category	share
Mission, vision	**1**	Decision	**13**	Task	**3**
Foreseen end states	**6**	Alternatives	**14**	Means	**6**
Anticipated futures	**5**	Possibilities	**8**	Resources	**6**
Action patterns	**1**	Restrictions	**5**	Environment	**8**
Features	**2**	Event model	**6**	Events	**15**

Secondly, the interviewees (Kuusisto 2005) estimated how the need to speed up the update frequency of information flows changes when moving from normal to disruptive situations and exceptional conditions. The

interviewees were asked to estimate by their experience the updating frequency of the information flows in normal conditions. Then they were asked to estimate the updating frequency in a crisis situation where immediate decisions and actions are required. As in the table above, changes in update frequency are reported in terms of percentage changes relative to those corresponding to normal conditions, and are summarized in Table 2. These percentage changes represent averages across the entire pool of interviewees.

Table 2: The need to speed up the updating frequency of the contents of information flows when moving from normal to disruptive situations and exceptional conditions (Kuusisto 2005)

Norms		Conclusions		Input facts	
category	share	category	share	category	share
Mission, vision	**13**	Decision	**1**	Task	**22**
Foreseen end states	**12**	Alternatives	**16**	Means	**1**
Anticipated futures	**12**	Possibilities	**2**	Resources	**8**
Action patterns	**4**	Restrictions	**4**	Environment	**0**
Features	**1**	Event model	**2**	Events	**1**

The results presented in Table 2 describe decision-makers' point of view on decision-making also. They show that information flows that need to increase the most in terms of relative update frequency in crisis situations include task and its analysis. Typically, a given task is immediately analyzed, and targets of activities are formed based on the task. Also clear from the table, there are high temporal requirements concerning mission and vision and the assessing of anticipated futures and foreseen end-states. In a fast changing situation, scenarios about the futures of the situation are outlined continually. Moreover, information about resources has to be updated more frequently in crisis situations. Together these three topics yield a requirement to form alternatives to act more frequently in crisis situations than in normal conditions. These information flow updating priorities appear to be quite logical. When a task is received, the alternatives to act have to be formed quickly based on mission and vision, anticipated futures and foreseen end-states, as well as resource information. Most needs to speed up the updating frequency of categories are focused on these six categories. Results of Tables 1 and 2 are combined in Table 3 via simple averaging. This reflects a simple but informative, combined measure that addresses both the priority and needed update frequency associated with the various information flows. As shown in Table 3, the most significant information flow appears to be alternatives to act. Mission and vision, anticipated futures and foreseen end-states, resources and means, as well as information about tasks, are needed to form alternatives to act. This is shown in Table 3 where these information flows have high priorities. These results follow the general model and philosophy literature, from which the general model was derived.

Table 3: Increase in the need to both speed up the updating frequency and receive more accurate and reliable information when moving from normal to disruptive situations and exceptional conditions

Norms		Conclusions		Input facts	
category	share	category	share	category	share
Mission, vision	**7**	Decision	**7**	Task	**12**
Foreseen end states	**9**	Alternatives	**15**	Means	**4**
Anticipated futures	**8**	Possibilities	**5**	Resources	**7**
Action patterns	**3**	Restrictions	**5**	Environment	**4**
Features	**1**	Event model	**4**	Events	**8**

When the results of Table 1 and 2 are compared, it is noted that there are differences between the temporal requirements and content priorities of information flows in crisis situations. For instance, task and its analysis temporal requirements are emphasized in one, but their more accurate and reliable content are not prioritized. In addition, mission and vision, foreseen end states, and anticipated futures have high temporal requirements in one, but reflect quite low content priority. On the other hand, events and decisions have high position in the content priority, but their updating frequency need not to be increased when moving into crisis situations. One reason for not emphasizing the time criticality of event and environment information is that in most cases the actors cannot affect the event and environment information collecting processes. A general system about the interaction between an entity and the world is presented in Figure 4. The results presented in Table 1 have a focus on decision-making. They describe observations and the moment of making decisions. In Figure 4, the decision-makers' point of view is located mostly on perceiving and remembering areas. The results presented in Table 2 describe the forming of basis for decision-making. In Figure 4, this is mostly located on imaginating and thinking areas. Together these two views provide a system capable for acting and having an effect on the world. One can over lay the relative priorities and needed update

frequencies reported in the tables above to differentiate how various flows of data, information and knowledge associated with crisis management would affect the activities in thus figure across time.

Both the forming of basis for decision-making and decision-making itself require a wide understanding of large systems having a structure, activities and information potential. When forming the basis for decision-making, this need is visible especially in task analysis. The forming of the basis for decision-making activities is supported by future study methods suitable for situations where immediate activities are needed. Hence information about resources is a prerequisite for future studies. In addition, information about features, action patterns and mission and vision are needed in future studies. The creation of new information and knowledge represents a critical capability needed for forming the basis for decision-making. This is different from capabilities needed for decision-making itself. The former capabilities include both combining existing information and willingness to make decisions.

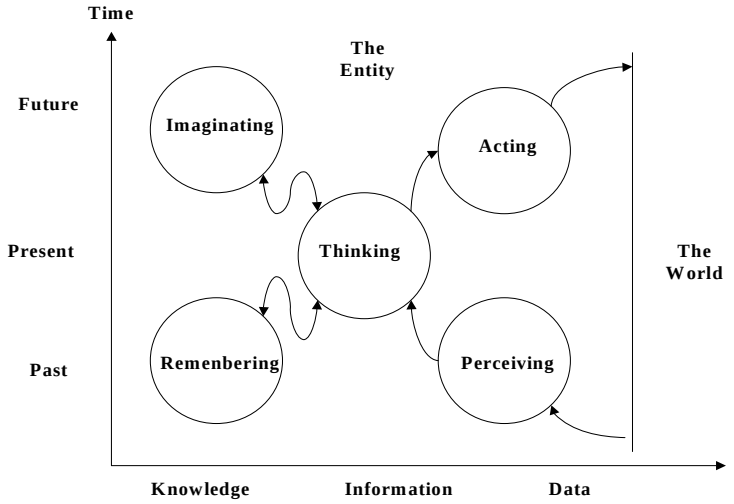


Figure 4: Interaction between the entity and the world (Kuusisto 2004), based on (Bergson 1911)

As a conclusion, the forming of basis for decisions as well as decision-making have to be supported by the following information based activities:

- analyses of the development of real-time situation
- presenting of the continually updated resource information
- practises for informing tasks immediately
- processing of future scenarios – finding of plausible development paths and foreseen end states
- realistic analysis of tasks and forming of missions
- forming, presenting and analysis of alternatives to act
- sharing of decisions

This study was based on the theoretical model outlined above, which was derived from the literature, and which provided organization and consistency to the interviews. The number of interviewees that participated in the study was limited. So, the empirical data summarized above give only rough view on the topic. However, the results appear to be logical and consistent with both theory and practice in terms of crisis management. We find it especially interesting how information flows are characterized differently depending upon whether they are evaluated in terms of relative priorities or update frequency. The development of the general planning and decision-making model will continue through future research along these lines. The model provides an information flow perspective on crisis management and situation awareness as well as interorganizational structure and processes. This perspective is novel and different from that reported in most research. It provides potential to discover novel phenomena, or too study recognized phenomena from different perspective. Both of these ways provide a potentiality to create new knowledge. The model supports research on management, management of networks, management culture, planning, and decision support information systems and data models.

4. Effects on IO planning and organization development

The findings presented above have implications for several areas of management, decision-making, planning, optimization and performing of information operations (IO). It seems that focusing simply on relative priorities of information flows is not enough when planning and conducting implementations or managing operations. Additional stress should be put on the issues concerning information flow updating possibilities and processes as well.

The results above, which focus quite generally on crisis management in an interorganizational context, apply equally well to IO.. In either case the same phenomena concerning information exchange at the level of general situational understanding exists. It seems that the sharing of information frames or certain information categories between interacting parties is beneficial whether conducting operations or activities. This can be found obvious on the basis of Habermas´s theory of communicative action (1984, 1989). He claims that to start communication, at least one common item must exist between interacting parties. Interaction and its development are based on this common item. The empirical results presented in this paper follow his statement. To conduct interaction between two or multiple actors, one or multiple common categories of information shall be present. To gain mutual understanding or to effect by information, or to protect one's own information space, interacting parties should have as common information flows. This includes both priority of the information flows and time aspects of information updating. In practice this means that information exploitation and sharing profiles should be studied to understand the complexity of these interacting situations.

This research resulted that information exploitation profiles differ from each others in the cases of making decisions and making preliminary planning work before actual decision-making. To perform interaction between these two different functions, at least one information flow should be common between those functions. In this research, in the case of crisis situation management, the alternatives to act turn out to be this common information flow. Planner product alternatives and the decision-maker choose one of them to be put into practice. So, alternatives to act information flows should be managed well, because they join planning and decision-making functions. In general this means that organizations should understand what information flows are important for the activities of the different parts and units of the organization. This can be used to produce and develop the working procedures and structures of organizations. Organization structure should support common information flows with interacting parties. Organizations' working flows should support information exchange procedures over interacting parties´ borders to assure the information flow priorities, and to take into account the temporal demands of exchanged information. Organizations should gain understanding about information exchange requirements to be able to form their existence to fulfil those challenges that they face when interacting with other organizations or actors.

Organizations should determine information exchange profiles for themselves and their counterparts to optimize interactivity. This interactivity may be internal or external. It may be directed to the directions of superior, subordinate or peer level. Those organizations or parts of organizations that are working with the same kinds of issues should have common information flows. Cooperation can be enhanced when information flow priorities and time frames of updating content are consistent across various, collaborating, interorganizational actors. It can be concluded that to develop iinterorganizational structures and their working processes, it is rather essential to find as interworking information flows. In crisis management, this means that all of the interacting actors (e.g., including military, governmental, non governmental and local actors) should have shared information flows (Figure 5). In addtion, they need technology based services supporting such information flows.

However, the development of interworking information flows is not only a technical issue, but an issue of interworking social systems phenomena as well. This means that both in IO and management cases the information flow phenomena, containing the importance of accurate priority and temporal requirements of information categories of all parties of the common operating environment should be figured out. This gives a good basis to the further development of organizations involved.

5. In conclusion

This paper considered the implications of information flow priorities for interorganizational crisis management. The paper presented the main results of a prior study that addressed the information requirements of high-level decision-making activities when moving from normal situations to those involving disruptive events and exceptional conditions. The study identified information flows that relative priority increases and information flows that relative update frequency need increases in crisis situations. To

prioritize information contents is a traditional approach to define information requirements for decision-making. This approach shows the decision-makers' point of view to information flow requirements. When considering retrieving and delivering of information in the networks of crisis management actors, the prioritizing of information contents approach shows those information flows that the actors need to share. According to the empirical study, the information flows to be shared are events, alternatives to act and decisions as well as resources and deriving of the possibilities to act. The results of the empirical study about the increase in relative information update frequency needs provide novel insights on the making of preliminary planning for decision-making in crisis management. The results show those information flows that the planners need to share. These information flows are task, alternatives to act, mission and vision, anticipated futures and foreseen end-states as well as resources. These information flows are promising to consider when planning influencing on decision-making or improving the sharing of information flows for interorganizational crisis management.

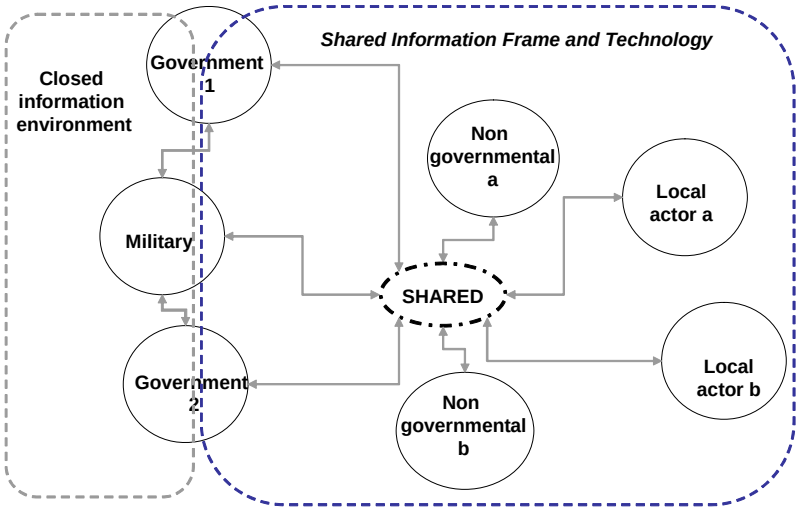


Figure 5: Actors in crisis management need shared information flows and technology environment

References

Alberts, D.S., Hayes, R. E. (2003) Power to the Edge: Command and Control in the Information Age, CCRP, USA.

Bergson, H. (1911) Creative Evolution, University Press of America.

DoD (2005) Department of Defence Dictionary of Military Terms, http://www.dtic.mil/doctrine/jel/doddict/, visited September 15, 2005.

Habermas, J. (1984) The Theory of Communicative Action, Volume 1: Reason and the Rationalization of Society, translated by Thomas McCarthy, Beacon Press, Boston.

Habermas, J. (1989) The Theory of Communicative Action, Volume 2: Lifeworld and System: A Critique of Functionalist Reason, translated by Thomas McCarthy, Beacon Press, Boston.

Kuusisto, R. (2004) Aspects on Availability, A Teleological adventure of information in the lifeworld, Edita Prima Oy, Helsinki. 124p.

Kuusisto, R. (2005) Availability of information in fast decision-making, Science and Weapon, number 63, 2005, in Finnish.

Kuusisto, R. (2006) "Flowing of Information in Decision Systems", Proc of HICSS-39, USA.

Kuusisto, T., Kuusisto, R., Yliniemi, T. (2005) Information Needs of Strategic Level Decision-Makers in Crisis Situations. Proc. of the 4th European Conference on Information Warfare and Security. Glamorgan, UK, 2005, pp. 187-194

Kuusisto, T., Kuusisto, R., Lund, O.-P., Mikkola, T. (2006a) Information Flow Aspect on the High-Level Tactical Managing. Proc. of International Conference on i-Warfare and Security, USA, pp. 131-136

Kuusisto, T., Kuusisto, R., Seppänen, H. (2006b) "Geographic Information Management in Brigade Level Decision-Making". Proc. of the 5th European Conference on Information Warfare and Security. Helsinki, Finland, pp. 151-159

Thinking Strategically about Information Operations

Irving Lachow and Robert Miller
National Defence University, Washington, DC, USA
lachowi@ndu.edu
millerr@ndu.edu

Abstract: There is widespread agreement that the Global War on Terrorism is, at least in part, a "war of ideas." There is also a consensus among both academics and policymakers that the United States has done a poor job in fighting this war. While the U.S. has struggled to develop and implement a coherent strategy for influencing key target audiences, Islamic extremists have effectively used modern communications technologies like the Internet to inspire, frighten, and manipulate target audiences, media organisations, and even entire countries. However, their heavy reliance on technologies like the Internet also makes extremist groups vulnerable to countermeasures. This paper explores general steps the U.S. can take to exploit the vulnerabilities found in the current information operations (IO) approaches being followed by Islamic extremists. It also examines options at the physical, informational, and cognitive "layers" of the information environment and describes advantages and disadvantages for each. The result is a recommended series of actions the U.S. can pursue to further its own strategic IO goals while degrading the IO efforts of its adversaries.

Keywords: Information operations, influence, extremists, terrorism, strategic.

1. Introduction

It is becoming clear that the information age has significantly changed traditional calculations of national power and national strategy. The old verities and assumptions of the past—that nation-states are the only significant actors on the international stage, that there is a clear distinction between military and diplomatic forms of influence, and so on—began to weaken during the Cold War. The global environment of the 21st century, with its rapid spread of news and ideas, integrated economies and world-wide audiences, has further attenuated these traditional views. These changes operate at both the strategic and operational levels. For example, the current crop of Islamic extremists uses the Internet along with cheap global travel to propagate their doctrines, and also as a fast and efficient way to communicate with each other. Combined with their ready access to cheap and effective weaponry, they have achieved strategic agility and operational flexibility without the need to develop their own infrastructure. They have also, to a limited but real extent, managed to gain an edge in the so-called "war of ideas." Ideas are the weapons of choice for the militarily weak, and history shows that they are often decisive. The pen—or the keyboard—can certainly be mightier than the sword if wielded effectively. Despite the many advantages that extremists gain by using the Internet, there are also disadvantages associated with this strategy. This paper explores the general steps that the United States (U.S.) can take to exploit the vulnerabilities found in the current information operations (IO) approaches being followed by Islamic extremists. It examines options at the physical, informational, and cognitive "layers" of the information environment (as defined by Joint Publication 3-13) and describes the advantages and disadvantages of each. The result is a recommended series of actions the U.S. can pursue to further its own strategic IO goals while degrading the IO efforts of its adversaries.

2. Extremist IO strategy: An overview

The IO strategy followed by extremists is primarily centred on use of the Internet (Lachow and Richardson, 2007). The Internet has five characteristics that make it an ideal tool for these organisations. First, it enables rapid communications. People can hold conversations in real-time using instant messaging or web forums. Instructions, intelligence information, and even funds can be sent and received in a manner of seconds via email. Second, using the Internet is a low-cost proposition. Extremist organisations can now affordably duplicate many of the capabilities needed by modern militaries, government organisations, and businesses: a communications infrastructure, an intelligence gathering operation, a training system, a media-savvy public affairs presence. Third, the ubiquity of the Internet means that small extremist groups can have a global cyber presence that rivals that of much larger organisations. Not only can their members communicate with each other from almost anywhere in the world, but a small terrorist cell may create a web site that is viewed by millions of people and possibly even examined daily by media outlets for news stories. For example, many news services reported the 2004 beheading video of Nick Berg, which was originally uploaded on the website of the militant Islamist group Muntada al-Ansar – a group associated with Al Qaeda. The popularity of the video can be inferred from the fact that the term "Nick Berg" was the second most popular search term in Google in May 2004 (Weimann 2006: 110). Fourth, the growth in bandwidth combined with development of new software has enabled unsophisticated users to develop and disseminate complex information via the Internet. For example, Rozen (in Weimann 2006: 126-127) has noted that "In December 2004, a militant

Islamic chat room posted a twenty-six-minute video clip with instructions on how to assemble a suicide bomb vest, along with a taped demonstration of its use on a model of a bus filled with passengers." Finally, modern encryption technologies allow Internet users to surf the web, transfer funds, and communicate anonymously – a serious (though not insurmountable) impediment to intelligence and law enforcement organisations trying to find, track, and catch terrorists. To do this, extremists can download various types of easy-to-use computer security software (some of which is commercial and some of which is freely available) or register for anonymous email accounts from providers like Hotmail or Yahoo!.

This combination of characteristics makes the Internet a valuable strategic asset for extremists. In fact, one could argue that the Internet, along with other modern communications technologies, is a *sine qua non* for modern global extremist movements. As noted by Sageman (2004: 158-161) successful extremism requires the transformation of interested outsiders into dedicated insiders. Once someone has become an insider, less intense but still continuous interactions are required to maintain the needed level of commitment to the cause. Before the advent of advanced communications technologies, this process was largely based on face-to-face interactions, which limited the scope of a given group. However, the Internet allows groups to create and identify dedicated insiders, and to maintain favour in those already dedicated to the cause, on a global scale. They also allow the extremists to deliver well-coordinated propaganda campaigns that increase the levels of support among the general public, which in turn allows the terrorists to operate freely in these societies. For example, according to Brachman (2006: 160) one of Al Qaeda's goals is to use the Internet to create "resistance blockades" in order to prevent Western ideas from "further corrupting Islamic institutions, organisations, and ideas." One technique they use is to distribute Internet browsers that have been designed to filter out content from undesirable sources (e.g. Western media) without the users' knowledge. In summary, the development and proliferation of the Internet has enabled the rise of loose, decentralised networks of extremists all working towards a common goal. Although their use of the Internet is at times tactical, the overall thrust of the extremist movements is strategic. In the words Brachman (2006: 150): "…it is the strategic—not operational—objectives of the jihadi movement's use of technology that engenders the most enduring and lethal threat to the United States over the long term."

3. Analysis of US IO options

This section explores different strategies that the United States can pursue to counter extremist IO strategies. It begins by examining the strengths and weaknesses of targeting the three components of the "information environment" as defined by the Department of Defence: physical (infrastructure), information (content), and cognition (perceiving and deciding).

3.1 Physical infrastructure

One approach to counter extremist use of the Internet is to target their communications infrastructure to either deny or disrupt their ability to communicate and/or maintain an Internet presence. While there are some benefits to this approach, there are also significant challenges. The vast majority of extremist organisations depend on commercially owned infrastructure for their communications needs. Most of that infrastructure, especially the elements that provide web-based services, are hosted in either the United States or Europe. As a result, a strategy of countering extremist activities by attacking their infrastructures would require the U.S. to target either itself or its allies. There may be cases where the infrastructure in question is owned and/or operated by company that resides in a country that is not allied with the U.S. In such cases, a direct attack (either via physical or cyber means) on the targeted information infrastructure might prove to be useful in certain circumstances. The U.S. would have to weigh the potential benefits of such an attack with the political and military risks associated with a potential act of war against another sovereign nation. Another policy might be for the U.S. government to ask infrastructure providers to identify extremist clients and selectively terminate or disrupt their activities. Unfortunately, this is harder than it sounds. Extremists often pose as legitimate companies or use false information to register for accounts. They also tend to move between providers on a regular basis. In the cases where infrastructure providers can identify extremists that are using their services, it might make more operational sense for the U.S. to monitor and/or eavesdrop on the extremists rather than just shut them down (in which case they would simply move to another provider). This approach—which could be called "tolerate, monitor and exploit" — could provide the U.S. with valuable intelligence. It could also open the door to the planting of disinformation. This approach will be discussed in greater detail below.

3.2 Information content

A second approach to countering extremist information operations is to target the content component of the information environment. The goal here is to affect one or more of the three components of information assurance: confidentiality, integrity, and availability. By attacking the confidentiality of information, the U.S. would deny the terrorists the ability to communicate securely. This could be accomplished by using wiretaps, breaking encryption algorithms, or using undercover agents to gain access to secure chat rooms and web sites (these sites are usually password protected and their locations are revealed only to trusted members via secure email or other covert means). Although there are some challenges inherent in this approach (see below), Harden (2006) has documented one case of a U.S. citizen successfully posing as a terrorist recruit and using that access to help catch several potential terrorists. By attacking the integrity of information, the U.S. would be able to do two things: secretly plant misleading information in order to get the extremists to take desired actions; and openly reveal that they had compromised critical information (e.g. by hacking into databases or websites) in order to raise doubts in the minds of the terrorists about the validity of all of their content. Finally, by attacking availability, the U.S. would deny the extremists effective and timely access to their information. This could be accomplished in numerous ways, including both physical and cyber means such as denial-of-service attacks. Given their heavy reliance on the Internet, limiting terrorist access to the content available via that medium alone (including but not limited to: training information, intelligence information about possible targets, command and control messages, fundraising and recruiting websites) would limit the effectiveness of these groups. While attacking extremist content is generally easier than taking out their physical infrastructure, this strategy still requires the U.S. to overcome several challenges. One problem described previously is that terrorist websites, and the servers that host their content, are often difficult to find. A related challenge is that the number of relevant websites is growing extremely quickly. Thus, significant resources would be required to keep track of the tremendous amount of extremist content appearing on-line. This also means that compromising some data, or denying access to a few websites or databases may have a small impact on the overall extremist movement. A final problem is that terrorists tend to use secure chat rooms and encrypted emails to transmit critical pieces of information. For example, they like to sign up for free anonymous email accounts provided by companies like Yahoo! and Microsoft. This allows them to use public computers at any location (e.g. a cyber café) to communicate. It is much harder to find and disrupt, alter, or deny these messages than it is to track websites, which is itself a challenge.

3.3 Cognition

A third approach to counter extremist information operations is to focus on the cognitive domain rather than on either the infrastructure or the content per se. In this case, the goal is to influence how people perceive information and/or how they make decisions (this is often referred to as “perception management”). Although achieving this goal relies on the use and manipulation of content, influencing perceptions also requires the consideration of factors such as language, culture, values, and context. This approach is often associated with activities such as psychological operations, public affairs, and public diplomacy. There are a number of advantages associated with perception management. First, it can be used to reduce the legitimacy and attractiveness of the extremist movement. This, in turn, would have a cascading effect on the ability of terrorist groups to recruit, raise funds, maintain operational security, influence the media, and operate training bases. Perception management can also be used to influence allies and non-aligned parties in order to build support for U.S. policies and actions targeted against these groups. Another benefit of using perception management, especially the public diplomacy component, is that it can help the U.S. spread its values around the world. This reduces the likelihood of military conflict and improves the chances for beneficial economic relations, both of which reduce factors that can contribute to the success of terrorist groups. There are two significant challenges facing the U.S. in the area of perception management. The first is that the current ability of the United States to operate a well-coordinated, government-wide strategic perception management campaign is limited. Shortcomings in the area of public diplomacy have been well documented (see References for two examples from the General Accountability Office), but many efforts on the military side have also come up short. Developing a robust strategic perception management capability will require time and resources at the agency level (specifically at the State Department and Department of Defence), and an effective interagency process for the development and coordination of coherent themes and messages—a significant challenge given the government’s current structure. Secretary of Defence Donald Rumsfeld (Rumsfeld, 2006) admitted as much when he said: “If I were rating, I would say we probably deserve a D or D+ as a country as how well we’re doing in the battle of ideas that’s taking place….So we’re going to have to find better ways to do it and thus far we haven’t as a government. The government’s not well organised to do it.”

Finally, perception management campaigns, even ones that are well funded and organised, take many years to reach fruition. Changing how people think is not easy; it may take a generation or more. The United States needs to take a long-term view of the problem, much as it did during the Cold War. Unfortunately, such long-term thinking is rare in the current political climate because incentives work against spending money now to achieve benefits that will not accrue for years or even decades.

4. Strategic information operations: An integrated approach

There are clear strengths and weaknesses associated with actions that target each element of the information environment. Depending on the specifics of a given situation, one of these strategies might prove dominant. However, it is also evident that, in many circumstances, an integrated all-forms approach to information operations appears to be the best bet. In this connection it is worth pondering the difference between strategy and tactics. The Department of Defence (2006a: 520) officially defines "strategy" as: "A prudent idea or set of ideas for employing the instruments of national power in a synchronised and integrated fashion to achieve theatre, national, and/or multinational objectives." Thus, strategy is designed to marshal all assets in a long-term, coordinated effort to achieve a lasting, sustainable success. Operational and tactical efforts contribute to strategic success, but are not the same thing. The obvious question is how to exploit the strategic weaknesses of the extremist movement cited above through a sustained use of information operations. Individual IO efforts and exploits, however valuable, should be seen through this strategic lens. This kind of strategic effort, which will require a changing mix of short-, mid-, and long-term operations, will also require careful coordination. It is not clear whether the current command structure for government-wide U.S. information operations, which is overwhelmingly dominated by the Department of Defence and often seems focused on tactical concerns, is the optimal way to organise the resources needed for such a long-term, sustained effort. What would a strategic, offensive IO campaign targeting extremists look like? Most likely it would include a mix of approaches focused all on three components of the information environment. Some plausible steps are listed below:

- Weaken the credibility of extremist leadership and doctrines. This can be done in a variety of ways. For example, one can leverage values that are grounded in Islamic theory and traditions, such as honour, to emphasise peaceful ways to achieve political ends [for an interesting discussion of the role of honour in Islam see Ahmed (2003)]. The United States can also help to publicise the many respected voices within Islam that oppose violent extremism.
- Confuse and confound the operational plans of extremists. This would involve a wide range of operations carried out by the military, intelligence community, and law enforcement agencies.
- To the greatest extent possible, sow fear, uncertainty and doubt among extremist leaders and their adherents. There are many ways to do this, ranging from psychological operations and infiltration of groups to misdirection and the planting of false information and orders.
- Where appropriate, disrupt the information infrastructure used by extremists to reduce their operational effectiveness. This may require careful coordination with allies and/or private sector service providers. A corollary is to work with these infrastructure providers in order to gain access to extremist communications.

Clearly, each of these suggestions carries advantages and disadvantages, as described previously. The critical foundation of a successful IO campaign against extremist foes is to think strategically and act in a coordinated fashion. This will require an impetus from the White House and a robust inter-agency process run by the National Security Council. There have been efforts to manage strategic communications in this way, but they have proven ineffective. In any event, those efforts focused primarily on one aspect of the information environment. The same is true of current efforts to coordinate psychological operations (which generally fall under the purview of the military and intelligence community): while they are certainly necessary and helpful, these efforts are not designed to develop, coordinate and oversee a comprehensive information strategy for the nation. Unless and until the highest levels of leadership in the U.S. government make strategic information operations a priority, the U.S. will be at a disadvantage vis-à-vis its enemies.

References

Ahmed, A. (2003) Islam Under Seige, Cambridge, UK: Policy Press.

Brachman, J. (2006) "High-Tech Terror: Al-Qaeda's Use of New Technology," The Fletcher Forum of World Affairs, Volume 30:2, Summer, pp. 149-164.

Department of Defence (2006), Information Operations, Joint Publication 3-13, Washington, DC: Department of Defence.

Department of Defence (2006a), Department of Defence Dictionary of Military and Associated Terms, Joint Publication 1-02, Washington, DC: Department of Defence.

Government Accountability Office (2005), U.S. Public Diplomacy: Interagency Coordination Efforts Hampered by the Lack of a National Communication Strategy, GAO-05-323, Washington, DC: Government Accountability Office.

Government Accountability Office (2006), U.S. Public Diplomacy: State Department Efforts Lack Certain Communication Elements and Face Persistent Challenges, GAO-06-707T, Washington, DC: Government Accountability Office.

Harden, B. (2006) "In Montana, Casting a Web for Terrorists," Washington Post, 4 June 4, p. A03.

Lachow I. and Richardson, C. (2007) "Terrorist Use of the Internet: The Real Story," Joint Forces Quarterly, Issue 45, April, forthcoming.

Rozen, L. (2003), "Forums Point the Way to Jihad," Wired News [Online], Available: http://www.wired.com/news/conflict/0,2100,59897,00.html

Rumsfeld, D. (2006), Remarks by Secretary Rumsfeld at the Army War College, Carlisle Barracks, PA, March 27, 2006, [Online], Available: http://www.defenselink.mil

Sageman, M. (2004) Understanding Terror Networks, Philadelphia: University of Pennsylvania Press.

Weimann, G. (2006) Terror on the Internet: The New Arena, the New Challenges, United States Institute of Peace Press, Washington, DC.

Intrusion Detection in open Source Software via Dynamic Aspects

William Mahoney and William Sousan
University of Nebraska at Omaha, NE, USA
wmahoney@unomaha.edu
wsousan@unomaha.edu

Abstract: Aspect-Oriented Programming (AOP) is an emerging software engineering methodology, which has been used to assist in the removal of crosscutting concerns from traditional methods of software development. As an example, software used to determine whether a user has appropriate security clearance might be scattered throughout the many modules, which require this check. Utilising AOP, “aspects” are “woven” into the software either in a “static” method, during compilation, or a “dynamic” method while the program is executing. The “join points” in a program are the points where these aspects are applied. The “aspect” code is written once and “woven” in to the modules at join points. Typical aspects involve logging changes to a database and monitoring memory usage. Our focus is on aspects related to security and intrusion incident detection.

Dynamic weaving allows aspects to be woven in and out as the program is executing. However the base code often must be compiled with additional “syntactic sugar” – additions that are required for the later connection of dynamic aspects. This paper presents a new technique to enable dynamically loaded security modules to be added into existing C/C++ code on the fly while the program is executing. Our tool is a Run-Time Event Monitoring System called “dynamicHook”, implemented on a standard Linux platform using existing Linux tools, which tests each potential join point for the required activation of advice. Our system does not need to modify the executable files, but instead we compile in special “linkage” between the base code and potential aspects which are then called as dynamically linked routines located in shared libraries. Our scheme does not require any new syntax or language extensions or rely on code transformations; we thus use it for adding intrusion detection methodologies to pre-existing off-the-shelf open source software.

Keywords: Open-source, intrusion detection, dynamic aspects, AOP.

1. Introduction and objectives

Aspect-oriented programming (AOP) is a method used to apply “advice” in the form of “aspects” across various portions of software. The idea behind AOP is to allow one to avoid the need to build in various checks, many of which are the same or similar, in modules which are potentially scattered throughout a large software project. As an example, Sommerville (2007) describes a hospital patient information system. The functionality needs to be changed so that it is necessary to re-authorise the user whenever a database update is made. Each function updating the database starts with the word “update”. Aspect-oriented programming allows one to apply the security enhancement to all functions, which match a certain criteria – those that start with “update”, for example – thus avoiding the more involved, and potentially buggy manual approach. Aspect-oriented programming systems often require what compiler specialists call “syntactic sugar” – additions to the normal syntax of the language in order to indicate the locations of the possible join points. Thus it is not applicable for large open-source software unless someone is willing to augment the code with the necessary join points. But a useful and exciting possibility for AOP is to add enhanced security to pre-existing open-source software. One could take an open-source software package and apply dynamic aspects for intrusion detection. Thus, adding dynamic aspects for instruction detection to existing C/C++ software is the point of this research. Our objectives are:

- Allow for existing C/C++ software to be recompiled in order to support dynamic aspects with minimal (no) changes to the source.
- No external “dynamic aspect manager” should be required.
- Provide support for dynamic aspects to be loaded and executed by the running process using a simple command line interface.
- We wish to use an existing tool chain. In our case this tool chain will be a Linux system and compiler.

- Allow the join points to be adapted to meet new requirements as the process is executing, by utilising dynamic aspects, which are instantiated at run time.
- Allow the user to apply advice aspects at both function entry and exit within the process, thereby providing intrusion checks, which may occur before, after, or around the function.

Our main focus of this research is the demonstration and description of a method, using available tools, of providing dynamic aspects tailored for intrusion detection in open-source code. The remainder of this document is organised as follows: In section 2 we introduce the reader to AOP concepts and terminology. In section 3 we review existing AOP systems. Section 4 introduces the reader to our AOP tool named "dynamicHook". Since our objective was to use AOP techniques for intrusion detection, the aspects we have written are detailed in section 5. Section 6 describes the performance penalty due to our techniques. Future work is listed in section 7, and conclusions in section 8.

2. Background

Complex software systems may contain millions of lines, thousands of functions, and numerous features. Thus large systems contain non-functional properties spread throughout the entire system. Security, logging, and error handling are some examples of aspects that may crosscut multiple modules. To aid in the correction of this problem, Aspect-oriented programming Kiczales (1997) was introduced as a means for modularisation of crosscutting concerns. By writing software specific to these concerns and encapsulating this code into separate modules, AOP creates software systems with increased maintainability, flexibility, and adaptability. These modules, the "aspects", are inserted at specific points called "join points" (Sommerville 2007) within the base code through a process termed "weaving". The advice code can be called before, after, or around the join point. So-called "pointcuts" are used to describe which advices are applied to what join points. Several aspect languages provide a pointcut language to facilitate a flexible system for describing the join points for a given aspect or aspect group. Aspects can be implemented either via a "static" means, when the program is compiled, or in a "dynamic" mode as the program is running. Static aspects tend to be more efficient, since the aspects can be woven into the code and optimised in advance. But any time different or modified aspects have to be applied, the base code must be recompiled. Dynamic aspects are more flexible - they can possibly change on the fly - but they are slower due to the need to insert and remove them as the system is executing. Dynamic aspects reduce the need of having to re-compile, install, and deploy the new system, and provide a method for rapid prototyping. It also promotes systems that can adapt during runtime to changing configurations, and environments.

3. Current methods

Java and C/C++ are the primary languages used in research on AOP methods. Multiple Java based systems have been developed that support dynamic weaving such as Popovici (2003), Sato (2003), Bonér (2004), Bockisch (2004), Vanderperren (2005), and Aussmann (2003). AspectC++ Spinczyk (2005) and DAO C++ Almajali (2004) are the main contenders for C++. The C language systems that offer dynamic weaving support are TinyC2 Zhang (2003), Toskana ,Arachne Douence (2005). AspectC++ and AspectJ (Kiczales 2001), both contain aspect extensions for describing join points and advice. AspectJ supports static time weaving as well as dynamic weaving for Java. Arachne is based on runtime weaving into the binary code and includes an extension to C for the creation of advice code, a joint point model, and a pointcut language. Toskana (Engel 2005) is an AOP toolkit, which focuses on dynamic aspects within an operating system kernel. It provides for before, after, and around advice for in-kernel functions, pointcut specifications, and the implementation of aspects as kernel modules. TinyC2 is a source-to-source translator. The programmer uses additional notation for the implementation of aspectual units known as "snippets". This is accomplished by inserting code directly into the target address space, and then jumping to this new code. The system has considerable overhead in its dynamic weaving process.

4. "DynamicHook"

Our focus is on the implementation of dynamic aspects for intrusion detection. With respect to dynamic aspects we note that there are three categories (Almajali 2006): Pre-run-time

instrumentation, run-time event monitoring, and run-time weaving. Instrumentation involves the activity of either weaving the code before the program is executed (by the use of a pre-processor), or during the loading of the application, or finally the use of just-in-time compilers for byte code manipulation. Event monitoring works by checking possible join points, at execution time, to determine if an aspect is to be applied at that join point. Run-time weaving actually weaves and un-weaves the aspects during program execution. Our system, "dynamicHook", is a run-time event monitoring system that tests each possible join point for the required activation of advice code. As a result, there is no need to modify executable files. Rather, we adjust the "linkage" between the base code and aspects through the use of dynamically linked functions located in shared libraries. Our interface does not require the use of any new syntax for aspect language extensions. We offer instead a utility for modifying a shared memory block that is scanned at each potential join point to determine if an aspect should be executed. The platform we use to implement our system is Linux, with the GCC compiler suite. GCC is utilised in order to have the original code make calls to the "dynamicHook" functions. The "dynamicHook" interface accepts a set of requests, including the names of dynamic modules to call, acts upon these requests.

4.1 The "dynamicHook" interface

In order to utilise "dynamicHook", it is not necessary to change the open-source or original source code; a recompile is necessary, but no additional "start up" code need be inserted into the program. Our interface is linked into the program at build time and adds initialisation code to set up an area of shared memory local to the process. This region is used to facilitate communication between the process as it is executing, and the requests to load and call various aspects. The shared memory area is created using the standard system calls for manipulating these objects so that no additional libraries are necessary. The organisation of this shared memory area is as follows:

- A header section contains housekeeping information including the number of elements in the table.
- A large table of structures is included; this table contains the data necessary for a particular request, including what the request consists of, the module to load (if any) for a dynamic aspect, and what function to use from that module.

During program execution, the executing process monitors this area for requests using event-monitoring calls at the prologue and epilogue of each function. This code includes the address where the call came from, and the address of the function that is being called. This data allows us to determine whether this function is a join point, and if so, what aspects apply at that particular join point. Each shared memory table element contains the name of the module and the aspect function, which may be called at that join point. The aspects are stored in a shared object (Linux ".so") files. These files can thus be organised by function – here's a file containing all the intrusion detection aspects, another that contains all of the database security calls, etc. In addition, there are a few aspects, which are self-contained within the "dynamicHook" interface itself. These include the ability to trace program execution, similar to our original work reported at the ICIW conference in 2006 (Mahoney 2006).

4.2 Pointcut expression tool

When the user desires to insert a dynamic aspect into the executing program, our command line interface is a front-end for manipulating the shared memory table. Specifically it inserts into the request table, at the correct place, the filename, function name, and an indicator to begin applying the aspect. When that table entry is examined by the instrumentation function, the shared object file is loaded (the first time only), the aspect is located within the file, and from that point onward it is called automatically. Our work includes a command line interface for inserting the aspects – a pointcut expression tool called "joinPoint". For example, a dynamic aspect is woven into the executing program by declaring the name of the module to load, and the name of the function to use within the module:

```
./joinPoint ./prog func dynamic ./intrude.so checkUID
Or:
./joinPoint ./prog -func dynamic ./intrude.so checkUID
```

The minus sign in advance of the function name indicates the aspect is to be applied to the end of the function. A supplied aspect in the "dynamicHook" interface is the ability to trace the execution of the program. In the case execution tracing, one can enable the trace at the end of a function and disable it at the beginning, thus tracing the functionality of the program in areas other than the function. Obviously different areas of the program can be independently traced, have dynamic aspects applied, and so forth. It is necessary to support a pointcut expression syntax so that aspects can be applied to multiple locations simultaneously. Recall the example of the hospital database; the user wished to apply the security aspect to all functions, which start with the word "update". To accomplish this, "joinPoint" uses an expression syntax based on regular expressions that is similar to the Linux shell "globbing" function. For instance:

./joinPoint ./prog "update*" dynamic ./intrusion.so reAuthorise

The string "update*" matches all functions in the executing program that have a function name beginning with the string "update", just as the Linux shell would expand this to include all filenames starting with "update". Since "dynamicHook" supports C and C++, The usual C++ scope resolution notation is supported as well:

./joinPoint ./prog "-myClass::func*" dynamic ./intrusion.so checkExit

will weave in the function "checkExit" in the module "intrusion.so" to the end of any function named "func" in the C++ class "myClass".

5. Implementation of intrusion detection aspects

The desire of this research, once the aspects are implemented, is to utilise dynamic aspects in order to detect intrusion in common open-source software. Our approach is twofold:

- To detect the intrusion as it is happening by inserting aspects into the executing program
- To allow logging to an external resource such as a process monitor or traditional intrusion detection heuristic or statistical package.

We have examined several methods used by hackers and crafted aspects for each. We tested these aspects with two open-source web servers: "thttpd" (Poskanzer 2006) which we have used previously for other security experiments, and "monkey" (Silva 2006). Both have the advantage that they are simple to configure and test, but as internet applications they would be representative targets for an assault against a computer system. As a point of reference the "thttpd" platform consists of approximately 10,800 lines of source code and 200 functions. "monkey" is smaller, with only 5,300 lines and just over 100 functions. We also wrote a custom application to demonstrate a stack overflow attack, since there were no known intrusion possibilities of this type in the open source we selected. We implemented three aspects and wove these in at appropriate join points in the executing servers. The three aspects were chosen as being representative of either methods currently in use to break into a system, or methods we could use to easily detect the break-in.

5.1 An aspect to detect stack overflow intrusion attempts

The stack overflow and buffer overflow attacks account for roughly half of all security vulnerabilities (Cowan 2000). In order to implement an intrusion using this method, first one injects attack code, often a small sequence of instructions that spawns a shell process, and second, one must change or alter the execution path of the program in order to execute the injected code. The intruder can force a snippet of executable instructions onto the program stack, as well as a return address, which vectors the execution to this code. We will refer the reader to a recent communications of the ACM article in (Kuperman 2005) for technical details. We also note that there have been numerous attempts to curtail this intrusion method; a few of these are stackguard (Cowan 1998), return address defense (RAD) (Chiueh 2001) (Prasad 2003), static detection methods in (Larochelle 2001), and the "libverify" project (Baratloo 2000). Our method for detection of intrusions is based on the observation that for most functions in an executing process (except a few special cases), following all of the return addresses should lead back to "main" where the program started. Thus,

an aspect, which is added at the epilogue of a function, can check this return code path prior to ending the function. We implemented an aspect named “checkStack”. This aspect is aware of the original “call site” to main – the location in the C/C++ startup code where the “main” function was initially invoked. It is also aware of the stack frame layout generated by the compiler. And finally it is aware of two special symbols available in the process; all of the instructions in an executing process will be between these two symbols. Consider the following scenario: there is an alert posted on a site such as US-CERT indicating that your web server has a potential stack overflow intrusion attack in function “get_input”. Since you opted to include “dynamicHook” when the project was compiled, you add this aspect dynamically to the process:

./joinPoint ./monkey ”get_input” dynamic ./intrusion.so checkStack

and at that point you are protected until the patch or update comes along.

We have written a simple test program, which facilitates a denial of service attack. A certain function optionally places an infinite loop on the stack and returns to it. The program, when linked with “dynamicHook”, is able to detect this overflow attack at the point where the program leaves the test function. This detection is reported using methods outlined below.

5.2 An aspect to detect pre-loaded library intrusions

Executable images on Linux and Windows include dynamically linked libraries. In Windows parlance these are the familiar “DLLs” and in Linux they are “SOs”. As a program on Linux is loaded into memory, the linker modifies various calls and references to these libraries so that the program runs correctly. There is a collection of these standard libraries that include all of the common C/C++ functions. An environment variable is available which, when set, overrides the default actions of the linker. This variable is called “LD_PRELOAD”. It allows one to specify a dynamic library that is to be loaded in advance of the standard runtimes. This also gives the opportunity of linking in a man-in-the-middle attack between the program and the needed libraries. As an example, suppose there is a shared library with a different version of the “crypt” function. Also suppose that LD_PRELOAD is set so as to load this library in advance of the standard version. Executing a program, which asks for a password, could now save the clear text version prior to encryption. For an example of this behaviour with actual source code, see (Isik 2005). Because of these security concerns, certain program permission settings cause the loader to ignore the flag unless the user is the super user. However, if an intrusion is made into a system by other means, setting this flag on scripts that start up system processes will leave that system vulnerable to additional intrusions in the future, as these system services are often started up from a position of privilege. Thus, a simple dynamic aspect we have implemented and tested is to check whether LD_PRELOAD is set at certain points in the process. This particular aspect makes the most sense when applied to the beginning of “main” but could of course be woven in at any point in the process:

./joinPoint ./monkey ”main” dynamic ./intrusion.so checkPreload

If the LD_PRELOAD variable has a value, our dynamic aspect notifies the external intrusion detection monitoring facilities and exits the program.

5.3 An aspect to monitor the user identification of the process

The simplest of our intrusion detection methods we have outlined last. When a process, particularly an internet server such as our web servers, is executing, it will have a certain user identification associated with it. For example a Linux process will have the real user-ID, the effective user-ID and real and effective group-IDs as well. Since the identifications we expect to see are known, it is a simple matter to design a dynamic aspect, which verifies the values for all four-process variables. We have implemented these by verifying that the identifications match what is expected for our test environment; however it is a simple matter to include software to read these from a configuration file, for example, and no recompilation of the aspects is even necessary. For example:

./joinPoint ./thttpd ”tmr_run” dynamic ./intrusion.so checkUID

Since this particular open-source software periodically calls the function “tmr_run”, any unexpected changes to the owner of the process will be caught within a few seconds.

5.4 Notification of intrusion

Our dynamic aspects write to an inter-process communications device, a UNIX-style named pipe. There is a program, which listens for data on this pipe and generates a report in real-time of any messages that the aspects need to pass along. The file handle for this interface is a global variable within the “dynamicHook” code, which makes it accessible to any aspects which happen to be loaded during run time and which need to log events. We outlined the fact that the “dynamicHook” system does not require an external process for aspect weaving; this is still true, as our external program is simply present to report the incidents. Also, this is not even necessary as the aspects might instead log the intrusion events via a standard interface, such as the Linux syslog server. Regardless, the aspects we have tested for intrusion detection generate a snippet of data, a “message”, which is passed to the external process via the inter-process communications channel. Other aspects generate other data types, such as “trace”. For our testing purposes the monitoring process prints these on the console as the process executes. In the case of a “real” intrusion detection system, the data would be available in real time to whatever pattern recognition software makes up the processing portion of the detection scheme.

6. Performance impact

The addition of dynamic aspects on an executing program has two components: what is the speed degradation incurred by adding the overhead of run-time event monitoring, and what is the memory overhead incurred by linking in the “dynamicHook” interface.

6.1 Execution speed

In order to determine the speed impact of “dynamicHook”, we performed two tests. First we measured the absolute slowdown of a function call due to dynamic aspects. This slowdown was between 10 and 20 times the execution speed compared to an unmodified program. For our second test we inserted a dynamic aspect into “thttpd” which instructed the program to exit when it reached a certain point. This point was after the server was up and running but was waiting for web connections and requests. We repeated the test again, but this time with a normal compile and no dynamic aspect code. An artificial call to the “exit” function was placed at the same point that it would have been executed via the dynamic aspect code. Although the performance impact on a per-function-call (10-20X) was significant, there was no measurable difference between the two tests using the web server. Our conclusion is that the overhead for these applications, as opposed to function call test applications, is negligible. However the overhead is determined solely based on the number of function calls within the program; HTTP servers would not contain a significant overhead because of this, but other applications such as back-end database servers, etc. would not qualify for our approach based on the overhead we have observed. An area of research is obviously to see what can be done to lower the overhead per call.

6.2 Memory overhead

The second aspect of overhead concerns the additional space required for inclusion of the “dynamicHook” interface. This consists of two components – the larger memory footprint of the actual executable program, and the addition of the shared memory table. The sise of the latter is entirely dependent on the value of the shared memory area referred to above, and this is in turn dependent on the symbol table. Our measurements have indicated that for the two open-source servers which we used for our intrusion detection testing, the memory overhead was reasonable – 21% in one case and 8% in the other case. As is also true with the speed overhead figures, the memory overhead is determined by characteristics of the program. A new method of implementing these hash tables has been designed which will reduce this overhead, and this method is currently under development.

7. Future work

At the moment our tool is capable of taking off-the-shelf software and adding dynamic aspects and dynamic event monitoring as the program is executing. There are additional capabilities, which we envision for our "dynamicHook" tool. These include the following:

- The sise of the shared memory table used to convey the dynamic aspect data can be excessive. We are implementing a second version of our system, which relaxes the restrictions that cause this.
- Other dynamic aspect systems allow for the exchange of function parameter data with the aspect that is woven into that function. This is possible with our system but requires detailed knowledge of the parameter passing mechanism and has not been implemented at this time.
- We are working on implementing external intrusion detection tools, which will monitor the real-time tracing data from the servers and use pattern matching techniques to watch for anomalies.
- Intrusion detection is a complex process. It is not clear how the selection should be made as to *what* functions should be monitored in a given collection of source code. Further research is warranted to determine if this task can be automated, or whether it will only be possible to manually select those functions of interest for intrusion detection. Obviously, selecting these functions by hand is not the desired method, but was used for demonstration purposes in this research.

8. Conclusions

We have implemented a novel approach to the problem of supporting dynamic aspects for an AOP system. We have used this as the foundation to implement several intrusion detection methods, which can be added to software as it is executing on a Linux platform. New security (or other) aspects can be created and loaded at any time without the need for re-starting or recompiling the base code. Our system does not require any rewriting of the binary code for aspect weaving. The "dynamicHook" code was tested with readily available open-source software, and the intrusion detection aspects were added to the executing system on the fly. With the exception of the stack overflow security aspect (there were no known overflow attacks on the servers we tested with) the aspects operated correctly and reported intrusion attempts. In the case of the stack overflow attack the aspect that was woven into the test application was able to determine whether there was a correct stack trace back to "main" or whether there was an overflow intrusion in progress. In summary, the "dynamicHook" interface has met our objectives – in particular, the use of off-the-shelf tools and available software to enable one to add dynamic aspects to unmodified source code, and to have these aspects operate as a security enhancement for intrusion detection.

References

Almajali 2004 Almajali, S., Lead, T.**: "**A Dynamic Aspect Oriented C++ Using MOP with Minimal Hook". Proceedings of the 2003 Dynamic Aspect Workshop (DAW04 2003), RIACS Technical Report 04.01, March, 2004, Lancaster, UK.

Almajali 2006 Almajali, Sufyan, Tzilla Elrad, "Benefits and Challenges of a Class-Based Design for Dynamic Aspects in DAC++", Proceedings of the 2006 Software Engineering Properties of Languages and Aspect Technologies Workshop (SPLAT'06) as part of AOSD'06 (Bonn, Germany, March 2006)

Aussmann 2003 Aussmann, S., Haupt, M.: "Axon – Dynamic AOP through Runtime Inspection and Monitoring". First workshop on advancing the state-of-the-art in Runtime inspection (ASARTI'03), 2003

Baratloo 2000 Baratloo, Arash, Navjot Singh, Timothy Tsai: "Transparent Run-time Defense Against Stack Smashing Attacks", Proceedings of the 2000 USENIX Annual Technical Conference, San Diego, California, 2000.

Bockisch 2004 Bockisch, C., Haupt, M., Mezini, M., Ostermann, K.: "Virtual Machine Support for Dynamic Join Points", Proceeding of the 3rd International Conference on Aspect-Oriented Software Development (AOSD 2004), March 2004, Lancaster, UK

Bonér 2004 Bonér, J.: “AspectWerkz – Dynamic AOP for Java”, Proceeding of the 3rd International Conference on Aspect-Oriented Software Development (AOSD 2004), March 2004, Lancaster, UK

Chiueh 2001 Chiueh, Tzi-cker, Fu-Hau Hsu, “RAD: A Compile-Time Solution to Buffer Overflow Attacks”, 21st IEEE International Conference on Distributed Computing Systems, 2001.

Cowan 1998 Cowan, Crispan, et al.: “StackGuard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks”, 7th USENIX Security Symposium, San Antonio, Texas, 1998.

Cowan 2000 Cowan, Crispan, Perry Wagle, et al.: “Buffer Overflows: Attacks and Defenses for the Vulnerability of the Decade”, DARPA Information Survivability Conference and Exposition, January, 2000.

Douence 2005 Douence, R. et al.: “An expressive aspect language for system applications with Arachne”, Proceeding of the 4th International Conference on Aspect-Oriented Software Development (AOSD 2005), Chicago, Illinois, March 2005.

Engel 2005 Engel, M., Freisleben, B.: “Supporting Autonomic Computing Functionality via Dynamic Operating System Kernel Aspects”, Proceeding of the 4th International Conference on Aspect-Oriented Software Development (AOSD 2005), Chicago, Illinois, March 2005

Isik 2005 Isik, “Reverse Engineering with LD_PRELOAD”, at http://www.security.nnov.ru/articles/reveng/.

Kiczales 1997 Kiczales, G. et al.: “Aspect-Oriented Programming”, in Proceedings of ECOOP ’97, Springer.

Kiczales 2001 Kiczales, G., E. Hilsdale, J. Hugunin, M. Kersten,J. Palm, and W. G. Griswold, “Getting Started with AspectJ,” in Communications of the ACM, 44(10):59-65, October 2001.

Kuperman 2005 Kuperman, Benjamin A., Carla E. Brodley, et. al. “Detection and Prevention of Stack Buffer Overflow Attacks”, CACM, November 2005, 51-56.

Larochelle 2001 Larochelle, David, David Evans: “Statically Detecting Likely Buffer Overflow Vulnerabilities”, Proceedings of the 10th USENIX Security Symposium, 2001.

Mahoney 2006 Mahoney, Bill: “Compiler Assisted Tracking of Hacker Assaults”, in Proceedings of ICIW 2006: International Conference on Information Warfare, March 2006.

Popovici 2003 Popovici, A. et al.: “Just-in-time aspects: efficient dynamic weaving for Java”, in Proceedings of the 2nd International Aspect-oriented software development conference (AOSD 2003), March, 2003, Boston

Poskanzer 2006 Poskanzer, Jef, “thttpd – tiny/turbo/throttling HTTP server” at http://www.acme.com/software/thttpd/.

Prasad 2003 Prasad, Manish, Tzi-cker Chiueh: “A Binary Rewriting Defense against Stack based Buffer Overflow Attacks”, USENIX Annual Technical Conference, 2003.

Sato 2003 Sato, Y. et al.: A Selective, “Just-In-Time Aspect Weaver”, Proceedings of GPCE’03, 2003, Erfurt, Germany.

Silva 2006 Silva, Eduardo: Monkey HTTP Daemon, at http://monkeyd.sourceforge.net/

Spinczyk 2005 Spinczyk, Olaf, Daniel Lohmann, and Matthias Urban, “AspectC++: an AOP Extension for C++”, Software Developer's Journal, pages 68-76, 05/2005.

Sommerville 2007 Sommerville, Ian, “Software Engineering” 8th ed. Addison-Wesley, 2007.

Vanderperren 2005 Vanderperren, W. et al.: “Adaptive Programming in JAsCo”. Proceeding of the 4th International Conference on Aspect-Oriented Software Development (AOSD 2005), Chicago, Illinois, March 2005

Zhang 2003 Zhang, C., Jacobson, H. A.: “TinyC2: Towards building a dynamic weaving aspect language for C”, Proceeding of the 2nd AOSD Workshop on Foundations of Aspect-Oriented Languages, Boston, March 2003.

Fine-Grain Security for Military-Civilian-Coalition Operations Though Virtual Private Workspace Technology

R William Maule and Shelley Gallup
Naval Postgraduate School, Monterey, CA, USA
rwmaule@nps.navy.mil
spgallup@nps.navy.mil

Abstract: Next-generation service-oriented architectures provide a means for unprecedented levels of collaboration. Security built upon LDAPv3 when coupled with virtual private database technology can enable secure operations within a domain while retaining need-to-know thresholds with fine-grain security. This paper discusses such a technology in use for navnetwarcom trident warrior experimentation. A simulated scenario is discussed in which 12 officers involved in a mhq with moc in mda scenario prototyped an operational application of a comprehensive suite of xml web services that provided a personalised portal for each user, email, chat, presence, instant messenger, web conference, threaded discussions, and secured virtual workspaces with libraries, discussion area, and task management. The focus was on methods to secure communications across military, civilian and coalition operations preliminary to more extensive testing to occur in Trident Warrior 07. Along with an introduction to the technology the study results are presented, addressing methodology and protocols for highly collaborative sessions with varying levels of security in highly dynamic scenarios.

Keywords: military, security, collaboration, community of interest, coi, workspace, wiki

1. Introduction

The current state of world defence requirements has dictated a paradigm shift in national policy. And, more significantly, has required cooperation between nations that is unprecedented. In addition to the normal configuration of predefined treaties and agreements, humanitarian assistance and disaster relief (ha/dr) operations, crisis response due to other events, and the global war on terror has changed the nature of cooperation. Partnerships between the United States and other countries can be defined in strategic relations, or in ad-hoc cooperative arrangements to support very near term needs. The basic principle in effect is one of requisite variety. That is, the variety of potential circumstances in the world that will require a unilateral response by the United States or in collaboration with other nations is much greater than pre-defined options and plans put into place far ahead of time to meet notional circumstances. Instead, ad-hoc configurations of nations, fit to the circumstances requiring action, will be much more effective and ensure that multiple contingencies can be met simultaneously. The overall impact is a more dynamic and effective national and military response.

Defining the collaborative relationships between nations and militaries based upon emergent and ever changing requirements is a necessary, but a difficult problem for information systems providers. The assumption used by operational planners and authors of the configurations in place may not have initially included an understanding of implications inherent in these designs. The United States Navy has embraced the concept of a centralised command and control (c2) capability described as a maritime operations centre, or moc. Although still in design, the concept of the moc is that this centre will become a “reachback” centre for information fusion and planning, thus freeing afloat units to execute missions with smaller personnel resources at sea. And, as these mocs reach initial operational capability, they will be able to collaborate with mocs of other nations, thus extending the operational reach necessary in global operations. Herein however is the principal issue. How to reach across the security boundaries between nations in a way that preserves the autonomy of those nations information security, while allowing acceptable interactions with sufficient information to construct executable plans for quick action and command and control of those actions. There are those information systems, which are already the subject of international agreements, but which will also require that classified information be added in such a way as to make this internationally shared information most useful. Architecting information systems and knowledge systems, in consonance with this security, environment diversity, and

national response paradigm, therefore has a multi-dimensional challenge. The purpose of this paper is to provide background to help frame technical considerations for this effort, and to provide potential solutions for implementation based on a tested approach.

2. Enterprise architecture

Modern enterprise architecture often uses "system-of-systems" methodology to integrate new and legacy operations (Jamshidi 2005). Web services exponentially compound the complexity, and military systems further complicate integration concerns through "need-to-know" and extensive security classification requirements—which may change at any moment. Prior research has addressed evolutionary design in the acquisition of complex adaptive systems (Sage and Cuppan 2001); integration issues in information intensive operations (Pei 2000); and provisions for information superiority (Manthorpe 1996) in architecture implementation. This paper extends the above research by showing a practical application of a means to dynamically provision security within enclaves to support evolving communities of interest. Historically military system-of-systems initiatives have pursued development, integration, interoperability, and optimisation to enhance performance in future battlefield scenarios (Jamshidi 2005; Pei 2000). Information superiority initiatives address interoperability and synergism of command, control, computers, communications, and information (c4i) and Intelligence, surveillance, and reconnaissance (isr) systems (Manthorpe 1996). Relatively new is the development of comprehensive collaborative technologies sufficient to integrate previous collaborative "stovepipes", structured within a service-oriented architecture (soa) (Maule and Gallup 2006a). Also new is a focus on the capacity for "hastily formed networks" to address disasters which involve international and non-governmental coalitions (Denning and Hayes-Roth 2006).

2.1 Knowledge security

Separate from network security is methods to secure data within databases and file structures, often based on content, context, or time (Thuraisingham and Ford 1995). Policies for multilevel relational databases often assign clearance levels to subjects, and classification levels to objects (Qian and Lunt 1990). Means have been advanced to guide policy development (Hosmer 1990) with a possible solution in meta-reasoning models (Tan, Poslad and Xi 2004). The "inference" problem that occurs when low-level data references data that is more highly secured has yet to be fully resolved (Marks 1996; Hinke and Delugach 1992; Smith 1990). Semantic meta-level architectures have been advanced to manage policy-based security by separating domain knowledge (what the system knows) from control knowledge (how the system uses this knowledge) within a loosely coupled approach that shares ontology across different systems (Harmelen 1989; Qian and Lunt 1997). Object-orientation in database design and security provides a supporting approach (Thomas and Ravi 1996; Jajodia and Kogan 1990; Keefe, Tsai, and Thuraisingham 1988). Limitations have been shown for traditional label-based approaches wherein data is tagged with appropriate security (Smith, Blaustein, Jojodia and Notargiacomo 1996; Gong and Qian 1995).

3. User-generated security protocols

Need-to-know, "for official use only", and secret-level classifications are not sufficient for knowledge-based operations. Fine-grain segmentation of content is necessary to "tune" data for analysis, and user-defined ontology provides the best available and most cost-effective means for categorisation due to the required subject matter expertise to properly categorise data. At a technical level, authentication will ensure appropriate expertise, and single sign-on (sso) permit smes (subject matter experts) to address content integration such that security protocols are passed at the data and item level. As the reader has no doubt already surmised, the underlying issue revolves about the strategy to shift content management to smes and users—still, this is the only feasible solution given cost constraints, highly specialised and technical information fully understood by only a very few and which changes on a daily basis, and the need to move experiment findings quickly into the procurement pipeline and to the war fighters.

3.1 Domain classification

Military experiments produce massive amounts of data and information, highly related in context, making processing time-consuming and complex. Assignment of security to specific data results in knowledge worker "overload" as all involved are overwhelmed by data types, issues of context, and integration of quantitative and qualitative variables. Distillation of appropriate security protocols is problematic at best. There is a need to allow users to self-organise such that their collective actions will provide a framework to both organise and secure content. The forcenet innovation and research enterprise (fire) system supports experimentation analysis, and the tactical applications for collaboration within fire (tacfire) the collaborative web services for fleet applications. Requirements include that knowledge practices support dynamic input from users [forms, reports, files], reside completely in a secured, partitioned database with encrypted communications, and support collaboration, remote acquisition via crawlers, abstract representation, and high-level metadata. Data and information is from large, highly complex experiments, smaller or limited-scope experiments, and technology-specific studies and field trials (Maule and Gallup 2006b). Additional requirements are that enterprise knowledge support strategic, operational, and tactical objectives, and that the archiving methodology support "thread-pulling" commonly used for retrieving information in the military. Within fire, thread pulling was initially accomplished through ontology-based codes or tags on each piece of data; later through a combination of AI tools to automatically generate ontology and extractions based on that ontology. Current approaches are evolving as described in this paper–user-generated content schemes, user-defined security—with administrators and knowledge workers to monitor high-level processes.

3.2 Workflow processes

A means to address user-managed resource development and security management is to design workflows contingent on completion of designated procedures. The fire system mentioned above successfully implemented such a procedure for experimentation design and management wherein resource development and access protocols were shifted to subject matter experts. Tacfire (tactical applications for collaboration in fire) extends this strategy through virtual workspaces that can be configured and self-managed by user groups. Key to the implementation of these workspaces and supporting web services is a workflow wherein tools are optimised. Complicating factors include logistics as various user groups and communities are merged into a new community; technical as communications are extended from chat and portals to a full suite of collaborative web services; and political as tools are positioned to compete for utilisation. Figure 1 provides a high-level workflow for tool utilisation recently tested by a group of officers at the naval postgraduate school. A similar scenario is currently being evaluated for the navnetwarcom trident warrior 2007 experiment. In this scenario a maritime operations centre (moc) is superimposed over a decision support to civilian authorities (dsca) scenario that was completed as part of trident warrior 06 and involved a successful alliance between the DoD, department of homeland security (dhs), FBI, state/county/local agencies, coast guard, and coalition partners—all in a maritime domain awareness (mda) scenario.

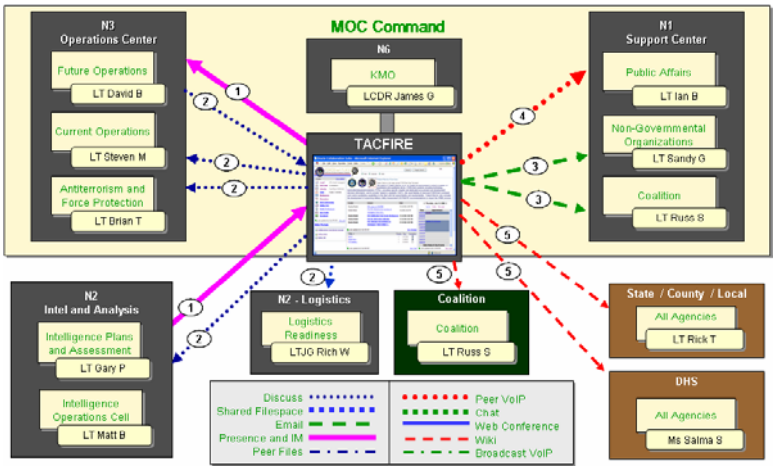


Figure 1: Tacfire support for mhq with moc in mda scenario.

Key to success in such an undertaking is selection of the most effective tool for a given task, and provisions for security in tool utilisation. Figure 2 illustrates a tested workflow when an mda event occurs. Intel receives the alert and uses the presence server to instant message those currently online. Future operations builds a core team through chat, launches a secure workspace and virtual private database with discussion threads. Resources within the workspace including the task manager and content manager send alerts via email. A peer-peer VoiP session to public affairs helps assemble the non-DoD participants. PA then launches a wiki to provide public information to be shared with coalition, dhs, hls, and state/county/local officials.

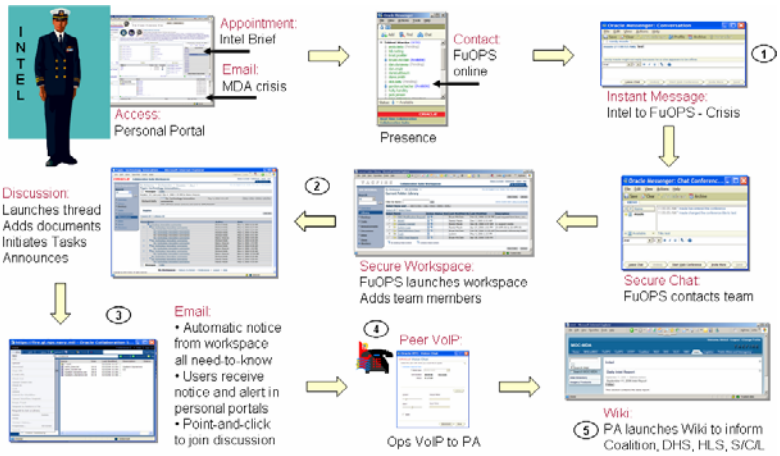


Figure 2: Tacfire tool flow in mhq with moc for mda scenario.

3.3 Workspace/coi security

In the tested scenario a need was established for three operational workspaces, each of which enabled a community of interest (coi) with security provisioned to the community. Each user accessed the workspace through their personal portals (figure 3). Instant communications were provided through integrated system-level services including instant messenger, chat, and peer-peer VoiP.

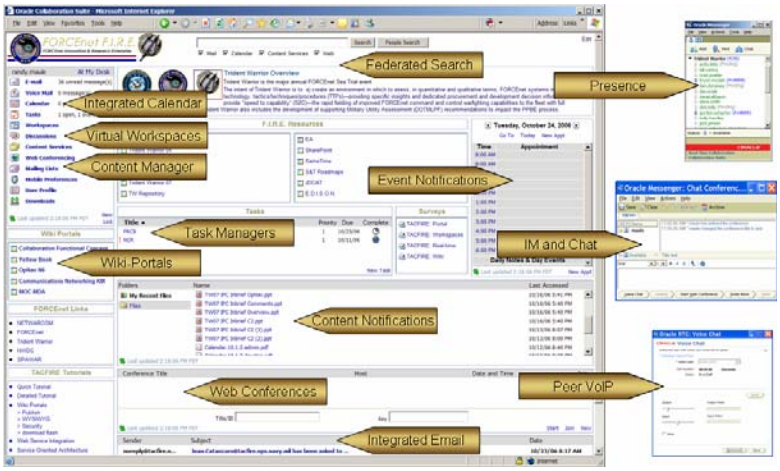


Figure 3: User personal portals and ancillary services.

The participants in each are identified in Figure 4 by position. Workspaces were provisioned dynamically during initial deployment of the scenario—with a workspace/coi specific to Intel, a workspace/coi for current operations, and a workspace/coi for non-DoD participants. Each workspace became a separate security enclave. To pass information across enclaves a user would need to be a member of both enclaves and then physically pass the information. Basic services provided in a workspace include an overview or notifications page with portlets showing new or changed information within the workspace such that a user can click one of the notifications to immediately access that item. The library contains documents or media for which workspace/coi

members are cleared for access. The discussion area organises by thread, and discussions can be entered via the workspace or email. A task manager allows members to assign tasks and receive updates as a percentage of completion with annotation. A meeting area can launch traditional or online meetings with visual streams, chat, and VoiP. COI events populate to member's personal calendars and appear in their daily events portlet within their personal portals. Figure 5 illustrates these workspace services, with the discussion capability and workspace search visible. The search will federate results across calendar, discussions, threads, web, and email.

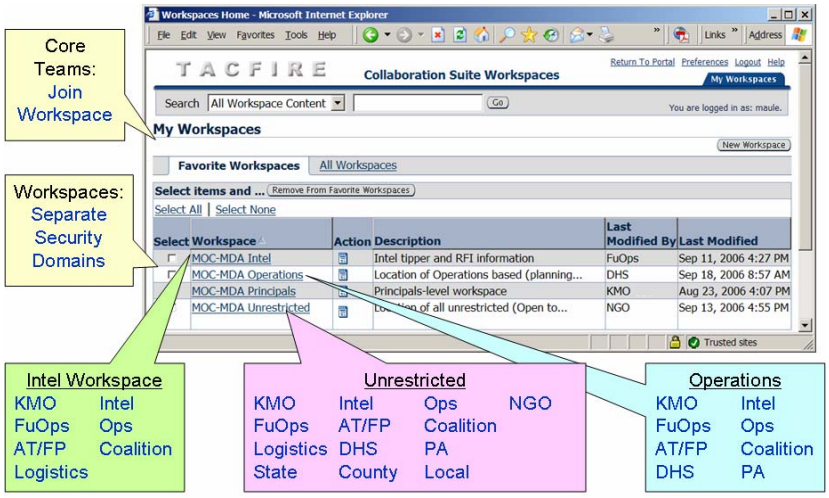


Figure 4: Secured tacfire workspaces and participants.

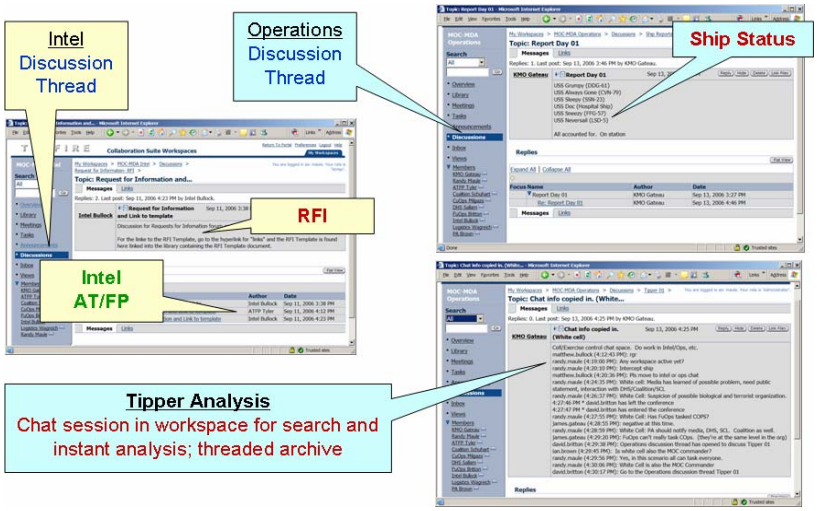


Figure 5: Internal workspace/COI and discussion services.

The task manager is a very powerful feature—but there is a learning curve. In the scenario, current ops used the task manager for assignment to appropriate personnel for tasks such as requests for information (rfi) and calls to action. A user that was assigned a task would receive notifications via email and also within the task notifications portlet on the workspace overview page—at which point comments were entered into the task and percentage completion assigned. Other members of the assigned group were able to collaboratively participate such that tasks could be segmented while status reports aggregated back to current ops. Lessons learned from utilisation of the task manager within a workspace are that members should be identified by current position to enable, for example, several officers to be assigned a task rather than an individual. In this situation as a user of an assigned group completes part of a task the portion complete can be designated and comments made such that another in the group can monitor and complete the task as needed—which can be particularly helpful in the event more than one group member is required to complete a task, or a change in shifts necessitates passing the task to another.

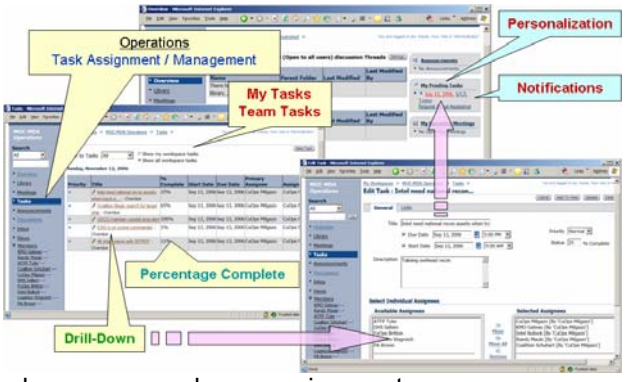


Figure 6: Workspace task manager and user assignments.

4. Assessment

Participants were assessed across the 4 major tool categories to determine functionality, usability, and usefulness in an operational setting. Figure 7 shows portal and wiki surveys, which indicate 91% of participants see an advantage in aggregating web and collaborative services into a personalised portal; 83% found the interface user-friendly or intuitive. Most users had not previously developed a wiki, but 77% found the wiki either satisfactory or very satisfactory with the remainder undecided. Users determined the wiki to be most appropriate for general information (27%), expertise location (22%), public information (16%), operational descriptions (14%), position descriptions (11%), and file transfers (11%).

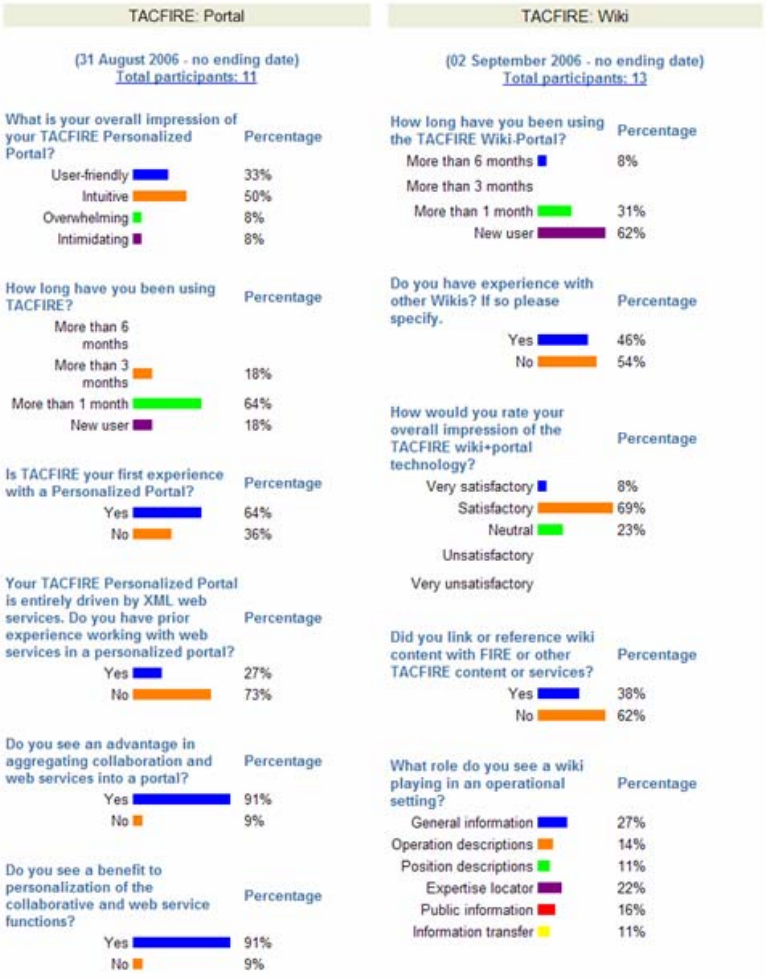


Figure 7: Participant assessment surveys: Personalised portal and wiki.

Similar but lengthier surveys were conducted for the workspace and real-time tools. 54% of the participants were either positive or very positive concerning the workspace in an operational setting, with 38% undecided. The workspace overview page contained notification portlets to show activity throughout the workspace and 83% found this useful. For task and workflow automation (the most complicated feature) the responses were evenly divided between positive and neutral. 75% of responses were positive on the workspace library in an operational setting. 100% of the respondents recommended use of the workspaces for virtual collaboration. Assessment of real-time tools covered web conferencing, instant messenger, multi-party chat, and both broadcast and peer-to-peer (server bypass) VoiP. Web conferencing received a 64% favourable response; instant messenger 81% favourable; chat 74% favourable; both peer and broadcast VoiP a 63% favourable rating. In all cases the remaining responses were primarily neutral. A feature of the presence capability, wherein the status of all participants could be readily ascertained (available, away from desk, etc.), and the ability to trigger real-time services from this presence interface, received a 90% favourable rating.

5. Conclusion

The ability to dynamically provision and assign members to secure virtual workspaces/cois enable secure communications within an enclave and domain. Members can be instantly added or removed as appropriate “need-to-know” is established or removed. The architecture supports single sign-on across several-hundred web services. LDAPv3 helps aggregate services to a portal personalised for each user. The tested situation involved 12 officers in a maritime domain awareness crisis in which a maritime operations centre was superimposed on a defence support to civilian authority scenario. Operations included federal, state, local and coalition participants—with communications and resources segmented and secured through the workspace/coi technology. Results indicate that this approach and technology eases prior security problems; provides ample security for mda and coalition operations; produces a common browser-accessible platform for moc, mda, coalition, and civilian interactions; can automate tasks needed to enact large-scale collaborative efforts; and most importantly can speed workflow while reducing costs and the number of technologies required to enact collaborative, knowledge-based, security-aware services.

References

Denning, P., and Hayes-Roth, R. (2006). “Decision-making in very large networks,” Communications of the ACM, Vol. 49, No. 11, pp19-23,

Gong, L. and Qian, X. (1995). “Enriching the expressive power of security labels,” IEEE Transactions on Knowledge and Data Engineering, Vol. 7, No. 5, pp839-841.

Harmelen, F. (1989). “A classification of meta-level architectures,” in Logic Based Knowledge Representation, MIT Press, Cambridge, MA.

Hinke, T. and Delugach, H. (1992) “Aerie: An inference modelling and detection approach for databases,” Proceedings of the Sixth IFIP Working Conference on Database Security, August, Vancouver, B.C.

Hosmer, H. (1990). “Integrating security policies,” Proceedings of the Third RADC Database Security Workshop, pp169-173.

Jajodia, S. and Kogan, B. (1990). “Integrating an object-oriented data model with multilevel security,” Proceedings of the IEEE Symposium on Research in Security and Privacy, May, Oakland, CA.

Jamshidi, M. (2005). “System-of-systems engineering - A definition,” 2005 IEEE International Conference on Systems, Man, and Cybernetics, URL: http://ieeesmc2005.unm.edu/SoSE_Defn.htm.

Keefe, T, Tsai, W., and Thuraisingham, M. (1988). “A multilevel security model for object-oriented system,” Proceedings of the 11th National Computer Security Conference, pp1-9.

Manthorpe, W. (1996). “The emerging joint system of systems: A systems engineering challenge and opportunity for APL,” John Hopkins APL Technical Digest, Vol. 17, No. 3, pp. 305-310.

Marks, D. (1996). “Inference in MLS database systems,” IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 1, pp46-55.

Maule, R., and Gallup, S. (2006a). “Quality of service in next-generation knowledge management”, Proceedings of the International Conference on Information Society (i-Society 2006), ISBN 0-9546628-1-4, 7-10 August, Miami, FL.

Maule, R., and Gallup, S. (2006b). TACFIRE: Enterprise Knowledge in Service-Oriented Architecture. Proceedings of the IEEE/AFCEA 2006 Military Communications Conference (MILCOM 2006), ISBN 1-4244-0618-8, 23-25 October, Washington, DC.

Pei, R. (2000). “Systems of systems integration (SoSI) - A smart way of acquiring army C4I2WS systems,” Proceedings of the Summer Computer Simulation Conference, pp. 574-579.

Qian, X., and Lunt, T. (1997). “A semantic framework of the multilevel secure relational model,” IEEE Transactions on Knowledge and Data Engineering, Vol. 9, No. 2, pp292-301.

Qian, X., and Lunt, T. (1990). “A MAC policy framework for multilevel relational databases,” IEEE Transactions On Knowledge And Data Engineering, Vol. 8, No. 1, pp3-15.

Sage, A., and Cuppan, C. (2001). “On the systems engineering and management of systems of systems and federations of systems,” Information, Knowledge, Systems Management, Vol. 2, No. 4, pp325-345.

Smith, K., Blaustein, B., Jojodia, S. and Notargiacomo, L. (1996). “Correctness criteria for multilevel secure transactions,” IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 1, pp32-45.

Smith, G. (1990). “Modeling security-relevant data semantics,” Proceedings of the IEEE Symposium on Security and Privacy, May, Oakland, CA.

Tan, J., Poslad, S. and Xi, Y. (2004). “Policy driven systems for dynamic security reconfiguration,” AAMAS’04, July 19-23, NY, NY.

Thomas, R. and Ravi, S. (1996). “A trusted subject architecture for multilevel secure object-oriented databases,” IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 1, pp16-31.

Thuraisingham, B., and Ford, W. (1995). “Security constraint processing in a multilevel secure distributed database management system,” IEEE Transactions on Knowledge and Data Engineering, Vol. 7, No. 2, pp274-293.

Creating Hardware-based Primitives that Facilitate the Exposure of State Information Useful for Security Related Monitoring

Stephen Mott and Paul Williams
Air Force Institute of Technology, Fairborn, OH, USA
stephen.Mott@afit.edu
Paul.Williams@afit.edu

Abstract: The Co-processor Intrusion Detection System (CuPIDS) is an intrusion detection system (IDS) we are using to explore ways in which security-focused hardware primitives can improve the security of modern computer architectures without compromising performance. While focusing on intrusion detection, CuPIDS also allows for application focused security policy compliance monitoring (SPCM). Previous work on CuPIDS (CuPIDS-Purdue) proved that leveraging multiple CPUs in an asymmetrically parallel manner can reliably detect a multitude of attack/exploit types in real-time for real world applications – a sought after capability in the security monitoring field. Our research - the CuPIDS-AFIT project - furthers this effort by exploring novel hardware-based primitives that support parallel monitoring for real-time intrusion detection (ID) and SPCM tasks. The availability of processors embedded within a field programmable gate array (FPGA) afford us the opportunity to develop practical means by which we can implement these primitives to gain and transmit state information in hardware by methods not possible previously. From this, we have shifted CuPIDS' reliance on the host operating system (OS) to dedicated hardware for viewing and reporting system state information. This makes the IDS itself more secure from attack, tangibly improves runtime efficiency and time to detection, as well as enables new means by which real-time ID and SPCM are implemented.

Keywords: Intrusion detection (id), security policy compliance monitoring (spcm), parallel monitoring, asymmetric computing

1. Overview

Security is an increasing problem in today's computing ecosystem. Every year the reported instances of vulnerabilities in software grow with no end in sight (CERT 2006). The computing industry is aware of this and tries to implement good programming practices, create safer programming constructs, and modify how operating systems (OS) interact with the software. However, programmers are only human and are bound to make mistakes, no matter how strong their resolve is to create non-exploitable code. Furthermore, as code becomes more complex to solve increasingly complicated problems, vulnerabilities are harder to detect and prevent. This is, in part, due to current computer architecture's primary design goal--the improvement of performance. We believe computer architectures themselves may be leveraged to prevent the exploitation of vulnerable code, as we cannot rely purely on the programmer to ensure this. Our research explores ways in which we can shift computer architecture design towards a security-centric paradigm without compromising the performance of modern systems. This is accomplished by providing new methods by which system state can be revealed and processed in parallel, enabling more robust and flexible intrusion detection (ID) and security compliance monitoring (SPCM). We propose incorporating a number of functional hardware-based primitives into contemporary computer architectures to expose and utilise state information in new ways at the hardware level. Using these hardware primitives to monitor system state in parallel as the monitored code executes allows finer-grained intrusion detection than was previously possible, with little to no impact on system performance. The architectural enhancements described here target a wide range of exploits and faults. Further, moving monitoring capabilities from software into hardware improves time-to-detect after an error or attack This also offers possibilities for higher levels of security assurance than is possible with software-only security systems and hardware-oriented systems that closely couple monitoring software to the software it monitors.

2. Related research and background

We classify current hardware-based security efforts into three main categories: Loosely-coupled coprocessor intrusion detection systems (IDS), tightly-coupled coprocessor IDSs, and hardware-based security mechanisms. In a loosely-coupled coprocessor IDS, the monitoring coprocessor does not reside at the same logical level as the production processor being monitored. As such, the coprocessor is limited to gathering state information only from other system peripherals (i.e. main memory, hard drive, etc.) and cannot exert direct control over the processor/system being monitored. Such IDS stem from research done on cryptographic add-in coprocessors like the IBM 4758 crypto processor (Zhang). Tightly-coupled IDSs tackle the shortcomings of loosely-coupled IDS architectures by residing at the same logical level as the processor

being monitored, thus having equal or greater privileges than the monitored processor. Our CuPIDS-AFIT system can be considered tightly-coupled IDS. Hardware-based security mechanisms are tightly coupled to the hardware in which they are designed to protect, but perform a more limited scope of security related tasks. Independent auditors (Molina 2003) and CoPilot (Petroni 2004) are examples of loosely-coupled IDSs. Both systems use coprocessors residing on an add-in board and execute their own dedicated software independent of the software being executed by the monitored system. While this protects the IDS itself from being compromised, this also limits the useful state information the IDS is able to gather. As such, the independent auditors system can only access filesystem information from the monitored system's hard drive to check the integrity of the filesystem (Molina 2003). CoPilot observes the monitored system's physical memory space in order determine if key invariants have been modified. However, as memory is being monitored by a peripheral on the system bus, only data that always resides at known memory locations (i.e. wired kernel pages) can be monitored. Thus, CoPilot can only protect the integrity of part of the kernel's address space. All user processes within the system utilise virtual memory and cannot be monitored due to paging (Petroni 2004).

Tightly coupled IDSs are enabled by the recent commercial availability of affordable multiprocessor and multi-core systems. Examples include CuPIDS-Purdue (Williams 2005) and the Security Enhanced Chip Multiprocessor (SECM) (Shi 2005). The monitoring coprocessors in these systems have direct access to the production processor being monitored as well as other system components such as memory. This improves the useful state information available to the IDS and allows the monitoring processor to intervene in the event that the monitored processor is compromised. CuPIDS-Purdue uses a shared memory multiprocessor system in an asymmetric manner. One processor is used for executing the monitored production code, while the second processor is used for executing the monitoring code. The tightly-coupled nature of CuPIDS-Purdue synchronises monitoring and production processors, providing truly parallel, real-time security monitoring. However, CuPIDS-Purdue uses a single operating system spread across both processors. This single OS approach allows the monitoring processor to have access to the production processor's virtual memory space, enabling the monitoring of both kernel and user space. System calls are used for intra-processor communication, however this introduces overhead that tangibly affects performance of the production processor as well as makes the IDS itself vulnerable in the event that the OS is compromised (Williams 2005). The SECM addresses some of these issues by being implemented on a multicore processor with each core executing its own OS – the privileged core executing a more privileged OS than the production core. The SECM, however, is more of a gatekeeper than an intrusion detection system, as the production core must go through the privileged core when performing operations that are not part of its privileges. The privileged core is connected directly to the production core's L1 and L2 cache, rather than using the memory bus to communicate state information - alleviating communication overhead. Furthermore, the dual OS nature of the SECM makes the IDS less vulnerable to compromise, (Shi 05). A large amount of research has been conducted in the hardware-based security mechanism field. Hardware-based branch checking uses hardware to ensure that branch instructions branch to known good memory locations. Examples of this type of mechanism are presented in (Divya 2005)(Zhang 2005). A hardware return address stack is a stack created in hardware that is compared to the software stack in memory when a function returns. Such systems are discussed in (Lee 2003) (Ragel 2005). Non-executable memory adds a bit to the page table entry that denotes whether or not an instruction can execute from a memory location (Intel). Additionally, there are some non-hardware-based intrusion detection concepts that have proven useful to our research. These include system/function call monitoring (Feng 2003) and branch source-destination checking (Kirainsky 2003).

3. High-level design

The CuPIDS-AFIT architecture combines the robust state gathering capabilities of tightly-coupled IDSs with the secure nature of loosely-coupled IDSs. The general architecture of our monitoring concept is shown in Figure 1 below. The production processor element (PPE) executes the host OS and/or the monitored process(es). We label the monitoring hardware ambiguously as "monitor hardware element" (MHE) because its implementation is very flexible and subject to the hardware designer. For prototyping, this flexibility is afforded by a field programmable gate array (FPGA) that allows the use of embedded hardcore processors, synthesisable softcore processors, real-time logic (RTL), or a combination of the three for implementing the MHE. As a result, the MHE does not necessarily have to be identical to the PPE – a first for tightly-coupled IDSs – nor does it have to run a dedicated OS or any code at all for that matter. The implementation of the MHE is partially dependent on, but not limited to, the hardware-based primitive(s) being implemented. Furthermore, it should also be noted that as we are proposing primitives, these concepts are flexible enough to implement the MHE in a number of ways (i.e. not limited to how we have implemented them).

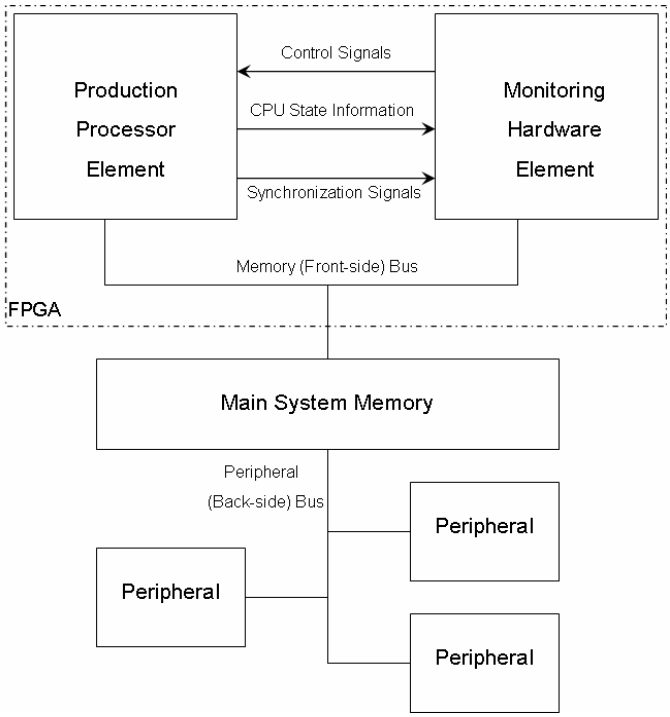


Figure 1: CuPIDS-AFIT general architecture

As with CuPIDS-Purdue, our design methodology is based on an asymmetric shared memory approach to allow for sharing of hardware resources (i.e. memory and peripherals) between the PPE and the MHE. Unlike CuPIDS-Purdue, however, we implement a separate OS/software executing on the MHE similar to the SECM in (shi 2005). This decreases the amount of software coupling between code executing on the PPE and the monitoring hardware. This decreased software coupling increases the security of the MHE itself, but makes communication between the PPE and MHE more difficult as system calls can no longer be used. Thus, the CuPIDS-AFIT architecture implements all communication between the PPE and the MHE using dedicated hardware paths. Using dedicated hardware paths not only decreases communication overhead, but also prevents any communication between the PPE and the MHE from being intercepted (as would be possible if communicating over the main memory bus). Rather than the PPE explicitly communicate state information to the MHE, state information is snooped (implicitly communicated) from the PPE by the MHE tapping directly into key parts of the production processor and it's supporting architectural components such as memory. Thus, depending on the implementation, the MHE has the potential to snoop any portion of the PPE without the PPE's knowledge. As the PPE has no knowledge of what state information is being gathered by the MHE, timing and other related attacks against the security system are made more difficult.
Additionally, critical to our parallel monitoring concept is the ability to monitor code on the PPE as it executes. This requires that the MHE be synchronised in some way to the code executing on the PPE. As a result, the OS executing on the PPE must have some, albeit little, knowledge of the MHE. An example of such a synchronisation signal is an interrupt that the PPE sends when a context switch occurs, requiring a slight modification to the OS' scheduler. We keep such explicit communication to a minimum which minimises the area of attack. The only other explicit communication between the PPE and MHE in CuPIDS-AFIT are control signals sent from the MHE to the PPE.

4. Onto hardware-based primitives

4.1 Execution policy enforcement

The program counter (PC) keeps track of the memory location containing the currently executing instruction. While monitoring the PC for security reasons is not a new concept, we have applied it in a new manner which adds capabilities that are not present in current computer architectures. By keeping track of the PC, we can locate where in memory an instruction is being executed from – an ability that benefits both intrusion detection and security policy compliance monitoring (SPCM) tasks. When having defined where in memory code is allowed to execute from and where it is disallowed, we can use the PC to check if such a policy is being adhered to or not. This knowledge about what code is allowable can have multiple granularities ranging from the global level to the basic block level. We propose a mechanism that specifically tracks the PC to provide ability similar to that of execute-disable bit (XD bit) technology from Intel Corp. XD bit

technology adds an extra bit to the page table entry that can be toggled to determine if code is allowed to execute from a particular memory address or not (Intel). However, many processors in use today still do not contain this type of capability. This mechanism specifically targets processors that would benefit from such a capability, but were not designed with it in mind. Our execution policy enforcement mechanism is depicted in Figure 2 below. Every instruction is automatically checked by the PC monitor. Checking is accomplished by treating the PC as if it is accessing that memory location. The allow-bit corresponding to the PC is then output to the CPU. Depending on the value of the allow bit, the PPE may be interrupted and the violation handled. Theoretically, we can keep track of every address in the memory space and then compare the PC to check if the executed instructions are allowed, however, it may be more beneficial to the system designer to monitor a range of addresses. Our mechanism is flexible enough to accommodate such a capability. Additionally, It should be mentioned that the capabilities described here cannot be implemented via an algorithm running on a processor as the instruction pointer must be checked immediately upon presentation to the monitoring hardware and violations are detected and handled before the violation is allowed to take place. Thus, this capability is enforced via RTL implemented within the FPGA fabric to provide a detection time of O(1).

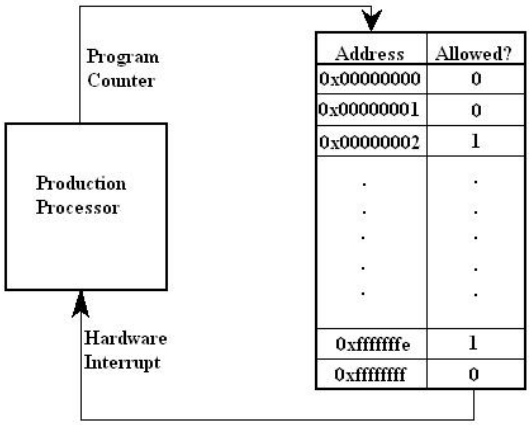


Figure 2: Program counter monitor

4.2 Multi-context hardware monitors

The execution policy enforcement from section 4.1 as well as the hardware-based security mechanisms discussed in section 2 has thus far all been limited by being application specific. As such, only one instruction stream context can be monitored by the implemented security mechanism. This limits these implementations to non multi-programmed environments. Not every executing process needs to be monitored, but currently there is no way for the security mechanisms to even differentiate between multiple processes at the hardware level. Thus, these mechanisms will not work correctly when trying to operate in a multi-programmed environment, even if only trying to monitor a single process out of many. As we desire monitoring hardware to be able to discern between different processes executing on the production processor, exposing the process identifier (PID) is the most logical piece of state information that can enable this. As no processor outputs the current PID to any external signals that can be monitored, we propose looking toward the OS on the PPE for such a capability. In an OS, the system scheduler keeps track of all processes and their state using a table stored in the kernel space of the OS itself. This portion of the kernel is paged, but stored at a fixed location in main memory, providing a constant location for the PID. By monitoring this location in memory, we are then able to keep track of what process is currently executing. We are not concerned with how the monitoring hardware knows which process or processes it needs to monitor, however we are concerned about whether the capability exists to use the PID to allow the monitoring hardware be able to differentiate between various executing processes. Once the PID has been made visible outside of the PPE, the MHE can take the appropriate action to enable multi-context hardware monitoring. There are three general ways we can do this: Monitoring a single process out of many currently executing processes, monitoring multiple processes with the same hardware, or monitoring multiple processes with multiple types of hardware.

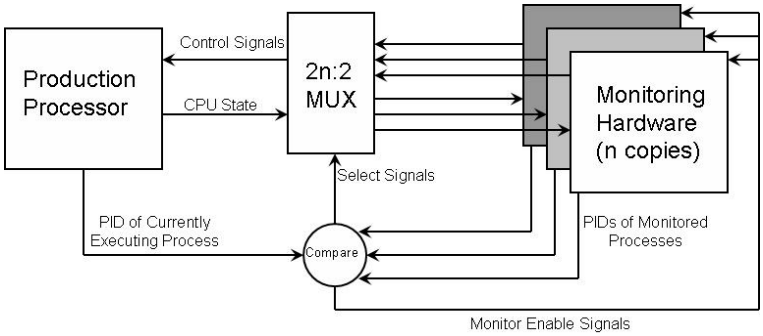


Figure 3: Multi-context monitoring for multiple processes using multiple hardware monitors

Monitoring a single process out of many processes is the simplest case. This requires the MHE to be enabled only when the PPE outputs the PID corresponding to the process to be monitored. Similarly, monitoring multiple processes with the same hardware occurs in the same manner; however the MHE contains a list of PIDs that correspond to monitored processes. This case also requires some method of storing state when switching between processes being monitored. As such, it is probably practical to implement the MHE in the form of code running on a processor core for the ease of writing to memory. Doing so also provides for flexibility by using the same MHE, but with different monitoring algorithms for processes requiring different types of monitoring. Finally, the most complex case monitors multiple processes, but uses multiple hardware monitors to constitute the MHE. This is depicted in figure 3. As can be seen, a multiplexer is used to select and connect the appropriate MHE signals to the PPE. The PID of the currently executing process is compared to the PIDs of the processes being monitored to generate the select signal that controls the multiplexer. The comparator also generates enable signals for the appropriate monitoring hardware, as the monitoring hardware should not try to control the production processor or retrieve state information while the process it is monitoring is not executing. As the PID is gathered in such a way that does not impose on the PPE, no additional compiler support is needed to enable such a capability (compiler support still apples to the specific hardware monitors implemented, however).

4.3 Hardware-based memory introspection

Intrusion detection systems such as CoPilot and CuPIDS-Purdue rely greatly on the ability to have insight into a system's memory space (Petroni 2004) (Williams 2005). We present two methods below by which we believe we can achieve similar capabilities in the CuPIDS-AFIT, but with improved performance, security, and abilities. It should also be mentioned that as the methods described below access main memory, we assume the MHE is being implemented as a processor or processor/RTL combination for easy access to memory.

4.3.1 Asymmetrically-partitioned physical memory

Since our platform is constructed on shared-memory architecture, both the production processor and the monitoring hardware can access the same main memory. This is ideal from a monitoring standpoint as both the production processor and the monitoring hardware can have visibility into the same physical memory space. However, we do not want this sharing to cause data collisions between memory accesses by the production processor and monitoring hardware. Thus, we need to partition memory in such a way that the production processor and the monitoring hardware do not step on each other's toes.

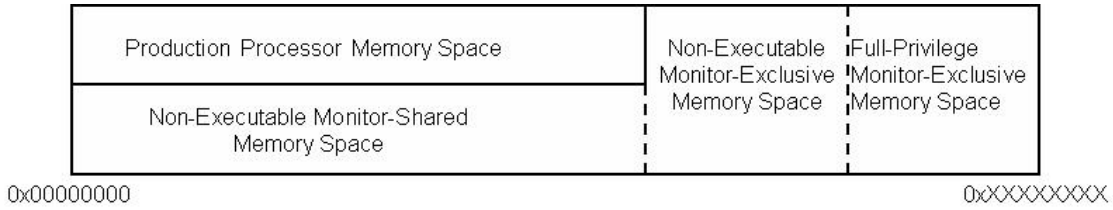


Figure 4: Asymmetrically partitioned memory map (not drawn to scale)

Figure 4 depicts how we partition memory. Since we don't want the production processor to be able to access the monitor's memory space for security reasons, we instruct the production OS to view only a portion of the total available physical memory space. We then allow the OS or monitoring software executing on the MHE to be able to have access to the entire available physical memory space. Thus the monitoring hardware has access to its own memory while still being able to access the PPE's memory space. Now that

the monitoring hardware has access to the entire physical memory space, we further partition the memory map (as viewed by the monitoring hardware) into the following three contiguous regions described below:

- No-execute shared memory: This region maps directly to the production processor's physical memory space. It provides the monitoring hardware with read/write access to the production processor's memory for monitoring and data restoration tasks. This region is non-executable, as we want to prevent the possibility of the monitoring hardware inadvertently executing any malicious code that may reside within the PPE's memory.
- No-execute monitor-exclusive memory: This region acts as a safety measure. As we leave the decision of what data is stored by the monitoring hardware to the developer, we provide this non-executable region for storing information that may be malicious in nature.
- Full-permission monitor-exclusive memory: This region is the only region of the monitor's memory space from which instructions can be executed. As such, it is used to store the monitoring OS and/or any software that executes on the monitoring hardware.

We use a Harvard-based architecture to implement the MHE in this case. This is because we can map the non executable regions of memory to the data bus and the executable region to instruction bus of the Harvard-based processor. Additionally, code executing on this architecture is particularly resilient to exploitation via malicious self-modifying code. It should be noted that using an integrated coprocessor to asymmetrically partition memory as we have described here can provide for a tightly coupled version of the CoPilot (Petroni 2004) system. Rather than use an add-in coprocessor with its own dedicated memory bank, our implementation would sacrifice some of the PPE's main memory to implement CoPilot in the same physical package as the processor it monitors. Our implementation would still be limited to only being able to monitor the production processor's kernel space, as monitoring concepts would operate in the same way as they did in CoPilot. However, one notable improvement is that we have the ability to route control signals from the monitoring hardware to the production processor, enabling the enhanced CoPilot to be able to exert control over the system in the event that the kernel is compromised. Additionally, this allows the monitoring hardware to have access to main memory without having to contend with other peripherals (except for the production processor itself) for control of the bus as was the case with the current iteration of CoPilot.

4.3.2 Co-opting the memory management unit

Whereas the CoPilot system is able to only monitor key data structures within the kernel space – a limitation brought about by gathering state via monitoring physical memory - the CuPIDS-Purdue prototype is able to monitor processes within user space by creating a monitoring shadow process that hooks into a production processes virtual memory space (Williams 2005). Although CuPIDS-Purdue's parallel monitoring occurs in real-time, it imposed a tangible performance penalty in addition to the IDS itself being susceptible to attack. To remedy such shortcomings, we propose modifying the memory management unit (MMU) of the PPE in such a way as to enable a similar capability in hardware rather than software. The benefit of which is two fold. 1) Implementation in hardware removes nearly all communication overhead, and 2) Implementation in hardware enables us to use a separate OS on each processor, increasing the security of the IDS itself. The MMU of a modern processor is responsible for controlling the translation of virtual memory addresses into physical memory addresses. In conventional computer architectures supporting virtual memory, the MMU resides within the processor itself. Therefore the MMU only services memory requests made by the processor core that contains it (i.e. contemporary multi-core processors have a MMU for each core). However, to monitor the state of user space processes in CuPIDS-AFIT we propose modifying the production processor's MMU such that the MHE can take control of (co-opt) it. We are not proposing the modification of the virtual memory system itself, but rather we are proposing to change how the MMU is controlled. This allows us to use the abstract concept of virtual memory, rather than worry about modifying the complex functionality of the virtual memory system itself.

The general architecture of our co-opted MMU concept is shown in figure 5. It should be noted that we are not concerned with how the virtual address being monitored is obtained, so we assume that the MHE already has knowledge of where key data structures to be monitored reside within a process' virtual memory space. As a result, the PC and PID can be monitored (see sections 4.1 and 4.2) to determine when the PPE accesses the data structure. When this occurs, the monitoring hardware triggers a hardware interrupt signalling the PPE that its MMU is to be co-opted. At this point the PPE is allowed to continue executing so long as it has not fetched a load or store instruction. In the event that a memory accessing instruction is encountered, the PPE will stall. The MHE then sends the virtual address in question to the production processor's MMU, at which point the virtual memory system looks up the physical address and transmits it back to the MHE. The MHE then performs a read at the physical address. This assumes the monitoring

hardware does not contain/use a MMU of its own. Once the data structure has been read by the monitoring hardware, the production processor is notified via a hardware interrupt that it can continue execution if it was stalled. Simultaneously, the monitoring hardware performs invariant checks and/or other state related tasks on the data structure. Throughout this entire process, the PPE is unaware the MMU is being co-opted as it is either executing or stalled, and as such requires no extra instruction or compiler support.

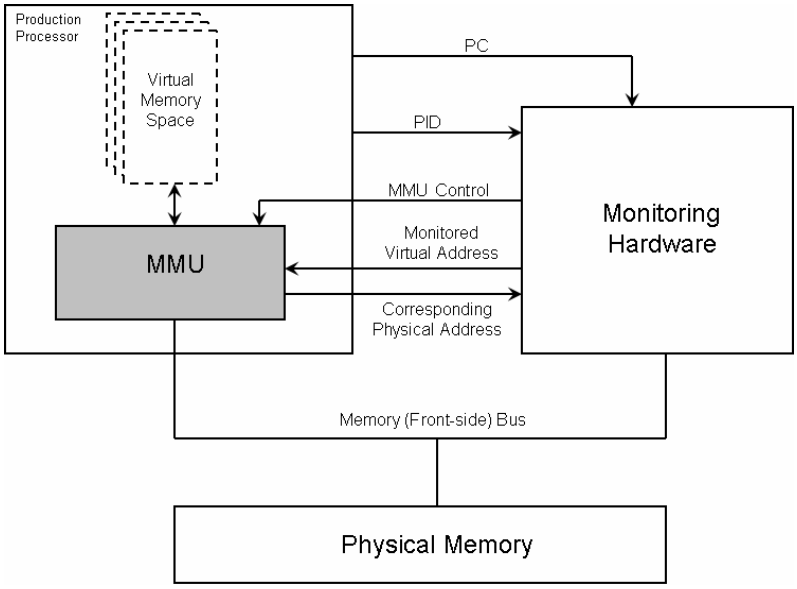


Figure 5: Co-opted memory management unit architecture

As virtual memory is very dynamic, the physical address where a data structure resides will most likely change over time, thus the MMU must be co-opted every time a data structure is to be monitored to ensure that the correct data is monitored. Additionally, as the MMU is often-used, such a system may hamper performance of the production CPU. This is because contention for the MMU is generated. This is dependant on how often the PPE accesses memory as well as how often the monitoring hardware co-opts the MMU.

5. Expected results

We used a Xilinx Virtex-II Pro for prototyping, allowing a flexible platform on which different hardware primitives can be implemented in an effective manner. We intend to test each primitive independently of one another, rather than as complete IDS. Since we are proposing hardware-based primitives, implementation is conducted as proof of concept, and as a result, testing at this stage is mostly functional in nature. Concerning performance, we expect the detection times of the proposed primitives to be significantly reduced compared to CuPIDS-Purdue. Depending on the primitive, we expect detection time to range anywhere from one instruction to a few thousand instructions at the most – a marked improvement over the hundreds of thousands of instruction incurred by CuPIDS-Purdue. This metric is measured in terms of instructions executed on the production processor, rather than time, as it allows us to generalise performance to all production processors/platforms no matter what their relative performance is. Detection sooner rather than later is the ultimate goal. With the added capabilities of CuPIDS-AFIT, we believe we can detect intrusions while presenting a minimal performance overhead to the user. Typical applications should be able to operate transparently with CuPIDS-AFIT, while high performance applications designed to take advantage of the entire system may still lose some performance. However, we feel that the gains in security far outweigh any potential performance degradation.

6. Future work and conclusion

Although we have conducted much research in developing hardware-based primitives for parallel security monitoring, there are a number of areas open to future research. As a number of our proposed primitives have access to main memory, we would like to look into any self-healing and data forensics capabilities such primitives could enable. Furthermore, while we touched on scaling intrusion detection mechanisms to multiple processes, with the emergence of quad core processors and beyond, we would also like to explore how we would need to modify hardware and software to provide parallel security monitoring for multiple cores simultaneously. Research done on the initial CuPIDS-Purdue prototype proposed concepts that challenge conventional wisdom by trading performance for security. CuPIDS-AFIT furthers this effort by creating and implementing hardware-based primitives that can provide efficient and practical real-time

intrusion detection and security policy compliance monitoring. Our research attempts to strike a balance between tightly coupled and loosely coupled IDS architectures in an effort to utilise the benefits of each in a single system. As a result, we have produced concepts that rely on tightly coupling hardware, while minimising software coupling as much as possible. This can decrease the vulnerability of the monitoring hardware itself, improve time-to-detect, and decrease communication overhead compared to CuPIDS-Purdue. Finally, we would like to thank the reviewers whose suggestions were incorporated into the paper/presentation and the Air Force research laboratory's anti-tamper software protection initiative technology office as well as the Air Force office for scientific research for supporting this work. The opinions expressed in this paper are those of the authors, and do not necessarily reflect the views of the US Air Force or the US Government.

References

Arora, D. (2005) "Secure Embedded Processing Through Hardware-Assisted Runtime Monitoring", Design, Automation and Test in Europe Conference and Exhibition

Computer Emergency Response Team (CERT) (2006) "CERT/CC Statistics 1988 2006", [online], http://www.cert.org/stats/cert stats.html

Feng, H. (2003) "Anomaly Detection Using Call Stack Information", SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy

Intel Corporation, "Ia-32 Intel architecture software developers manual volume 3: System Programming Guide", Intel Corp

Kiriansky, V. (2003) "Secure Execution Environment via Program Shepherding," Master's thesis, Massachusetts Institute of Technology

Lee, B. (2003) Enlisting Hardware Architecture to Thwart Malicious Code Injection, Security in Pervasive Computing, Springer Berlin, Berlin, pp. 237–252.

Molina, J. (2003) "Using Independent Auditors as Intrusion Detection Systems", Information and Communications Security: 4th International Conference

Petroni N. (2004) "Copilot - A Coprocessor-Based Kernel Runtime Integrity Monitor," Proceedings of the 13th USENIX Security Symposium, USENIX

Ragel, R. (2005) "Micro embedded monitoring for security in application specific instruction-set processors," Proceedings of the 2005 international Conference on Compilers, Architectures and Synthesis for Embedded Systems

Shi, W. (2005) "An intrusion-Tolerant and Self-Recoverable Network Service Using A Security Enhanced Chip Multiprocessor," CAC'05

Williams, P. (2005) CuPIDS: Co-Processor based Intrusion Detection System, PhD thesis, Purdue University, West Lafayette, IN

Zhang, T. (2005) "Anomalous Path Detection With Hardware Support", CASES'05

Zhang, X. "Secure Coprocessor-based Intrusion Detection", IBM Corp.

Graphical Based User Authentication with Embedded Mouse Stroke Dynamics

Kenneth Revett[1], Asim Zia[1], Sérgio Tenreiro de Magalhães[2], and Henrique Santos[2]
[1]University of Westminster, London, UK
[2] Universidade do Minho, Portugal
K.Revett@westminster.ac.uk
hsantos@dsi.uminho.pt

Abstract: Research on character based login details reveal that they tend to be simple and relatively easy to guess. As an alternative, graphical user authentication schemes have been suggested, as research in cognitive psychology indicates that images are more memorable than characters. We combine this advantage that graphical passwords provide, along with an added layer of security which incorporates mouse stroke dynamics. In this paper, we examine a graphical user authentication scheme with respect to length as a function of embedded noise and image size to determine the optimal values for these parameters. We then add a mouse stroke dynamics biometric with the optimal graphical password length and graphic image size. The results indicate that a ratio of 1:4 may be optimal with respect to the number of elements in a graphical password and the total number of graphics presented to select from. The size of the images did not change the FRR. Lastly, mouse stroke dynamics did not produce any change in the FRR, but reduced false acceptance to levels approaching keystroke dynamics, depending on the acceptance threshold criteria employed. These results are quite promising – with FRR/FAR values in the range of 1-4% - consistent with the most accurate biometric systems reported for this technology. These results indicate that graphical based authentication schemes (with embedded mouse stroke dynamics) are viable alternatives to character based password schemes. In addition, our technology could be directly applicable to a mobile computing device that required a stylus for interaction.

Keywords: Graphical authentication, mouse stroke dyanmics, biometrics, false acceptance, flase rejection

1. Introduction

Traditionally, most computer systems are protected through the installation of some form of user authentication details. These of course include the traditional username and password sequence. Even if users have only one or two accounts, many fail to realise that the entire security of the computer system they are a entrusted to use is dependent on the rigour of their selected login details. Generally speaking, most users generate simple and easy to remember login details, which may be potentially easy to crack with sophisticated dictionaries. In one often-cited case study of 14,000 Unix passwords, almost 25% of them were found by searching the words from a carefully formed "dictionary" containing only 3 $x10^6$ words (Jerym et al., 1999). Since the password space of an 8-character password is on the order of 2×10^{14}, this indicates that only a small subset of the character space is utilised for a large percentage of users. The principal reason for this lack of rigour in the selection of usernames/passwords is the difficulty of remembering them – especially if a user has many password-protected accounts. The question for IT managers then is how to ensure that users select passwords that are more difficult to crack? One possible solution, first proposed by Blonder in 1995 is the deployment of a graphical password (Blonder, 1996).

It has been reported in the cognitive science literature that image based systems (such as those proposed for graphical passwords) are easier to remember than text-based passwords (Suo et al., 2002, de Magalhaes et al., 2006a). This result implies that users can select more complicated passwords if they are formed from a collection of images/pictures. If the image space is sufficiently large, then the password space is on the same order if not larger than that of character based systems. In addition, with the ubiquitous deployment of hand help computing devices such as PDAs, which present a clearly graphical interface, the deployment of a graphical password based authentication system is a natural one.

There are two basic modes of operation with graphical passwords: recall based and recognition based systems. The most prevalent graphical password systems are recognition based. Dhamija and Perrig (Dhamija & Perrig, 2000) proposed a graphical authentication system based on the Hash Visualisation algorithm (Perrig & Song, 1999). In this system, the user is asked to select a certain number of images from a set of random pictures generated by a program (Figure 1). For authentication, the users are required to click (select) the images that they chose for their password. The results showed 90% of all participants succeeded using their graphical authentication system, only 70% succeeded using text-based passwords and pins (Dhamija & Perrig, 2000).

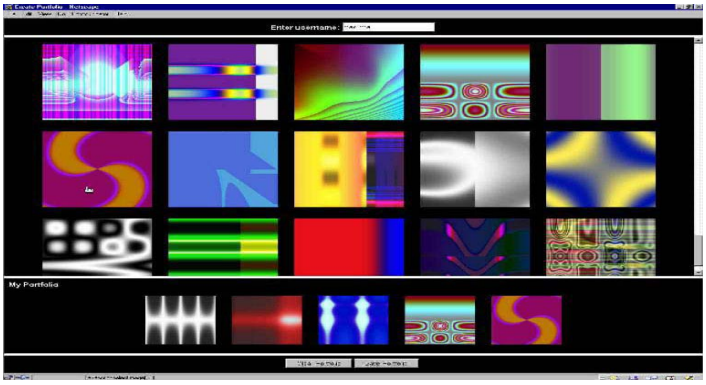


Figure 1: A sample of randomly selected images from the Hash Visualisation of Dhamija and Perrig. The images in the bottom panel are those that will be matched to the random images displayed in the top panel.

Another popular recognition based graphical authentication system is Passfaces™, developed by real user

Figure 2: The Passfaces™ graphical authentication system is based on selecting a set of four faces for their password. A user selects a single face from four sets of nine face panels.

Corporation (www.realuser.com). In this system, a collection of faces is presented and the user creates their password by selecting a series of images from the face database (see Figure 2). This approach relies on the assumption that people have an almost innate capacity to recognise faces easier than they can recognise images. Valentine (Valentine, 1998, Valentine, 1999) has presented strong evidence through a set of longitudinal studies that Passfaces™ are memorable over very long intervals. Further studies indicate that the login failure rate (FRR) of Passfaces was a third less than conventional text-based passwords. There are many more examples of recognition-based graphical authentication schemes and the interested reader is referred to (Sobrado & Birget, 2002, Thorpe & Oorschot, 2004a, Paulson, 2002, Wiedenbeck et al., 2005a,2005b) for more details on this topic.

The principal recognition based graphical authentication system is Draw-A-Secret (DAS), proposed by Jerym (Jerym et al., 1999). In this system, the user draws a unique password using a stylus or mouse (see Figure 3). For the authentication the user is required to re-enter their drawing within the same coordinate frame, which is generally a 5x5 grid (see Figure 2). Thorpe and van Oorschot provide evidence that the password space for a 5x5 grid is larger than the password space for an 8-character text based password (Thorpe & Oorschot, 2004b). Syukri has proposed a system where the user is required to draw their signature using the mouse (Syukri et al., 1998). Again, the user's input is digitised into a coordinate space, which is stored as the BIR (biometric information record) for their password. There are several other recall-based techniques and the interested user can consult (Thorpe & Oorschot, 2004, Paulson, 2002) for further details on these systems.

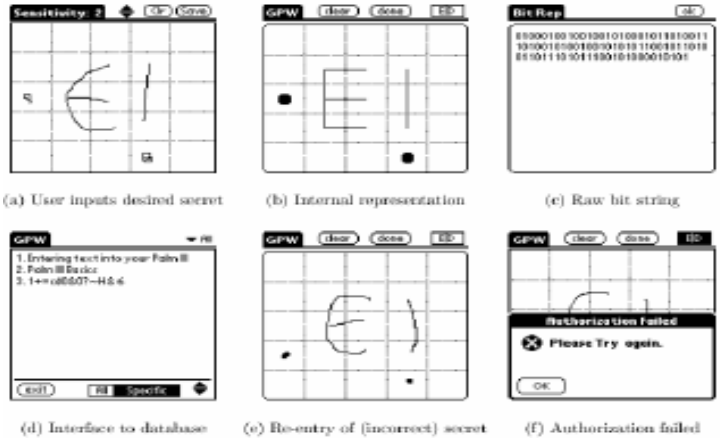


Figure 3: Draw-a-Secret user interface. The user enters their freehand drawing (see top left panel) which is then digitised (top middle panel) into a set of (x,y) coordinates.

The graphical based systems presented in this paper contain the principal elements reported in the literature. Their principal driving force is that graphical (image based) passwords are easier to remember than text-based passwords, so users are expected to select more complicated passwords. If the search space is equivalent to that of text-based passwords – which most are – then this will enhance the security level of a graphical based passwords. In addition, cracking such a system may be more difficult than text based passwords – where there is an abundance of useful "dictionaries" which have been used successfully in many cases. Yet, as the awareness of graphical based passwords spreads, the sophistication of the attacking mechanism(s) spiral upward(s) in a predator-prey fashion. Our solution to this dilemma is the addition of a biometric element in *conjunction* with a graphical password based authentication system.

We have developed a prototype graphical based authentication system called MouseLock. It is a recognition based authentication system that incorporates a biometric

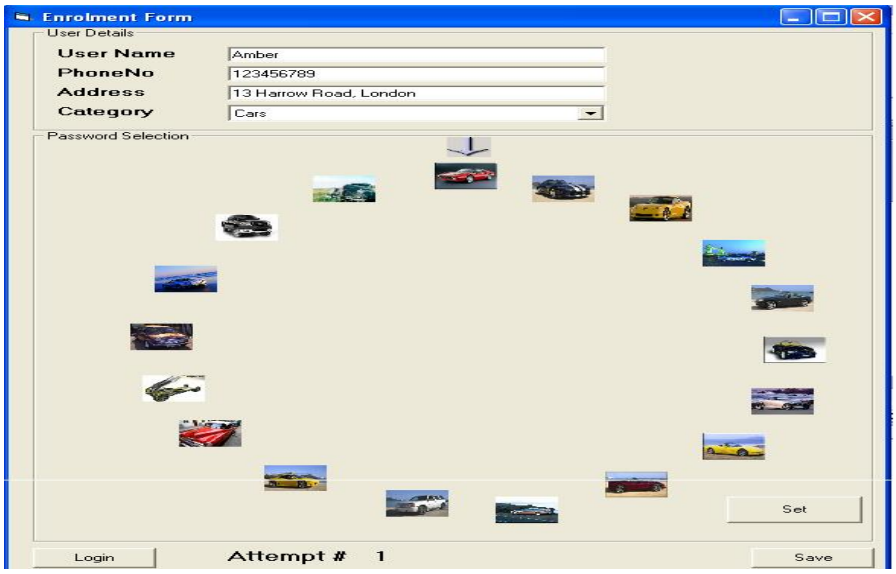


Figure 4: The enrolment form for the MouseLock graphical authentication system.

The user is required to select a minimal number of images for their password. The user selects an image and then must move it to the top pointer by selecting either the Left/Right button, just as is done with a conventional safe combination. based on mouse dynamics. The basic enrolment form of the system is presented in Figure 4. The user must select a minimal number of images for their password (typically 6) and then decide if the image should be moved to the pointer (top if the circle of images) in either the clockwise (left mouse click) or the anti-clockwise (right mouse click) direction. This system is very similar to a safe – where the user must move the dial with numbers into the appropriate direction. In the case of MouseLock – the order of image selection can be made superfluous – this decision is made during the enrolment process. This allows MouseLock to simulate the temporal/spatial independence found in the DAS system (Jerym et al., 1999).

Since the user interacts with MouseLock via a mouse or stylus, we can capture some basic biometric information. For instance, in our prototype, we capture the mouse clicks when they select an image. We use the first mouse click as time 0 – and then calculate the time between successive clicks (to the nearest ms). In addition, the total time taken to enter their password is stored. The user enrolls into the system (typically ten trials) and basic statistical information is acquired, which is stored in the BIR for this particular user account. This information is used during the authentication process to decide whether the user should be authenticated. Note that there are a number of additional attributes that can be extracted from this system, which can be used in the authentication process (see de Magalhaes et al., 2006b). In addition, it should be noted that the images in Figure 4 are from a single category – cars. There is a range of categories – and there is also the ability to randomly select images from the picture database. In the next section, we present a series of experiments that were performed to evaluate the effectiveness of MouseLock.

2. Results

In this preliminary study, we evaluated the content of the pictures on the FAR/FRR and the signal to noise ratio. In the first experiment, we selected a series of three image categories and compared that to a random selection of images from the same three categories. The three image categories were: cars, flowers, faces, and a combination selected randomly from each category. What we wanted to determine is whether category type influenced the recognition ability of the user (whether it increases FAR/FRR). Our database has 1,000 images for each category and we selected the required number from each category for the enrollment and authentication process. We had a total of 4 users for this system – each logging into their own accounts 50 times in order to provide an estimate of the FRR. We employed the same image categories in this

Table 1: Summary of FAR/FRR results for four users. The 'FAR/FRR 1' corresponds to the car category, 'FAR/FRR 2' the flower category, 'FAR/FRR 3' the people category, and 'FAR/FRR 4' the mixed category.

	Car	Car	Flower	Flower	People	People	Mixed	Mixed
User	FAR 1	FRR 1	FAR 2	FRR 2	FAR 3	FRR 3	FAR 4	FRR 4
1	3.5%	2.2%	4.9%	4.0%	1.7%	1.9%	3.7%	3.1%
2	2.8%	1.6%	2.2%	2.5%	3.9%	0.9%	1.1%	2.8%
3	1.7%	1.0%	4.7%	3.6%	3.0%	1.4%	1.0%	1.5%
4	2.7%	1.4%	3.8%	3.2%	4.6%	2.3%	1.3%	1.9%
Mean	**2.7%**	**1.2%**	**3.9%**	**3.3%**	**3.3%**	**1.6%**	**1.8%**	**2.3%**

Note these results are from 50 trials for each of the 4 accounts. These results were obtained using the mouse dynamics data.experiment that was employed that the "wheel" can vary in the number of images that it can hold – we used the default value of 18 images. The users were required to select 6 images for their passwords and enroll ten times. The statistical attributes acquired during enrollment to generate the data presented in Table 1. We also examined the impact of the mouse dynamics – to determine the significance of this feature on the authentication results. This was performed in a retrospective manner – by re-assessing the authentication results without the addition of the mouse dynamics component. These results are depicted in table 2.

Table 2: This data was generated from the same data presented in Table 1 – except that no mouse stroke dynamics or other non-image selection criteria were employed. The FRR X columns correspond to the same categories presented in table 1 (Car, Flower, People, Mixed).

User	FRR 1	FRR 2	FRR 3	FRR 4
1	3.3%	3.8%	2.2%	4.2%
2	3.5%	3.4%	2.3%	3.2%
3	4.1%	2.4%	4.2%	3.4%
4	2.4%	3.6%	3.8%	4.6%
Mean	**3.4%**	**3.3%**	**3.1%**	**3.9%**

In addition, we examined the signal-to-noise ratio. To do this, we varied the number of images that were presented to the user during enrollment and authentication. More specifically, we varied the number of images in increments of 6 from 12 to 30 (12,18,24, & 30). This was performed in two different ways: the images occupied a constant circumference, which meant the image size was reduced as the number of images increased. In the other –the image size remained constant and hence the circumference increased with increasing number of images. We kept the number of images in the password constant at 6. This experiment was designed to determine if increased noise (i.e. increased number of non-password images)

would have an impact on the FRR/FAR. The results from th experiment with a constant circumference is presented in Table 3 and that for a variable circumference are presented in table 4.

Table 3: This data represents the results from 4 users (each logging into their own and the other accounts 50 times each). The 12-30 in the first row represent the number of images from which the user was required to select the password. For this experiment, the circumference was held constant, and so each image was reduced in size accordingly. Note that the mouse dynamics was activated during this experiment. Note the trend was for the 24 FRR column had the smallest FRR value (not statistically significant though $p < 0.07$).

User	12 FAR	12 FRR	18 FAR	18 FRR	24 FAR	24 FRR	30 FAR	30 FRR
1	2.4%	1.5%	1.5%	2.1%	1.3%	1.9%	2.1%	3.5%
2	1.2%	2.8%	1.0%	2.0%	2.0%	1.8%	2.9%	3.4%
3	1.6%	3.5%	2.4%	2.6%	2.6%	2.2%	1.5%	2.8%
4	1.5%	2.7%	1.7%	3.2%	3.3%	2.0%	1.8%	3.4%

Table 4: This data represents the results from 4 users (each logging into their own and the other accounts 50 times each). The 12-30 in the first row represents the number of images from which the user was required to select the password. For this experiment, the circumference was allowed to vary so each image size remained constant and the circumference increased with increasing number of images. Also note that the mouse dynamics was activated during this experiment.

User	12 FAR	12 FRR	18 FAR	18 FRR	24 FAR	24 FRR	30 FAR	30 FRR
1	2.1%	1.9%	2.2%	1.9%	1.8%	2.0%	2.5%	2.4%
2	2.2%	3.2%	1.0%	2.5%	2.0%	1.0%	1.8%	2.8%
3	1.8%	2.2%	2.5%	2.6%	1.2%	2.4%	1.0%	2.0%
4	2.1%	2.6%	2.5%	3.0%	1.0%	2.1%	1.2%	1.8%

3. Discussion

The results from this preliminary study indicate that using a graphical password scheme without any biometrics added yields an FRR of approximately 3% (see Tables 1 and 2). We employed a set of image categories such as cars, flowers, faces, and a combination selected randomly from each category. The results in Table 1 indicate that there was no significant difference for the FRR based on category ($p = 0.10$). This result indicates that at least for a six-image password, the category used in this study had no significant impact on the ability of the users to select their passwords correctly. This result of course may not hold across all image classes, although the last category – which consisted of a random selection, was not significantly different from any of the other categories. The addition of a collection of mouse dynamics did significantly impact the FRR – and yielded an FAR on the order of 2% - a value comparable to "traditional" text based biometric algorithms (de Magalhaes et al., 2006, Revett et al., 2006). The mouse dynamics that we included were based principally on click times – generating a signature of how each user selected their password elements, along with the total time and the mid-point time. We generated the mean, standard deviation (SD) and the coefficient of variation (CV) for each of these elements and stored them in the BIR with the password entry. The acceptance/rejection selection criterion was statistical in nature – where the values collected during authentication must be within the mean +/-CV*α where α ranged from (0.5 – 1.0).

We also investigated how noise – in the form of the addition of non-password images impacts the FAR/FRR rate. In Table 3, we present data from an experiment where the total number of images varied from 12-30 (in increments of 6), with a constant password length of 6 images (the same password was used throughout). The results indicate that the FRR was not significantly affected – whereas the FAR decreased with increasing noise (number of images). In addition, we examined the effect of image size on the FRR/FAR. If our system is to be employed on mobile computing devices, images size may have an impact. Mobile phones obviously have a small screen size – which places a limit on the number of images that can be displayed on the screen. The results indicate that image size (constant circumference versus variable circumference – constant image size) had no significant impact on the FRR rate. There was a statistically insignificant increase in the FRR rate as the signal-to-noise ratio increased. This effect may be dependent on image class – and this effect will be examined in more detail in future studies. The FAR rate was also effected – but not in a statistically significant manner in this experiment. It may be that a larger study may be required – both in terms of subject numbers and image selection category. This effect will also be examined in future work.

In summary, this study provides a very direct measure of the impact of mouse dynamics and its impact on a graphical password based authentication system. The FRR – a measure of the repeatability of the password was not significantly affected by the introduction of mouse dynamics. In addition, there was only a small but statistically insignificant effect produced by the addition of noise to the password selection scheme. This effect is important as it provides an operating range for the number of images that can be successfully displayed. This is a critical factor in small mobile computing devices such as mobile phones. The FRR rates were very low in these experiments – comparable to those obtained using more traditional keystroke dynamics based authentication schemes (Revett et al., 2006). In order to examine whether these results are robust, we are in the process of repeating these experiments with a group of 20 subjects. We will publish those results when they become available.

References

Blonder, G.E., *Graphical password*, U.S. Patent Number 5.559.961, 1996.

de Magalhães, S.T., Revett, K., and Santos, H.: "Generation of Authentication Strings From graphic Keys," International Journal of Computer Science and Network Security, vol 6(38), March 30, 2006a, ISSN 1738-7906.

de Magalhães, S.T., Revett, K., and Santos, H., "Keystroke Dynamics - Stepping forward in Authentication", GESTS International Transactions on Computer Science and Engineering, Volume 29, Number 1, pp. 113-119, March 2006b. ISSN: 1738-6438. Volume's ISBN: 89-953729-5-8.

Dhamija, R. and Perrig, A., "Deja Vu: A User Study Using Images for Authentication," in *Proceedings of 9th USENIX Security Symposium*, 2000.

Jermyn, I., Mayer, A., Monrose, F., Reiter, M.K., and Rubin, A. D., "The Design and Analysis of Graphical Passwords," in *Proceedings of the 8th USENIX Security Symposium*, 1999.

Paulson, L.D., "Taking a Graphical Approach to the Password," *Computer*, vol. 35, pp. 19, 2002.

Perrig, A. and Song, D., "Hash Visualization: A New Technique to Improve Real-World Security," in *Proceedings of the 1999 International Workshop on Cryptographic Techniques and E-Commerce*, 1999.

RealUser, www.realuser.com, last accessed 10 November, 2006

Revett,K., de Magalhães, S.T., & Santos, H., Enhancing Login Security Through the Use of Keystroke Input Dynamics, IAPR International Conference on Biometrics 2006, Hong Kong, 5-7 January, 2006 (ISBN3-540-31111-4)

Sobrado, L. and Birget, J.-C., "Graphical passwords," *The Rutgers Scholar, An Electronic Bulletin for undergraduate Research*, vol. 4, 2002.

Syukri, A.F., Okamoto, E., & Mabo, M., A User Identification System using Signature Written with Mouse, Third Australasian Conference on Information Security and Privacy (ACISP): Springer-Verlag Lecture Notes in Computer Science (1438), pp. 403-441, 1998.

Suo, X., Zhu, Y., & Owen, G. S., Graphical Passwords: A Survey, Department of Computer Science, Georgia State University, 2002.

Thorpe, J., and v. Oorschot, P. C., "Graphical Dictionaries and the Memorable Space of Graphical Passwords," in *Proceedings of the 13th USENIX Security Symposium*. San Deigo, USA: USENIX, 2004a.

Thorpe, J., & v. Oorschot, P.C., Towards Secure design Choices for Implementing graphical Passwords, 20th Annual Computer Security Applications Conference (ASCAC), Tucson, USA, IEEE 2004b.

Valentine, T., An evaluation of the Passface personal authentication system, Technical Report, Goldsmiths College, University of London, 1998

Valentine, T., Memory for Passfaces after a long delay, Technical Report, Goldsmiths College, University of London, 1999

Wiedenbeck, S., Waters, J., Birget, J. C. , Brodskiy, A., and Memon, N., "Authentication using graphical passwords: Basic results," in *Human-ComputerInteraction International (HCII 2005)*. Las Vegas, NV,2005a.

Wiedenbeck,S., Waters, J., Birget, J.C., Brodskiy, A., and Memon, N., "Authentication using graphical passwords: Effects of tolerance and image choice," in *Symposium on Usable Privacy and Security (SOUPS)*. Carnegie-Mellon University, Pittsburgh, 2005b.

Planning Cost-Effective Deceptive Resource Denial in Defense to Cyber-Attacks

Neil Rowe
Cebrowski Institute, U.S. Naval Postgraduate School, CA, USA
ncrowe@nps.edu

Abstract: Cyber-attacks against computer systems that provide valuable services can often be effectively defended by tactics of deliberately deceptive resource denial. Delaying in response to suspicious requests is one example; it permits time to develop a good defense, facilitates analysis of the attacks and formulation of a response, and may little affect legitimate users. But delays can look suspicious; a better tactic can be for the operating system to falsely claim unavailability of some critical resources that the attacker needs (files, directories, access rights, network connections, or software). This can be more effective than using "security policy" as an excuse to deny those resources because it is unexpected and more flexible. We formulate a decision-theoretic approach to the problem of deciding when to deceive by resource denial in a sequence of interactions with a user of an operating system, and provide general formulae for decisions in planning deceptions. Our theory covers both reactive and proactive deception, and both single-session and multi-session attacks. We also provide additional criteria to ensure logically consistent tactics. We provide some evidence from a survey of users to support our modeling.

Keywords: Deception, cyberspace, decision theory, resources, denial, lies

1. Introduction

Rapid cyber-attacks can often be best defended by delaying tactics. Delaying facilitates analysis of the attacks and formulation of a response, particularly for a new kind of attack. This is valuable for systems like those of the critical national infrastructure that must be kept running. At the same time, delays will rarely affect legitimate users if they are only applied after suspicious behavior is observed. Delays can be accomplished by waiting to respond to commands, as in "wrappers" around operating-system commands that delay when informed of suspicious behavior by an intrusion-detection system (Rowe, 2004). But delays by themselves look suspicious; a more sophisticated delay is wasting of a user's time. This can be done by an operating system falsely claiming unavailability of resources that the attacker wants (files, directories, access rights, network connections, and software) at some time into a session. This can be more effective than denying access rights based on security policy because it works against insiders with privileges as well as outsiders, discourages less a repeat of the same (foiled) attack, and encourages underestimation of defenses by the attacker. They may later find they have been deceived, but it will take them some time. By then the system administrators can be aware of the attack, and vulnerable software and ports can be shut down. False resource denial is a form of deception. Deception is an important human strategy and tactic to accomplish goals involving manipulation of people (Ford, 1996). Deception, used sparingly, is often more cost-effective social manipulation than direct attempts to influence people, as is attested to by its many dramatic uses in military history. However, deception has a price: People do not like being deceived. If they discover deception, they may react strongly and negatively. Furthermore, legitimate users of a computer system who are deceived about resource availability may be unable to accomplish their goals, so deception may increase the cost of doing legitimate business. Thus we need a model to analyze the tradeoffs involved in deception. While there are ethical concerns about deception, most ethical theories permit deception to prevent a significantly greater harm, and destruction of the software of a computer system is a serious harm.

After surveying previous work, we build a decision-theoretic model of online deception in section 3. We show how to introduce probabilities and costs (section 3.1), and argue for statelessness of the analysis (section 3.2). We then evaluate anticipatory deception (section 3.3) and multi-session deception (section 3.4). We then point out issues in logical consistency of deceptions (section 4) as a way to make them more convincing. We conclude with some observations and directions for future work.

2. Previous work

Deception is a common social phenomenon. People use deception frequently for worthwhile ends (Nyberg, 1993), and it would be impossible for societies to function without it in many areas such as law, politics, business, entertainment, and psychology. Attackers of computer systems use deception themselves in their identities and software tools, so it would seem fair to use deception in defending our systems too. Deception has the advantage of being a generally unexpected defense for computers, and one that can be difficult for attackers to perceive.

Deception has been used effectively for "honeypots", computer systems intended solely for collection of data about attacks (The Honeynet Project, 2004). For defenders to collect useful data with a honeypot, the attacker must think that the system is an ordinary one. To this end, it helpful to create fake files and data on the honeypot to suggest that real users have been using it in normal ways (Cohen & Koike, 2003; Gerwehr et al, 2000). Such fake data can include passwords to other machines, credit-card numbers, and other things for which we can confirm use. A variety of models of deception have been constructed for psychology (Heuer, 1982) and counterintelligence (Whaley, 1982). Recent computational models include the descriptive models of (Cohen, 1999), the linguistic speech-act model of (DeRosis et al, 2003), agent-based deception-planning models (Christian & Young, 2004), and reputation-systems assessment of deceptiveness (Barber & Kim, 2001). Our previous work built probabilistic models of belief in "generic excuses" for why the attacker could not achieve their goals, plus the belief of the attacker in the hypothesis "I am being deceived" which we do not want them to have (Rowe, 2004); excuses are a versatile strategy for refusing to do things (Snyder, Higgins, & Stucky, 1980).

3. Choosing if and when to deceive

In what follows we assume a computer system is under attack by a relatively alert attacker who is trying to achieve particular goals (Chirillo, 2001). These are reasonable assumptions for insider attacks, state-sponsored or organization-sponsored outsider attacks, and sophisticated hackers. At the same time, our deceptive responses will be unexpected events for any attacker, and thus will easily stop most automated or scripted attacks. We can initiate deceptions about resource availability at many possible times in a sequence of commands. However, some opportunities are better than others because (a) some deceptions are more convincing, and (b) some deceptions cause less harm to legitimate users who are accidentally suspicious. We will use a decision-theoretic model because this has been effective in understanding deception in sociobiology (Lachmann & Bergstrom, 2004). A range of strategies for using deception to defend computer systems are possible (Rowe & Rothstein, 2004). One major strategy is to try to "scare away" the attacker by making the system seem dangerous to attack. This can be done, for instance, by making a system appear to be a honeypot when it actually is not (a "fake honeypot"), because most attackers know honeypots record their actions and are difficult to exploit (Rowe, Duong, & Custy, 2006). Another strategy is to encourage the attacker to leave by making it seem like the attack target is not exploitable, as when key features necessary for their attack are absent. This can be done, for instance, by telling the attacker that the network is down so that they think they cannot download their attack tools. Lies like this about system resources are generally simple and easy to accomplish, and hard for an attacker to disprove. We investigate this second strategy in this paper.

3.1 Incorporating probabilities and costs

For the simplest model (Figure 1), we assume that the user is malicious with probability p_m, a malicious user will continue attacking the system after the deception with probability p_{mr}, and a legitimate user will continue working after the deception with probability p_{lr}. The first will be obtained from an intrusion-detection system (Monteiro, 2003) and will vary based on user behavior. The others will be personality- and task-dependent because some users are just more determined than others (Lydon & Zanna, 1990). Assume also the cost of allowing an attack to succeed is c_m, and the cost of a legitimate user not achieving their goals is c_l. Costs can be time, but could include subjective factors like aggravation too.

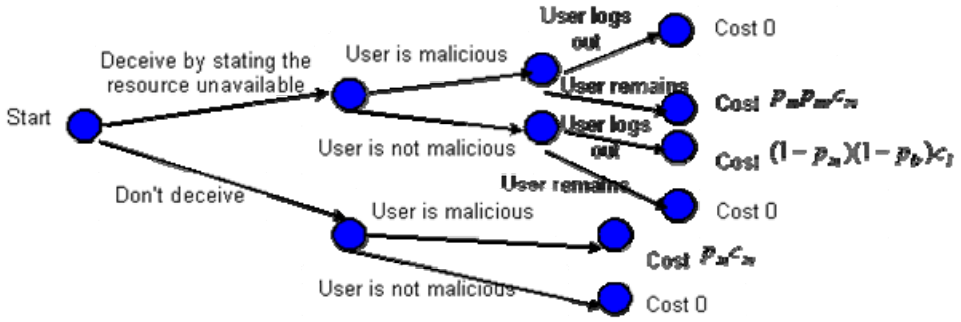


Figure 1: The simplest model of the decision to deceive at some point in a command

Then the expected cost of deceiving will be less than the cost of not deceiving if

$p_m p_{mr} c_m + (1 - p_m)(1 - p_{lr})c_l < p_m c_m$.

If we define

$g(x) = x/(1+x)$

this can be written as

$p_m > g[c_l(1 - p_{lr})/c_m(1 - p_{mr})]$

This g is the function that converts odds to probabilities, so the expression is brackets is analogous to an odds measure. The inequality says that if a deception is completely ineffective in encouraging the malicious user to go away, it is never desirable; if it completely ineffective in encouraging the legitimate user to go away, it is always desirable; and if the deception no more affects the attacker's goals than the legitimate user's goals, the inequality reduces to

$p_m > c_l/(c_l + c_m)$

so even then the deception may be desirable. For example, suppose we lie that the network is down (nonfunctional) in response to an attack involving attempted rootkit installation. Assume the cost of a successful rootkit installation is 2 hours of administrator work to reinstall the operating system and fix bugs. Assume the cost to a legitimate user of the network being down is 5 minutes of work because they can copy to a portable storage medium to do local file transfers. Because network file transfer is essential to most rootkit installation methods, a malicious user will have a low probability of continuing their attack after this deception, say 0.1. On the other hand, legitimate users probably have other tasks not needing the network such as editing, so their probability of continuing is something like 0.5. Therefore

$p_m > g[(5*0.5)/(120*0.9)] = 0.023$

and we should deceive if the user has at least 2.3% chance of being malicious.

We conducted a survey (see Appendix) to estimate some key probabilities for this analysis. The survey was administered to seven students and five faculty members at our school. Geometric means of the response values for each subject ranged from 0.187 to 0.000002, but *ratios* of responses between questions were much less variable, indicating that conditional probabilities were more consistent across subjects than a priori probabilities (which may reflect personality). The survey confirmed that deception was considered a possible explanation for rare events on a computer system, with conditional probabilities of 0.32, 0.54, and 0.09 for deception with the reported conditions "network down", "local network down", and "network messing up files". Thus subjects understood that while deception is rare a priori, it is not conditionally rare. Our subjects also were quite persistent in trying the same action again when it failed, so p_{mr} and p_{lr} should be close to 1; this suggests we can often use the approximate criterion

$p_m > c_l/(c_l + c_m)$.

3.2 Refining the deception model

A useful tool is:

> *The Statelessness Assumption: The resources of a computer system remain constant in their availability status through a session of a user.*

This is often reasonable because most sessions are just minutes long. Then the above model implies we should wait as long as possible to deceive an attacker to be more sure that they are malicious. However, (Josang, 2001) suggests that distrust grows more easily than trust; waiting gives the attacker time to become distrustful. (Rowe, 2004) proposes that the believability of a deception for an attacker is a product of the a priori likelihood of the deception event, the a priori likelihood that the system does not engage in deception, and the intrinsic suspiciousness of what the attacker has done previously. Thus attackers engaged in suspicious behavior are more suspicious of unusual events that thwart them on an important system. We will postulate that attackers recognizing deception will not log out because they figure they can find a way around it, whereas unsuspicious attackers will log out because deceptions like “network down” can be chosen to suggest impossibility of most attacker plans.

This gives us enough constraints to create a Bayesian model (Korb & Nicholson, 2004) of p_{mr}, the probability the malicious user remains logged in after a deception that would imply that they cannot achieve their goals. Using Bayes' Rule where d is the occurrence of a deception action and c is the condition that the attacker believes they are being deceived,

$p(c \mid d) = p(d \mid c)p(c)/p_d$ and $p(\neg c \mid d) = p(d \mid \neg c)(1 - p(c))/p_d$,

So if
$p_{mr} \approx p(c \mid d)/(p(c \mid d) + p(\neg c \mid d))$, $p_{mr} = g[p(c)p(d \mid c)/(1 - p(c))p(d \mid \neg c)]$.
Thus for the deception "network down" and the probabilities
$p(c) = 0.01$, $p(d \mid c) = 0.1$, and $p(d \mid \neg c) = 0.01$,
we estimate $p_{mr} = 0.9$. On the other hand, we assume p_{lr} is independent of circumstances because many deceptions can be chosen to be in features inessential to a legitimate user.

The p_{mr} can also be calculated even when statelessness is not assumed. For example, the condition "network down" is around 0.02 on computer systems the author uses; but if the network was down one hour ago, it is likely to be down now. A down period for the author's network has an expected duration of one day. In general, we can use a Poisson model much like that for M/M/1 queuing models, where λ_D is the number of times that the condition D would legitimately occur sometime during a day (the arrival rate for a queue), and λ_U is the number of times that D would be remedied during a day (the service rate). Then if we reported to the user that D was true at some time, the probability that D is still true at a time t units later is $p_U = e^{-\lambda_U t}$; if we reported to the user that D was false at some time, the probability that D is still false at time t is $p_D = e^{-\lambda_D t}$.

Another reason to avoid waiting too long to deceive is that the number of available deceptions decreases during a session as resource availabilities are confirmed if we maintain statelessness. For instance, "network down" is less convincing when the user has encountered unrelated software problems already. This is similar to the factor of “mobility” in chess, where of two positions with equal piece count and danger, the one with more possible moves is better because it is less likely to lead to a forced bad move. We can incorporate a mobility factor in our decision model if we choose.

3.3 Anticipatory deception

We now analyze anticipatory deception, where we deceive when we are mildly suspicious to permit a consistent deception later if necessary. For instance, we can say the network is down in response to an attempt to transfer an innocuous file, in anticipation of a later attempt to transfer a rootkit. The first deception enables consistency of the "network down" hypothesis and reduces attacker suspicion. If we deceive more than once, we should use the same deception each time because deceptions will be rare and two unrelated deceptions will be particularly suspicious. Thus generic excuses like "network down" are useful as deceptions because they explain failures of many separate things.

We now refine our decision model (Figure 2). Here we distinguish p_{m1}, the probability the user is malicious at the first deception, from p_{m2}, the probability at the second deception. We similarly distinguish p_{mr1}, the probability a malicious user will remain after the first deception, from p_{mr2}, the probability after the second. We also distinguish p_{lr1} from p_{lr2}, and both from p_{mr2d} and p_{lr2d}, the probabilities of the malicious or legitimate user remaining when the first deception is not done but the second deception is done. We can assume that $p_{lr1} = p_{lr2} = p_{lr2d} = p_{lr}$ because if deceptions are designed with attackers in mind, each is equally likely to make a legitimate user give up and go away. We also assume that $p_{mr2d} > p_{mr2}$ and $p_{mr2d} > p_{mr1}$, because the "mr2d" case most obviously suggests deception, and apparent deception should make a malicious user want to remain. We can then calculate expected costs for three strategies for a particular deception D in the course of a short session (else include the exponential decay):

- S1: No deception: $p_{m2}c_m$
- S2: Deceive only at the second opportunity: $p_{m2}p_{mr2d}c_m + (1 - p_{m2})(1 - p_{lr})c_l$
- S3: Deceive at both opportunities:
  $p_{m2}p_{mr1}p_{mr2}c_m + (1 - p_{m1})(1 - p_{lr})c_l + (1 - p_{m2})(1 - p_{lr})p_{lr}c_l$

Note it is useless to deceive only at the first opportunity, as this does not foil an action necessary to the attack, creates a suspicious inconsistency, and risks antagonizing legitimate users.

There are no simple formulas comparing the strategies, but there are special cases. As before, S2 and S3 will be suboptimal if the malicious user is never fooled, because then $p_{mr2d}=1$, $p_{mr1}=1$, and $p_{mr2}=1$, hence the first terms of the expressions for S2 and S3 are the entire expression for S1. Usually, however, p_{mr2d} will be close to 1 because an attacker will find it hard to believe if they are paying attention, and will be equal to 1 if the attacker is not paying attention, so S2 will be usually undesirable compared S1.

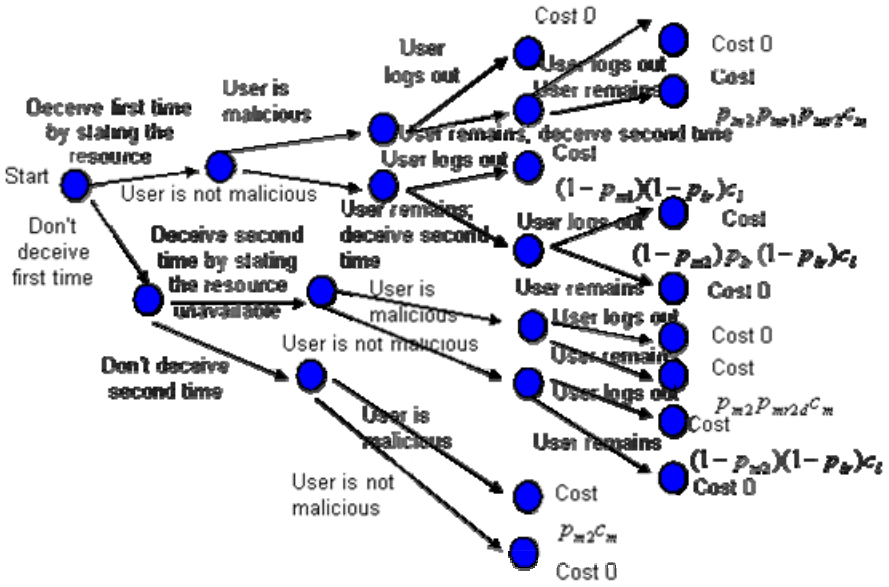


Figure 2: Analysis of anticipatory deception

3.4 Multi-session reasoning

Thusfar we have assumed a single-session attack. However, an attacker could log in multiple times as in "low and slow" attacks. We can make a small change to the above model to handle this. Each pair of sessions can have an associated probability that they are due to the same user. If we trust our user authentication, this is 1 for sessions under the same name and 0 otherwise. However, attackers can masquerade as legitimate users, and this can be suspected when measured by anomaly-focused intrusion-detection systems.

Again we can use the exponential model. A user logging in again after time t after logging out under a different name is likely to be the same user with probability $p_S = p_S e^{-\lambda_{Sm} t}$ if they are malicious and zero if they are nonmalicious, where λ_{Sm} is the malicious-user "decay" rate and p_S is the probability that a malicious user logging out will immediately log in again in a unit time interval. If R users logged out recently, each with probability of maliciousness p_{mi} and at time t_i before the present, the probability a new user is now malicious is $p_{m0} + \sum_{i=1}^{R} p_S p_{mi} e^{-\lambda_{Sm} t_i}$ where p_{m0} is the a priori probability a new user would be malicious. But this is only an initial value; once the new session begins, this probability of being malicious will change with user actions as per the intrusion-detection system. Note we exclude users that are still logged in. Some of these could be conceivably opening separate login windows, but unless commands to the operating system are being generated by a software script, which can be detected by unusually fast commands, a malicious user cannot do more than one thing at a time.

For instance, suppose user "tom" logged in at 10PM and used the file system and the network connection to site "foobar" and transferred what looks to be a rootkit, logged out at 10:30PM, and then another username logged in at 10:35PM with no one else on the system. Suppose $\lambda_{Sm} = 0.1$ for times in minutes, $p_S = 0.4$, $p_{m1} = 0.8$, and $p_{m0} = 0.001$. The probability the new user is malicious is $0.001 + 0.4 * 0.8 * e^{-0.5} = 0.195$. Suppose the cost of a successful malicious attack is 20, the cost of hurting a legitimate user was 1, the probability of an attacker remaining after the particular deception

"network not authorized" is 0.5, and the cost of a legitimate user remaining after this deception is 0.2. By our simple decision model, 0.195 > g((1*0.8)/(20*.5)) = g(0.08) = 0.08 and this deception is justified.

4. Consistency of deceptions

Repeated deceptions are a good way to foil many attacks when the defender is not at first sure of the nature of the attack. However, a key clue to deception that the attacker can recognize is inconsistent information (Vrij, 2000). Thus, besides applying the cost criteria of section 3, we should avoid deceptions that would logically contradict previously supplied information about the state of a computer system.

The resources associated with each possible command issued to an operating system are the "material" used in attacks (Templeton & Levitt, 2000). The main categories are:

- The directories and files of the computer;
- Peripheral devices to which the computer is attached;
- Networks to which the computer is attached;
- Other sites accessible by the networks;
- The executables for the commands run by an operating system;
- Status markers such as "logged-in" and "administrator privileges" (tokens that must be acquired);
- People associated with the computer systems. People are not always cooperative, although "social-engineering" attacks exploit them.

Resources can be identified as certain arguments to commands, as the site server23 for the command "ftp server23". Other resources are entailed by those already mentioned, like the “network” when transferring files with “ftp”. The executable associated with each command is also a resource, like the "ftp.bin" resource that implements the "ftp" command. It is also useful to distinguish the new resources created by commands, because resource-denial deceptions are more convincing on resources untested over time.

For each resource, we propose six facets of its status, each with an associated predicate:

- Existence, exists(X): Whether the resource X exists (in the place you are looking);
- Authorization, authorized(X): Whether the user is authorized to use the resource X (as by passwords and access control);
- Readiness, initialized(X,A): Whether the initialization of resource X is sufficient for the associated action A to be done;
- Operability, working(X): Whether the resource X is functionally sound;
- Compatibility, compatible(X,Y): Whether the two resources X and Y are mutually compatible (for instance, a text editor is incompatible with an image);
- Moderation, moderate(X,A): Whether the action's demands on a resource are within acceptable limits.

As an example consider the successful action by user Bob of downloading a file "foobar.doc" of size 50,000 bytes from "remotesite" to "homesite" across network "localnet" via the FTP file-transfer utility on homesite, at a time when localnet has five simultaneous users already. The resources necessary to accomplish this are file systems on remotesite and homesite, the network localnet, and the executables for ftp. Successful completion of the actions says that:

- File systems on remotesite and homesite exist.
- File systems on remotesite and homesite are authorized access by Bob.
- File systems on remotesite and homesite are initialized for access.
- File systems on remotesite and homesite are working.
- The network localnet exists.
- The network localnet is authorized use by Bob.
- The network localnet is initialized for file transfers.
- The network localnet is working.

- Localnet is compatible with remotesite and homesite.
- Executable ftp exists on homesite.
- Bob is authorized to use ftp on homesite.
- Executable ftp on homesite is initialized.
- Executable ftp on homesite is working.
- Executable ftp is compatible with the file system on homesite.
- Executable ftp is compatible with localnet.
- Executable ftp is compatible the file system on remotesite.
- The file system on homesite can hold files of 50,000 bytes.
- Localnet can transfer files of 50,000 bytes.
- Localnet can handle six simultaneous users.

Then any convincing deception we do after the download cannot violate any of these. For instance, if the user tries another download, we cannot now say that the download executable is missing, broken, or uninitialized.

5. Conclusions

We have examined a relatively new approach to defending computer systems, that of having them deliberately deceive attackers about resource availability to waste their time and eventually make them go away. Our models are straightforward to construct and can handle a wide range of situations. Because deception risks hurting legitimate users, we must carefully assess the costs and benefits, and our modeling will help us do this more precisely. Our models will be especially useful in automated responses to attacks because their precision permits quick use; automated responses are essential for critical information-system infrastructure such as military networks.

Acknowledgements

This work was supported by the U.S. National Science Foundation under the Cyber Trust Program. The views expressed are those of the author and do not represent policy of the U.S. Government.

References

Barber, R. S., & Kim. J. (2001) “Belief revision process based on trust: agents evaluating reputation of information sources,” in Falcone, R., Singh, M., & Tan, Y.-H. (eds.), *Trust in Cyber-Societies*, LNAI 2246, Berlin: Springer-Verlag, pp. 73-82.

Chirillo, J. (2002) *Hack attacks revealed,* New York: Wiley.

Christian, D., & Young, R. C. (2004) “Strategic deception in agents,” Proc. 3rd Intl. Joint Conference on Autonomous Agents and Multiagent Systems, New York, NY, pp. 218-226.

Cohen, F. (1999, May) “Simulating cyber attacks, defenses, and consequences,” http://all.net/journal/ntb/simulate/ simulate.html.

Cohen, F., & Koike, D. (2003) “Leading attackers through attack graphs with deceptions,” *Computers and Security*, Vol. 22, no. 5, pp. 402-411.

DeRosis, F., Castelfranchi, C., Carofiglio, V., & Grassano, R. (2003) “Can computers deliberately deceive? A simulation tool and its application to Turing's imitation game,” *Computational Intelligence*, Vol. 19, No. 3, pp. 235-263.

Ford, C. V. (1996) *Lies! Lies!! Lies!!! The Psychology of Deceit*, Washington, DC: American Psychiatric Press.

Gerwehr, S., Weissler, R., Medby, J. J., Anderson, R. H., & Rothenberg, J. (2000, November) “Employing deception in information systems to thwart adversary reconnaissance-phase activities,” Project Memorandum, National Defense Research Institute, Rand Corp., PM-1124-NSA.

Heuer, R. J. (1982) “Cognitive factors in deception and counterdeception,” in *Strategic Military Deception*, ed. Daniel, D. C., & Herbig, K. L., New York: Pergamon, pp. 31-69.

The Honeynet Project (2004) *Know Your Enemy, 2nd Edition,* Boston: Addison-Wesley.

Josang, A. (2001, June) “A logic for uncertain probabilities,” *Int. Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems*, Vol. 9, No. 3, pp. 279-311.

Korb, K., & Nicholson, A. (2004) *Bayesian artificial intelligence,* Boca Raton, FL: Chapman and Hall/CRC.
Lachmann, M., & Bergstrom, C. (2004) "The disadvantages of combinatorial communication," *Proc. of the Royal Society of London B*, Vol. 271, pp. 2337-2343.
Lydon, J., & Zanna, M. (1990, June) "Commitment in the face of adversity: A value-affirmation approach," *Journal of Personality & Social Psychology*, Vol. 58, No. 6, pp. 1040-1047.
Monteiro, V., (2003, March) "How intrusion detection can improve software decoy applications," M.S. thesis, U.S. Naval Postgraduate School, www.cs.nps.navy.mil/people/faculty/rowe/oldstudents/monteiro_thesis.htm.
Nyberg, D. (1993) *The varnished truth: truth telling and deceiving in ordinary life*, Chicago: University of Chicago Press.
Rowe, N. (2004, December) "Designing good deceptions in defense of information systems," Computer *Security Applications Conference*, Tucson, AZ, pp. 418-427.
Rowe, N., & Rothstein, H. (2004, July) "Two taxonomies of deception for attacks on information systems," *Journal of Information Warfare*, Vol. 3, No. 2, pp. 27-39.
Rowe, N., Duong, B., & Custy, E. (2006, June) "Fake honeypots: a defensive tactic for cyberspace," 7th IEEE Workshop on Information Assurance, West Point, NY, pp. 223-230.
Snyder, C., Higgins, R., & Stucky, R. (1983) *Excuses: masquerades in search of grace,* New York: Wiley.
Templeton, S., & Levitt, K. (2000, September) "A requires/provides model for computer attacks," *Proc. of the New Security Paradigms Workshop*, Cork, Ireland.
Vrij, A. (2000) *Detecting lies and deceit: the psychology of lying and the implications for professional practice,* Chichester, UK: Wiley.
Whaley, B. (1982, March) "Towards a general theory of deception," *Journal of Strategic Studies*, Vol. 5, No. 1, pp. 179-193.
Zeller, A. (2002) "Isolating cause-effect chains from computer programs," *Proc. 10th ACM SIGSOFT Symposium on Foundations of Software Engineering*, Charleston, SC, pp. 1-10.

Appendix: Questionnaire and geometric mean of the responses

1a. What is the probability of finding a random computer's Internet connection down on a random day? .0416
1b. What is the probability of finding a random computer's local-area network down on a random day? .0422
1c. What is the probability that communications defaults are messed up so that binary file transfers across the Internet make the files unopenable? .0079
1d. What is the probability that a random computer will say that you are not an authorized user when you give it a correct password? .0057
1e. What is the probability that a random computer will deliberately deceive you? .0043
2a. What is the probability that a random system is wrong when it says the Internet is down? .0165
2b. What is the probability that a random system is wrong when it says its local-area network is down? .0086
2c. What is the probability that a random system will inadvertently mess up your files in binary transfers across the Internet? .0113
2d. What is the probability that a random system will lie to you when it says the Internet is down? .0072
2e. What is the probability that a random system will lie to you when it says its local-area network is down? .0032
2f. What is the probability that a random system will deliberately mess up your files in binary transfers across the Internet? .009
3a. Supposing you need to download a file as the major part of your job today, what is the probability you will log out immediately when the system says the Internet is down or it appears that the Internet is down? .0196
3b. Supposing you need to download a file as the major part of your job today, what is the probability you will log out immediately when you have tried three times to transfer files and the system says the Internet is down or it appears that the Internet is down each time? .1012
3c. Supposing you need to download a file as the major part of your job today, what is the probability you will log out immediately when the system says the local area network is down or it appears that the Internet is down? .0566
3d. Supposing you need to download a file as the major part of your job today, what is the probability you will log out immediately when you have tried three times to transfer files and the system says the local area network is down or it appears that the Internet is down each time? .0702
3e. Supposing you need to download a file as the major part of your job today, what is the probability you log out immediately when the files you transfer are messed up? .0052
3f. Supposing you need to download a file as the major part of your job today, what is the probability you log out immediately when you have tried three times to transfer files and the files you transfer are messed up each time? .0225

Experiments with a Testbed for Automated Defensive Deception Planning for Cyber-Attacks

Neil Rowe, Han Goh, Sze Lim, and Binh Duong
Cebrowski Institute, U.S. Naval Postgraduate School, CA, USA
ncrowe@nps.edu

Abstract: A key problem in research in cyberwarfare is the difficulty of conducting experiments with real attackers; science requires experiments, and it is desirable to make information warfare scientific where we can. Some data is available regarding attacks but not much on countermeasures. We report on a testbed we are developing for conducting defensive deception experiments with the normal random background of attacks on the Internet. The testbed is built on top of a honeypot, a computer system that deliberately invites attack to yield useful intelligence about attack methods, but modified to use various deception methods to fool an attacker. Unlike earlier attempts at deception testbeds, ours permits full interaction of an attacker with our system, and thus a wide range of deceptions is possible. We present data obtained by running a partial implementation of our testbed, showing patterns in attacks over time and with system modifications. We show analysis of situations in which attackers may have been induced to leave by thinking the system was not attackable or not in their interest to attack. We also discuss design directions we are exploring.

Keywords: Deception, internet, testbed, attack, honeypot, statistics

1. Introduction

Deception is common in cyberspace attacks (Hutchison & Warren, 2001), most often in the forms of identity deception, privilege escalation, Trojan horses (Chirillo, 2002), and various forms of social engineering. Defensive deception is used in honeypots, computer systems designed solely to collect information about attacks (The Honeynet Project, 2004). Honeypots need to be deceptive to fool attackers into thinking they are normal-use computer systems and thereby provide more data and more-useful data on their attack methods. However, honeypots generally only deceive attackers in a few limited ways: in the primary function of their systems (data collection on attacks), in surreptitious recording of attacker activity, and in concealing the presence of a protective firewall by techniques such as failing to decrement the time-to-live of packets passing through. These three are done for instance by the Sebek honeypot software (The Honeynet Project, 2004), a kind of defensive "rootkit".

Contrary to popular belief, deception is acceptable and important in a wide range of human activities, including law, business, entertainment, politics, and military activities (Ford, 1996; Whaley, 1982). Many deception tactics apply to cyberspace. (Rowe, 2006, ICIW) enumerates 23 methods for defensive deception in cyberspace derived from case grammar in linguistics. Deception could scare attackers away; it could slow attackers down, giving time to organize a defense; it could put imaginary obstacles in their way; it could fool attackers into thinking that they have succeeded in an attack when they have not; or it could fool attackers into revealing information about themselves. Previous work has developed theories for choosing good deceptions, both logical and decision-theoretic (Rowe, 2004), and has implemented some simple experiments such as deceptive Web sites. This can include rating of the effectiveness of the deception class in general (as for the semantic cases); rating the suitability of the resource or resources on which the deception is performed; and rating the method of presentation of the deceptive ploy. For instance, unavailability of a network connection is more likely than unavailability of a file system on a computer, so deception in the former would be more convincing than deception in the latter. Such analysis must also include ways for assessing the penalty of hurting legitimate users who might occasionally do something suspicious and get caught by our deceptions. Too much information-warfare research involves broad claims for theoretical advantages of an approach without testing against real attacks. Theoretical security weaknesses do not always get exploited by attackers, and many attacks we do not yet understand occur on computer systems. So it is important for a discipline like information warfare to try to test against the real world, not just laboratory simplifications of threats. The work reported here is a modest step in this direction.

2. Previous work

Besides in honeypots, deception is being explored as a defensive tactic in cyberspace by several projects. The Deception Toolkit (Cohen, 1999) is the first important example, a system that faked the first steps of key protocols for a number of virtual machines, in the hopes of tying up an attacker in fruitless attacks on nonexistent machines. Such tactics are forms of deceptive mimicry, and can be effective because many attacks are highly specific to operating-system versions. This work was extended to "attack graphs" that

permitted some simple game-playing with the attacker as to which machines and services to fake based on what an attacker had done so far (Cohen and Koike, 2003). Other pioneering work was that of (Gerwehr et al, 2000) that created fake intelligence information for spies to find. Recent related work that has proposed deceptive methods for defense of computer systems includes (Vrable et al, 2005) that built whole deceptive networks called "honeypot farms". (Nazario, 2004) proposed methods for fooling worms into avoiding sites by mimicking their messages for already-infected sites. Several tools such as Morph (www.synacklabs.com/projects/morph) can modify outgoing packets to make a machine appear to have a different operating system or version than the one it really has; this is a good way to fool common reconnaissance tools such as NMAP. (Rowe, 2006, HICSS) proposed deception methods for creating convincing-looking fake file directories. An interesting recent development has been the appearance of tools for attackers to detect honeypots. (Holz & Raynal, 2005) suggested a range of specific clues in the operating system that attackers could use, many of which could be faked to defend a computer system. Other methods for defeating honeypots and responses to them are mentioned in (Krawetz, 2004) and (McCarty, 2003).

Developing good deception techniques benefits from good data on attack patterns. Various attack-record datasets are available for testing defenses such as the TGS data from Skaion, but they generally provide only historical data that does not cover recent trends in a fast-changing environment. The DETER testbed developed by the National Science Foundation can be used to test malicious code experimentally, but exploits are generally tested one at a time with it and this does not much capture the dynamics of real Internet attacks. Using DETER also requires bureaucracy: A test must be submitted for approval in advance, checked for safety, and then time must be allocated on the testbed. This makes it hard to test new exploits quickly or respond quickly to Internet trends. And of course, new attacks need not bear any similarity to previous attacks. Thus it seems important to analyze live attacks and countermeasures on a real machine. We made a start recently by implementing a honeypot network ("honeynet") with some additional features (Rowe, Duong, and Custy, 2006). We stationed a honeynet on the Internet unimpeded by any firewall, with the goal of manipulating it in a variety of ways to deceive attackers. The honeynet mainly collects a wide variety of common speculative attacks on the Internet, what has been termed the "background radiation" of the Internet (Pang et al, 2004). We discuss here the data we have since obtained about it and the first deceptions we have tried with it.

3. Testbed and data-collection setup

3.1 Honeypot network design

We used three machines running Linux: a router, a honeynet, and a data collector. The honeynet itself simulated two virtual machines using the VMWare Workstation product, a regular Windows machine and a Windows Web server. The router was connected to a digital subscriber data line that did not go through our school's firewall. The purpose of the router was to host an intrusion-detection system, Snort, that would not be affected directly by exploits on the victim machine. The reason for the data collector was to provide secure archival data. All machines were kept current with automatic updates for Linux and Windows. Intrusion-detection rules were also updated periodically from our school's Snort-rule collection.

3.2 Summary data collection

We collected data from Snort for our honeypot network for 41 weeks from January 22 to November 7 2006. In this period, Snort raised 175,516 alerts, including a few on traffic intended for neighboring systems. Internet traffic is quite bursty, but by clustering on alerts of the same class and alert message that occurred within 10 minutes of one another, we got much less bursty data that looks much more like a Poisson distribution over time. Thus we preferred the clustered data for analysis, and a 10-minute clustering window appeared the most useful. Clustering on remote IP address gave similar results (around 2/3 of the successive-line pairs involved the same IP addresses) but not quite as good. Analysis is clearer it we remove the effect of day of the week. Figure 1 shows the number of alerts in the first 39 weeks as a function of day of the week, clustered in two ways. The first way was the time clustering described above; the second was by finding sequences of alerts from the same external IP address in the same 10-second time window, and counting each such sequence as a cluster, a simplified versions of the ideas of (Julisch, 2003). An example two-alert sequence that was common was a string of NOPs followed by a buffer overflow attempt. (Alert sequences occurring less than four times in the sample were split in two if the subsequences occurred at least once, since we guessed that such sequences were coincidental concatenations.) Figure 1 suggests that many attackers think of their work as a Monday-to-Thursday job.

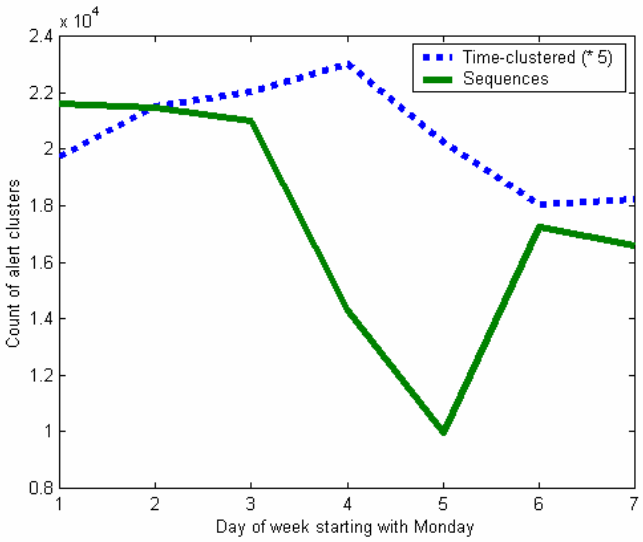


Figure 1: Alert counts as a function of data of the week for the first 39 weeks.

3.3 Trends per week

Figure 2 shows the alert counts per week over the first 41 weeks as clustered three ways, and Figure 3 shows the counts of the six major alert categories clustered only by time. The honeypot network was not working in weeks 8-12 and for several days in weeks 25, 26, and 37 (as discussed in the next section). Clustering was done of alerts using the two methods described above plus by using the K-Means algorithm. The latter clustered events having similarities on 12 boolean metrics including date, IP addresses, port numbers, and alert type; the clustering threshold was adjusted so that we obtained a mean of 306.4 and a geometric mean of 71.3 clusters per day. We can see that most kinds of attacks decreased significantly over time. We infer that attackers did initial reconnaissance to detect vulnerabilities (as often in BAD-TRAFFIC attacks, which either work or don't), and then mostly ignored us when they saw we were not an easy target. In this time period, the system only crashed once from attack, and we saw little evidence of permanent changes to the operating system. The BAD-TRAFFIC, NETBIOS, and SHELLCODE alerts (and to a lesser extent, MS-SQL) decreased significantly over time independent of the offline periods, and resulted in an over downward trend with a linear fit of losing 3.53 alerts per week. Apparently attackers do not return to a system often once they have decided it is not interesting. This suggests that an effective way to obtain reduced attacks is to use existing IP addresses for new systems; deception works well when your adversary underestimates your ploys (Vrij, 2000).

The decreasing trend in attack traffic was not shared by the Internet in general in the 41 weeks, but there were moderate similarities in traffic patterns on shorter time scales. This we confirmed when we obtained data for Snort alerts for our school's network for the time period from August 10 through September 22 (roughly weeks 29 through 35). The mean number of alerts per day was 1351.6 and decreased 30.5 per day; in the same time period, the honeypot saw a mean of 136.6 alerts per day and decreased by 1.952 per day, a comparable rate.

We used the Durbin-Watson test to examine the autocorrelation (Bowerman, O'Connell, and Koehler, 2005) of the count of alerts on the honeypot. We first fit the data to a linear trend and extracted its residuals. Autocorrelation of the residuals for the weekly data (31 weeks, excluding the weeks with days in which the honeypot was turned off) was significant at a lag of 1 week (0.983) but not beyond that. Consistent with this, autocorrelation of the daily data (215 days) was only significant to a 99% level at lags less than a week: 1 day (0.555), 2 days (0.898), 3 days (1.152), 4 days (1.227), 5 days, (1.276), and 6 days (1.489). This suggests definite multiweek similarities, probably due to attack "fads" or repeated attacks of the same type by the same users. As for the five largest categories of alerts, autocorrelations for one-week lags were 0.384 for BAD-TRAFFIC, 0.980 for ICMP, 1.068 for MS-SQL, 0.977 for NETBIOS, and 0.815 for SHELLCODE. So BAD-TRAFFIC was the most faddish. As the graphs suggest, we did not see much "scale-free" behavior as has been seen on the Internet in other work: Attacks in a broader time window were more regular than in a smaller time window, not the same.

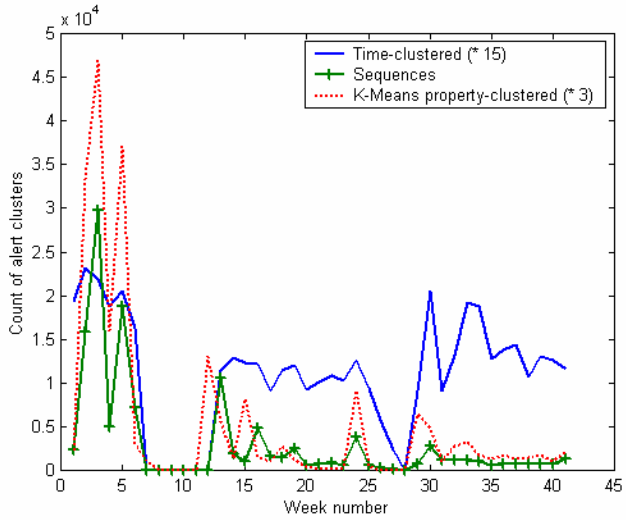


Figure 2: Alert counts per week for the first 41 weeks of alerts, clustered by time (times 15), by sequences, and by K-Means clusters on boolean properties (times 3).

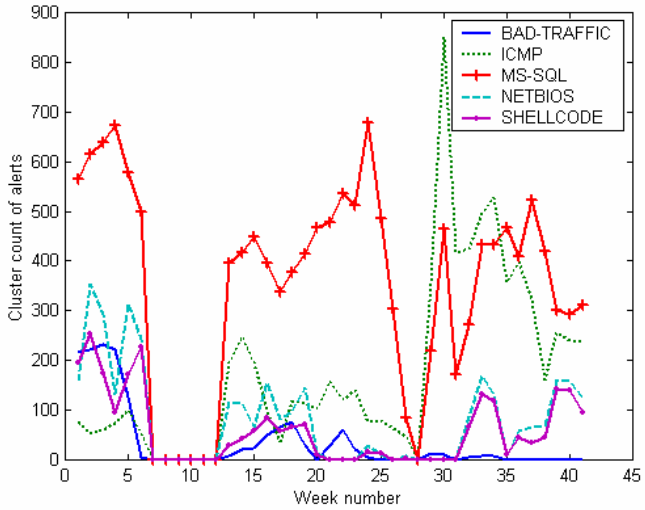


Figure 3: Alert cluster counts per week of the major categories.

3.4 Effects of system manipulations

The down time of the honeypots clearly did affect the attackers. ICMP alerts increased both times in the time-clustered data after the system had been offline for several days, both at week 8 and at week 28. Note that the variety of alert sequences decreased steadily, so it was the number of attacks of the same type that increased at both week 8 and week 28. This suggests automated reconnaissance that needs to keep in frequent contact with our honeypot, perhaps increasing the priority of contacting us when it has had repeated failures. Another explanation is a Trojan horse bot on our honeynet, but we saw no evidence when comparing checksums over time that our honeypot had been compromised in this way. This suggests that deliberately taking a system offline is a good way to encourage reconnaissance traffic, and higher rates of traffic should yield clearer data trends. Only a small amount of this traffic, by the way, was classic ECHOs; most of it was more sophisticated.

There were differences in the outages. Weeks 8-12 just involved disconnecting the Internet hardware connection without changing any software. Thus the reduced level of non-ICMP attacks likely represents attackers "losing interest" in a site that is no longer working. But in weeks 27 and 28, crashes of the system were due to our own mistakes worsened by configuration problems. We then reinstalled the operating system of the honeypot. We had been patching the operating system with the latest Microsoft updates in the meantime, but even so, a fresh operating system seemed to trigger a higher rate (though not variety) of attacks. Finally, in weeks 39-41 we reinstalled the operating system and virtual machines, excluded a small

amount of non-honeynet traffic from calculations, and added new software to the honeypot that suggested monitoring (TCPDump). MS-SQL and ICMP alerts decreased but BAD-TRAFFIC and SHELLCODE alerts increased. We recently conducted experiments where we deliberately dropped (ignored) ICMP packets for a week, but saw no significant effect on attacks, so either we need to do experiments over a longer period to see effects or attackers look for more substantial clues than dropped packets. We will explore this further.

More differences in the response of the attackers to changing circumstances can be seen on other metrics. Figure 4 shows the histogram of interarrival times for successive alerts over all 41 weeks and Figure 5 shows it for October 19 through November 7. Several important peaks are different in the data over the shorter period despite reasonable sample sizes (175,000 and 6,200). The peak at 3.7 (representing 0.04 sec.) is higher, the peak at 8.6 (representing 7 sec.) is lower, and the peak at 18.3 (representing one day) has disappeared. (The peak at 6.9 represents 1 second exactly, and is important in both graphs – apparently there are a lot of programmed one-second delays.) Differences could be due to changes in the honeynet in the last period. Since the peaks suggest automated attack timings, differences in them suggest that automated attacks are responding differently to the honeynet changes. In general, occurrences of common time gaps are good clues to an automated attack, and facilitate identification of more-interesting manual attacks. Note there was very little evidence of attack plans requiring more than a minute, despite the worries of some researchers about "low and slow" attacks.

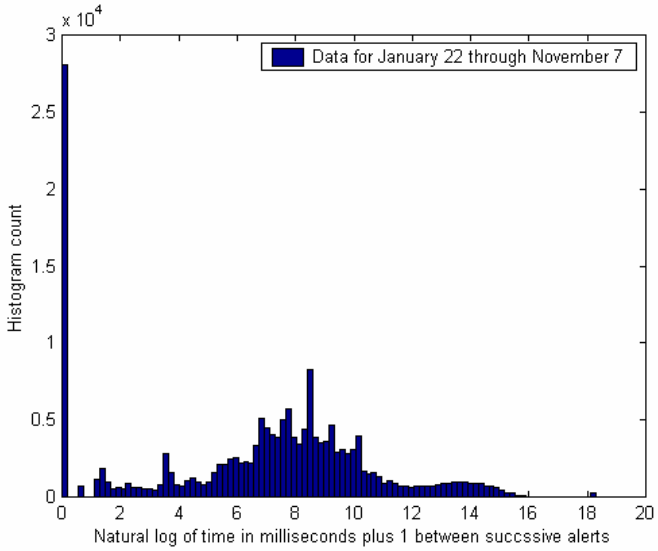


Figure 4: Histogram of interarrival times between alerts over 41 weeks.

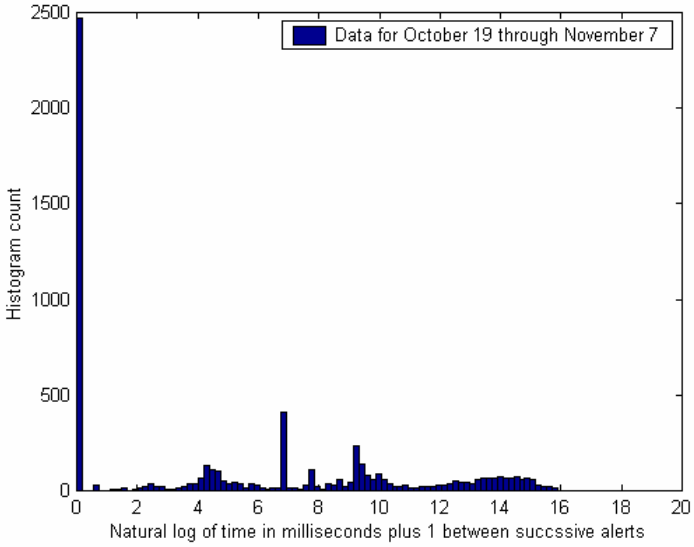


Figure 5: Histogram of interarrival times for alerts over three weeks of system changes.

Differences in the external IP addresses used in attacks are also seen over time. Figures 6 and 7 compare the first byte of the IP address over two successive periods of three weeks before and after we made deliberate changes. 63 is the first byte of our own address and so makes a big peak in both time periods. But a new larger peak occurs at 67 and a new smaller peak at 210. These probably indicate responses to changes in the honeynet as we did not see these peaks in the previous 35 weeks.

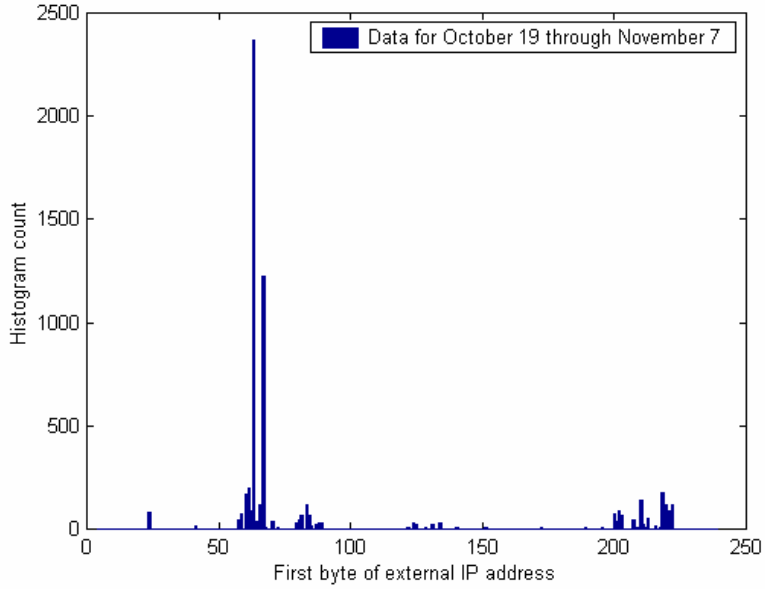


Figure 6: Histogram of IP addresses used in attacks, October 19 through November 7.

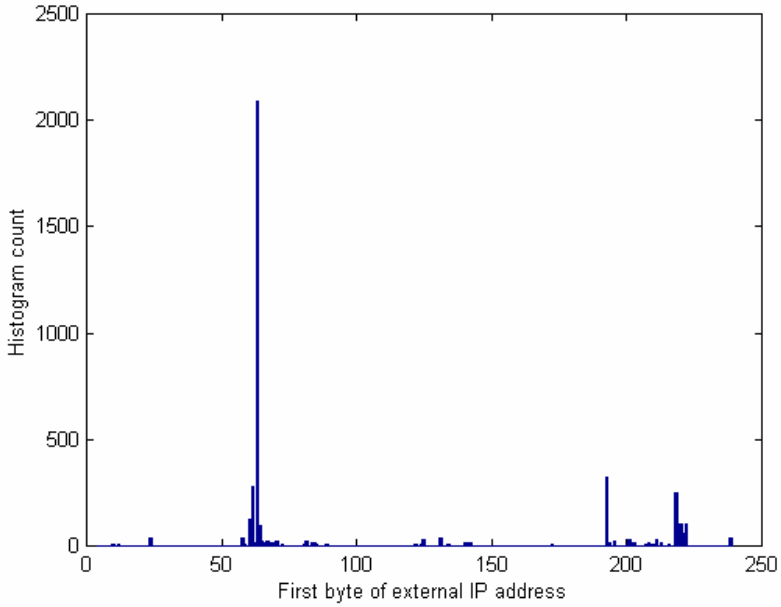


Figure 7: Histogram of IP addresses used in attacks, September 29 through October 19.

4. Learning to scare attackers away

Currently we are developing methods for automatically learning good deceptions to induce attackers to go away. The method is to observe behavior of attackers and record the last few things they did before they went away, then use concept learning to generalize these circumstances into countermeasures we can implement, perhaps eventually automatically within seconds. Methods for automatically learning patterns in intrusion data have been studied previously (Qin & Hwang, 2004; Wuu & Chen, 2003) but they have been focused on finding new intrusion-detection patterns rather than new countermeasures.

Our rationale for this approach is that at some point in interacting with out honeynet, attackers have decided either it is not worth attacking or they personally are unable to attack it. Attackers may be persuaded to

reach either of these conclusions, both equally useful to us, by deception. If we can see what features of the honeypot they were examining just before they gave up, those features could be useful features to put on other machines or make more obvious for the attacker on this machine. For instance, most attackers want to avoid honeypots, so they could be finding features suggestive of honeypots just before they log out; if this is the case, we could add more features of honeypots to encourage them to log out sooner. Such deceptive ploys could act as a "vaccination" of a computer system, reducing the severity and likelihood of success of an attack.

4.1 Experiments

To guide our selection of "scaring" tactics, we extracted from Snort the last alert generated each day for each source IP address appearing in some alert that day. We took counts over these alerts over 39 weeks and compared them to the total observed counts of the alerts. This ratio represents a conditional probability of the alert being last in the sessions for the day for that IP address. Our hypothesis is that alerts with high conditional probabilities are more likely to represent "scaring-away" phenomena. The top-rated alerts are shown in Table 1 (with their conditional probability of being last, the associated count ratio, the Snort alert category, and the description of the alert).

Table 1: The top-rated session-ending alerts, with their conditional probabilities and counts of occurring last in a session.

Prob.	Count ratio	Alert type	Alert message
1.0	3/3	WEB-MISC	Cisco IOS HTTP configuration attempt
1.0	3/3	POLICY	FTP 'MKD / ' possible warez site
1.0	1/1	SHELLCODE	x86 setgid 0
1.0	1/1	NETBIOS	DCERPC Messenger Service buffer overflow attempt
1.0	1/1	BACKDOOR	netthief runtime detection
0.769	10/13	ICMP	traceroute
0.722	13/18	SCAN	FIN
0.667	2/3	BAD-TRAFFIC	tcp port 0 traffic
0.625	35/56	POLICY	SMTP relaying denied
0.625	10/16	WEB-MISC	robots.txt access
0.5	1/2	WEB-MISC	net attempt
0.5	1/2	WEB-CGI	printenv access
0.5	1/2	WEB-CGI	icat access
0.5	1/2	ICMP	Destination Unreachable Network Unreachable
0.5	1/2	DNS	named version attempt
0.488	699/1431	ICMP	Destination Unreachable Port Unreachable
0.433	13/30	POLICY	FTP anonymous login attempt
0.425	37/87	INFO	web bug 1x1 gif attempt
0.4	2/5	WEB-CGI	wrap access
0.359	110/306	ICMP	Time-To-Live Exceeded in Transit
0.354	17/48	INFO	web bug 0x0 gif attempt
0.333	3/9	WEB-CGI	calendar access
0.333	1/3	WEB-FRONTPAGE	/_vti_bin/ access
0.333	1/3	NETBIOS	SMB-DS srvsvc NetrPathCanonicalize WriteAndX little endian overflow attempt
0.333	1/3	EXPLOIT	WINS name query overflow attempt TCP
0.275	11/40	NETBIOS	SMB-DS umpnpmgr PNP_QueryResConfList unicode little endian attempt
0.273	6/22	WEB-MISC	weblogic/tomcat .jsp view source attempt
0.219	1562/7133	BAD-TRAFFIC	udp port 0 traffic
0.216	187/867	NETBIOS	DCERPC ISystemActivator path overflow attempt little endian unicode
0.200	1130/5646	MS-SQL	version overflow attempt

Some of these counts are based on just a few instances and thus may be misleading, although rare events with strong conditional probabilities still can indicate attacks by experts. But some interesting strong conditional probabilities are supported by significant evidence like that for "ICMP traceroute" representing apparently failed attempts at using classic network reconnaissance tools. Several explanations are possible for the strong conditional probabilities:

- That these alerts are generated by popular attack tools and just happen to be the last alerts generated by use of those tools in many instances. However, many attackers jealously guard their attack tools and do not like to share.
- That the alerts represent logically necessary last steps in common multistep attacks. This can occur if they are the "payoff" steps that accomplish important attack goals. However, most of the Table 1 alerts cannot be so construed.
- The alerts represent that last thing seen after the attacker seized control of our honeypot and turned off Snort. However, we saw no evidence of that from our integrity checker.
- That the alerts represent things that an attacker did that convinced them that a machine was not worth attacking.

If the last is the case, a "vaccination" might mean applying up-to-date patches of the operating system to prevent an exploit from succeeding (as for ICMP traceroute and SCAN FIN events). But in other cases this data guides us to do specific things to the operating system that we may not have thought about otherwise (such as turning off SMTP relaying and robots.txt access). And instead of doing those things, which may require work to implement, it may be sufficient to deceive about them – just respond to the specific way the attacker is querying them, and give a false response.

Alerts are only the "tip of the iceberg" in malicious user behavior. So our current work is using the TCPDump network-packet extractor to obtain packets destined for our honeypot, and then studying the last packet involving each external IP address each day. Such "last packets" comprised a small fraction, 88 for four weeks versus 100,000 total packets per day. We have seen clear correlations in the size of the last packet with the type of attack, more so than with packets in general; for instance, last packets of size 922 were strong indicators of a particular Trojan horse spread through email. We have seen interesting differences in the TCP flags set in last packets (Table 2). As could be expected, we saw many RST (reset) flags and no flags indicated forthcoming packets. The hope of this approach is to use distinctive features of last packets to design quite specific deceptions, perhaps automatically in response to new threats.

Table 2: Flag-pattern statistics on the last packets seen from an external IP address.

Packet flag pattern	Count in one day in all packets received	Count in 4 weeks in last packets
TCP/ACK	126	12
TCP/ACK-DON'T FRAG	141	0
TCP/ACK-PSH	115	0
TCP/ACK-PSH-DON'T FRAG	1440	0
TCP/ACK-PSH-FIN	4	0
TCP/ACK-RST	7	15
TCP/ACK-SYN	36	0
TCP/ACK-SYN-DON'T FRAG	88	0
TCP/ACK-FIN	2	0
TCP/ACK-FIN-DON'T FRAG	180	0
TCP/PSH-SYN-DON'T FRAG	2	0
TCP/SYN-DON'T FRAG	88	0
TCP/RST	0	2
UDP	2382	59
UDP/DON'T-FRAG	13	0

5. Conclusions

This is a difficult area in which to do research because attackers are not especially systematic. Several useful conclusions are still possible with even these preliminary results.

- Attacks are fewer on Friday, Saturday, and Sunday. That suggests those days are good times to bring up a new system.
- Attacks on a newly used IP address are high at first and then decrease significantly over a few months. This suggests that a good way to reduce attacks on a new computer is to reuse an existing IP address if you can.
- A few common vulnerabilities are repeatedly attacked. It is good to have up-to-date patches on these (www.cert.org often provides good warnings), or at least good fake indicators of the patches. Then

many common attacks will simply terminate when they see evidence of patches installed. Note that while we may be more vulnerable to rarer attacks, reducing the traffic of common attacks can make it easier to handle the rarer ones.

- Taking a system offline increases alerts due to ICMP traffic, packets used for network-management communications. Fortunately, there are not many ICMP vulnerabilities and most are straightforward to fix. Thus taking a system offline irregularly may encourage attackers to waste more time fruitlessly attacking an impervious machine instead of searching for a more susceptible target.
- Many of the commonly attacked vulnerabilities refer to features of the configuration of a site, its parameters and services, like SMTP relaying. The last-alert data provides good clues to how to set these parameters to discourage interest by attackers, by setting them to uninteresting values or turning off the services interesting to attackers.
- The last packet received from a particular IP address shows a limited range of flags, and it may be possible to exploit this in defensive deception.

In future work we will test the effect of more elaborate types of deceptions on attackers with Snort-Inline, an intrusion-prevention system variant on the Snort intrusion-detection. We will test in particular the effects of changing suspicious packets since lack of an expected response was a major prod to attackers in the work reported here.

Acknowledgements

This work was supported by the U.S. National Science Foundation under the Cyber Trust Program. The views expressed are those of the author and do not represent policy of the U.S. Government.

References

Bowerman, B., O'Connell, R, & Koehler, A. (2005) *Forecasting, time series, & regression: an applied approach, 4th ed*., Belmont, CA: Thomson Brooks/Cole.

Chirillo, J. (2002) *Hack attacks revealed*. New York: Wiley.

Cohen, F., and Koike, D. (2003) "Leading attackers through attack graphs with deceptions," *Computers and Security*, Vol. 22, no. 5, pp. 402-411.

Ford, C. V. (1996) *Lies! Lies!! Lies!!! The Psychology of Deceit*, Washington, DC: American Psychiatric Press.

Gerwehr, S., Weissler, R., Medby, J. J., Anderson, R. H., & Rothenberg, J. (2000, November) "Employing deception in information systems to thwart adversary reconnaissance-phase activities," Project Memorandum, National Defense Research Institute, Rand Corp., PM-1124-NSA.

Holz, T., and Raynal, F. (2005, June) "Detecting honeypots and other suspicious environments," *Proc. 6th SMC Information Assurance Workshop*, West Point, NY, pp. 29-36.

The Honeynet Project (2004) *Know Your Enemy, 2nd Ed.,* Boston: Addison-Wesley.

Hutchinson, W., and Warren, M. (2001) *Information warfare: corporate attack and defense in a digital world,* London: Butterworth-Heinemann.

Julisch, K. (2003, November) "Clustering intrusion detection alarms to support root cause analysis," *ACM Transactions on Information and Systems Security, 6* (4), pp. 443-471.

Krawetz, N. (2004, January-February) "Anti-honeypot technology," *IEEE Security and Privacy*, Vol. 2, No. 1, pp. 76-79.

McCarty, B. (2003, November-December) "The honeynet arms race," *IEEE Security and Privacy, 1*(6), pp. 79-82.

Nazario, J. (2004) *Defense and detection strategies against Internet worms*, Boston, MA: Artech.

Pang, R. Veqneswaran, V., Barford, P., Paxson, V., and Peterson, L. (2004) "Characteristics of Internet background radiation," *Proc. 4th ACM SIGCOMM Conference on Internet Measurement,* Taormina, IT, pp. 27-40.

Qin, M., & Hwang, K. (2004) "Frequent episode rules for Internet anomaly detection," *Proc. 3rd IEEE Symposium on Network Computing and Applications,* Los Angeles, CA, pp. 161-168.

Rowe, N. (2004, December) "Designing good deceptions in defense of information systems," *Proc. Computer Security Applications Conference*, Tucson, AZ, pp. 418-427.

Rowe, N. (2006, January) "Measuring the effectiveness of honeypot counter-counterdeception," *Proc. 39th Hawaii International Conference on Systems Sciences*, Poipu, HI.

Rowe, N. (2006, March) "A taxonomy of deception in cyberspace," *Proc. International Conference on Information Warfare and Security*, Princess Anne, MD, pp. 173-181.

Rowe, N., Duong, B., and Custy, E. (2006, June) "Fake honeypots: a defensive tactic for cyberspace," *Proc. 7th IEEE Workshop on Information Assurance*, West Point, NY, pp. 223-230.

Tirenin, W., and Faatz, D. (1999, October) "A concept for strategic cyber defense," *Proc. Conf. on Military Communications*, Atlantic City, NJ, Vol. 1, pp. 458-463.

Vrable, M., Ma, J. Chen, J., Moore, D., Vadekieft, E., Snoeren, A., Voelker, G., and Savage, S. (2005) "Scalability, fidelity, and containment in the Potemkin Virtual Honeyfarm," *Proc. ACM Symposium on Operating Systems Principles*, Brighton UK, pp. 148-162.

Vrij, A. (2000) *Detecting lies and deceit: the psychology of lying and the implications for professional practice,* Chichester, UK: Wiley.

Whaley, B. (1982, March) "Towards a general theory of deception," *Journal of Strategic Studies*, Vol. 5, No. 1, pp. 179-193.

Wuu, L.-C., & Chen, S.-F. (2003, October) "Building intrusion pattern miner for Snort network intrusion detection system," *Proc. 37th IEEE Canadian Conference on Security Technology*, pp. 477-484.

Possible Risks Analysis Engine: A Prototype Tool for Managing IT Security Safeguards Acquisition

Robert Sainsbury and Richard Baskerville
Georgia State University, Atlanta, GA, USA
rsainsbury@cis.gsu.edu
baskerville@acm.org

Abstract: Risk analysis provides a cost-benefit analysis of information security controls and safeguards in economic terms. Despite serious flaws in its fundamentals, approaches to calculating risk have changed little over the past decades. The publicly available frequency data that does exist is generally incompatible and unusable. Theories of mathematical evidence indicate that probability theory is inappropriate where frequency data is unavailable. While alternative theoretical frameworks have been suggested, practical vehicles for the use of such frameworks have yet to materialize. This paper reports on design science research that employs fuzzy sets and possibility theory as kernel theories to develop and demonstrate a prototype of such a practical vehicle. This vehicle opens avenues for testing and operating risk analysis methodologies based on alternative mathematical theories of evidence.

Keywords: Risk management, risk analysis

1. Introduction

Success in the implementation of information security in most organizations involves the satisfaction of two goals. The first goal is technological in nature (developing working technical solutions to security vulnerabilities). The second goal is managerial-economical in nature (making the right decisions about what security features to acquire). While myriad technical solutions are available, management principles and techniques for deciding which solutions to acquire for a particular organizational setting are rather limited. Risk analysis is often promoted as a central technique used for identifying and managing organizational information technology risks. It essentially permits a cost-benefit analysis of information security controls and safeguards in economic terms. Risk analysis has been an essential component in information security management for over two decades. Figure 1 represents the typical model for approaching information security based on an analysis of the information security literature by Baskerville (1993). This model illustrated the centrality of risk analysis in the typical six stages of the process of specifying security. The basic process consisted of identifying assets, threats and controls, and then performing risk analysis to determine which controls provided the best return on investment when used to protect valuable assets from frequent threats. The results of the risk analysis were used to prioritize the list of possible controls. This prioritized list was then implemented to the degree possible with the available funds.

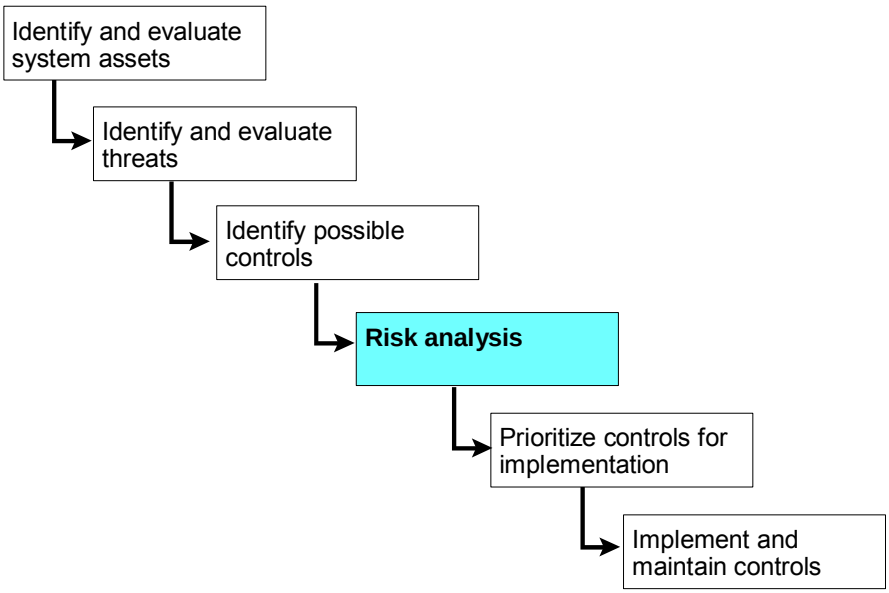


Figure 1: A general model of security specification processes (Baskerville, 1993).

Although published nearly 13 years ago, the fundamental principles of this approach to information security in business are still present. The basic elements of risk analysis are perennial, and still consist of two measures: the cost of the impact of the occurrence of a particular threat, and of the frequency or probability

of this occurrence. These elements were central components in the information risk analysis methodology proposed 30 years ago by Robert Courtney (1977), and are still the key measures used by responsible risk managers today (cf. Jones & Ashenden, 2005). For example, the relatively recent Octave methodology is still focused on assets, vulnerabilities, and a strategy developed from the use of risk analysis. Figure 2 illustrates the major processes in this methodology. Phase 1 is focused on assets, phase 2 is focused on vulnerabilities, and phase 3, and process seven still emphasizes risk analysis.

PPhase 1: Build Asset-Based Threat Profiles
- Process 1: Identify Senior Management Knowledge
- Process 2: Identify Operational Area Knowledge
- Process 3: Identify Staff Knowledge
- Process 4: Create Threat Profiles

PPhase 2: Identify Infrastructure Vulnerabilities
- Process 5: Identify Key Components
- Process 6: Evaluate Selected Components

PPhase 3: Develop Security Strategy and Plans
- Process 7: Conduct Risk Analysis
- Process 8: Develop Protection Strategy

Figure 2: The octave methodology (Alberts & Dorofee, 2001)

A second example is the IT Infrastructure Library (ITIL) which contains a volume on security management. While this approach is focused on managing information technology services, the ultimate best practices framework is initially founded on risk analysis. Figure 3 diagrams this approach, clearly showing risk analysis is the foundation of the initial security efforts. The cyclical management process that follows this includes a form of risk analysis in the iterative requirements feasibility analysis that is dependent on the continued maintenance of the results of the initial risk analysis (Weil, 2004).

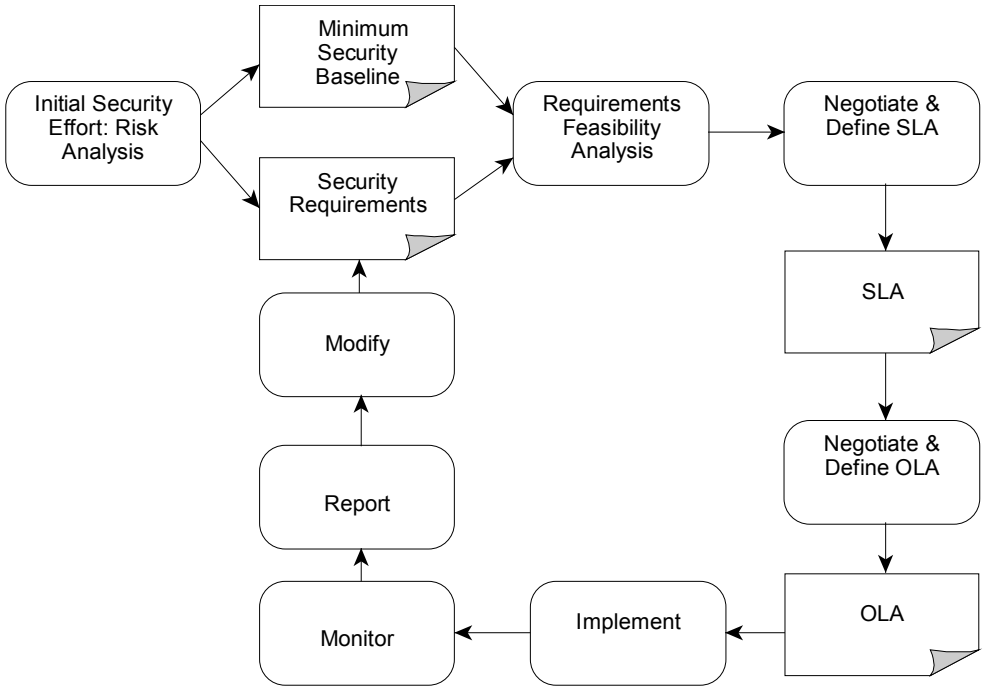


Figure 3: ITIL risk management methodology (Weil, 2004)

Despite the prevalence of risk analysis in the approaches to information security management, the technique has serious flaws. The most essential problems arise from the questionable use of probability arithmetic in theoretic terms. Probability theory requires rather specific kinds of data. It requires actual frequency evidence that is randomly distributed. For the purposes of information security, it is nearly impossible to acquire such data. Where the data actually exist, these are largely regarded by the owners as extremely

confidential, since it has been shown that disclosure of actual loss data has a negative impact on the capital markets of the organizations that make such disclosures (Hovav & D'Arcy, 2003). In many organizations the actual data are never generated, much less retained. Organizations in the throes of recovering from a serious security event rarely have the excess capacity to develop careful records about the event. Most organizational resources at the time are dedicated to recovering from, not accounting for, the loss.

The publicly available data that does exist is generally incompatible and unusable. The unit of analysis is almost invariably the firm rather than the event, and consequently the relationships between frequency of events and costs of events are not developed in these public surveys (Baskerville & Sainsbury, 2006). In addition, it is impossible to depend on actual frequency data in the case of rare or unique events. A unique event necessarily develops no frequency data. Information security risk management is an intractable problem in the sense that the available solutions are founded on a theoretically insufficient theory for the purposes of risk analysis. It is not an ideal solution. Indeed, there is evidence that the technique is practiced seriously by fewer than 25% of IT Risk Managers (Baskerville, 2005a).

Theories of mathematical evidence have been used to suggest alternative theoretical frameworks for the common risk analysis in settings where actual frequency data is unavailable. For example, possibility theory, with its foundations in fuzzy sets, has been described as an alternative basis for those typical risk analysis and information security settings where opinions must be used instead of actual event data (Baskerville & Portougal, 2003). This approach has been shown to particularly apply to risk management efforts in support of defensive information operations (Baskerville, 2005b). While research in the area of the use of fuzzy sets and possibility theory as a foundation for risk analysis dates from almost the earliest days of the development of this theory (cf. Rine, 1978), actual implementations of practical risk management methodology based on such evidence are rare.

2. Research methodology: Design science

The approach taken in addressing this intractable problem is design science research. The research is aimed at providing a practical solution, as evidenced by an operable artifact, which resolves the intractable problem in risk analysis. Design science research is a theory-driven, means-ends research framework governing the search for solutions among potential design features. Design science research has been described as,

> *"The proper study of those who are concerned with the artificial is the way in which that adaptation of means to environments is brought about -- and central to that is the process of design itself." (Simon, 1996, p. 113)*

A key element that distinguishes design science research from the simple construction of artifacts regards the search for appropriate assemblies with which to construct problem solutions. These problem-solving systems and design procedures are more sophisticated and rigorous as a form of "artificial" science (Simon, 1996). Design science research operates with prescriptive and functional design theories that explain phenomena in terms of their function within design solutions, and describe reality in prescriptive terms (as the qay things ought to be). Design theories are grounded on kernel theories (which are other natural and social science theories) that determine the testable hypotheses by which a design artifact may be evaluated.

Walls, et al. (1992) provides an extensive description of design theories for information systems. Their ideas are illustrated in figure 4. The design theory is divided into two major aspects. One major aspect is the design product itself (the left part of the diagram in figure 4), while the other major aspect is the design process (the right part). The same or different kernel theories they underlie each of the aspects. The design process aspect consists of a design method which is a procedure for constructing the artifact, and hypotheses for testing the results of the process against the meta-design. The design product aspect consists of meta-requirements, a meta-design, and product hypotheses. Requirements and design are regarded as operating at a meta level because these regard a class of products rather than just a specific product. Once the design theory is perfected, it can be used to repeatedly develop designs within this class of products. It is because it operates at this meta-level that we are producing a design theory, rather than just simple, one-off, design practice.

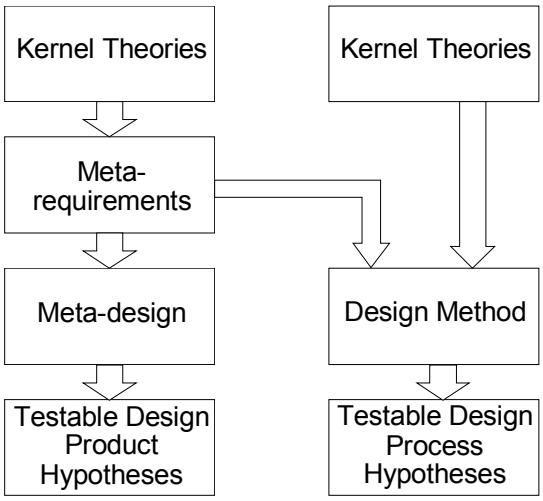


Figure 4: Extensive description of design theories for information systems

Work in the area of design science research for information systems has been developing steadily over the past decade. For a review of work that has specifically built on information systems design theory, see Walls, et al. (2004) and Vaishnavi & Kutchler (2004). In this case, the kernel theories consist of mathematical theories of evidence and include probability theory and possibility theory. Our meta-requirements include an ability to operate using professional opinion and uncertainty rather than actual frequency data. The need to use professional opinion in settings of uncertainty arises from the unavailability of actual frequency data. The only available data for the purposes of analyzing risk is that of experienced professionals working in the area of IT security who are regarded as knowledgeable enough to make reliable estimates of potential threats. Interestingly enough, this is usually the data that is used for probability-based risk analysis even though theories of evidence discount the legitimacy of such arithmetic operations (Klir, St. Clair, & Yuan, 1997).

In our case another meta-requirement is the need to demonstrate the feasibility of using alternative forms of arithmetic evidence in this analysis. Our aim is not to construct a final, production quality, risk analysis methodology or tool. Instead, our aim is to construct a prototype of such a tool along with a simple methodology for applying the tool. Based on these meta-requirements, we turned to a design method known as prototyping. Prototypes are preliminary versions of computerized systems. A prototype models aspects of the final system, but remains limited in many unimportant ways. Prototyping is a system development approach that involves construction and test of prototypes to demonstrate feasibility. Prototyping is sometimes viewed as a mechanism for improving the effectiveness of analysis and design in loosely structured, high technology development projects needed in adaptive situations (Baskerville & Stage, 2004). It is a venerable and well known approach that alleviates many of the practical problems that arise in requirements definition (Boar, 1984; Mason & Carey, 1983; Naumann & Jenkins, 1982). These features make the approach highly suitable for the development of an experimental risk analysis tool that is founded on an alternative theory of mathematical evidence.

Finally, we need a set of testable design product hypotheses and testable design process hypotheses by which to evaluate the product and the design method.

> *Hypothesis 1: The possibility based risk analysis tool will increase the meaningfulness of risk analysis when examining rare and emerging threats*

This hypothesis deals with the nature of threats the possibility tool is designed to analyze. Without historic frequency data, probably based analysis will not be fundamentally sound, giving meaningless results. By analyzing the available evidence with possibility based logic, the same data should give more meaningful results

> *Hypothesis 2: The possibility based risk analysis tool will encourage input from more sources than traditional risk assessment tools.*

The tool's ability to store and display the input from multiple sources while also tracking the sources' reliability allows organizations to solicit the opinions of experts who may not have been included in the process using other tools. Being able to easily distinguish the different contributions and relative weight of the sources will make the interpretation of many viewpoints manageable, helping to increase the diversity of data sources without increasing complexity.

These Hypotheses will guide the future testing of the possibility based risk assessment tool. By comparing the results obtained with current risk assessment tools to the results of the possibility based prototype the validity of the hypotheses can be established. Based on fundamental theories of mathematical evidence, the research was anchored in possibility theory (rather than probability theory). We describe a working prototype of a software-based IT risk analysis tool that eschews probability theory, yet provides a valid mechanism for the analysis of rare and unpredictable risks along with risks for which no actual frequency data is available.

3. Possible risks analysis engine

Development of the risk analysis engine based on possibility theory begins with the adaptation of this mathematical theory of evidence for the purposes of gathering data, analyzing the data, and representing the results within the context of information security risk analysis. Possibility theory is based on the relationship between evidence and uncertainty, and separates decisions about quality and uncertainty of the evidence from the decisions about the uncertainty of the phenomenon being represented by the evidence. This framework is effective for studying nonrandom events and considerably better for situations with improbable but possible events. As with other forms of fuzzy sets, possibility theory operates with opinions, ranges, and subjective information rather than operating with point estimates (such as is common with probability-based arithmetic). This means that data should be able to be collected from multiple experts and the analysis should provide a means of comparing and contrasting the estimates and opinions of these experts. It also means that the analysis must take into account estimates and opinions of experts on several different kinds of evidence relating to a particular risk or threat.

Collecting data from multiple experts, on multiple points of evidence, each of which may involve a range of opinion and confidence will require fuzzy set arithmetic to analyze and compound the available evidence. For simplicity, the engine will use simple additive analysis of this evidence. Fuzzy set addition is usually represented as the union of two fuzzy sets. Fuzzy set union means taking the maximum value at every point. Figure 5 depicts this union. The graph at the left of figure 5 represents a set of evidence about a particular value ranging from one to six. The evidence consists of two fuzzy sets, one ranging from 1.0 to 6.0, and the other ranging from 2.0 to 5.5. The graph at the right of figure 5 represents the sum of the two sets, which is the maximum point of either of the two sets in point of evidence. The new risk analysis engine will use fuzzy sums to represent the sum of opinions about particular risks.

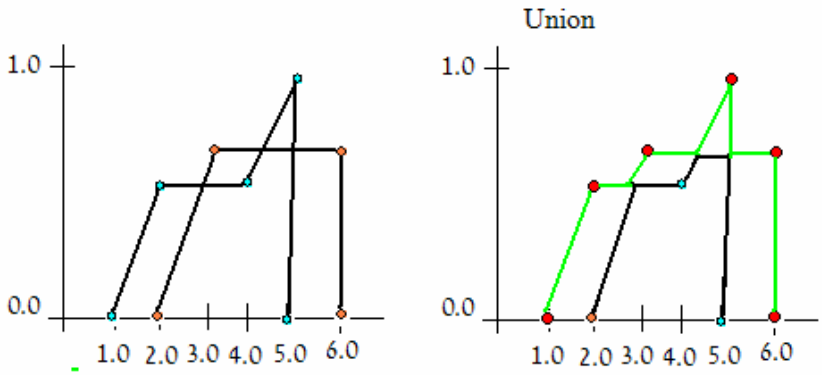


Figure 5: Fuzzy set addition.

Based on the design theory, we developed an interactive prototype in Visual Basic. We will describe the features of this interactive prototype by providing a demonstration of its operation in the next section.

4. Demonstrating the engine

To demonstrate the operation of the risk analysis engine we will look at the risk of a data spill. A data spill occurs when organization loses sensitive records to an intruder or to an innocent receiver as a result of an accident. There have been many disastrous data spills that have become publicized over the past years. Table 5 lists a few of these companies and the number of records each has improperly disclosed due to a data spill. These are serious events.

In order to estimate the risk of the data spill in any organization, we would need to collect and analyze evidence about this risk. Given that frequency data that is correlated with impact is unavailable, we will settle for fuzzy evidence collected from experts concerning the risk of the data spill in a particular

organization. For the purposes of our demonstration we propose to use four different categories of evidence.

Table 1: Number of records improperly disclosed in recent data spills

CardSystem Solutions	40,000,000
Citigroup/UPS	3,900,000
Bank of America	676,000
LexisNexis	310,000
HSBC	180,000
San Jose Medical Gp	185,000
Tufts University	105,000
Time Warner	600,000
Ameritrade	200,000
Department of Justice	80,000
ChoicePoint	160,000
Bank of America	1,200,000
Berkeley	98,000
Cal State	23,000
Total (and growing….)	47,717,000

Evidence that might indicate the likelihood of a data spill in a particular organization could include (1) the quality of encryption of the backup media, (2) the quality of protection for systems, (3) the effectiveness of security procedures, and (4) the degree to which the security procedures are followed. In order to gather opinion data about these four kinds of evidence, we could ask experts the following four questions; the answers to which would embody the evidence above.

1. What is the risk of the backup media will be compromised?
2. What is the threat for breaching these systems?
3. What is the risk that security procedures are ineffective?
4. What is the risk that security procedures will not be followed?

The possibility risk analysis engine collects data from each expert on these four elements of evidence in the following way. Each expert is asked to enter their opinion on each element of evidence by adjusting five slider bars to represent the lowest and the highest *possible* levels, the lowest and the highest *likely* level, and the degree of their confidence in the value of their evidence. The screen representing the interaction of the user with the risk analysis engine in answering question one above is shown in figure 6. In this case the expert has entered an opinion that the lowest possible level is very low and the highest possible level is fairly low. This is represented in the bottom line of the trapezoid in the figure, and corresponds to the two slider bars for “possible” levels. The expert has also estimated their opinion that the lowest likely level is slightly higher than very low, and the highest likely level is slightly lower than fairly low. This is represented by the top line in the trapezoid, and by the two bottom sliders in figure 6. Finally this expert has set their degree of confidence in this evidence to be very confident represented by the slider to the left of the trapezoid, and also represented by the height of the trapezoid itself.

Figure 7 represents the entry of a second element of evidence by the same expert. This represents the response to the second question, “What is the threat for breaching these systems?” This expert has responded with a belief that the lowest possible level is slightly lower than fairly low while the highest possible level is slightly lower than very high. This entry is represented by the top two slider bars in the figure for lowest and highest possible level respectively, and is represented by the bottom line in the trapezoid area. The expert has also estimated that a lowest likely level is slightly lower than average and the highest likely level slightly lower than fairly high. This entry was made using the bottom two sliders in the interactive screen is represented by top line in the trapezoid. With regard to this evidence, this expert has expressed a fairly low degree of confidence by setting the confidence slider in the bottom third of its range. This is represented by the height of the trapezoid (which is rather short).

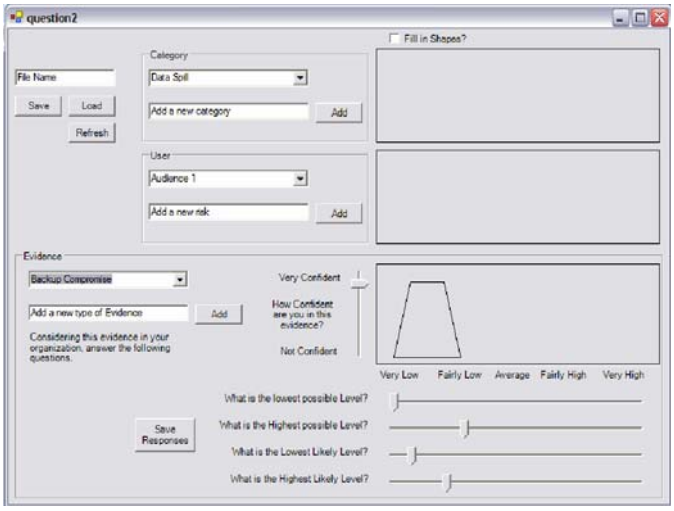


Figure 6: Screen representing the interaction of the user with the risk analysis engine in answering question

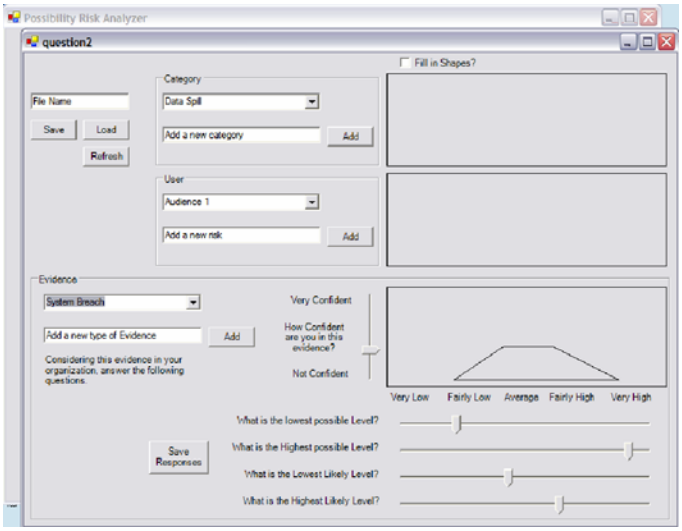


Figure 7: The entry of a second element of evidence by the same expert

As this evidence is entered, the engine accumulates an analysis by providing a representation of the compound opinions of multiple elements of evidence by multiple experts. This analysis appears on the screen as illustrated in figure 8. The expert currently working the system is represented by blue trapezoids in the center display. But we see that a previous expert has already entered opinions in an earlier evidence session. This previous expert is represented by red trapezoids in the top display. The top display also includes the evidence that has just been entered by the current expert in blue. This overlapping display enables analysis of the different forms of evidence offered by different experts. This enables the analyst to separate confidence (uncertainty) about the evidence from uncertainty about the event itself. For example, the analyst can make decisions about the uncertainty in the evidence from expert number one (in red) in comparison with a higher degree of certainty in those cases from expert number two (in blue).

Finally, after all adjustments have been made, it is possible to obtain the fuzzy sum of all of the evidence from the prototype possibility risk analysis engine. This is represented in figure 9. Essentially, all of the discrete representation of different kinds of evidence from different experts is obscured in the sum total represented as a single geographic fuzzy shape. This overall analytical representation illustrates how the evidence, while quite broad, has the highest confidence (least uncertainty) at the very low to fairly low values. Indeed the evidence slopes from the lower right to the upper left indicating how the most certain evidence is in the low range, while the most uncertain evidence is in the higher ranges.

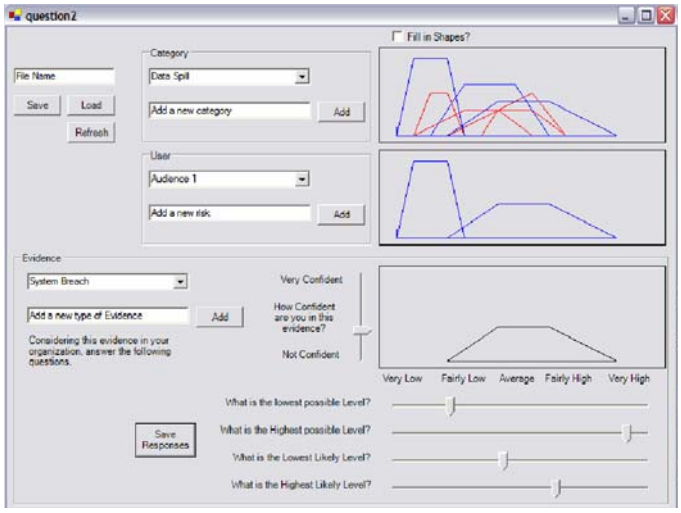


Figure 8: Representation of the compound opinions of multiple elements of evidence by multiple experts

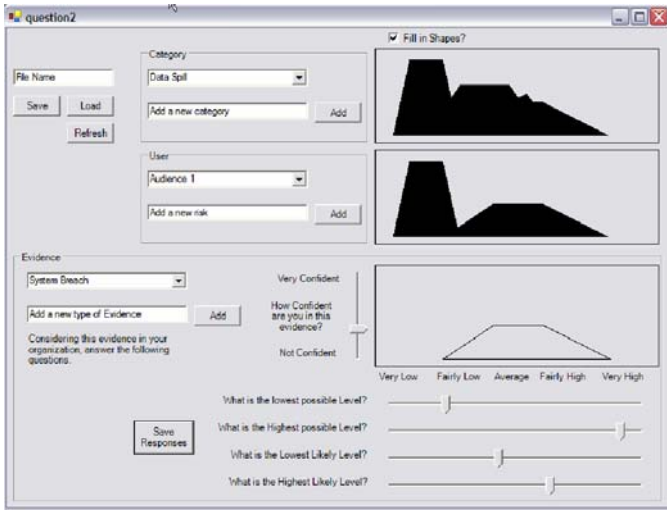


Figure 9: The fuzzy sum of all of the evidence from the prototype possibility risk analysis engine

5. Conclusion and future research

The prototype possibility risk analysis engine demonstrates that the design theory appears to operate correctly in the production of risk analysis artifacts based on alternative arithmetic frameworks. Work remaining includes introducing the artifact into real-world settings and evaluating the ability of the artifact to produce effects that support the hypotheses.

Thus far, the early indications are quite promising. The prototype was demonstrated to a group of practicing risk managers, and their informal evaluations were gathered. In general the responses were supportive of the hypotheses. But further research is needed to more rigorously evaluate the artifact. One avenue of future research includes the training of risk managers on the artifact and observation of their subsequent use and opinions about the tool. Using current risk managers will allow those who currently use probability based tools to compare the relative strengths of the prototype verses current risk assessment tools. In addition, further development could be undertaken to fix and improve the prototype based on the users feedback.

Overall this research demonstrates the feasibility of using possibility logic to analyze uncommon and currently unknown threats. While further work is needed to confirm the benefits of possibility based arithmetic in these situations, the current prototype indicates the promise of a fundamentally accurate method of examining non-frequency data in risk assessment exercises.

References

Alberts, C. J., & Dorofee, A. J. (2001). *OCTAVE Criteria, Version 2.0*: Software Engineering Institute.

Baskerville, R. (1993). Information systems security design methods: Implications for information systems development. *ACM Computing Surveys, 25*(4), 375-414.

Baskerville, R. (2005a). Best Practices in IT Risk Management: Buying safeguards, designing security architecture, or managing information risk? *Cutter Benchmark Review, 5*(12), 5-12.

Baskerville, R. (2005b). Information Warfare: A Comparative Framework for Business Information Security. *Journal of Information Systems Security, 1*(1), 23-50.

Baskerville, R., & Portougal, V. (2003). A Possibility Theory Framework for Security Evaluation in National Infrastructure Protection. *Journal of Database Management, 14*(2), 1-13.

Baskerville, R., & Sainsbury, R. (2006, 15-16 March). *Analyzing Risk of Improbable Events: Managing Business Continuity and Information Warfare.* Paper presented at the International Conference on i-Warfare and Security, University of Maryland, USA.

Baskerville, R., & Stage, J. (2004). Prototyping. In W. S. Bainbridge (Ed.), *Encyclopedia of Human-Computer Interaction* (Vol. 2, pp. 582-586). Great Barrington, MA: Berkshire Publishing Group.

Boar, B. (1984). *Application Prototyping: A Requirements Definition Strategy for the 80's.* New York: J. Wiley.

Courtney, R. (1977). Security risk assessment in electronic data processing. In *AFIPS Conference Proceedings NCC* (Vol. 46, pp. 97-104).

Hovav, A., & D'Arcy, J. (2003). The impact of Denial-of-Service attack announcements on the market value of firms. *Risk Management and Insurance Review, 6*(2), 97-121.

Jones, A., & Ashenden, D. (2005). *Risk Management for Computer Security : Protecting Your Network & Information Assets*. Oxford: Butterworth-Heinemann.

Klir, G. J., St. Clair, U. H., & Yuan, B. (1997). *Fuzzy Set Theory: Foundations and Applications*. Upper Saddle River, N.J.: Prentice-Hall.

Mason, R., & Carey, T. (1983). Prototyping interactive information systems,. *Communications of The ACM 26*, 347-354.

Naumann, J., & Jenkins, A. (1982). Prototyping: The New Paradigm for Systems Development,. *MIS Quarterly*, 29-44.

Rine, D. (1978). *Possibility theory: As a means for modeling computer security and protection* Paper presented at the Proceedings of the eighth international symposium on Multiple-valued logic table of contents, Rosemont, Illinois, United States.

Simon, H. A. (1996). *The Science of the Artificial* (3rd ed.). Cambridge, Mass.: MIT Press.

Vaishnavi, V., & Kuechler, W. (2004). Design Research in Information Systems. Retrieved October 2, 2004, from http://www.isworld.org/Researchdesign/drisISworld.htm

Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an information system design theory for vigilant EIS. *Information Systems Research, 3*(1), 36-59.

Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (2004). Assessing Information System Design Theory in Perspective: How Useful Was Our 1992 Initial Rendition? *JITTA : Journal of Information Technology Theory and Application, 6*(2), 43-58.

Weil, S. (2004). How ITIL Can Improve Information Security. *Security Focus* Retrieved 25 June 2006, from http://www.securityfocus.com/infocus/1815

Modified RSL as a Countermeasure against Differential Power Analysis

Minoru Sasaki, Keisuke Iwai and Takakazu Kurokawa
National Defense Academy, Japan
g44037@nda.ac.jp
iwai@nda.ac.jp
kuro@nda.ac.jp

Abstract: Ever since embedded cryptosystems face threats arisen from side channel attacks such as Differential Power Analysis: (DPA), many countermeasures have been proposed. However, these countermeasures are usually vulnerable in the realistic circuits because of the distributions of gate delays. Among those, Random Switching Logic: (RSL) takes account of such delays, and masks internal variables with random numbers. This countermeasure is classified as the primitive gate level countermeasure against DPA, and has already been proved to have high resistant to DPA. However, a special circuit design for generating enable signals is necessary for RSL, which fact increases its circuit scale. This paper proposes an improved method of RSL named as "Modified Random Switching Logic: (MRSL)". MRSL requires no enable signals, which fact reduces a vast amount of time and effort to design RSL circuit, and also can reduce the circuit scale of MRSL drastically from that of RSL. Considering all possible state changes of MRSL gates, MRSL can be proved to accept any distributions of gate delays. As an evaluation for the proposed MRSL, 2-input MRSL NAND gate as well as multi-input MRSL gates are confirmed to have enough DPA resistance compared with other representative DPA countermeasures. Furthermore, S-BOX in DES circuit using composite fields on Field Programmable Gate Array: (FPGA) based on MRSL is proved to have an excellent ability to hide internal variables. The leakage voltage of the proposed MRSL after DPA is almost the same as that of RSL, and is confirmed to be far less than that of Trichina's gate, and normal gate. The circuit scale of MRSL can be reduced drastically from that of RSL as well as of Trichina's gate, which fact leads to its possible usage to the embedded cryptosystems.

Keywords: DPA, side channel attack, DES, FPGA

1. Introduction

Most modern cryptographic algorithms separate their secret keys from themselves, so that it is not necessary to keep these algorithms in secret. These open cryptographic algorithms can reveal any design flaws faster by public debate and constant challenges from a community of cryptanalysts. It is indeed only their secret keys on which the securities of modern cryptographic algorithms depend.

There are several methods for code-breaking such as Brute Force Attack, Cryptanalysis, and Side Channel Attacks as shown in Figure 1. Brute Force Attack requires an enormous amount of computing so that code-breaking is prevented in realistic time. The Cryptanalysis is a method for obtaining the significance of encrypted information with specific analysis in order to find the secret key. Differential cryptanalysis proposed by Biham et al.(Biham, Shamir 1991) analyzes a pair of DES's ciphers (NIST 1999) each of which is encoded from different plain-texts. Matsui (Matsui 1993) proposed linear cryptanalysis to find affine approximations which require 243 known plaintexts. However, these methods are not usually practical in realistic cryptosystems. They are known to be more realistic (in terms of time, memory requirements, etc.) than Brute Force Attack, nonetheless they require a great number of plain / cipher texts and long computational time to find a secret key.

Side Channel Attacks are practical methods to obtain a secret key in realistic time. They use information from the physical implementation of cryptosystems, rather than theoretical vulnerability in the cryptographic algorithm. As shown in Figre1, there are several practical methods in Side Channel Attacks such as Timing Attack, Power Analysis, Electromagnetic Analysis or even Acoustic Analysis. Among them, Cache Attack, and Timing Attack are methods to reveal a secret key by carefully measuring the amount of time required to perform key operations (Kocher 1995) (Kocher 1996). Cache Attack uses correlation between memory access time of data and their hit rate in cache memory (Tsunoo, Tsujihara, and Minematsu, Miyauchi 2002). Power Analysis focuses on relating changes of power consumption to changes in the internal state of a cryptosystem (Kocher, Jaffe, and Jun 1999).

Power Analysis can be divided into SPA (Simple Power Analysis) and DPA (Differential Power Analysis) (Kocher, Jaffe, and Jun 1999). These methods are the most attractive methods in Side Channel Attacks. Modern cryptographic devices change their power consumption level according to the switching frequency of logic gates. These power analyzing methods exploit the relationship between the circuit switching and their

power consumptions.

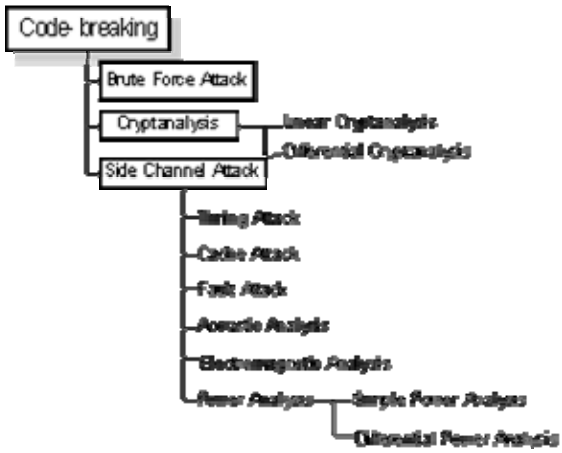


Figure 1: A classification map of code-breaking methods

SPA is a technique based on a single power trace of a cryptographic circuit, since their power consumption depends on operations of a cryptosystem. As a typical example, SPA wave trace of DES is shown in Figure 2. DES algorithm consists of the same 16 stages, each of which we can easily find out from the SPA wave trace in this figure. As a result, the inside operations of cryptosystem are apparently leaked to its power consumption.

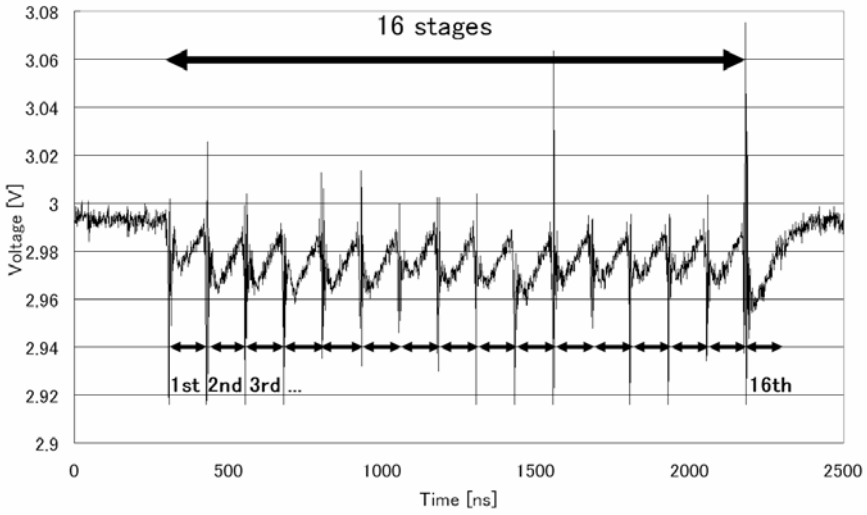
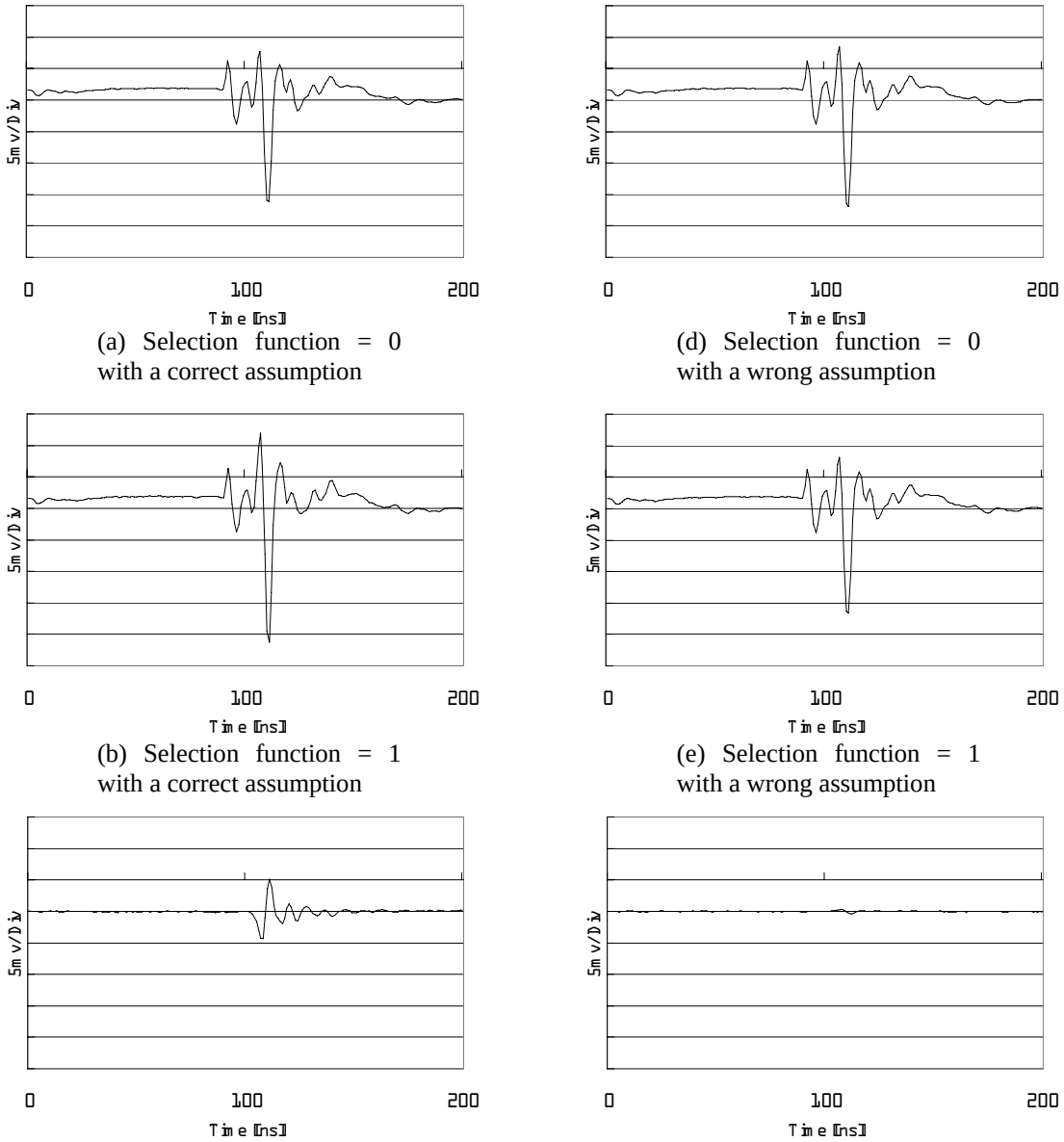


Figure2: SPA wave trace of DES

DPA also attacks cryptographic devices by tracing their power consumption during their cryptographic operations. DPA is different from SPA terms that it uses using statistical analysis of large number of traces. These traces are first divided into two groups according to a value of internal key which is calculated using a selection function (Kocher 1996). This function outputs a value of internal key as a certain assumption using cipher text and plain text as its inputs. DPA wave traces are then averaged to two traces. Finally the differences of these two traces are calculated as DPA wave trace. DPA wave traces of S-BOX1 in DES are shown in Figure 3 as an example. Here F function in DES consists of 8 S-BOXes (S-BOX1, S-BOX2…). Each S-BOX which has 6-bit inputs and 4-bit outputs works as substitution function.

Figure 3 (a) and (b) show the average traces of groups whose selection function are 0 or 1 with a correct assumption of an internal key respectively. Figure 3 (c) shows the subtracted result of Figure 3 (b) from Figure 3 (a) which accords with a DPA result of S-BOX with a correct assumption of an internal key. In this case, we can observe a group of pulses on DPA wave trace. On the other hand, Figure 3 (d) and (e) are the average traces of groups whose selection function are 0 or 1 with a wrong assumption of an internal key respectively. Figure 3 (f) shows their subtracted result which accords with a DPA result of S-BOX with a wrong assumption of an internal key. In this case, we cannot observe such a pulse. In case of common key cryptosystem, the number of cipher texts required for DPA is much smaller than that for cryptanalysis. It is possible to guess the secret key of DES using no more than ten thousand pairs of cipher text and traces of

power consumption.



(a) Selection function = 0 with a correct assumption

(d) Selection function = 0 with a wrong assumption

(b) Selection function = 1 with a correct assumption

(e) Selection function = 1 with a wrong assumption

(c) Subtraction (b) from (a) with a wrong assumption

(f) Subtraction (e) from (d) with a wrong assumption

Figure 3: DPA wave trace against S-BOX1 in DES

2. Countermeasures against DPA

Since Kocher et al. (Kocher, Jaffe, and Jun 1999) proposed side channel attacking methods in 1998, several countermeasures have been introduced to hide internal variables of a cryptographic device from power consumption.

Major countermeasures against DPA in hardware level are not only transistor level but also logic gate level. Countermeasures in transistor level such as SABL (Sense Amplifier Based Logic) (Tiri, Akmal, and Verbauwhede 2002) (Tiri, Akmal, and Verbauwhede 2003) are easy and efficient for developers to implement. The power consumption of a SABL does not change in its transition, including the degenerated events in stable state. In every logical operation, a SABL gate charges its capacitance to a constant value. SABL has a transition time and it consumes a fixed power that is not related with transition probability following to the change of its input signals. However, SABL sometimes costs more than logic level countermeasures, because there are no SABL gates in the usual standard CMOS (Complementary Metal Oxide Semiconductor) cell libraries. In case of implementation on FPGA (Field Programmable Gate Array), we can not use such libraries either. Furthermore, this method is more power consuming and requires twice

as many transistors than conventional static CMOS gates. Tiri et al. proposed WDDL (Wave Dynamic Differential Logic), (Tiri, Verbauwhede 2004) as a logic gate level countermeasure. WDDL uses the dual-rail pre-charge logic, and has DPA resistance. However, it is necessary to design a pair of WDDL gates which switches complementary(Suzuki, Saeki, and Ichikawa 2004) to equalize their load capacity. As logic level countermeasures, masking methods to cover internal variables by random value are often adapted as a countermeasure for common key cryptosystems. Internal variables are masked with random value, so that the power consumption does not depend on internal variables, nonetheless it relates to the random value. Among many masking methods, we are especially interested in RSL method (Suzuki, Saeki, and Ichikawa 2004), and we will propose MRSL as a modification of RSL with easier timing management and smaller amount of hardware.

2.1 Masking methods

The basic idea of masking methods to hide input values from DPA is the addition of exclusive-or operations with a random value to the operation by a module or by a primitive gate as shown in Figure 4. There are the following two reasons for the leakage of internal variables to power consumption, unequal probability of signal transitions and hazards generated by signal delay. A random masking value should be statistically independent of input values and uniformly distributed. The input values (X) to a module are hidden by the exclusive-or operation with a random value (R) as masking. Therefore, the cryptosystem should handle the masked value (X ^ R) instead of input values (X). As a result, internal variables can not be guessed from traces of power consumption. In order to satisfy a condition that the transition probability of a module should be even for all the inputs, several masking methods such as bitwise Boolean functions(Messerges 2001), Masked AND(Trichina 2003), MAND(Shimizu 2004), and RSL (Suzuki, Saeki, and Ichikawa 2004) shown in Figures 5, 6, 7, and 8 respectively are proposed.

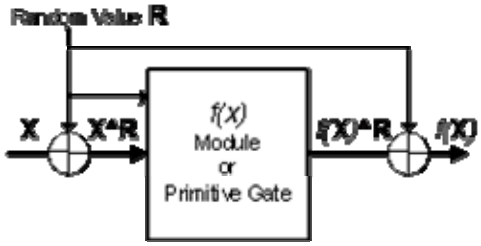


Figure 4: A model of masking method

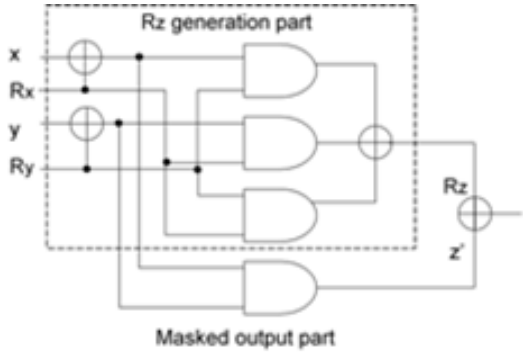


Figure 5: A bitwise Boolean function

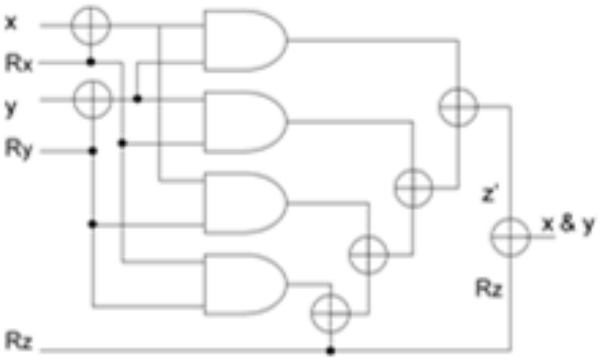


Figure 6: Masked AND

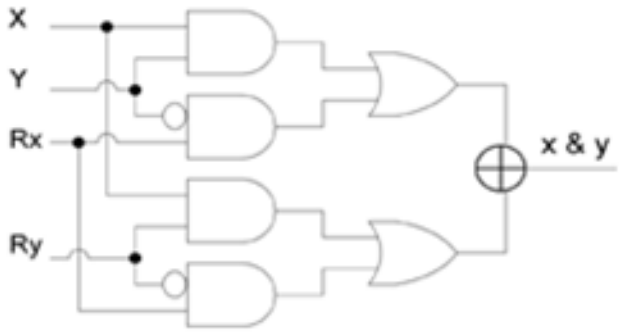


Figure 7: MAND

These methods take a countermeasure of transition probability into consideration effectively, however, there are still possibilities for generating hazards by the signal delays. Since hazards influence leakage of internal variables significantly, it is necessary to prevent hazards caused by signal delays.

2.2 Random switching logic (RSL)

Random Switching Logic (RSL) (Suzuki, Saeki, and Ichikawa 2004) was proposed to solve problems caused by signal delays which are inherent in the conventional masking methods. RSL is a countermeasure using a single rail logic based on CMOS technology. RSL can be implemented as CMOS transistor circuit as well as LUTs (Look up Tables) of FPGA (Suzuki, Saeki, and Ichikawa 2004). Figure 8 shows a basic construction method of RSL module. With a slight modification, the function of this RSL module can be equal to a masked AND gate or a masked NAND gate. In this section, we will explain an RSL NAND gate which consists of an RSL module and three XOR gates. This module has 4-inputs (x), (y), an enable signal (en), and a random value (r), and one output (z). Table 1 shows the truth table of RSL NAND gate when en = 1. Since RSL is a random masking method, inputs (x) and (y) are masked with a random value (r) before the operation at RSL module. The internal operation of RSL NAND module can be defined as $c = en \cdot \left(\overline{a} \cdot \left(\overline{b} + \overline{r}\right) + \left(\overline{b} \cdot \overline{r}\right)\right)$. Therefore the output of RSL NAND gate (z) can be generated by X-ORing the output of RSL module (c) with the random value (r) when (en) = 1. When en = 0, the output of RSL module (c) becomes 0, in order to prevent uncertain hazards caused by signal delays. It is essential to manage the timing of enable signal (en) by a controller to prevent hazards made by signal delays.

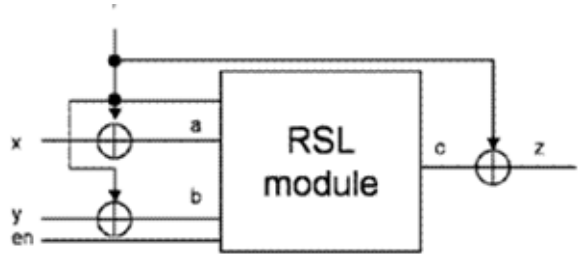


Figure 8: RSL module

Table 1: Truth table of RSL-NAND gate when (en) = 1

x	y	r	a	b	c	z
0	0	0	0	0	1	1
0	0	1	1	1	0	1
0	1	0	0	1	1	1
0	1	1	1	0	0	1
1	0	0	1	0	1	1
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	0	0	1	0

RSL has the following two properties:

1. RSL executes masked-operations for all the input/output signals using one bit of random value denoted as (r) in Figure 8.
2. RSL executes operations while enable signal (en) is 1, otherwise outputs 0.

The first property prevents inclined signal transition probability. As a result, the transition probabilities of the output (c) to be 1 or 0 always become even. The second property prevents transitions which do not depend on random value. By enabling RSL module after all the input signals (a, b, r) arrive to RSL module, the transient hazards can be suppressed. As a result, RSL can be safe against DPA, if the random value (r) is uniform. To implement multi staged-RSL module, we have to manage the timing of enable signals accurately. The easiest method for its implementation is the usage of a clock signal to which all the enable signals for RSL modules synchronize. However, this approach needs the same number of clock cycles as the stage number of this module, which fact decreases latency of this module. Multi staged-RSL module can also be implemented as an asynchronous combinatorial circuit for high latency. However, this approach needs precise measurements of all the gate delays and also needs timing optimization of the entire enable signals. According to the increase of stages, a management of enable signals becomes harder and also the amount of hardware to manage timing increases. Even if RSL is an effective countermeasure against DPA, it is very difficult to manage timing of all the enable signals.

3. Modified random switching logic (MRSL)

To implement effective multi staged-RSL module as an asynchronous combinational circuit, a precise timing design of enable signals is necessary. The easiest management method for enable signals of RSL modules is applying synchronous design using a clock signal. Since this approach may decrease the latency of the multi staged-RSL module, there still remains room for improvement.

3.1 Basic construction of MRSL

Here, we will propose a modification method of RSL named as “MRSL” that does not need any enable signals. Figure 9 shows a basic construction of an MRSL module and its truth table is shown in Table 2. MRSL does not generate hazards as long as it is initialized before its operation. The transition probabilities of the output (c) in Table 2 to be 1 or 0 become even to all the possible input patterns. As a result, the power consumption of an MRSL module depends on the value of the random number (r) only. A new requirement that MRSL must be initialized before its operation can delete an enable signal for RSL. As an initial condition, (a, b, r) must be set to (0, 0, 0) or (1, 1, 1).

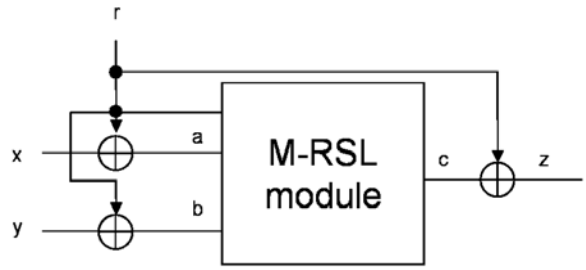


Figure 9: MRSL module

Table 2: Truth table of MRSL-NAND

x	y	r	a	b	c	z
0	0	0	0	0	1	1
0	0	1	1	1	0	1
0	1	0	0	1	1	1
0	1	1	1	0	0	1
1	0	0	1	0	1	1
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	0	0	1	0

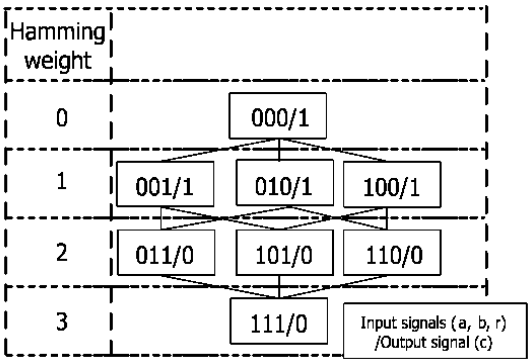


Figure10: Transition diagram of MRSL-NAND gate

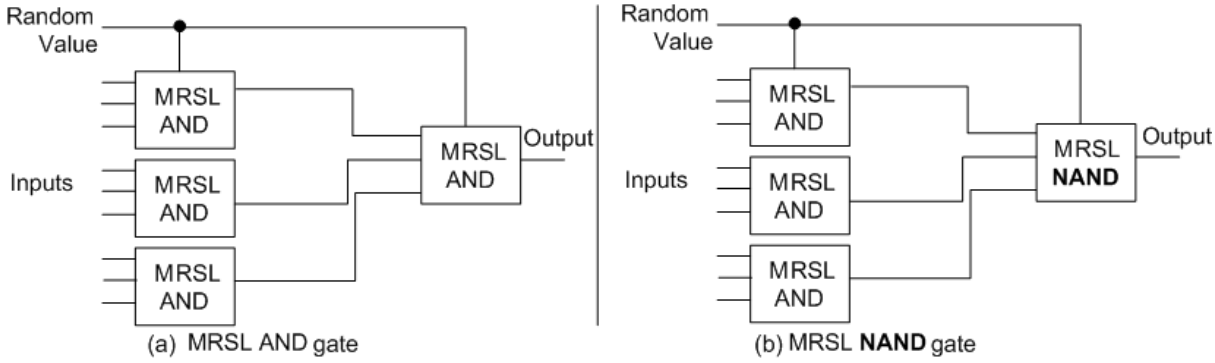


Figure 11: Multi-input MRSL gates

Figure 10 shows the transition diagram of MRSL NAND gate as an example. There are eight possible input patterns, all of which are classified into four levels according to their Hamming weight in this figure. Initially, all of the input (a, b, r) are set to (0, 0, 0), so that MRSL outputs 1. Starting from this initial point, MRSL changes its state following to the change of its input values. MRSL always generates no hazards regardless of the arrival timing of signals to its inputs. Even if 2-bit inputs change at a same time, MRSL generates no hazards too. It is possible to construct multi-input AND/NAND gates using multi staged-MRSL AND/NAND gates as shown in Figure 11. Multi-input AND gates can be composed of fundamental MRSL AND gates which are implemented on each LUT. In cases of multi-input NAND gates, they can be composed of a multi-input AND gate, which has the same number of inputs, by replacing an MRSL AND gate at the last stage by an MRSL NAND gate. MRSL method requires initialization of their input values every time before its operation without enable signals. Since it is difficult to manage many enable signals which are necessary for multi staged-RSL gates, our MRSL approach can free from such confusing management.

3.2 Initialization of MRSL module

A multiplexer is necessary in front of the MRSL module for the initialization. By changing the ‘‘Select’’ signal to this multiplier (MUX) as shown in Figure 12, MUX selects an initial value or the operation value. Additional requirement of a multiplexer increases total amount of MRSL hardware. However, MRSL needs no enable signal, which fact drastically reduces the total amount of hardware required for MRSL from that for RSL.

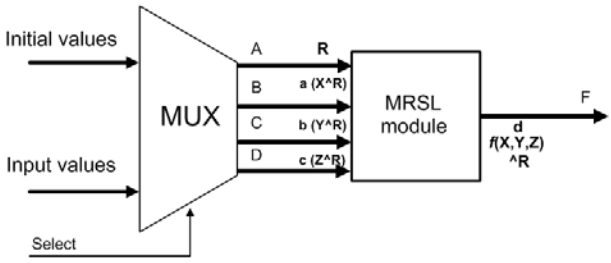


Figure 12: An initialization of MRSL

3.3 Advantage of MRSL

Here, features of MRSL are compared with those of RSL. First of all, it can be said that MRSL is advantageous in a circuit scale and an easy design. Because the multi staged RSL requires multiple enable signals, the hardware to generate these enable signals and their control circuit requires more circuit scale than MRSL.

On the other hand, the multi staged-MRSL only requires all zero inputs as a simple initialization. Moreover, MRSL can use additional inputs instead of control signals for RSL. This fact reduces the total required number of LUTs in FPGA for the entire circuits of MRSL compared with RSL as shown Figure 13. Most FPGA applies LUTs each of which has 4 inputs and 1 output. When RSL AND gate is implemented on an LUT as shown in Figure 13 (a), it can only have 2-input (a, and b). On the other hand, MRSL AND gate on an LUT can have 3-input (a, b, and c). As a result MRSL module can have an additional one input compared with RSL module.

Though MRSL requires initialization process, its design procedure can be drastically simplified from that for RSL. This fact will be shown in detail in the following subsection 4.3.

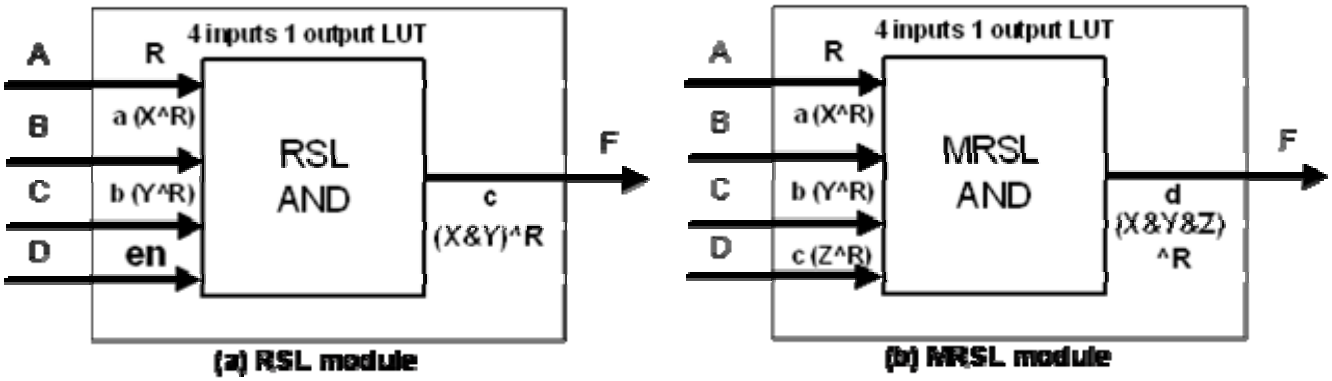


Figure 13: Implementation methods of RSL module and MRSL module on a 4-input LUT in FPGA

4. Implementation and experimental results

4.1 Evaluation environment

In this section, we will explain how to implement MRSL on FPGA. Table 3 shows our evaluation environment. We evaluated MRSL using a SCAPE (Side Channel Attack Platform for Evaluation) board developed by Mitsubishi Electric Corporation (Ichikawa, Suzuki, and Saeki 2004).

Table 3: Evaluation environment

Design Environment	Language	Verilog-HDL
	Logical synthesis and Place and Route	ISE version 6.1.03i
Measurement Environment	Target Board	SCAPE(FPGA) (XCV1000-5-BG560C)
	Oscilloscope	IWATSU DS-4354ML

4.2 DPA results of a 2-input NAND gate

First, we implemented a 2-input MRSL NAND gate, and evaluated its validity as a countermeasure for DPA. Several random masking gates as countermeasures were also evaluated to compare with MRSL gate. Figure 14 shows the DPA results of these gates. We used one input signal to each gate as selection bit. Random numbers generated by LFSR over GF (2^{16}) were used as another input signal and a random masking value. Power consumption of these gates was measured 80,000 times for each gate. The resolution of the oscilloscope was set to 500M samples/second. The entire wave traces presented in this paper are shown under the same condition. Figure 14 (a) is a DPA result of an ordinary 2-input NAND gate without countermeasure. Figure 14 (b) and (c) are both DPA results of 2-input RSL NAND gate. Their

difference is the arrival timing of the enable signal. Figure 14(d) shows a DPA result of the proposed MRSL.

Following these traces, (a) showed a big peak and so did (b). This fact accords with the leakage of information through DPA when using these gates. While, MRSL trace in (d) and RSL trace in (c) did not show such peaks, which fact proves the DPA resistance of these gates.

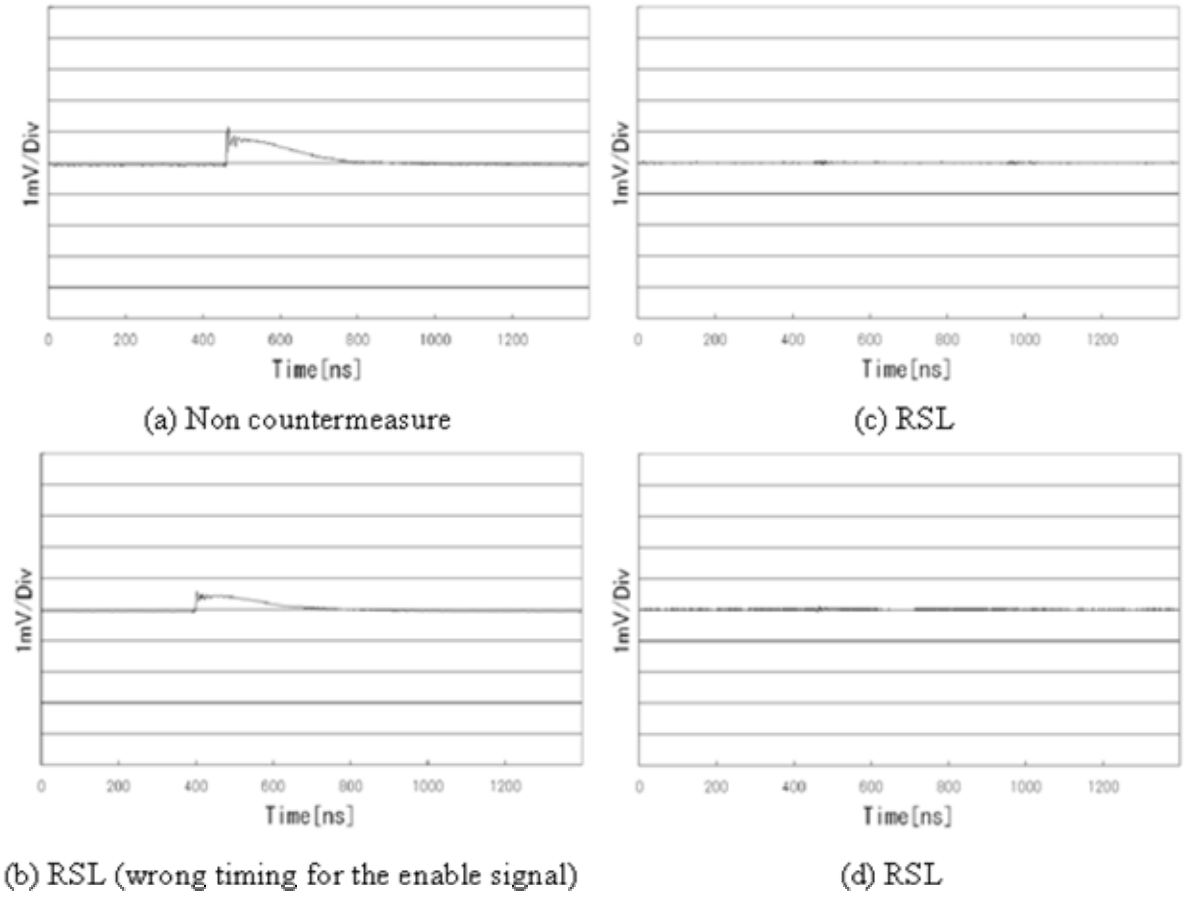


Figure 14: DPA results for 2-input NAND gates

4.3 DPA results of multi-input gates

Many available FPGA chips on the market are composed of SRAM-based LUTs which have 4-input signals and 1 output signal. Since RSL needs at least 4-input signals (input a, input b, enable signal, and a random value) for a logical function, one LUT can realize 2-input RSL logic. On the other hand, we can design 3-input MRSL logic using the same LUT, since it does not have an enable signal. As a result, 3-input MRSL gate consisting of one LUT has an additional one input signal compared with RSL gate.

Referring to the transition diagram for a 2-input MRSL NAND gate shown in Figure 10, we can make a transition diagram for a 3-input MRSL NAND gate for its implementation on one LUT in FPGA. Multi-input MRSL NAND gates as well as MRSL AND gates can be composed of MRSL AND gates and one NAND gate as shown in Figure 11.

Following to this construction method, we implemented multi-input RSL NAND gates and MRSL NAND gates, and evaluated their validity as countermeasures for DPA. All the traces did not show any peaks by DPA as shown in Figure 15. This fact proves the DPA resistance of these gates. In addition, the amount of LUTs for MRSL is reduced from RSL, since MRSL requires no enable signal.

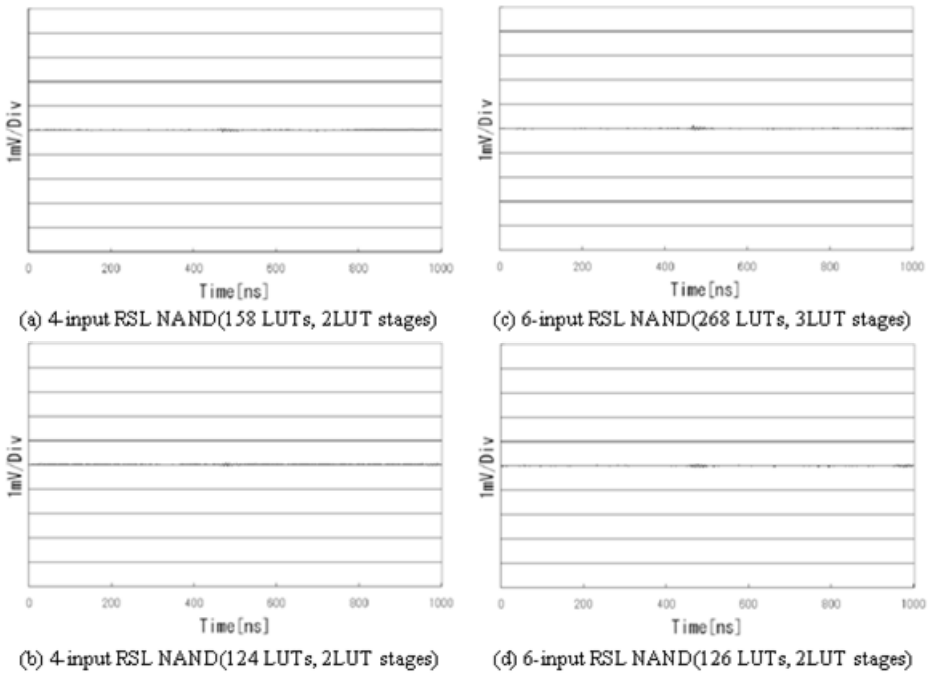


Figure 15: DPA results for 4-input NAND gates and 6-input NAND gates

4.4 DPA results against SBOX in DES

In practice, we evaluated effects of DPA against SBOX in DES by implementing four kinds of S-BOX1 composed of SOP (sum of product) as no countermeasure modules, RSL gates, 2-input MRSL gates, and 3-input MRSL gates. Figure 16 shows their DPA results. Figure 16 (a) is a DPA result of traces of an ordinary S-BOX without countermeasure. This figure showed a big peak after DPA. However, the others showed no such peak. These DPA results prove the effectiveness of RSL and MRSL as DPA countermeasures. Furthermore, the amount of hardware required to implement MRSL is obviously less than that for RSL. S-BOX1 with no countermeasure requires the least amount of LUTs, however this method is weak against DPA. As a result, 3-input MRSL became the best countermeasure as a cost-effective method.

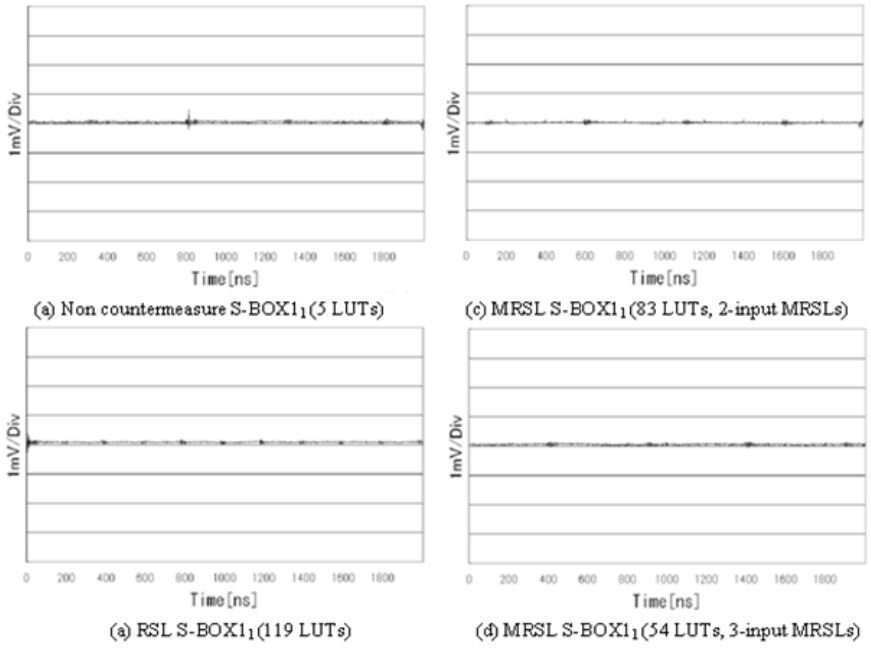


Figure 16: DPA results for S-BOX1 in DES

5. Conclusions

In this paper, we proposed MRSL following RSL's properties to equalize transition probabilities and suppress transient hazards. MRSL was proven to be effective as a countermeasure against DPA through realistic experiments. MRSL gates require no enable signals, and are easy to be implemented as a FPGA circuit. If nonlinear transformation circuit such as S-BOX is implemented with MRSL instead of RSL, we can decrease the amount of hardware drastically. The implementation results of 2-input MRSL gate, multi-input MRSL gate, and S-BOX1 showed effectiveness of the proposed MRSL as a countermeasure against DPA.

References

Biham,E. Shamir,A. (1991) "Differential cryptanalysis of DES-like cryptosystems", Vol 4, No.1, pp3-72.

Ichikawa,T. Suzuki,D. and Saeki,M (2004) "An Attack on Cryptographic Hardware Design with masking Method", ISEC 2004, Vol.104, No.200, pp.119-126, in Japanese.

Kocher,P. (1995)
"Cryptanalysis of Diffie-Hellman, RSA, DSS, and Other Systems Using Timing Attacks", [online], CiteSeer.IST, http://citeseer.ist.psu.edu/kocher95cryptanalysis.html

Kocher,P. (1995) "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems", [online], Faculty of Information Technology, http://www.fit.vutbr.cz/~cvrcek/cards/timingattack/timingat.htm.en

Kocher,P. Jaffe,J. and Jun.B. (1999) "Differential power analysis", [online], Cryptography Research, http://www.cryptography.com/resources/whitepapers/DPA.pdf

Matui,M. (1993) "Linear cryptanalysis method for DES cipher", EUROCRYPT'93, LNCS 765, Springer-Verlag, pp.386-397.

Messerges,T. (2001) "Securing the AES finalists against power analysis attack", FSE2000, LNCS 1978, pp.150-164.

NIST. (1999) "Data Encryption Standard (DES)", [online], FIPS,http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

Shimizu,H. (2004) "Countermeasure against Side Channel Attack using Mask Logic Elements", ISEC 2004,Vol.104, No.315, pp.15-19, in Japanese.

Suzuki,D. Saeki,M. and Ichikawa,T. (2004) "Switching Logic: A Countermeasure against DPA based on Transition Probability", [online], Cryptology ePrint Archive, http://eprint.iacr.org/2004/346.pdf

Tiri,K. Akmal,M. and Verbauwhede,I. (2002) "A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on SmartCards", [online], UCLA, http://www.ee.ucla.edu/~tiri/files/esscirc2002.pdf

Tiri,K. Akmal,M. and Verbauwhede,I. (2003) "Securing Encryption Algorithms against DPA at the Logic Level: Next Generation Smart Card Technology", CHES 2003, LNCS 2779, pp.125-136.

Tiri,K. Verbauwhede,I. (2004) " A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation", [online], UCLA, http://www.ee.ucla.edu/~tiri/files/date2004.pdf

Trichina,E. (2003) "Combinational logic design for AES subbyte transformation on masked data", [online], Cryptology ePrint Archive, http://eprint.iacr.org/2003/236.pdf

Tsunoo,Y. Tsujihara,E. and Minematsu,K. Miyauchi,H. (2002) "Cryptanalysis of Block Ciphers Implemented on Computers with Cache", [online], College of Engineering, Oregon State University, http://web.engr.oregonstate.edu/~aciicmez/osutass/data/Tsunoo02.pdf

A Framework for Relating Cyberspace Operations to the Cognitive and Physical Domains

Tiffiny Smith[1], Pamela Woolley[2], Robert Mills[1], and Richard Raines[1]
[1]Air Force Institute of Technology, Fairborn, OH, USA
[2]Fairchild AFB WA, USA
Tiffiny.Smith@afit.edu
Pamela.Woolley@fairchild.af.mil
Robert.Mills@afit.edu
Richard.Raines@afit.edu

Abstract: The United States Air Force recently released a new mission statement adding cyberspace as an area of operations to achieve military goals and objectives. The ability to clearly articulate exactly what is and what is not cyberspace will greatly simplify the assignment of roles and missions, as well as the development of cyber warfare doctrine. With that said, there remains much debate about what cyberspace really is, and whether it is truly a domain in which warfare can be waged. In this paper we offer a new perspective on cyberspace that addresses many of the stumbling blocks of previous definitions. We begin with the network centric warfare model that identifies physical, cognitive, and information domains. By focusing on information flows, we develop a model of cyberspace which does not stand on its own but rather is part of a larger whole that encompasses both virtual and physical domains. We go on to show how this framework overcomes limitations of existing cyberspace definitions and will also help clarify discussions about cyber roles and missions, organizational responsibilities, and force development.

Keywords: Cyber warfare, information warfare theory.

1. Introduction

> The mission of the United States Air Force is to deliver sovereign options for the defense of the United States of America and its global interests -- to fly and fight in Air, Space, and Cyberspace. – *United States Air Force Mission*

The above mission statement for the Air Force was officially announced by Air Force leadership on December 7, 2005. (Wynne, 2005) The addition of cyberspace as a domain of operations is a monumental step, which has been likened by many to the early days of air power. As the United States Air Force prepares to take on its new mission area of cyberspace, it is faced with many questions of how to implement the new mission. These questions include, but are not limited to:

- What are the roles and missions of a cyberspace force?
- Under what conditions does a cyberspace force support other military functions?
- Under what conditions is the cyberspace force supported by other forces?

To begin to answer these and many more questions, the Air Force must first understand what cyberspace is (and is not), warfighting capabilities and effects that can be achieved in and through cyberspace, how cyberspace can support and be supported by other warfighting functions, and the relationship cyberspace has with other warfighting domains. For our purposes, we will use the following cyberspace definition, awaiting approval by the Chairman, Joint Chiefs of Staff and Secretary of Defense: *Cyberspace is a domain characterized by the use of electronics and the electromagnetic spectrum to store, modify, and exchange data via networked systems and associated physical infrastructures.* (NMS Cyberspace, 2006)

Cyberspace is a relatively new concept. As with most new concepts, a visual representation of cyberspace may aid in understanding what cyberspace is and how it relates to other domains that are more familiar to us. Once we understand how cyberspace relates to other domains, we are then better prepared to plan, organize, train and equip a cyber force.

The purpose of this paper is to relate cyberspace to familiar domains and describe the interactions of these domains. To aid us in this endeavor, we will first review existing models developed for uses similar to cyberspace. We will see that although these models provide a starting point, they all have limitations when representing cyberspace and its relationship with other domains. Because of the limitations in existing models, we will introduce a new model and explain how it will aid discussions in understanding the cyberspace domain as it relates to other domains. Then we will discuss how this model can be used for

planning, organizing, training, and equipping a cyber force. Finally, we will propose some future studies that build from this foundation that will be useful to the cyberspace force.

2. Domains of warfare

Existing domain models present a good, fundamental beginning for understanding cyberspace, its capabilities, and relationships between the domains of warfare. However, previous models were not intended to depict cyberspace, which means they have limitations and are therefore only a starting point. These models come from the Information Operations (IO) and Network Operations realms. The IO realm is one that is accepted and used by the military, and IO doctrine is fairly mature. We will review the Network Centric Warfare (NCW) model proposed by Alberts *et al*, then Romanych's model, followed by Waltz', and finally Joint Publication 3-13. After each model is introduced and a brief description is provided, we will identify the short-comings of those models when applied to cyberspace.

2.1 Network centric warfare model

The first model is the Network Centric Warfare model, shown in Figure 1. This model is composed of four domains, social, cognitive, physical and information. Each domain by itself is a "place" where warfighting happens. By *warfighting*, we mean targets are attacked and defended in order to obtain or preserve some position of relative advantage over the adversary.

Perhaps the easiest to understand is the *physical* domain, which includes sea, space, air and land. This is where people and weapons (kinetic) actually exist and function. The *cognitive* domain is also familiar, being the domain that includes thoughts, experiences, decision making and intangible thought processes of individuals. It is important to note, however, that the cognitive domain is not strictly limited to humans, but also includes computer-assisted and automated decision support systems. According to Alberts and Gartska, "These intangibles are crucial elements to Network Centric Operations." (Alberts and Gartska, 2004) The *information* domain is where "information is created, manipulated, value added and shared. It can be considered the 'cyberspace' of military operations." Finally, the *social* domain is intended to overlap the cognitive and information domains but remains its own domain. The social domain is where thoughts, experiences, knowledge, and beliefs of one person or group are shared with those of another person or group to enhance decision making. It takes the beliefs and values of many, combines it with the thoughts and makes decisions. Once one understands what each domain entails, it is important to understand that true network centric capabilities actually occur where these four domains overlap.

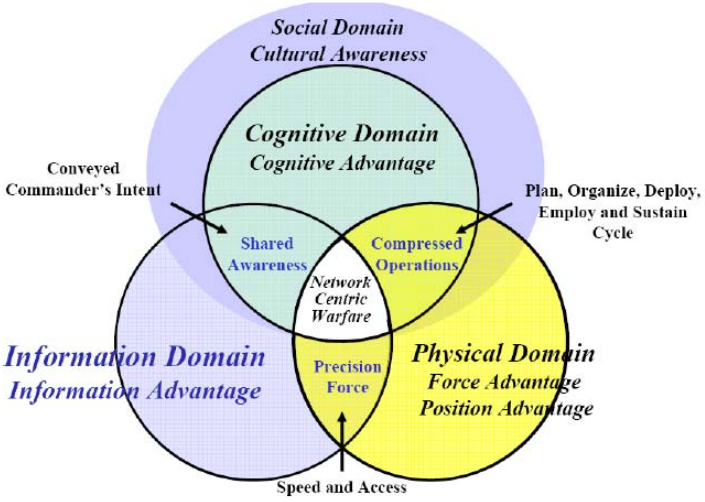


Figure 1: Network centric warfare domains (Alberts and Gartska, 2004)

When used as a model for explaining cyberspace, this model has one large limitation. The model appears to confine information to a specific sphere. The model shows little overlap of information with the other domains and implies that the other domains can exist without information. We believe this is a major flaw of this model when used to discuss cyberspace. Information is everywhere, and discussing information as a domain that can, but does not necessarily, interact with the other domains introduces confusion. For example, without information there is no cognition. We cannot think, decide, or act without information to use in our cognitive processes. As for the physical domain, we must have information to make this domain

useful. We must know what physical components are available for use, what capacity and limitations the components have. Without this information, the physical aspects are useless.

2.2 Romanych model

The second model we consider is "Domain Relationships" as shown in Figure 2 (Romanych, 2005). This model uses the cognitive, physical and information domains but dismisses the cultural domain of the NCW model. Romanych defines the cognitive domain as the human mind, the physical domain as the real world and the information domain as the intersection of the physical and cognitive domain. Romanych argues that physical domain activity generates data that is collected by information systems, which in turn create and direct the flow of information through the information domain. Finally, information is used to form perceptions and to ultimately make decisions (cognitive domain). These decisions may be subsequently communicated through the information domain via information systems to the physical domain, resulting in physical activity. Essentially Romanych is saying that all three domains must interact and that the information domain is the only way the cognitive can interact with the physical.

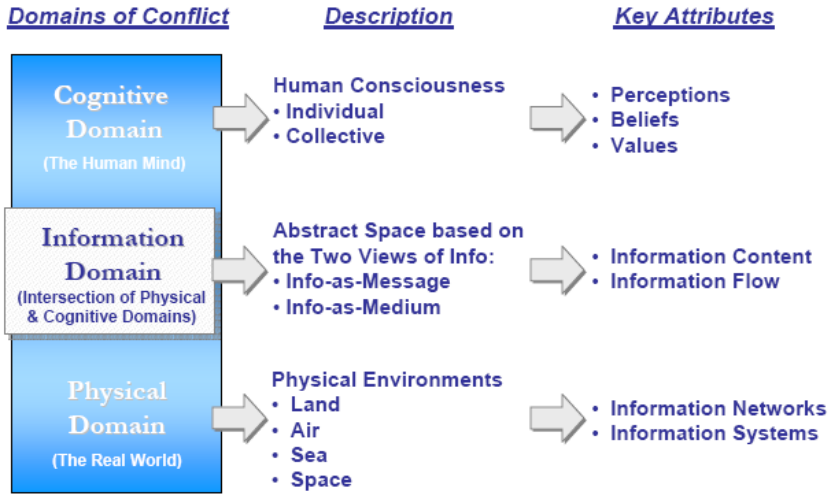


Figure 2: Romanych's domains of conflict (Romanych, 2005)

While this is a useful construct, we find it limiting when attempting to study cyberspace operations, because of the defined separation of the physical and cognitive domains connected only by the information domain. This model has a similar weakness to the NCW model in that information resides in both the cognitive and the physical world.

2.3 Waltz' model

The previous model is similar to that presented by Edward Waltz (Waltz, 1998). The biggest difference is that Waltz proposes a *perception* domain rather than a cognitive domain. Waltz also introduces the domain of *will* which he sees as "…resolve, determination, and human will…" Additionally, Waltz overlays Boyd's observe-orient-decide-act (OODA) loop (Boyd) onto these domains to demonstrate how it is used in a situation such as a conflict between two warring opponents. Waltz portrays the cycle starting in the physical domain where the first side acts. This prompts the second side to observe information. The second side orients themselves according to their perceptions, and then they decide on a course of action based on their will. Finally, the second side will act which starts the whole cycle again but this time with the roles of the first and second sides reversed.

Just as with Romanych's model, this portrays a linear model of the domains where one leads to the other, and unless the domains are next to each other, they cannot interact. Furthermore, Waltz's model shows Boyd's OODA loop as a linear process; however, the true illustration of Boyd's OODA Loop is not linear. Boyd's loop (Figure 3) actually shows that each point on his loop can influence all the other points on the loop. Boyd's OODA loop specifically illustrates these influences and demonstrates that information gathered at any point on the loop potentially impacts all other points.

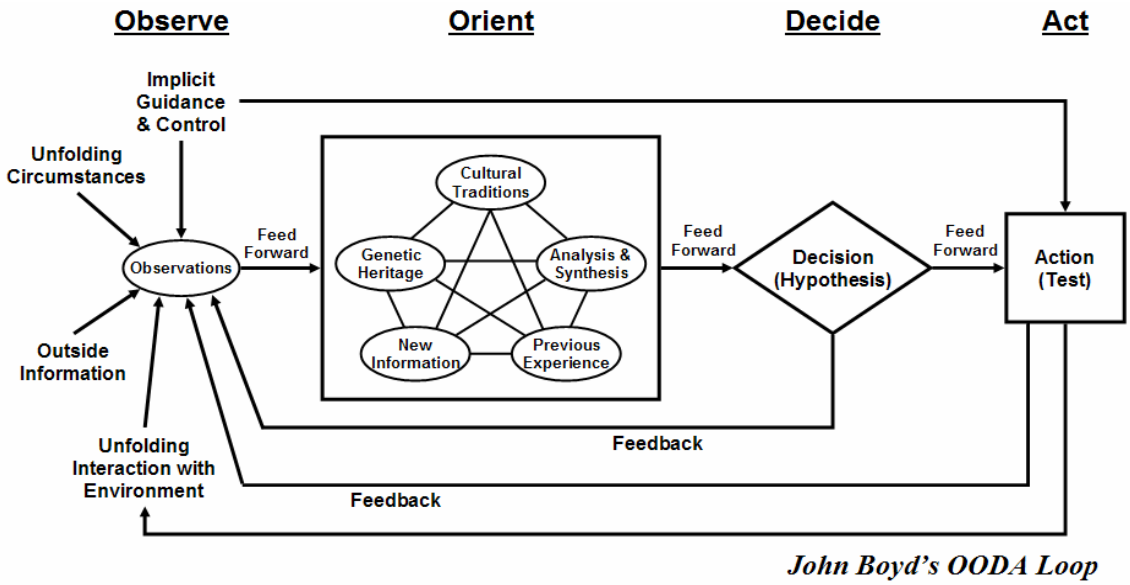


Figure 3: Observe-orient-decide-act

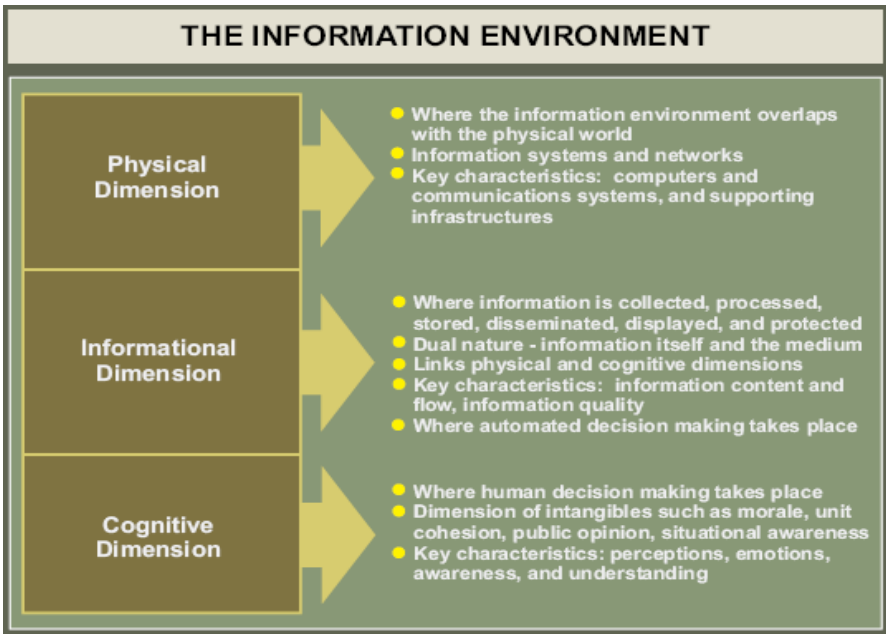


Figure 4: Information environment (JP 3-13)

2.4 Joint publication 3-13

Joint Publication (JP) 3-13, *Information Operations*, introduces the construct of an Information Environment in a similar fashion to the previous models (Figure 4). JP 3-13 defines the information environment as "…the aggregate of individuals, organizations, and systems that collect, process, disseminate, or act on information." (JP 3-13, pg I-1) The Information Environment contains three dimensions, the physical, informational, and cognitive. These are also defined similarly to the previous definitions. The concern regarding this model for cyberspace is that the cyber domain as defined earlier is not really captured well and cuts across parts of both the physical and informational dimensions. It also fails to capture some of the electromagnetic spectrum issues associated with cyberspace.

3. A model for cyberspace

When used as a starting point to represent cyberspace, these models all have the same flaw in that they consider information a separate domain from the physical and cognitive domains. However, these models do provide some beneficial features. The NCW model shows that domains overlap and work together. The JP 3-13 model acknowledges that information is everywhere and in every domain.

Based on the strengths and weaknesses of these models and the definition of cyberspace, we now propose a model for cyberspace as shown in Figure 5. As shown, information exists in each of the physical, cognitive and cyber domains. JP 3-13 acknowledges that "Even though the information environment is considered distinct, it resides within each of the four domains." By considering the definitions of information in the previous examples and the statement from JP 3-13, it is clear that information permeates all domains; therefore, it is necessary to distinguish between information and another domain that interacts with the physical and cognitive domains and also resides within the information environment; we call this the cyber, or digital, domain.

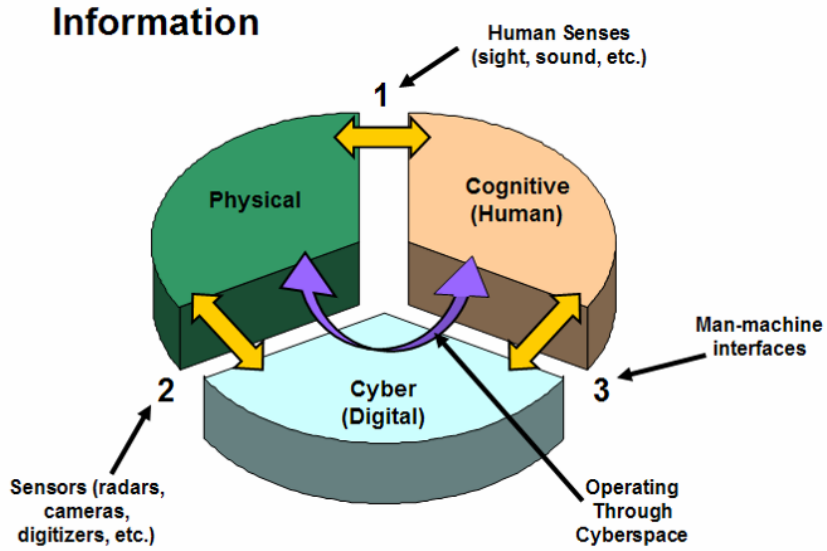


Figure 5: Cyber domain

The new model defines the physical and cognitive domains essentially the same as in the previous models. The cyber domain is inserted into the information environment, and we remove information as a domain in itself. Information is everywhere and can be in more than one place and in different forms at the same time. Another difference is that the new model does not portray an overlap of the domains but demonstrates the interactions of the domains.

3.1 Information transfer

Remembering that information is everywhere, we will focus on how information is transferred from one domain to another. The transfer occurs through sensors and interfaces between domains. Information in the physical domain can be transferred to the cognitive domain directly via the human senses: touch, sight, hearing, smell and taste. An example of this transfer can occur through two individuals speaking face to face. One individual can share information with another person using speech, facial expressions, and body language.

Information from and about the physical domain can be captured in the cyberspace domain. This is achieved by using sensors (radars, cameras, digitizers) that sense the physical environment and then represent the information in digital form. Once the information has been digitized, it can be stored, processed, or transmitted elsewhere for use. Information is also passed between the cognitive and the cyber domains. Information from the cognitive (human) domain can also be entered into the cyber domain. The mechanisms for this conversion are typically the keyboard or other human-computer interfaces. That information can be used as is, combined with other information, stored for future use or used immediately. These cross-domain transfers are represented by the yellow arrows in Figure 5.

There is also the possibility that all three domains can be used simultaneously. One example of this is when information about the physical world is captured in the cyber domain and passed to humans who can take that information and use it to make decisions. These operations *through* cyberspace can occur in both directions and are denoted by the purple arrow in Figure 5. For example, consider a remotely piloted unmanned aerial vehicle used for reconnaissance. Ground personnel control the aircraft's operation and mission systems remotely by sending signals and information through the cyber domain (tactical radio data links, satellite communications, etc.). Likewise, the aircraft sensors capture information in the physical

domain and generate digital representations (imagery for example) that are transferred to decision makers via the cyber domain.

3.2 Cyber as a domain of operations

We believe this model, despite its relative simplicity, may help aid the Air Force as it further defines its cyberspace mission. Cyberspace has two distinct components. First, it is a domain that supports and enables operations and activities in the physical and cognitive domains. Second, it is a domain in which military operations can be conducted (targets are attacked and defended) in order to achieve or preserve some relative position of advantage over the adversary. The two components are very closely related, and our model helps discuss both aspects.

When viewed as a supporting or enabling domain, cyberspace supports all military daily tasks that rely upon information transfer. Our military has become so dependent on technology, and "…information networks have become so important that many modern military tacticians, commanders and logisticians would be crippled without them and many modern military capabilities would be degraded." (Knight and McIntyre, 2005)

With the increasing need to interact among and between organizations that are geographically separated, cyberspace is essential to accommodate information exchange and collaboration between organizations. Cyberspace is a foundation for all the other functions the military performs. It is easy to see that cyberspace is the focal point of the model when considering business support and day-to-day functions of the military. Information contained in one domain can be shared with one, two, or three other domains. For example, information regarding the weather to plan an intelligence, surveillance, and reconnaissance (ISR) mission can be obtained using cyberspace to access information stored in databases and web servers. Likewise, current weather information may come from sensors in space, it may be a composite of sensor data and historical data, or it may be a prediction based on someone's knowledge.

Cyber is also a domain of conflict. Targets are attacked and defended, operations are planned, defenses are prepared, and so on, just as in the physical domains of land, sea, air and space. What differs is the nature of the targets. Targets in the cyber domain include information and data, as well as the systems that generate, store, and transmit that information. These targets can be attacked via a network attacks (denial of service) or electronic warfare, or a combination thereof.

Clausewitz noted that war is an "instrument of policy" (Clausewitz), and that while it is desirable to avoid conflict if at all possible, it is sometimes unavoidable. Even then, operations are carried out with an objective of minimizing collateral damage and loss of life. Because of this, cyberspace becomes an attractive battle environment. Waging war in cyberspace does not result in the immediate and horrific results of physical domain conflict. When we use cyberspace as a primary battlefield, our intent may even be to remove the physical domain from warfighting. Instead we can attempt to shape the cognitive domain by altering, using, or denying information to enemy decision makers.

Attacking targets in the cyber domain may result in effects in the other domains. For example, suppose we know what information systems and communication paths an adversary commander uses to develop and plan his actions. If we can gain access to those systems and alter or delete information, we may be able to influence his decision making process in the cognitive domain. Ultimately, the intent our deception (using cyberspace as a domain of operations) is to cause the adversary commander to make decisions that are advantageous to our own cause. We are trying to alter the enemy's perception of the information surrounding him. If the enemy believes, through non-kinetic means, that he is already defeated or he can do no more harm to us, then we have beat him with information in the cognitive and cyber domains, while leaving the physical domain unaffected.

Because cyberspace is a domain in which we conduct military operations, we must be able to clearly define the battlespace and understand the connections and overlaps with the other domains. This could prove challenging because of the virtual aspects of the domain. It is much easier to see how ground, air and space interrelate, and we also have procedures and mechanisms for dividing the physical space for command and control purposes. For example, we divide the terrain into regions (Pacific, Europe, and Middle East for example) and assign these areas to commanders responsible for military operations within those regions. While certain aspects of cyberspace will have regional ties, such as physical location of networking equipment, much of the domain is virtual and infinite. Cyberspace is best looked at as a global domain of operations.

4. Cyber missions

Now that we recognize cyberspace as a domain of operations, we can consider various types of cyber missions. Missions (activities) are carried out to achieve some desired *effect* (possibly many effects). While these effects can also be realized through physical (kinetic) attack, we will limit our discussion to cyber operations. Although cyberspace is a virtual domain, it is helpful to use air and space power mission areas as a basis for discussion. Cyber operations can thus be divided into the following areas:

Strategic Cyber Attack. This mission refers to the desire to attack selected targets in such a manner that the adversary will be unable to respond in force. Alternatively, the adversary may realize the futility of any such response and would decide to capitulate. There are clear parallels between the idea of a strategic cyber attack and strategic air attacks as advocated by early air power theorists such as Billy Mitchell and Giulio Douhet. Strategic cyber attacks would occur in the cyber (digital) domain of Figure 5. Targets would include critical networking infrastructure, servers, databases, and other equipment or data that is critical to the adversary's ability to function. Means for conducting a strategic cyber attack could include network attack, distributed denial of service, compromising (hacking) high value equipment, altering or destroying the adversary's data, etc. Note that these same means could be used for other missions and effects—it is not the capability or platform that determines the mission type; rather it is the intended effect.

Suppression of Enemy Cyber Defenses. This is a support mission that enables other cyber operations, such as strategic attack and counter cyber. In air operations, the adversary's air defense system can pose considerable threat to friendly aircraft and pilot safety. To improve chances for mission success, the air defense system is suppressed, either through direct kinetic attack against the command and control elements and/or electronic warfare (jamming) operations against the enemy's radar and communication links. Similarly, in the cyber domain, the adversary may have a defense network in place that can either detect or defeat our ability to operate freely. Therefore, a subset of cyber operations would involve the defeat or temporal suppression of those defenses to allow greater freedom of movement for our own operations. These operations could occur in both the cyber domain and at the interfaces between the domains of Figure 5.

Offensive and Defensive Counter Cyber. These mission areas deal with establishing the ability to act freely in the cyber domain without interference from the adversary. We deny that ability to the adversary as well. In the air and space domains, counter air operations would include destroying the enemy's aircraft and the infrastructure that supports his ability to conduct air operations. In the cyber domain, these missions would include the offensive capabilities needed to neutralize adversary defenses (firewalls, intrusion detection systems, etc.), as well as the defensive capabilities to prevent the adversary from doing the same to our systems. These operations could occur in both the cyber domain and at the interfaces between the domains of Figure 5.

Cyber Interdiction. Interdiction operations refer to those actions taken to "divert, disrupt, delay or destroy the enemy's military potential before it can be brought to bear effectively against friendly forces" (DOD Dictionary). This could involve direct attack on troops and equipment in rear assembly areas, disrupting lines of communication, and attacks on supply routes. In the cyber domain, such operations are geared towards interdicting the flow of information. Information is a commodity in much the same way that rations, ammunition, and fuel are. Therefore, attacks on the information supply chain can disrupt or delay the adversary's decision making process, which in turn makes it more difficult for him to command and control forces. Examples of how this might be done include attacks on networking infrastructure (similar to attacking roads and bridges in the physical domain), interception and denial of key communications links (radio, fiber optic, or copper), and attacks against specific servers that serve as bottlenecks or chokepoints in the adversary's OODA loop. As in the previous cases, these operations would occur in the cyber domain or at the interfaces between domains.

Cyber Exploitation and Intelligence. The *Intelligence Preparation of the Battlespace* (IPB) process will need to be refined or improved to help us better understand cyberspace as an operational domain. This includes understanding the "terrain" and how it can affect battle; enemy forces, capabilities, and intentions; friendly forces and capabilities; and other outside factors such as political climate and international law. The challenge of identifying friendly, hostile, and neutral parties cannot be underestimated. In cyberspace, it is difficult at best to determine exactly who the adversaries are, let alone their intentions and capabilities.

Visualizing cyberspace—at least in a way that will be meaningful to operational commanders—is difficult. Cyberspace is a global—and joint—area of operations. While the parallels to early air power theory are attractive, it is probably advisable to avoid looking at cyberspace as a "silver bullet" that will render previous forms of warfare obsolete. Rather, cyber operations will be most effective and useful if they can be integrated into the rest of the toolkit that is available to the operational commander. Tying these domains together will be a huge challenge, and a cyber common operational picture (COP) will be needed to help commanders better understand this battlespace. More importantly this COP will need to integrate with existing battlespace management systems used for air, space, land and naval operations.

Finally, the very nature of cyberspace makes intelligence, surveillance, and reconnaissance difficult due to the short time spans available. In the physical domain, it takes time to build, equip, and move forces in preparation for battle—perhaps months and years. In cyberspace, it is difficult to tell if someone is developing a cyber capability, and worse yet, there may be little or no warning that an attack is underway.

5. Conclusions

While the notion of cyberspace has existed for years, it has only recently been officially recognized as a domain of military operations. The US Air Force has changed its mission to flying and fighting in *air, space, and cyberspace.* There are many questions to be asked and much research to be completed regarding what can and cannot be done in cyberspace, and we expect to see much research activity in this area. Many liken today's discussions of cyberpower to discussions about air power in the early 1920s.

Unlike physical domain warfare, significant investments in materiel and people are not required to yield an effective capability. The cyber domain is one in which potential adversaries (especially ones that are not as technically dependent as the United States) can successfully fight an asymmetric campaign. If our enemy will use cyberspace as a battlefield, we must be there first, although it may already be too late. Hostile entities are already chipping away at our information infrastructure, looking for weak points and vulnerabilities. We do not even know who our enemies are or what their intentions are. We do know that we must protect our systems and be prepared to fight in this new domain.

In the early days of air power, aircraft were used for reconnaissance, but soon, the benefit of planes to support ground troops was recognized as well as the advantage of being able to go where ground troops could not with ammunition. Cyberspace is similar in that we first viewed it as a way to do business but now we are starting to see its potential for warfighting. We must understand that cyberspace is not the only battlefield and that sometimes cyberspace will support other missions while sometimes other battlefields will support cyberspace as the primary mission.

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

References

Alberts, D.S. and Garstka, J.J., Network Centric Operations Conceptual Framework, version 2.0, 2004, Evidence Based Research, Inc.

Boyd, J.R., Discourse on Winning and Losing (unpublished manuscript), available online at http://www.d-n-i.net/second_level/boyd_military.htm

Clausewitz, Carl von, On War, translated by J.J. Graham, 1873, http://www.clausewitz.com/CWZHOME/VomKriege2/ONWARTOC2.HTML

DOD Dictionary of Military Terms, http://www.dtic.mil/doctrine/jel/doddict

Joint Publication 3-13, Information Operations, 13 Feb 2006, http://www.dtic.mil/doctrine/jel/new_pubs/jp3_13.pdf

Knight, R. and McIntyre, M. (2005). "An Operational Framework for Battle in Network Space" 10th International Command and Control Research and Technology Symposium the Future of Command and Control, McLean, VA.

National Military Strategy to Defend Cyberspace (draft, 2006)

Romanych, M. (2005) "Applying the Domains of Conflict to Information Operations", 10th International Command and Control Research and Technology Symposium the Future of Command and Command and Control, McLean, VA.

Waltz, E. (1998), Information Warfare Principles and Operations, Artech House, Aug 1998.

Wynne, M.W. and Moseley, T.M., SECAF/CSAF Letter to Airmen: Mission Statement, December 7, 2005, http://www.af.mil/library/viewpoints/secaf.asp?id=192

Similarity Analysis of Malicious Executables

Anthonius Sulaiman, Sandeep Mandada, Srinivas Mukkamala and Andrew Sung
New Mexico Tech, Socorro, NM, USA
ais@cs.nmt.edu
sandeep@cs.nmt.edu
srinivas@cs.nmt.edu
sung@cs.nmt.edu

Abstract: Malware seemingly leads the myriad of security threats on the Internet today. In this paper, we categorize and examine numerous malwares in order to identify the parts that cause malicious activities. Our paper relies on the factor that malware with similar symptoms share a common signature. We analyze and extract snippets of API sequences that appear frequently in a number of malicious samples. Our technique uses Windows API as its basis. Hence, its usage is currently limited to Windows systems. Experimental results from a large set of recent spy ware, adware, and viruses are presented.

Keywords: Malware, adware, spy ware, anti virus, similarity analysis

1. Introduction

The internet saw an increase in malicious codes that exposed confidential information as online criminals sought to make a profit [SC Magazine, 2005]. Lately the susceptibility of online financial data to such threats has been in the foray. Symantec's report on malware said that 74% of the 50 malicious code samples they received were comprised of malware related to gaining of the credit card and banking information [SC Magazine, 2005]. Online criminals today are motivated by profit, rather than curiosity or interest in showing off their technical skill as was the case a few years ago [SC Magazine, 2005]. Not to mention, the number of viruses has consecutively grown an average of 40 percent per year for the past two years - all this in step with the growth of spam [F-Secure, 2005].

The upsurge in confidential information threats is most probably due to the rapid growth of bots - compromised systems that are hijacked by attackers to send spam or denial-of-service attacks. Hence, in this paper unlike one of our previous work, NMTBRAVE, we have categorized the numerous malware as per their individual functionality into various categories which include the latest specific sets like denial of service, adware, data miners, financial data miners, toolbars etc. This has given us a better perspective on the working of each of the code from the same set, thereby allowing us to predict the codes of the next generation of mutants and variants of the current malware.

As the source codes of these malware are ubiquitous, it does not take too much programming skill to create the new variants for which exactly the script-kiddies are known for. The virus writing community is frequently visited by the law enforcement personnel. However, the law is still being considered and enacted [Gordon, 2000]. Currently, not all countries in the world are on the same level of enforcement.

With the current anti-virus static scanning techniques, it is not possible to detect these kinds of variants or mutants. In this paper, we propose one such technique in order to detect them.

Since Windows has been one of the most popular operating system to be targeted often by malware, our paper is concentrated towards the detection process of the malware for this operating system. The main idea is to create API snippets from the API sequences of the malware, making sure that they are not similar to the ones found in the normal programs else it might spring up false positives, which would identify normal programs as malicious.

API (Application Program Interface) is a set of routines or protocols to build applications. A set of API's is available for most operating systems. Since our focus is on Windows, the API's mentioned in this paper are Windows API's. These API's are provided in the form of dynamically linked libraries (DLL). One of the most commonly used libraries is kernel32.dll in the Windows system directory.

In the following sections, we discuss related works. Section 3 briefly discusses about the different categories into which the malicious programs have been classified into from our collections (which include the classic ones to quite a number of the recent ones). In section 4, we describe the experimental procedures. The results are provided in section 5. Finally, we present our conclusions and future work.

2. Related works

In this paper we focus on the sequence of the API snippets. Working on the API snippets gives us an insight on how to categorize malware according to their behavioral patterns. Some of our previous works also dealt with API's.

NMTSAVE built a foundation to our work by starting to use API sequences as its signature to detect malicious programs and their variants. This project uses only the full API sequences [Sung, 2004].

NMTMEDiC uses an entirely different method of detecting malicious programs and variants. This project relies on disassembled code of the malicious programs. Since it uses disassembly, it can be used in any operating system environment [Sung, 2005].

NMTBRAVE uses snippets too. The most important difference of this paper is that we categorized and studied the snippets based on more detailed behavioral patterns rather than taking all of our malware collection to find the snippets. The malware categorization is used to extend the results that we received from our analysis in NMTBRAVE, where the resulting snippets apply to the malware in our entire collection.

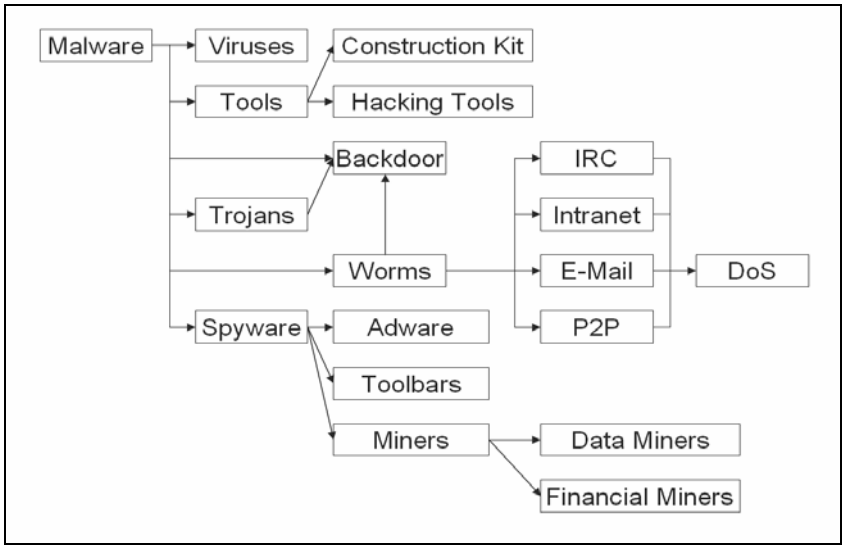


Figure 1: Malware taxonomy

3. Malware categories

Our analysis set consists of around 700 latest malware and their variants. From these samples, we have analyzed and characterized them according to their behavioral patterns. Figure 1 shows the full break down of our categories.

From the full set of malware in our collection, we divided them based on their functionalities. We collected all mass-mailing worms into one category and Trojans into another, etc. There may be overlaps between each category since a malicious program may be a Trojan and a mass-mailer at the same time. We also put the normal programs into one category so that we can be certain that the snippets we find do not appear in normal programs.

We first divided up malware as viruses, tools, backdoors, Trojans, worms, and spyware. This categorization is based on what the basic functionality of the malware:
Viruses: Viruses (vital information resource under siege) are software programs that interfere with computer operations by attaching themselves to files, usually executables, so that they are run whenever the hosts are run.

Tools: Tools can be further categorized into hacking tools and construction kit. Tools are programs that can be used to create malware. These programs by themselves are not malicious.

Hacking tools: These are programs used primarily to create, manipulate, modify, or analyze other programs, such as a compiler, an editor, or a cross-referencing program.

Construction kits: There are four kinds of kits in our database: C1, C2, C3, and C4. Each has different functionalities. They may not be malicious by themselves, but they can be used to create malicious programs.

Backdoor: A backdoor is a malicious code designed to connect to the authors/distributors website to download various files.

Trojan: It is a destructive program that masquerades as a benign application. Unlike viruses, Trojan horses do not replicate themselves but they can be just as destructive. One of the most insidious types of Trojan horse is a program that claims to rid your computer of viruses but instead introduces viruses onto your computer. Since Trojans can be used as a backdoor, we have some Trojans that have backdoor capability incorporated in them [Webopedia, 2005].

Worms: Worms are software programs deliberately designed to interfere with computer operations, records, corrupt, or delete data, or spread themselves to other computers and throughout the Internet, often slowing things down and causing other problems in the process [F-Secure, 2005]. Rather than injecting itself to another program like viruses, worms are stand-alone programs that are designed to propagate [Whalley, 2000]. Worms are further characterized based on their propagation method: IRC, intranet, e-mails, or P2P. We also included worms that propagate using two or more methods. We call this type as proxy worms.

IRC worms: IRC (Internet Relay Chat) is a special protocol, which was elaborated for communications between Internet users. Powerful and branched IRC clients' command system and scripts allow creating malware, which spread via IRC networks, that we call "IRC worms." The first report about IRC worms was registered at the end of 1997 from mIRC users [Kaspersky, 2005].

Mass-mailers: A mass-mailing worm is designed to trick recipients into opening a malicious program attached to the message. If a recipient opens the attachment, the worm sends itself to all of the contacts in the e-mail program's address book. An example of a mass-mailer is Bagle.A [F-Secure, 2005].

Intranet worms: An intranet worm is usually a standalone program that tries to copy itself to other computers connected to the same LAN (Local Area Network). Such worms travel from one computer to another using shares. A share is a media (hard drive for example) or part of it that can be accessed by everyone or only by users with specific access rights.

P2P: These worms spread via file-sharing networks. They either use peer-to-peer (P2P) file-sharing networks, such as Kaazaa or Morpheus, as a means to infect innocent victims, or they use P2P technology to construct worms that can communicate with one another [F-Secure, 2005a].

Denial of Service: A "denial-of-service" attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. An example of this is attempts to "flood" a network, thereby preventing legitimate network traffic. A lot of our worms are included in this category [CERT, 2005].

Spyware: Spyware are programs that spy on the users' behavior. A lot of spyware are piggy-backing other seemingly legitimate programs. Spyware can be further divided based on their appearance as adware, toolbars, and miners: both data miners and financial miners [I Am Not a Geek, 2005].

Adware: Adware is software that is installed on your computer to show you advertisements. Adware can slow your PC by using RAM and CPU cycles. Adware can also slow your Internet connection by using bandwidth to retrieve advertisements [Ars Technica, 2005].

Data miners: A data miner is a software application that monitors and/or analyzes the activities of a computer, and subsequently its user, for the purpose of collecting information that typically will be used for marketing purposes [Webopedia, 2005].

Financial data miners: A financial data miner is similar to the data miners, except that it is concerned with online financial data of the user.

Toolbars: Toolbars plug into internet browsers such as Internet Explorer to provide additional functionalities such as search forms or pop-up blockers. Google and Yahoo! toolbars are probably the most common legitimate examples. Malicious toolbars often attempt to emulate their functionality and look. Malicious

toolbars almost always include characteristics of the other malware categories, which are usually what gets them classified as malwares [Ars Technica, 2005].

We also included a special category called Polymorphic. This is the type of malware that can change its appearance code-wise so that anti-virus software cannot detect them. This rather interesting functionality can be applied to any of the malware categories we mentioned earlier. So, we included in this category any malicious codes from any categories that have this capability.

4. API snippets

An executable is usually built around API calls. A bag of API's only shows all API's that an executable may call during its lifetime. The order of API execution is of no importance in the bag. An API call in a bag is only shown once. However, a sequence shows a list of all the API calls being made in the order of which they are called during execution. An API call in a sequence can be made more than once. Hence, a sequence may show a particular API call over and over again. API snippets are parts of an API sequence that appear in an executable. What we are concerned here is not the full API sequence, but just the snippets.

To get the API snippets of one category, we first need to find shared characteristics between all the malicious codes of a particular malware category. These characteristics should be available throughout most of the malicious codes used in the experiments. The other condition to be satisfied is that they must not appear in normal programs.

Before we began, we developed a small tool to aid the process of listing the API sequence. This tool is called API Lister Tool. Given a PE (Portable Executable) binary, this tool returns a list of its API sequence. This tool can also return a bag of API's.

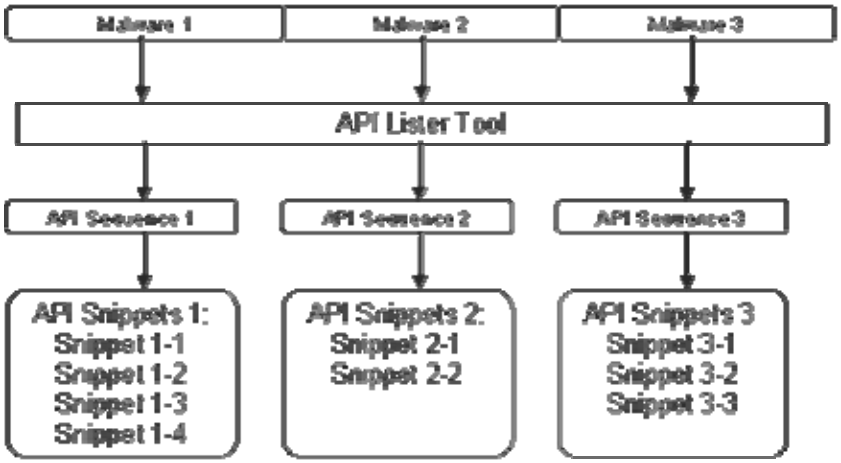


Figure 2: Creating snippets list for each malware in a category

PE is the standard Win32 executable file format. There is also NE (New Executable), a standard Win16 executable file format, but we are only focusing on PE file format.

A disassembler can also be used to provide an API bag. To get the API sequence, we must analyze the assembly code itself. We only use the disassembler method when the tool fails.

There is a third method to get the API's that a PE calls. By using a built-in command in Windows XP called TaskList, Windows XP will return all the API's called by a running executable. We avoid using this method as it requires us to run the executable. By running the executable, we are faced with the risk of infection. By running the executable, we are also deviating from our goal of static scanning.

For our first step, we looked at one category. Using the API Lister Tool, we can get the sequence of API's for each program under this category. Figure 2 shows the progression of creating snippets list for each malware. Each snippet contains the same number of API calls. The number of snippets each malware yields depends on the number of API's we use to compose the snippets.

During our experiments, we assembled a subsequence of two API's, three API's, up to ten API's. We found that two API's yields more false positives while ten API's yields almost no similarities. We decided to use only five API's in a subsequence and started gathering our snippets. From the full API sequence, we cut it up

and assemble much shorter subsequences consisting only of five API's each. For example, the malware Bagle has the following API sequence:

LStrLenA
LStrLenA
GlobalAlloc
StrStrIA
LStrLenA
LstrCpyA
LStrCatA
LStrCatA
GlobalFree
GlobalAlloc
GlobalAlloc
GlobalAlloc
GetModuleFileNameA
…and so on.

The subsequences of five API's or snippets derived from this sequence are:

LStrLenA+LStrLenA+GlobalAlloc+StrStrIA+LStrLenA
LStrLenA+GlobalAlloc+StrStrIA+LStrLenA+LStrCpyA
GlobalAlloc+StrStrIA+LStrLenA+LStrCpyA+LStrCatA
StrStrIA+LStrLenA+LStrCpyA+LStrCatA+LStrCatA
LStrLenA+LStrCpyA+LStrCatA+LStrCatA+GlobalFree
LstrCpyA+LStrCatA+LStrCatA+GlobalFree+GlobalAlloc
LStrCatA+LStrCatA+GlobalFree+GlobalAlloc+GlobalAlloc
LStrCatA+GlobalFree+GlobalAlloc+GlobalAlloc+GlobalAlloc
GlobalFree+GlobalAlloc+GlobalAlloc+GlobalAlloc+GetModuleFileNameA
and so on.

Table 1: Process to find "common" malicious snippets that are "uncommon" to normal programs

Process #2	Normal 1	Normal 2	Normal 3
Malware 1:			
Snippet 1-1	0 (Not Found)	0 (Not Found)	0 (Not Found)
Snippet 1-3	0 (Not Found)	1 (Found)	0 (Not Found)
Malware 2:			
Snippet 2-1	1 (Found)	0 (Not Found)	0 (Not Found)
Malware 3:			
Snippet 3-3	0 (Not Found)	0 (Not Found)	0 (Not Found)

For our example in Table 1, we compare the snippets that we got from normal programs. Now we look for zeroes across the table. We get snippets 1-1 and 3-3. We then concluded that these two snippets are the ones we call "common" malicious snippets that do not exist in normal programs. If a number one occurs in a snippet, it means that the snippet is a false positive.

We repeat this process for each malware category to reveal snippets for each. The same snippets may occur in different categories as malware in different categories may use the same functions at one time or another.

5. Analysis results

After performing the analysis on normal programs, we are presented with a few snippets for each category that match our goal of "common" to malware and not "common" to normal programs. In this section, we will present the results of our analysis.

Table 2 shows each of the malware being compared with the rest to find how similar they are to each other based on the API's used. The variants of each malware are grouped together, indicated in dark gray. Since the authors of these malware are different and they most likely do not work together when creating their malwares, we do not find striking similarities from one malware to another. Most matches found are most likely caused by their similarity in trying to spread by network or by the use of SMTP engine.

From table 2, we can conclude that most malware variants still stay true to their original code. For example, the four variants of Bagle show at least 80% similarity with each other. MyDoom also shows 90% similarity with its variants.

Table 2: Percentage of similarity based on the API's used in the malware

	Bagle.A	Bagle.I	Bagle.O	Bagle	Klez.E	Klez.H	MyDoom.A	MyDoom.E	MyDoom.F	Wozer.C	Wozer.E	ZMist.C1	ZMist.C2	ZMist.C3
Bagle.A	-	100	80.81	100	32.48	32.77	42.39	43.68	43.68	33.33	48.19	15.94	15.94	15.94
Bagle.I	100	-	80.81	100	32.48	32.77	42.39	43.68	43.68	33.33	48.19	15.94	15.94	15.94
Bagle.O	95.24	95.24	-	94.2	36.75	36.97	46.74	48.28	48.28	33.33	50.6	21.74	21.74	21.74
Bagle	82.14	82.14	65.66	-	23.08	23.53	36.96	37.93	37.93	27.78	42.17	15.94	15.94	15.94
Klez.E	45.24	45.24	43.43	39.13	-	98.32	51.09	54.02	54.02	51.85	53.01	12.32	12.32	12.32
Klez.H	46.43	46.43	44.44	40.58	100	-	53.26	55.17	55.17	53.7	54.22	12.32	12.32	12.32
MyDoom.A	46.43	46.43	43.43	49.28	40.17	41.18	-	97.7	97.7	31.48	43.37	14.49	14.49	14.49
MyDoom.E	45.24	45.24	42.42	47.83	40.17	40.34	92.39	-	100	33.33	44.58	15.22	15.22	15.22
MyDoom.F	45.24	45.24	42.42	47.83	40.17	40.34	92.39	100	-	33.33	44.58	15.22	15.22	15.22
Wozer.C	20.24	20.24	17.17	20.29	23.08	23.53	18.48	20.69	20.69	-	60.24	12.32	12.32	12.32
Wozer.E	38.1	38.1	34.34	42.03	29.91	30.25	30.43	32.18	32.18	75.93	-	11.59	11.59	11.59
ZMist.C1	26.19	26.19	30.3	31.88	14.53	14.29	21.74	24.14	24.14	33.33	25.3	-	100	100
ZMist.C2	26.19	26.19	30.3	31.88	14.53	14.29	21.74	24.14	24.14	33.33	25.3	100	-	100
ZMist.C3	26.19	26.19	30.3	31.88	14.53	14.29	21.74	24.14	24.14	33.33	25.3	100	100	-

Another important observation is on the variants of ZMist. ZMist can be considered one of the polymorphic malwares and yet, they still use the same API's. It indicates that ZMist variants we have in our collection are very likely polymorphic copies of one another. While they may look different, they have the same functionalities.

The non-polymorphic malwares have differences in each variant because their authors tried to fool the anti-virus software into believing they are not the same by changing their source codes and recompiling their malwares. A variant may be different from its original by attacking a different port, attacking a different website, etc. They are most likely still exploiting the same vulnerabilities.

A lot of the variants are created by other people too. The script-kiddies are using freely available scripts to alter the appearance of the malwares. Essentially, this type of variants is like a polymorphic malware without the polymorphic codes embedded into the malware (in other words, a manually-morphed variant). We have already shown that polymorphic malwares can be detected, so the variants from the script-kiddies can be detected as well.

During the experiments, we ran into several problems. One problem we have for the Windows socket library is that our program does not have detailed information regarding which API's are being used. To remedy this problem, we also disassembled the worms to analyze its API usage. Unfortunately, even a disassembler cannot recognize the socket-related API's.

In addition to malwares, we also have several spywares on our hands. We also did a similar experiment with the spywares. The numbers representing percentages of similarity of a particular spyware when compared to the signatures provided to our tool are presented in table 3.

Table 3: Analysis of similarity matches for spyware

	Signatures used									
	180SA.Sauhook	Bargains	Begin2Search	Claria.A.DbAu	Ebates	Gator.CMESys	ShopAtHome	Sidefind	WebRebates	WhenU.Sync
Bargains	6.5	100	5.41	4.3	7.9	5.5	79.2	8.3	63.4	5.9
Claria.A.DbAu	63.9	16.8	46.2	100	51.3	80.3	18.9	33.3	16.7	41.7
Gator.CMESys	60	12.5	34.2	36.8	40.9	100	13.6	27.7	13.1	34.6
Gator.GMT	70.8	30.6	67.9	99.2	63.5	84.9	26.6	72.2	29.3	55.2
Save	66.9	16.8	48.4	36.4	42.6	67.3	21.8	22.2	22.7	69.4
ShopAtHome	6.0	82.5	5.4	4.8	7.7	5.0	100	8.3	86.2	7.0
WebRebates	6.5	66.9	5.7	4.9	8.1	4.8	82.2	5.5	100	6.9

After analyzing the result for matches around and above 65%, there are several interesting points we can make from this experiment:

1. Gator e-Wallet (Gator.CMESys and Gator.GMT) and Claria DashBar (Claria.A.DbAu) have high similarity to each other because they come from the same company. Gator Corporation changed its name to Claria in October 2003. The results are highlighted in red. Hence the claim here is that in the future any similar product or versions made by this company is bound to be detected by NMTMEDIC.
2. Bargains, ShopAtHome, and WebRebates have high similarity matches because they are categorized as data mining programs. All these programs watch the user as and when they are surfing the internet and report the findings to the author. The results are highlighted in green.
3. Save and WhenU.Sync are also from the same company. The result is highlighted in blue.
4. Since Gator and Claria function libraries are so large, they are recognized using the signatures of other malicious programs. The results are highlighted in gray.
5. 180SA also recognizes a lot of Gator/Claria functionalities, indicating the possibility of 180SA being a part of Gator/Claria. The results are highlighted in yellow.

We ran our tool equipped with malware and spyware signatures and scanned normal programs. We set the virus threshold to 65%, which means a particular code must be 65% similar to any of the signatures to be considered a variant of a malware or a spyware. In these experiments, we have found no false positives so far, because the chance of malicious code islands to be present in a normal program is very slim. Although, there is one program called PuTTY (a telnet program) that came very close: 60% similar to 180SA and 58% similar to Klez. This is caused by the amount of network functionality in PuTTY.

Interestingly, the Polymorphic category reveals that most of the snippets are not "common" in the normal programs. In a sense, we can use any of the snippets found in this category to identify malware with polymorphic behavior.

Unfortunately, the remaining categories are inconclusive because of a few reasons. Some categories lack the number of samples necessary to ensure that we can conclude the snippets work. Some other categories simply do not have “common” malicious functionality that does not trigger false positive when compared to normal programs.

6. Conclusion

We present an extended behavior-based analysis for detecting malicious codes that may escape detection from anti-virus software. This method can also be used to identify the probability of an unknown program to be part of a malicious category. Since this method currently can only be used in Windows, we focused on getting the API snippets of Windows programs. The main use of this analysis is to warn users of suspicious executables that may be malicious. Based on the information a user is given on the likelihood of a program is malicious, he or she then can decide whether to continue executing that suspected program.

Our technique has a tremendous potential and can be used as a foundation to a more robust detection method based on its behavioral patterns. We do not rely on placement of signature code like most anti-virus do, so we do not have to worry about script-kiddies changing the parts that anti-virus software look for as long as the functionality of the executable does not change. The only information that we use for detection is an API sequence of the program in question. The signature for detection is the API snippets. The optimization of the snippets could also be achieved since we heuristically approached at the idea of picking up the snippets of length 5 or 6. With a more mathematical approach, it is possible to find snippets with an optimal length.

Acknowledgments

Support for this research received from ICASA (Institute for Complex Additive Systems Analysis, a division of New Mexico Tech) and a DoD IASP grant is gratefully acknowledged. The authors would also like to acknowledge Mr. Kartik for his assistance.

References

Ars Technica (2005) *Malware: What It Is and How to Prevent It*, http://arstechnica.com.

SC Magazine (2005) *Malware Targeting Confidential Data on the Rise*, http://www.scmagazine.com.

F-Secure (2005) *Data Security Summary*, http://www.f-secure.com.

Gordon, S. (2000) *Virus Writers: The End of Innocence?*, Virus Bulletin.

Sung, A., Xu, J., Chavez, P., Mukkamala, S. (2004) *Static Analyzer for Vicious Executables (SAVE)*, Proceedings of 20th Annual Computer Security Applications Conference (ACSAC), IEEE Computer Society Press, ISBN 0-7695-2252-1, pp. 326-334.

Sung, A., Sulaiman, A., Ramamoorthy, K., Mukkamala, S. (2005) *Malware Examiner using Disassembled Code (MEDiC)*, Proceedings of 6th IEEE Information Assurance Workshop, IEEE Computer Society Press, pp. 428-430.

Webopedia (2005) *Trojan Horse*, http://www.webopedia.com.

F-Secure (2005) F-Secure Virus Description: Worm, http://www.f-secure.com/v-descs.

Whalley, L., (2000) *An Environment for Controlled Worm Replication and Analysis (Internet-ina-Box*, IBM Antivirus Research, Virus Bulletin.

Kaspersky (2005) *IRC Worms*, http://www.avp.ch.

F-Secure (2005) *F-Secure Virus Description: P2P Worm*, http://www.f-secure.com/v-descs.

CERT (2005) *Denial of Service Attacks*, http://www.cert.org.

I Am Not a Geek (2005) *Spyware Removal Guides*, http://www.iamnotageek.com.

Using Deception for Assuring Security

Stilianos Vidalis, Eric Llewellyn and Christopher Tubb
University of Wales, Newport, UK
stilianos.vidalis@newport.ac.uk
eric.llewellyn@newport.ac.uk
christopher.tubb@newport.ac.uk

Abstract: For each layer of information security there are a number of techniques and tools that can be used to ensure information superiority. Indeed some experts would argue that you can not have the former without the later. In today's technological & interconnected world though, information superiority is very hard to achieve and almost impossible to maintain. This paper will argue that the art of deception is a reliable and effective technique that can ensure and maintain the security of an infrastructure. The paper will conclude by presenting a technical solution of the above statement.

Keywords: Information security, information superiority, deception, virtual honeypots

1. Introduction

This paper will attempt to associate the concepts of information superiority and information security while further developing the argument presented by Cohen (Cohen 1998) by saying that deception should be the main technique to use for assuring the main goals of security.

Information Security (InfoSec) as a concept is an utopia. According to Gollmann (Gollmann 1999) InfoSec is concerned with the integrity, availability, confidentiality and non-repudiation of assets (for definitions see (Pfleeger 1997). Based on (Kabay 1996), (Carroll 1996), (Kove, Seger et al. 1995), (Neumann 1995), (Nosworthy 2000), (Summers 1977), (Casey 2000), the following asset categories have been identified: Software, Communications, Data, Hardware, Physical & Environmental, Personnel, Administrative & Organizational. InfoSec has been defined by Waltz (Waltz 1998) as: *"…the activities to protect hardware, software and intangible information at the hardware and software levels"*. Figure 1, based on a figure in (Carroll 1996), illustrates the different layers of InfoSec.

Figure 1: Information security layers (source (Carroll 1996))

People debate about InfoSec, and indeed there is a lot of contradiction in the field (Anderson 2003), but they (the people) are sure neither about their words (blame the ever-changing technology), nor about the solutions they are proposing (blame the risk that is involved in the process). Surprisingly enough though, companies are still spending money for having not to spend money, and for justifying that expenditure they pay even more money! This is not right, especially when we take under consideration the accepted facts of security being a continuous process (Carroll 1996); (Nosworthy 2000); (Wright 1999); (Smith 1993); (Barber 2001) and of security being a responsibility of all the stakeholders (defined in (Blyth and Kovacich 2001)) no matter their importance and/or strength (Wood 1997).

Over the years the academia and the industry has approached the concept of information security in a number of ways (Eloff and Solms 2000), (Computer_Fraud_Security 2002), (Forte 2000), (Nosworthy 2000), (Kabay 1996), (Solms 2001), (Jones 2002), (Finne 1998), (Vidalis, Pilgermann et al. 2006). The question we have to ask (being good hackers…) is why do we observe this drop? Is it because 'Threat Agents' (Vidalis 2006) have moved on and changed their targets, effectively misleading the InfoSec professionals as to what they should be looking for? Or is it because threat agents now recognise they can achieve more by not

attacking the availability of information assets, effectively deceiving the InfoSec Officers and the rest of the world altogether?

The above questions initiated the research presented in this paper, and the idea of the virtual honeypots, a technology that make use of the art of deception in order to ensure information security in a computing infrastructure. A technology that aims to move the "point of reference" in the field of Information Assurance from risk assessment (Tregear 2001), (Bayne 2002), (Baskerville 2003) to threat assessment (Jones 2002), (Vidalis 2001). The technology makes use of specialised hardware and software in order to simulate computers at a software level, between the 3rd and 7th leyer of the OSI model.

Deception is better realised under the context of Information Warfare and Information Operations. According to (Waltz 1998) the two concepts have been defined as:

> *"IO are the continuous military operations within the military information environment that enable, enhance, and protect the friendly force's ability to collect, process and act on information to achieve an advantage across the full range of military operations."*
>
> *"IW includes actions taken to preserve the integrity of one's own information system from exploitation, corruption, or disruption, while at the same time exploiting, corrupting or destroying an adversary's information system and the process achieving an information advantage in the application force."*

Naturally, IW & IO have gone through a number of evolution changes. Homer's Iliad saga (Homer, Kirk et al. 2004) is an "early" example and the deception around the various psychological operations involved in the "war against terror" is a modern example. The main reason for these changes is information and the way it is being handled and transmitted over the various mediums.

2. The role of information

What is information? A very short and straight forward question that hasn't got a straight forward answer? Most people use the word without really knowing what it means. When the author was asked to define 'information' he replied using the word 'knowledge'. Indeed, information can be knowledge, but cannot be defined as that. For defining information we will go back to 1995 when Menou (Menou 1995) wrote that

> *"…information encapsulates a wide range of concepts and phenomena…". "They relate to both processes and material states which are closely inter-related…"*

According to the same author, information can be:

- "A product, which encompasses information as thing, as object, as resource, as commodity,
- what is carried in a channel, including the channel itself,
- The contents."

Based on Waltz (Waltz 1998) information has three abstractions: data, information and knowledge. Data can be individual observations and low level primitive messages. Threat agents make use of various techniques and technologies (see (Hoath and Mulhall 1998; Barber 2001; Scambray, McClure et al. 2001)) to collect data. The next step is to sort, classify or index them into organised sets. These sets are referred to as information. An information gathering methodology that is very popular amongst the 'softer' threat agents is presented by Scambray (Scambray, McClure et al. 2001). The goal is to put the data elements in relational context for further analysis, which will result in understanding the information. Quoting Waltz (Waltz 1998):

> *"Understanding of information provides a degree of comprehension of both the static and dynamic relationships of the objects of data and the ability to model structure and past (and future) behaviour of those objects. Knowledge includes both static content and dynamic processes."*

Once a threat agent has acquired this knowledge then he is in the position of being able to launch an active attack with a high level of impact (see (R. Kalakota and Whinston 1997); (Daughtrey 2001); (G. Johnson and Scholes 1999)). Knowledge is the main factor that distinguishes internal from external threat agents. Internal threat agents otherwise known as 'insiders' have been defined by Vidalis (Vidalis 2006) as:

> *"An **insider** to an organization is a threat agent who is directly or indirectly employed by that organization, and has access to the system or sensitive information not otherwise disclosed to him and to the general public."*

An insider prediction tool was presented by (Maglaras and Furnell 2002) where it is quoted that: *"…49% of the respondents faced IT security incidents due to the actions of legitimate users"*. Statistics on 'insiders' in the UK can be seen in the UKs National High-Tech Crime Unit survey (NHTCU 2004) and in the US on the FBI survey (Richardson 2003).

Once a threat agent acquires the status of the 'insider' then he is able to understand the 'whys' and the 'hows' of the system, which will result in him acquiring superiority over the defenders of that system. Information superiority is the aim of Information Operations and has been defined in Waltz (Waltz 1998) as:

> *"Information superiority is the capability to collect, process, and disseminate an uninterrupted flow of information while exploiting or denying an adversary's ability to do the same."*

Just by having the right information is not going to grant information superiority. In our modern interconnected world the issue of time has great importance. Today IO officers say that having the right information on the right time can possibly grant information superiority. Under the same context, providing the adversary with the right information in the wrong time, or with the wrong information in the right time, effectively undermining his decision-making processes, can possibly grant information superiority. Fred Cohen (Cohen 1998) has analysed extensively the role of deception in information protection.

3. The role of deception

Deception has been defined in the consise Oxford dictionary (Sykes 1981) as: *"misrepresentation, persuade of what is false, mislead purposely"*. Deception is an ancient art, and an art it is indeed, as noted in many sources, one of them being Dearth (Campen and Dearth 1998). It goes back to the 10th century BC when King Solomon said: *"A wise man has great power, and a man of knowledge increases strength; for waging war you need guidance, and for victory many advisers."* The more information one has the better he will be able to assess a situation in taking advantage of certain variables for achieving information superiority. When one talks about Information Operations and deception he cannot but quote the famous Chinese general Sun Tzu from the 6th century BC:

> *"In respect of military method, we have, firstly, measurement; secondly, estimation of quantity; thirdly, calculation; fourthly, balancing of chances; fifthly, victory."*
>
> *"Thus, what enables the wise sovereign and the good general to strike and conquer, and achieve things beyond the reach of ordinary men, is foreknowledge."*
>
> *"All warfare is based on deception [of the enemy]."*

The following table was taken from (Waltz 1998).

Table 1: Taxonomy of information operations

IW Model Layer		Function	NETWAR
Offence	Perceptual	Manage perception, Disrupt decision processes	PSYOPS, Deception
	Information	Dominate information infrastructure	NETOPS
	Physical	Break things…, Incapacitate/kill people	Physical destruction
Defence	Perceptual	Protect perceptions and decision-making processes	Intelligence, Counterintelligence
	Information	Protect information infrastructure	INFOSEC
	Physical	Protect operations, protect people	OPSEC

NETWAR has been defined by (Waltz 1998) as:

> *"The information-related conflict waged against nation states or societies at the highest level, with the objective of disrupting, damaging, or modifying what the target population knows about itself or the world around it."*

Agreeing to Zhou (Zhou, Burgoon et al. 2004): "*deception in human communication occurs when information senders attempt to create a false impression in receivers.*" Through deception we can manage our adversary's perception and disrupt his decision-making processes. These processes feed into his (the adversary's) defensive INFOSEC processes which when disrupted will allow the success of our offensive NETOPS (Waltz 1998) that will ensure out information superiority. Knowledge in military terms is Intelligence. We have already defined knowledge and how it relates to deception. If we are able to disrupt the Intelligence and Counterintelligence operations of our adversary then we can achieve information superiority. The common operation is deception and agreeing with Cohen (Cohen 1998) it is believed to be the future of IO and Information Security.

Deception allows subduing the enemy without fighting. Fighting cost resources and resources cost money, which is a very scarce resource… Security though should not be an expensive commodity. The weakest link of a game should be able to afford the same level of security the strongest link have. It is accepted (see

(Vidalis 2004)) that the weakest link does not get thrown out of the game, it destroys the game altogether. Furthermore, fighting is usually a reactive action. It has been proven though (see (Vidalis 2004)) that in the area of security reacting is expensive, instead we want to be proactive and somehow prevent attacks from happening or minimising/nullifying the impact of the threats.

The purpose of deception is to surprise the adversary. If the threat agent is in a state of surprise the outcome can be twofold: either the defenders have time to react and deploy the necessary countermeasures (or finely tune the existing ones), or the threat agent will call off the attack and return to the information gathering process in order to re-examine his plan of action.

Agreeing with Mitnick (Mitnick and Simon 2002), technology has allowed for an increased capability for information gathering, but perceptions and the nature of decision-making have one common vulnerability: the human factor. Humans sit behind monitors, typing and/or communicating commands. Humans are in charge of automated procedures and can shut them down if they perceive that something is wrong and that the computer reactions do not make sense.

The responsibilities of the network administrator is summarised in the following list:

- Design a network which is logical & efficient
- Deploy large numbers of machines which can be easily upgraded later
- Decide what services are needed
- Plan and implement adequate security
- Provide a comfortable environment for users and keep them happy…
- Develop ways of fixing errors and problems which occur
- Keep track of, and understand how to use, technology

By designing a logical network though, the administrator makes the life of the threat agents easier, as they can follow the same logic and enumerate the infrastructure. Deception can be used to hide the real computers amongst the false. By having easily upgradeable computers the administrator possibly introduces a critical threat against his infrastructure. Should the upgrade procedures get compromised then threat agents will be able to launch catastrophic active attacks. Again deception can be used to masquerade the procedures and/or produce confusion about what is real. Some would argue that you can never have enough security, a statement that has been argued from the threat and risk assessment professionals. Agreeing with Mitnick (Mitnick and Simon 2002), the network administrator will always have to fear the users of his system. System users are probably the bigger vulnerability of that system as they are susceptible to social engineering attacks. If we consider that deceiving our own users is acceptable then deception can offer a solution to the social engineering vulnerability. User deception and the ethical issues involved are examined in another paper developed by the author. To summarise, deception can be used in two ways for ensuring security in a computing infrastructure:

- Simulating: showing the false, drawing attention away from the real
- Dissimulating: hiding the real, producing confusion about what is real

Deception must employ people from unconventional backgrounds, people who understand the world. It is a psychological exercise, not a technological one. The following section presents a technology that makes use of psychological profiles in order to affect the decision-making process of the threat agents, effectively deceiving them as of the nature and layout of the infrastructure. Please note, that the presented solution is not just another type of honeypot; rather, it is a system which is making use of certain honeypot principles.

4. Virtual honeypots

According to (Sadasivam, Samudrala et al. 2005):

> ***Honeypots*** *are closely monitored decoys that are employed in a network to study the trail of hackers and to alert network administrators of a possible intrusion.*

Another definition comes from Provos (Provos 2004):

> *A* ***honeypot*** *is a closely monitored computing resource that we intend to be probed, attacked, or compromised.*

By combining the two definitions we can identify a paradox. We try to improve security by introducing a vulnerability to our infrastructure. Furthermore, in the case of a highly capable threat agent compromising the honeypot and launching a successful active attack, the company owning the honeypot would have to face a series of legal implications. The drawback list of the real honeypot includes its static nature and its inability to dynamically address change. If we add to the list the need of a close and strict management and the resources needed to setup and maintain the honeypot, we can start understanding why the technology never reached its expected success.

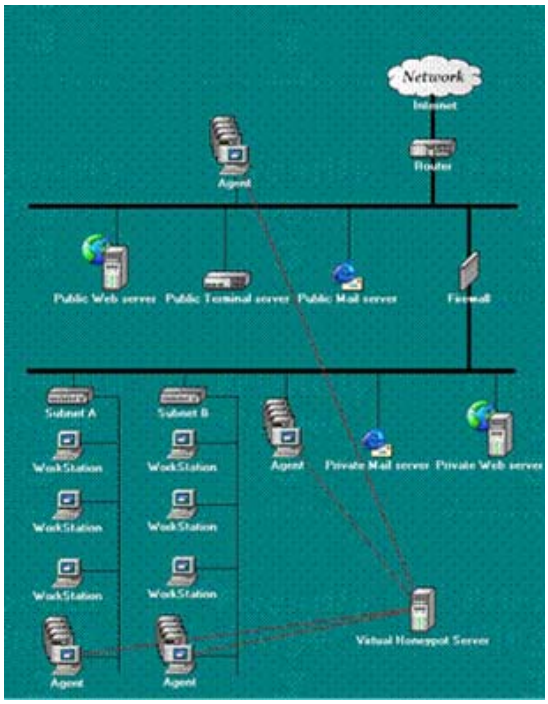


Figure 2: Example infrastructure

The Virtual Honeypot technology is designed to address the above drawbacks by interoperating with G4DS (Vidalis, Pilgermann et al. 2006). It takes the concept of honeyd (Provos 2004) and exploits it further by truly applying to the technology the art of deception. Figure 2 presents an infrastructure making use of the technology.

The infrastructure has the expected layout of an SME offering a number of online services. The first router can be combined with a simple firewall offering a first layer of security protecting from nothing but the obvious attacks, the public servers of the enterprise. A bastion-host sits between the DMZ and the inner-network screening every single Ethernet packet. The inner-network consists of a number of subnets and the private servers of the enterprise. The network is 100Tx fully switched with Cat5 cables.

The Virtual Honeypot technology makes use of host machines called “Agents”. The Agents are real machines, part of the primary infrastructure that run all the virtual machines called “Virtual Honeypots”. The Agents never respond to network probes with their real credentials. The primary infrastructure does not even know their existence. Instead, each Agent runs in promiscuous mode, having a list of virtual MACs, IP and TCP addresses. If the Agent traps a packet intended for a virtual NIC that it is responsible for, it replies with the necessary “virtual” data. Agents cannot operate across switching devices, which means that an agent can only “alter” the subnet that it is part of.

To address issues of integrity and confidentiality of communications, all the agents are connected to a central management machine called the “Virtual Honeypot Server”, via dedicated lines which we will call “secondary infrastructure”. The use of the secondary infrastructure allows the management of the Agents without revealing the operations to the rest of the world. Indeed, the only persons, in any given time, that know the real layout of the infrastructure should be the two information security officers that sit in front of the Virtual Honeypots’ dual monitors. Furthermore, threat agents are not able to identify the Agents and the Server by monitoring the traffic in the primary infrastructure, and the primary infrastructure is not getting encumbered with the management traffic.

The Virtual Honeypot Server is part of both infrastructures, and by combining data collected from both, it constructs the infrastructure that is presented to the rest of the world. The server does not offer any service through its primary NIC. Periodically it interrogates the primary infrastructure in order to update the layout. Similarly to the Agents, the Server never replies to any probes with its real credentials. The difference is that while an Agent could reply with a set of "virtual" credentials, the Server has none. A threat agent could potentially identify this black hole in the network and use it to identify the Server, hence considerations are being made for masquerading the Server as another server (potentially a printer server).

The front-end application runs on top of the management engine in the Server and is very similar to the Windows Explorer application. With the use of the mouse and keyboard the user is able to dynamically add or delete virtual machines in the same way that he can use the Explorer to add and delete files & folders. The user then has a choice of selecting the type of the hardware & software type of the virtual machine, the type of the virtual NIC and an associated directory tree.

5. Technology cost evaluation

This paper argues that the art of deception is a reliable and cost effective technique that can ensure and maintain the security of an infrastructure. An obvious question to ask is how a technology that introduces extra hardware and software on top of the existing one can be cost effective. If we think of the cost involved though, depending on the size of the organisation, and the number of different subnets, we will understand that really the cost is minimal. A server machine, running an open source OS, and freeware applications, communicating with a limited number of clients, running either open source or proprietary OSs, cost only a fraction of the potential losses.

Comparing the presented technology with existing defence mechanisms we can reach the following conclusion. We are still observing great financial losses every year due to electronic crime, and although the figures are not as high a they used to be at the beginning of the decade, they are alarming. Hence, we can safely say that current defence technologies do not offer the best solution. During the Meteore2000 project (www.meteore2000.net) it was realised that reacting to threats is not cost effective. Instead there is a need to be proactive, and guide the threat agents to behave in a specific way. Furthermore, it has been accepted by the scientific community that the risk is managed by the threat agents, so it is logical to say that only if we can in our turn manage the threat agents we can truly manage our risks.

As the technology is not yet fully developed, there are no hard figures to be presented in this paper. After the design and development phases, there will be an operational experiment where the technology will be properly evaluated and compared against other current defence mechanisms.

6. Conclusion

The aim of Virtual Honeypots is the near real-time threat agent deception, so the ten principles of strategic deception (Campen and Dearth 1998) were also considered to be the objectives of the technology:

1. All of the infrastructure should be dedicated to the execution of the deception
2. Intelligence will be brought fully to the picture via interoperating with G4DS (Vidalis, Pilgermann et al. 2006)
3. Intelligence will be cross-checked and combined in G4DS (Vidalis, Pilgermann et al. 2006)
4. Secrecy will be enforced via the use of anonymity, confidentiality and authentication procedures of G4DS (Vidalis, Pilgermann et al. 2006) and by the design of the Virtual Honeypot Technology
5. The deception plan will be designed at the top levels of the enterprise
6. Full implementation & consistency of all elements of deception via the use of a central management plan throughout the supply chain of the enterprises involved in the plan.
7. Virtual honeypots will introduce no hindrances to existing security policies & countermeasures
8. Deception via the Virtual Honeypot technology will be a continuous process
9. Long term aim. Individual short term security aims are fulfilled by other countermeasures
10. The technology will stimulate the threat agent imagination as virtual honeypots will be able to differentiate between each threat agent category.

The nature of deception was examined in previous sections. It requires the co-operation of all the involved actors, be they human, hardware or software. In order to be successful the technology will have to be used against the users of the infrastructure that it is deployed. We cannot let the employees of a company to acquire knowledge or information that will be able to jeopardise the deception plan of that company. This action will minimise the effects of any social engineering attacks, as well as help identifying internal threat agents. Because virtual honeypots will be offering no real services, any attempt to contact one will be suspicious and declare an act of an information gathering process.

Virtual Honeypots are designed to interoperate with G4DS (Vidalis, Pilgermann et al. 2006). G4DS is a new generation IDS system that makes use of the GRID technology and brings together organisations by forming virtual communities. The system allows the identification and monitoring of threat agents and their actions while attacking the infrastructure protected by G4DS. The threat knowledge gathered by G4DS can be used as intelligence in the Virtual Honeypot system. The intelligence can be considered secure and accurate without any need for further analysis as G4DS has ensured its integrity and confidentiality.

References

Anderson, J. M. (2003). "Why we need a new definition of information security." Computers & Security **22**(4): 308-314.

Barber, R. (2001). "Hacking Techniques." Computer Fraud & Security **2001**(3): 9-12.

Baskerville, R. (2003). Four Complexions of Information Security Risk Assessment. 2nd European Conference on Information Warfare & Security, University of Reading, UK, MCIL.

Bayne, J. (2002). An Overview of Threat and Risk Assessment, SANS Institution. **2002**.

Blyth, A. J. C. and L. Kovacich (2001). Information Assurance: Computer Communications & Networks. UK, Springer-Verley, 185233326X.

Campen, A. D. and D. H. Dearth (1998). Cyberwar 2.0: myths, mysteries and reality. Fairfax, Virginia, AFCEA International Press, 0-916159-27-2.

Carroll, J. M. (1996). Computer Security, Butterworth-Heinemann

Casey, E. (2000). Digital Evidence and Computer Crime, Academic Press

Cohen, F. (1998). "A note on the role of deception in information protection." Computers & Security **17**(6): 483-506.

Computer_Fraud_Security (2002). "Information Security: the great balancing act." Computer Fraud & Security **2002**(2): 12-14.

Daughtrey, T. (2001). Costs of trust for e-business, Quality Progress

Eloff, M. M. and S. H. v. Solms (2000). "Information Security Management: a hierarchical framework for various approaches." Computers & Security **19**(3): 243-256.

Finne, T. (1998). "A Conceptual Framework for Information Security Management." Computers & Security **17**(4): 303-307.

Forte, D. (2000). "Information Security Assessment: Procedures & Methodology." Computer Fraud & Security **2000**(8): 9-12.

G. Johnson and K. Scholes (1999). Exploring Corporate Strategy, Prentice Hall Europe

Gollmann, D. (1999). Computer Security. Cambridge, UK, John Wiley & Sons, 0-471-97844-2.

Hoath, P. and T. Mulhall (1998). "Hacking: motivation & deterence, part 1." Computer Fraud & Security **1998**(4): 16-19.

Homer, G. S. Kirk, et al. (2004). The Iliad. UK, Oxford University Press, 0-1928-3405-3.

Jones, A. (2002). "Protecting the Critical National Infrastructure - Developing a Method for the Measurement of Threat Agents in an Information Environment." Information Security Technical Report **7**(2): 22-36.

Kabay, M. E. (1996). Enterprise Security: Protecting Information Assets, McGraw-Hill

Kove, D., K. Seger, et al. (1995). Computer Crime: A crimefighter's handbook, O'Reilly & Associates

Maglaras, G. B. and S. M. Furnell (2002). "Insider Threat prediction Tool: evaluating the probability of IT misuse." Computers & Security **21**(1): 62-73.

Menou, M. J. (1995). "The impact of information II: Concepts of information and its value." Information Processing & Management **31**(4): 479-490.

Mitnick, K. D. and W. L. Simon (2002). The Art of Deception. Indianapolis, USA, Wiley Publishing, 0-471-23712-4.

Neumann, P. G. (1995). Computer Related risks, Addison-Wesley

NHTCU (2004). NHTCU web site, NHTCU. **2004**.

Nosworthy, J. D. (2000). "Implementing Information Security in the 21st Century - Do you have the balancing actors?" Computers & Security **19**(4): 337 - 347.

Nosworthy, J. D. (2000). "A Practical Risk Analysis Approach: managing BCM risk." Computers & Security **19**(7): 596-614.
Nosworthy, J. D. (2000). "A practical risk analysis approach`." Computers & security **19**: 596-614.
Pfleeger, C. P. (1997). Security in Computing, Prentice Hall Int.
Provos, N. (2004). A Virtual Honeypot framework. 13th Usenix Security Symposium, San Diego, USA.
R. Kalakota and A. B. Whinston (1997). Electronic Commerce: a manager's guide, Addison Wesley, 0201880679.
Richardson, R. (2003). "2003 CSI/FBI Computer Crime & Security Servey." Computer Security Journal **19**(2): 21-40.
Sadasivam, K., B. Samudrala, et al. (2005). "Design of network security projects uding honeypots." Journal of Computing Sciences in Colleges **20**(4): 282-293.
Scambray, J., S. McClure, et al. (2001). Hacking Exposed: network security secrets & solutions, Osborn/McGraw Hill
Smith, M. (1993). Commonsence Computer Security: your practical guide to information protection, McGraw-Hill
Solms, B. v. (2001). "Information Security - A Multidimensional Discipline." Computers & Security **20**(6): 504-508.
Summers, R. C. (1977). Secure Computing: threats & safeguards, McGraw-Hill`
Sykes, J. B. (1981). The Concise Oxford Dictionary, Clarendon Press
Tregear, J. (2001). "Risk Assessment." Information Security Technical Report **6**(3): 19-27.
Vidalis, S. (2001). TAMMPS: a threat assessment model for micro-payment systems. School of Computing. Pontypridd, University of Glmaorgan**:** 1-152.
Vidalis, S. (2004). Threat Assessment of Micro-payment Systems. School of Computing. Pontypridd, University of Glamorgan.
Vidalis, S. (2006). "Threat Agents: what InfoSec officers need to know." Mediterranean Journal of Computers and Security.
Vidalis, S., M. Pilgermann, et al. (2006). "Security in Heterogeneous Large Scale Environments." International Journal of Innovative Computing **2**(2).
Waltz, E. (1998). Information Warfare. Norwood, USA, Artech House, 0-89006-511-X.
Wood, C. C. (1997). "Policies alone do not constiture a sufficient awareness effort." Computer Fraud & Security **1997**(12): 14-19.
Wright, M. (1999). "Third Generation Risk Management Practices." Computer Fraud & Security **1999**(2): 9-12.
Zhou, L., J. K. Burgoon, et al. (2004). "A comparison of classification methods for predicting deception in computer-mediated communication." Journal of Management Information Systems **20**(4): 139-165.

The Design Space of Metamorphic Malware

Andrew Walenstein[1], Rachit Mathur[2], Mohamed Chouchane[1], and Arun Lakhotia[1]
[1]University of Louisiana at Lafayette, LA, U.S.A.
[2]McAfee Avert Labs, Beaverton, OR, U.S.A.
walenste@ieee.org
rachit_mathur@avertlabs.com
mohamed@louisiana.edu
arun@louisiana.edu

Abstract: A design space is presented for metamorphic malware. Metamorphic malware is the class of malicious self-replicating programs that are able to transform their own code when replicating. The raison d'etre for metamorphism is to evade recognition by malware scanners; the transformations are meant to defeat analysis and decrease the number of constant patterns that may be used for recognition. Unlike prior treatments, the design space is organized according to the malware author's goals, options, and implications of design choice. The advantage of this design space structure is that it highlights forces acting on the malware author, which should help predict future developments in metamorphic engines and thus enable a proactive defence response from the community. In addition, the analysis provides effective nomenclature for classifying and comparing malware and scanners.

Keywords: Metamorphic malware, virus scanner.

1. Introduction

Metamorphism is the ability of malware to transform its code. This ability was first introduced in viruses and was later used by worms, Trojans, and other malware. There now exist several metamorphic engines—programs that implement only the logic for transforming code—that can simply be linked to any program to make it metamorphic. Metamorphic malware can be classified into four broad groups defined by two parameters. First, metamorphic malware may be either closed-world or open-world. Closed-world metamorphic malware is malware that is self-contained; in contrast, open-world malware may extend its capability by communicating with the world, say by downloading plug-ins from the web. To our knowledge, all metamorphic malware thus far has been closed-world, though open-world engines have been hypothesized (Driller 2002; Z0mbie; Ször and Ferrie 2001). Second, metamorphic malware may be either a binary-transformer or an alternate-representation-transformer. The former class transforms the binary image that is executed, whereas the latter class carries its code in a higher level representation, which is used for transformation.

Metamorphic engines, historically, have been written primarily by self-trained individuals, and have tended to be complex and buggy. However, with nations adopting cyber-warfare doctrines, one can anticipate an organized effort to develop state-of-the-art robust metamorphic engines. Besides, even teenaged criminals have begun to make a handsome, albeit illegal, living from their craft and hence have an incentive to improve the potency of their wares. Thus, it would be reasonable to predict that, in the not too distant future, one would see highly stable metamorphic malware in the wild. This paper takes a proactive stance in dealing with closed world binary transformer metamorphic malware by analyzing its design space. It restricts attention primarily to the design choices relevant to scanners that use only statically extractable information for detection. Design spaces are frequently formalized as a collection of goals, actions and constraints, and their implications or tradeoffs. Our design space analysis is from the view of the malware author, whose primary goals are to have their malware function as intended, to effectively replicate, and to avoid recognition by scanners. This analysis differs from prior treatments in that, instead of enumerating design solutions in a relatively decontextualized way, it situates design choices in the context of the malware designer's goals. It also traces tradeoffs between these choices and how their interactions with one another affects the design goals of metamorphic malware. Making the design space explicit in this way offers several benefits, including: providing ways of fundamentally changing discourse by allowing discussion of abstract classes of malware designs, and laying the groundwork for predictive models for future metamorphic engines by situating author options in the context of their goals.

As is with any warfare technology, metamorphism has defensive applications as well. Dube, Edge at al. (2006) have investigated the use of metamorphism as a means of protecting software. With software rapidly becoming one of the key elements of modern defense, including warheads and bombs, methods to protect software takes on a different level of significance. Indeed, Edge, Dube et al. (2006) explicitly studies the protection mechanisms used by computer viruses and how they may be used for protecting software. While

the discussion in this paper considers the design choices for an adversary using metamorphism, it is obvious that the same set of choices is equally applicable when metamorphism is used for defense. The paper is structured as follows. Section 2 introduces the prototypical anatomy of closed-world, binary-transforming metamorphic malware. This anatomy is used to structure our model of the design space for such metamorphic malware. This space is outlined in Section 3. Section 4 lays out the design space by traversing through the anatomy and listing options designers have for implementing their goals, drawing implications and tradeoffs arising from these options along the way. Section 5 concludes the paper by summarizing the advantages of the structure, and by listing directions for future work.

2. Prototypical anatomy of metamorphic malware

The classes of design options available to a malware author are related to the logical structure of the metamorphic programs they create. Thus, in order to organize a presentation of design choices, this logical structure must first be described. Our model of the anatomy of a prototypical closed-world, binary-transforming metamorphic engine is constructed as an analogue to Singh's anatomy of a computer virus (Singh 2002). It is more granular than that introduced by Lakhotia *et al*. in (Lakhotia, Kapoor et al. 2005), however. The anatomy of a metamorphic engine identifies functional units (or organs) which are conceptual units of tasks that must be performed by the engine but are not required to be implemented as modules or functions. Figure 1, adapted from the model of Lakhotia et al. (Lakhotia, Kapoor et al. 2004), depicts our simplified model of the anatomy. The features of the model are as follows.

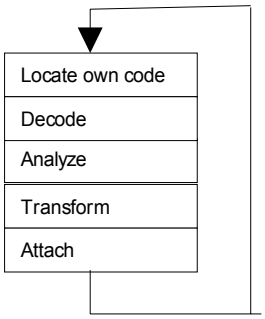


Figure 1: Anatomy of a metamorphic engine

Locate own code. A metamorphic engine must be able to locate the code to be transformed each time it is called upon to transform it. Parasitic metamorphic malware which transforms both its own code and its host's must be able to locate its own code in the new variant.

Decode. The metamorphic engine will need to decode information that is required to perform the transformations. A key piece of information is the metamorphic program itself since, in order to transform itself, some representation of itself is needed so that it knows how to make the transformations. In essence, this involves disassembly. However a metamorphic engine may also need to decode other types of information it requires for analysis or transformation. The information can be encoded in the malware body, either in the data segments, or in the code itself. Choice including using flags, bit-vectors, markers, hints, and so on. For example, the Miss Lexotan engine encodes register liveness information by storing it in the second argument of an "`xor ebp, immed`" instruction (Vecna 1998). Ordinary unpacking or decrypting of the whole executable is not within the scope of this Decode organ.

Analyze. In order for the metamorphic transformations to work correctly, certain information will normally need to be available. For example, certain transformations are semantics-preserving only when a given register is known to be *dead.* In order to perform these transformations correctly, then, the engine must have register *liveness* information available. When such information is not made explicitly available (and hence decoded), it must be constructed by the engine itself. In the case of register liveness, this can be constructed, in part, via a "def-use" analysis. The control flow graph (CFG) of the program is one piece of information that is frequently required for analysis and transformation. It may be used, for example, to rewrite the control flow logic of a program if a transformation expands the size of code.

Transform. The Transform step transforms the malware code into equivalent code, usually by replacing instruction blocks in the code to be transformed with equivalent ones. Well known metamorphic transformations include register-renaming, code substitution, NOP insertion, garbage insertion, and instruction reordering within a block.

Attach. Parasitic metamorphic malware need to attach the new version to a host file.

The ordering of the organs represents the direction of information flow, which is not required to be the execution order. The feedback loop acknowledges that the output of a metamorphic engine may become its input in the next generation. This feedback has previously been deemed as the Achilles' heel of a metamorphic virus (Lakhotia, Kapoor et al. 2004). It is critical for understanding the implications of design choices and the tradeoffs that typically must be made during engine design.

3. The structure of the design space

There are possibly many ways of constructing useful models of a design space such as the one facing authors of metamorphic engines. Significantly, a design space is more than just a listing of possible solution features: it must relate to resolution of designer's goals, and make explicit the tradeoffs implicit in various solution strategies. We define a model of the design space for metamorphic malware in terms of *goals*, *options*, and *implications*. Goals are things the designer intends to achieve. Options are ways of achieving goals and typically have both negative and positive implications. Implication analysis follows options to how they affect other options or goals. We therefore treat the design space as a graph of design choices connecting goals and other design choices via design implications.

Goals. The typical goals of metamorphic malware are to thwart reliable detection (even when the scanners are given one of its variants), propagate effectively, and perform its intended functions. Another important goal common to malware authors is simplicity or parsimony, i.e., they usually wish to choose the simplest and easiest option for achieving their other goals. There are two main goals relating to thwarting detection. First and foremost, the central purpose of metamorphism is to reduce the number of statically identifiable patterns in that can be used for signature detection. Second, the author may wish to make it difficult for scanners or human analysts to properly analyze its code.

Options. The key goals of a metamorphic malware author are in thwarting detection by reducing the number of identifiable patterns, and by raising the level of difficulty for analysis. For the former purpose, program transformations are selected that alter the form of the code without affecting its functionality. In general, the aim is for maximal change in form between all variants created. For the latter purpose, obfuscating transformations can be used, or transformations can be selected that require the solution of problems too difficult for scanners to perform. In the general case, the obfuscations may make only minimal impact on the form but still disable recognition by scanners. Key design choices available to the metamorphic malware designer relate to how these transformations are implemented. Any metamorphic engine must have access to information about the code to transform. This information allows the engine to determine whether or not some program transformation can be performed at a given point in the code. For the purposes of this closed-world engine design space, there are three possible ways of furnishing this information: (1) encode it into the malicious program itself, (2) assume it, or (3) construct it through self-analysis. These strategies yield multiple options, depending upon what the information is needed for. These options are enumerated in a principled manner by consulting the anatomy of the metamorphic engine from Section 2.

Implications. The complexity of the design space is greatly affected by how the goals, options, and implications interact. In a simple design space no design choice restricts or interferes with another (i.e., there are no cross-cutting constraints), no tradeoffs need to be made, and no options impact multiple goals. In a complicated design space there are such interactions and tradeoffs. Metamorphic malware presents an especially complicated design space because of the "meta": the space is a recursive one—one in which design choices affect prior design choices. If the metamorphic engine did not need to transform its output, the space would be less complicated. Instead, because it must transform its own output, a single design decision may create many implications, both positive and negative.

Consider an example. A designer's goal may be to defeat disassembly by malware scanners. One option would be to insert jumps over junk bytes, which can defeat so-called "linear-sweep" disassembly. However this decision has consequences: now the malware itself cannot use linear-sweep disassembly in its own analysis. Inserting junk bytes is one option, but notice the design space is not understood until the implications are elicited and placed within the context of the designer's goals. Some initial remarks about the options are required before continuing. Recall that we listed three different strategies for defeating anti-malware analysis: removing consistent patterns, defeating analysis methods through obfuscation, and requiring scanners to solve difficult problems. Ordinarily, the latter type of attack is possible because the

engine is able to more readily access or compute information needed to perform the analysis and transformations. A scanner may not always be able to construct the same information. For example, a metamorphic engine may capitalize on the existence of problems which are widely believed to be hard on average. These problems are currently relied upon in many cryptographic schemes and can be used by a metamorphic engine to introduce obfuscations only it can efficiently solve (using some sort of "certificate" or "extra knowledge") and thus be able to efficiently analyze its code in order to transform it. A metamorphic engine may also capitalize on the existence of problems which are widely believed to be hard in the worst case. These include the NP-complete problems. Coming up with “efficient exact” solutions to these problems is an immediate ticket to fame and fortune and is obviously not a reasonable expectation of a generic anti-malware scanner. Designers of anti-malware scanners can at most apply “efficient approximate” solutions to such problems (or efficient exact solutions to their sub-problems).

4. The design space

Given the anatomy of a metamorphic engine and the formula for its structure the design space can be present. The following sections list some of the significant design choices in the space. It is organized by the organization of organs in the prototypical malware.

4.1 Locate own code

Parasitic metamorphic malware must locate its own code in order to transform it. There are two main possibilities for transformation: keep it in a fixed location (the host executable's entry point), or change its locations. While the former option satisfies goals of simplicity, it does little to advance the goal of thwarting detection. The latter forces the engine to discover the locations. One option is to encode it; frequently this is done by using special markers or code segments known to indicate the beginning of the malware code. The marker can remain constant across generations of the malware or it can be transformed. There are tradeoffs to either choice. If the marker is required to remain constant across generations, then the transform step may not transform it. This is obviously the easiest choice for the engine, but it leaves it open to detection techniques which look for that specific marker. If the marker is transformed by the engine, simplicity is traded off since the engine now needs some method for creating new marking techniques programmatically.

4.2 Decode

Information may be encoded in the malware body to enable it to successfully carry out any of the tasks identified in its anatomy. To further harden the task of malware scanners, the engine could deliberately require the scanner to “answer hard questions” at any of these steps. We use the word “hard” here very loosely to refer to problems which are unsolvable (e.g., the Halting problem), *known* not to have an efficient exact solution (e.g., provably intractable problems), *believed* not to have an exact efficient solution (e.g. NP-complete problems), or simply whose known solutions still require an unacceptably high amount of space or time to carry out their computations on modern computers. These questions, of course, should not be as hard to answer for the engine. To circumvent the hardness of some general question, the engine may carry—or otherwise know how to infer from the code—some extra information that enables it to quickly answer this question and move on to the next functional unit and eventually be successful in efficiently generating a semantically equivalent variant of the malware. The Miss Lexotan virus, for example, embeds register liveness information in the code to be transformed so that it can transform the code without the need to perform an imprecise and inefficient def-use analysis on the entire program. But if the embedded information is constant it simplifies the scanner's problem. Transforming the encoding during replication adds complexity to the engine and may restrict the set of transformations it can successfully apply.

The simplest disassembly technique, the one commonly used by disassembly tools and debuggers, is linear sweep (Linn and Debray 2003). While meeting the goal of simplicity, the choice of using linear sweep significantly limits the code for which the malware can yield correct disassembly. For instance, for architectures such as the Intel IA32, linear sweep cannot correctly disassemble programs with code and data interspersed, with data bypassed using jump instructions (Z0mbie; Lakhotia, Kapoor et al. 2004). Another option is to use other, more powerful, disassembly methods such as recursive traversal or knowledge-based disassembly. They both expand the class of programs that can be disassembled, but still may not yield correct results with programs using jump tables for indirect transfer of control (Linn and Debray 2003). The choice of a disassembler has several important implications. An important *overall* effect is that it can limit the transformations that the metamorphic engine may perform. Since the malware needs to disassemble its own code, it cannot afford to use obfuscating transformations that thwart its chosen disassembly method. The disassembler choice can also constrain the choice of variation-inducing transformations. For example, using

a linear sweep method may simply rule out adding variety to the control logic via jump tables. The choice of a disassembler also has an effect on the set of target host files that a metamorphic virus can infect. For instance, a metamorphic engine that does not disassemble floating-point instructions will not be able to correctly disassemble any payload that contains such instructions. As a result such programs cannot acquire metamorphic capabilities by linking to that engine. Metamorphic malware can also deliberately use anti-disassembly obfuscation techniques that only it can quickly circumvent, normally because of assumptions it makes. The opcode shifting metamorphic transformation was shown by Dube et al. (Dube, Edge et al. 2006) to be quite effective at thwarting disassembly of the malicious code.

4.3 Analyze

Metamorphic engines have several options to choose from to construct the information needed for successful transformation and attachment. As described in Section 3, the designer can choose not to provide the information, or can choose to assume properties, encode the required information, or extract it.

4.3.1 Provide no information.

Choosing not to provide the information will restrict the designer's options for transformations. For example, one of the most commonly required pieces of information is a control-flow graph (CFG). A CFG is required to "rewire" the control flow if the starts of basic blocks are moved during transformation. So if a CFG is not available the malware author may not, in general, be able to add transformations that shrink or grow the program. Similar information, including register liveness and block-boundary information, are also commonly needed during transformation. Without them the engine designer is analogously restricted in the payload and engine code, and in the choice of transformations relating to registers and code blocks. This restriction is felt particularly keenly when implementing *conditional transformations* (see Section 4.4). Restrictions on the transformations normally constrain the effects of metamorphism, reducing the variation and simplifying the problem of scanning. For example, without liveness, many expression transformations are not feasible, and then compiler techniques such as constant folding might be used to counteract variation-inducing transformations, thus assisting the scanner. In general, A number of other exact and approximate techniques were developed to assist anti-malware scanners in detecting metamorphic malware that uses these techniques. Code normalization techniques have been proposed to counter the code substitution and instruction reordering transformations (Lakhotia and Mohammed 2004; Bruschi, Martignoni et al. 2006; Walenstein, Mathur et al. 2006). Techniques based on analyzing the structure of the malware's code and its control-flow graph have also been studied to counter the effects of these and other transformations (Kruegel, Kirda et al. 2005; Bruschi, Martignoni et al. 2006) (Jha, Seshia et al. 2005).

4.3.2 Assumptions.

The designer can choose to encode certain assumptions within the engine, obviating the need for the information. Since code is written by the malware author herself, she can ascertain that the assumptions hold throughout her code. This choice simplifies construction, but restricts the code to only that which follows the assumptions. Furthermore, the choice may also make certain transformations more difficult, or even impossible. For example, if Analyze and Transform steps make use of the property "register `eax` is not live after a `push eax`", it needs to be preserved for it to work correctly again in its offspring. Thus the Transform organ must generally ensure that this property holds across all variants of the malware.

4.3.3 Encoding.

Hints or data may be encoded using a certain pattern of code. For instance, as it propagates, metamorphic malware can also carry a data structure encoding the malware's full CFG. Alternatively, it might encode only some hints in the code that tells it how to obtain the targets of an indirect jump or call instruction. The metamorphic engine can use the CFG for its analysis, transform the code, and create an encoding of the new CFG. The construction of the initial CFG can be done with special tools, or manually, before the malware is released. Such an encoding of information can make analysis a breeze for the engine but not for an anti-malware scanner that is not aware of the existence or significance of the hint. For example, as mentioned before the Miss Lexotan engine encoded liveness information in special instructions. For any scanner not aware of these, a time consuming def-use analysis is required on the entire code. Moreover, such analysis is usually imprecise and hence unhelpful in detecting the inserted garbage code. More generally, metamorphic engines which encode information needed in transformation succeed in performing semantics-preserving transformations correctly because such engines know how to interpret (and what to infer from) such encodings. Furthermore, the engines can introduce obfuscations that cannot be

deobfuscated by a general deobfuscator. The engine would use such encodings (which are unintelligible to a general scanner) to understand its obfuscations. This constitutes an explicit attempt by the engine to defeat deobfuscation. Current engines do not appear to explicitly try to trip deobfuscators. They carry additional information (or assumptions) so they can do their own work correctly.

4.3.4 Extraction.

Extracting the information frequently means implementing complicated analysis routines; these require skill and are difficult to correctly implement. They run counter to the designer's goals of parsimony. Several implications are created by the choice of information to extract. Choosing to extract a CFG can place other restrictions on the type of code that can be used in the payload or engine, and can restrict the range of transformations it can make that affect control flow. Ignoring the impact of exceptions or signal handling, the completeness of the CFG depends on the correct decoding of jump and call instructions. There are two types of jump and call instructions: direct and indirect. In the direct category are instructions in which the address to which control is transferred is available in the instruction itself. Indirect jump/call instructions are those which obtain

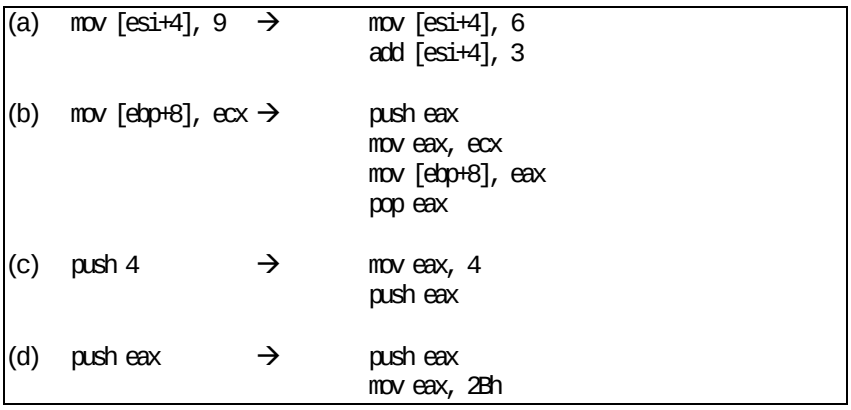


Figure 2: Conditional and unconditional transformations

the address for transferring control from either register or memory location. CFG construction is easy if only direct transfers of control are used. When indirect jumps are present, it may be extremely difficult to discover all possible branch targets; in the worst case, it is not solvable. Thus the decision often involves a tradeoff involving three issues: simplicity or ease in analysis, flexibility in the code that can be used, and choice of transformation types to implement scanner defeats. Other information, including register liveness and block-boundary, may also be needed, and these normally also require construction of a CFG. One alternative strategy that can be used to ease the difficulty of generating the engine is to limit static analyses to relatively simple ones. This choice forces the engine to live with the potential impreciseness of the resullts. Clearly these will limit the Transfom organ since *conditional* transformations (see Section 4.4) may need to rely in precise information (such as “is the value n this register a constant?”) to be semantics-preserving.

4.4 Transform

A simple choice to implement the Transform step is as a collection of transformation rules, such as those of a term-rewriting system (Walenstein, Mathur et al. 2006). When a code segment matches an instance of the left hand side (LHS) of a rule, it may be replaced by an instance of the right hand side (RHS) of the rule. The rule-based transformations can be further classified along two dimensions: a) the preconditions necessary for applying transformations and b) the transformation unit or the chunk-size of the LHS. Each dimension provides design options.

4.4.1 Preconditions for applying transformations.

The transformations can be classified into two categories: unconditional or conditional. The former type can be applied whenever a code segment matches the LHS of the rule; application of the latter transformation requires a match with the LHS *plus* certain other conditions must hold. Some of these conditions are exemplified by the rules in Figure 2 implemented in the virus W32.Evol. Rules (a) and (b) of this figure are unconditional rules, whereas the rules (c) and (d) are conditional. The RHS of rule (c) modifies the register `eax`, which is not modified by the LHS. To ensure that the transformed program has the same behavior as the original program it is necessary that the register `eax` not be live at the instruction where the rule is applied. The same condition also applies for (d), even though the LHS also modified `eax`. Nevertheless, the

condition is not required for rule (b) in which eax is modified in the RHS, used, and then restored. Clearly, applying only unconditional transformations is easier for the metamorphic engine. However, this choice also makes it easy for an anti-malware scanner to reduce the various generations to a single form, provided the transformation rules are known (Walenstein, Mathur et al. 2006). To perform conditional transformations, the metamorphic engine must have the information required to verify the condition so that it can preserve the overall semantics of the program. In Figure 1 the task of computing this information is allocated to the Analyze organ.

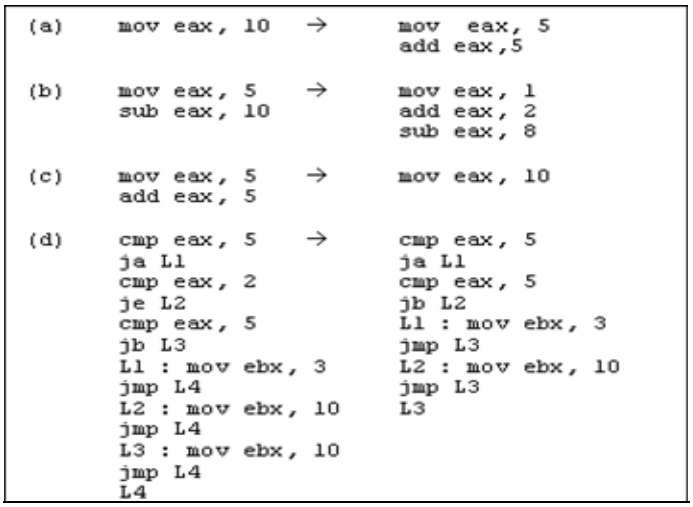


Figure 3: Single, block and SESE sub-graph transformations

4.4.2 Transformation chunk size.

To ensure correctness of transformations, it is ordinarily easiest if transformation rules are applied on single-entry single-exit (SESE) code segments (Lakhotia and Deprez 1998). Rules that are applied on SESE segments can further be classified on the chunk size of the SESE segment, i.e., whether it is a single instruction, a block of instructions, or a SESE subgraph. Figure 3 provides examples of these categories. Rule (a) transforms a single instruction; Rules (b) and (c) transform blocks of instructions; Rule (d) transforms a SESE subgraph. By definition, a single instruction is SESE, if exception conditions are ignored. So when the chunk size is a single instruction a rule may be applied if the LHS matches (and preconditions, if any, are satisfied). However, for chunk sizes greater than a single instruction it also becomes necessary for the metamorphic engine to ensure that the code segment matching the LHS is SESE, which in turn depends on correct construction of the CFG. The choice of chunk size can significantly affect the type of transformations that can be performed in hopes of thwarting detection. W32.Evol only applies single-instruction transformations; as a consequence its code monotonically increases over subsequent generations. On the other hand, Metaphor transforms single-instructions and blocks, thus causing its code to expand and shrink (Driller 2002). Though transformation of non-SESE segments is possible, the transformations also increase the theoretical challenges needed to ensure that the rules preserve correctness.

5. Conclusions

Metamorphism is known to have the potential to make detection of malicious code extremely difficult. While metamorphic malware to date have tended to be proof-of-concept and buggy, the threat of metamorphism is not to be taken lightly. We classify metamorphic malware as open-world vs closed-world and binary-transformer vs alternate-image transformer. Our analysis of the design space for closed-world binary transforming metamorphic malware yields an understanding of the space of design choices available to a malware author. Care was taken to link design choices available to the designer's goals and the implications and tradeoffs that the choices imply. The analysis makes it clear that the space is primarily structured by the needs of the metamorphic engine (its organs) and by the tradeoffs between transformation power and difficulty of implementation and analysis. The information needed to correctly implement the various transformations is a central theme. This analysis of tradeoffs lays a foundation for anticipating future directions in metamorphic engine design. At this time, one possible conclusion that can be made is that in the near-term, the main analysis battlefield is likely to be over encoded or assumed properties. The tradeoff analysis provided suggests that as a reasonable conclusion because of a confluence of forces: encoding or assuming properties meets parsimony goals, it can force scanners to solve hard problems, and it frees the

metamorphic malware designer to consider more powerful obfuscating or variation-inducing transformations that improve the ability to thwart detection. This then gives a malware author the upper hand and counters the earlier observation that “if a metamorphic malware can analyze itself, so can the scanner” (Lakhotia, Kapoor et al. 2005).

Further research is needed to improve the model of design space. One improvement that can be entertained is to create a more complete inventory of designer goals such that additional realistic design rationales can be constructed. A second advance would be to refocus the space on a more principled deconstruction of the information needed during transformation, and how it can be provided. This more detailed space is expected to be especially beneficial for predicting what properties are likely to be assumed, which information may be hidden and where, and how each choice influences the tug-of-war between the ‘vxers’ and ‘avers.’ It should also provide a more fine-grained vocabulary for discussing the classes of malware.

Acknowledgments

This work has been sponsored in part by funds from Louisiana Governor’s Information Technology Initiative.

References

Bruschi, D., L. Martignoni, et al. (2006). Detecting Self-Mutating Malware Using Control Flow Graph Matching. Proceedings of the Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA), IEEE Computer Society.

Bruschi, D., L. Martignoni, et al. (2006). Using Code Normalization for Fighting Self-Mutating Malware. Proceedings od the International Symposium of Secure Software Engineering, IEEE Computer Society.

Driller, M. (2002) Metamorphism in practice or "How I made MetaPHOR and what I've learnt". VX Heavens. http://vx.netlux.org/lib/vmd01.html.

Dube, T. E., K. S. Edge, et al. (2006). Metamorphism: A Software Protection Mechanism. International Conference on Inforfmation Warfare (ICIW2006), University of Maryland Eastern Shore, USA.

Edge, K.S., Dube, T.E., et al. (2006). A Taxonomy of Protection Used in Computer Viruses and Their Application to Software Protection, International Conference on Inforfmation Warfare (ICIW2006), University of Maryland Eastern Shore, USA.

Jha, C. M., S. Seshia, et al. (2005). Semantics-Aware Malware Detection. IEEE Symposium on Security and Privacy: 32-46.

Jordon, M. (2002). Dealing with metamorphism. Virus Bulletin**:** 4-6.

Kruegel, C., E. Kirda, et al. (2005). Polymorphic Worm Detection Using Structural Information of Executables. Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID), Springer-Verlag.

Lakhotia, A. and J.-C. Deprez (1998). Restructuring Programs by Tucking Statements into Functions. Journal of Information & Software Technology **40**: 677-689.

Lakhotia, A., A. Kapoor, et al. (2004). Are Metamorphic Viruses Really Invincible? - Part I. Virus Bulletin**:** 5-7.

Lakhotia, A., A. Kapoor, et al. (2005). Are Metamorphic Viruses Really Invincible? - Part II. Virus Bulletin**:** 9-12.

Lakhotia, A. and M. Mohammed (2004). Imposing Order on Program Statements and its implication to AV Scanner. Proceedings of 11th IEEE Working Conference on Reverse Engineering, Delft, The Netherlands, IEEE Computer Society Press, Los Alamitos, CA.

Linn, C. and S. Debray (2003). Obfuscation of Executable Code to Improve Resistance to Static Disassembly. Conference on Computer and Communications Security, Washington D.C., USA, ACM: Association for Computing Machinery.

Singh, P. K. (2002). A Physiological Decomposition of Virus and Worm Programs. Center for Advanced Computer Studies, University of Louisiana, Lafayette.

Stepan, A. E. (2005). Defeating Polymorphism: Beyond Emulation. Virus Bulletin Conference.

Ször, P. and P. Ferrie (2001). Hunting for metamorphic. Virus Bulletin Conference, Prague, Czech Republic.

Vecna (1998). Miss Lexotan 8, 29A E-Zine.

Walenstein, A., R. Mathur, et al. (2006). Normalizing Metamorphic Malware Using Term Rewriting. Sixth IEEE International Workshop on Source Code Analysis and Manipulation (SCAM 2006).

Z0mbie. About reversing. VX Heavens. http://vx.netlux.org/lib/vzo14.html.

Z0mbie. Some ideas about metamorphism. VX Heavens. http://vx.netlux.org/lib/vzo20.html.

Information Terrorism in the new Security Environment

Ken Webb
RNSA and Edith Cowan University, Perth, Western Australia
k.webb@ecu.au

Abstract: Over the years there have been many interpretations of what constitutes Information Terrorism. This paper reviews literature on Information Warfare and Terrorism to deduce what the threat of Information Terrorism is considered to be now in the new security environment. It achieves this by outlining the threat itself, and its potential impact, capability and advantages. The positives that can be derived to counter it are then examined and, based on the literature reviewed, a deduced interpretation/definition of Information Terrorism is provided. The paper concludes with remarks that Information Terrorism is a major dynamic contributing to a new national security environment.

Keywords: Information terrorism, information warfare, counter-terrorism, national security, future terror

1. Introductory remarks

This paper provides a review of literature to inform delegates of what constitutes Information Terrorism and the considerations surrounding this threat. The review was conducted as part of an Australian government-funded research project investigating how to better plan for and manage national security from asymmetric threats using terrorist information operations as the foundation. This forms the basis of the paper and highlights that a new capability/advantage for groups considering terrorism exists, thereby burgeoning the new national security environment.

2. The threat

According to Schwartau (2001, p.4), "asymmetry is what gives terrorists their strength". Therefore, combine the asymmetric threats of terrorism and information warfare (IW), and a compounding hybrid form of asymmetric warfare threat is 'information terrorism'. Also, it appears information terrorism has become one of the unintended consequences of the 'information revolution' and society's increased interconnectivity (Henych et al, 2003). The relationship between terrorism and information warfare broadens because terrorists achieve their sought after effect by conducting information terrorism even though information system abuse does not necessarily result in direct violence against humans (Devost et al, 2002). Although, as physical violence is a necessary component of terrorism then many acts of criminal computer abuse may not be considered terroristic if they do not result in direct physical violence. Further to this, Devost et al contend that as technology's implications on society and politics increase then the outcome of terrorist action in this field will achieve greater effect. Therefore, the transition into the acceptance of Information Terrorism is still occurring and its future impact is concerning. This view is supported to some extent by Jones et al (2002) who argue that the action of terrorism is to cause 'terror' to the people and for this reason it is difficult, given the current state of information technological advancement, for information terrorism to be effective. They clarify this by noting that this will change into a new situation as cultural values change and society becomes more dependent on technology. Considering the date of this viewpoint and the rate of information technological advancement then change can be considered near. Supporting this impending change is Clarke (2003, p.1), the former US presidential adviser for cyberspace security, who gives evidence of the probable conduct of IW by terrorists by noting that:

> *"Investigators have accumulated intelligence about Al Qaeda's interests and skills in using cyberspace to launch an attack. Many experts believe terrorists could likely combine such a cyber-based disruption with a real-world physical attack to amplify the impact. We're troubled by the fact that a number of people related to Al Qaeda -- including Khalid Sheikh Mohammed, who was recently arrested and was the chief operating officer...have [a] technical background. Recently, a student at the University of Idaho was arrested by the FBI for alleged terrorist connections, and he was studying in a PhD program on cyber security. So, I think, similarly to the fact that some of the Sept. 11 hijackers had training in flight training, some of the people that we're seeing now related to Al Qaeda had training in computer security."*

Despite this claim, Clarke, only a year before and contrary to the above testimony, is cited via Conway (2003, p.39) saying "most terrorists have not engaged in information warfare…instead, he admits, terrorist groups at this stage have only used the internet for propaganda, communications and fundraising". This is an indicator of the threat's prompt evolution. Dearth (2001, p.67) describes and accentuates the new situation by stating:

> *"Advanced societies are moving to a new place; call it the 'infosphere' or 'cyberspace'. In the journey to this new place, humankind will bring with it old impulses and needs: the need to connect, to converse, to share intimacies, to conduct commerce…will also bring with it the impulse to violence, conflict and war."*

Underlying the new situation and threat is that information terrorism is a relatively new phenomenon yet to be seen on a massive scale. However, it is worth noting that Hoffman (in Weimann, 2006, p.ix) outlines that one of the enduring axioms of terrorism is that it exists to attract attention and uses publicity to communicate its message. In the past, this has been relatively restricted because weapons to do this consisted mainly of guns and bombs. However, terrorists no longer have this restriction because the modern terrorist's arsenal includes computers, CD burners, email accounts, the Internet and the world wide web. Further to this observation, Sloan (2006) adamantly contests that at the very least terrorism is primarily a form of psychological warfare. Therefore, as psychological warfare is a component of IW and when considering its relationship with terrorism, it is clear that information terrorism appears to be the underlying principle behind the conduct of terrorism in the first place. Confirming this aspect is Weimann (2006) who concludes that, virtually without exception, all terrorist sites studiously avoid drawing attention to the violence or destruction that the group is responsible for. Instead, issues such as freedom of expression and human rights are used, and this indirectly suggests the psychological warfare purpose of information terrorism. Another consideration for the threat of information terrorism is the side effect explained by Wilson (2005). He contends that considering recent terrorist events have produced tighter physical and border security then terrorists have a reason to more heavily engage in information terrorism to achieve their aims and infiltrate nation states. Hinde (2001) confirms the existence of the threat in this regard by noting that shortly after the major physical terrorist attacks of 2001 in the US there were immediate concerns from data security experts that coordinated cyber-attacks might be launched against the telecommunications grid, which was already overloaded and slow because of the heavy demand caused by the physical attacks.

Bayliss et al (2002) support this assertion. They claim policymakers now fear that those conducting terrorism will electronically raid vital national systems controlled by computers. They also allege that fear is no longer based on the prospect of violence because information and the ability to control it has become a form of power, thereby amplifying the status of information terrorism. Another observation made by them is that terrorist cells can now share information and coordinate action without a hierarchical organisation that is vulnerable to penetration and subversion. Also of particular interest is that terrorism has been rooted consistently in Chinese society with no lapses in operation like much of the rest of the world (Thomas, 2004). Considering China is a prominent nation and now the world's largest supplier of information equipment, and the momentum and techniques of its information industry is increasing, further reasons to justify the increasing relationship between terrorism and IW are evident. Clarifying this effect and possible change required for national security is an indirect example given by Schwartau (2001, p.2), who cites that:

> *"While Western Militaries struggle for a decade on average to acquire new weapons, a country with commercially available computer equipment and less rigorous democratic and accounting processes could field new systems within a few years. It is the stuff of military nightmares."*

Therefore, at the very least and in anticipation of the above observations, the social and political definitions of terrorism should likewise broaden to accommodate information terrorism, and the current semantic vacuum of a universally accepted comprehensive definition suggests that IW is a probable new facet of terrorist activity (Devost et al, 2002).

3. The threat impact

The potential impact of information terrorism is considered severe because many countries now heavily use information systems to manage and operate critical services. Electricity, gas, water, sewerage and other services are now highly automated and computerised. These systems, in addition to banking, defence, government, telecommunications and transportation systems, form part of a society's critical information infrastructure (Armstrong, 2001). Such systems have been major terrorist targets in the past, and now that they are automated and more critical to society than before makes them increasingly vulnerable and attractive. Additionally, as attested by Wilson (2005), the computer networks that operate critical infrastructures now represent that nation's 'underbelly'. However, the software that operates them has proven vulnerable to attack through cyber crime, viruses, worms and other malicious code. Wilson underlines the increasing impact of such an attack by claiming that, because of the interdependencies among infrastructure sectors, a large-scale cyber attack that affects one sector may also have disruptive, unpredictable and perhaps devastating effects on other sectors, and possibly long-lasting effects to the national economy.

This observation conjures several devastating scenarios of information terrorism. These include altering medication formulas at pharmaceutical plants, 'crashing' telephone systems, misrouting passenger trains, and disrupting operations of air traffic control. The fact that this would apply to most nations makes it a universal consideration. Jones et al (2002) give another dimension to the impact of information terrorism in the form of denial of service attacks. They explain that when a terrorist group cannot achieve its objective by physical means then it has the potential to gain the desired impact by using the internet and the connectivity of other systems that the target audience relies upon. Conducting actions to prevent legitimate users of a service from using that service - such as network flooding, preventing direct access, and disrupting connections and systems - achieves that. Terrorists can do this relatively anonymously and without much effort from themselves, which provides further advantages. Considering that many critical infrastructures now interact electronically with users this brings further frightening scenarios. Also, information terrorism can directly or indirectly be used in conjunction with or to supplement physical acts of terror, thereby magnifying the desired impact. This conclusion is implied by Emery (2005), who outlines that wherever human activity occurs physically such activity also takes place simultaneously in the information dimension. Additionally, Wilson (2005) asserts that many security experts agree that a cyber attack would be most effective if it occurred to amplify a conventional terrorist attack. This means it is important to recognise that the residual effect from physical actions taken will shape the information environment. Enhancing the future impact of information terrorism is that new technologies will pose new risks and demand new responses to those risks. This includes terrorist usage of information for attacking nations (Anderson, 1999). Therefore, the potential impact and associated risks of information terrorism in the future will only increase with the advances in the information industry.

4. The threat capability

A primary enabler of information terrorism is the internet, which is a reason for the current transition into a globalised society. While the internet is a creation of the 'West', it is also attractive to terrorists for some of the same reasons it is attractive to society generally. It may be used anonymously so identity is masked; it is global, which allows access to huge audiences around the world; and it is inexpensive and subject to little regulation. Therefore, the Internet enables global terrorism (Deeks et al, 2005) and terrorist groups appear to be using it to their advantage. Confirming this observation, Thomas (2005, pp.34-44) in discussing terrorist methodologies, identifies 16 measures for consideration, as the Internet:

- Can be used to put together profiles;
- Access can be controlled or its use directed according to the server configuration, thus creating a true ideological weapon;
- Can be used anonymously, or as a shell game to hide identities;
- Produces an atmosphere of virtual fear or virtual life;
- Can help a poorly funded group to raise money;
- Is an outstanding command and control mechanism;
- Is a recruiting tool;
- Is used to gather information on potential targets;
- Puts distance puts distance between those planning the attack and their targets;
- Can be used to steal information or manipulate data;
- Can be used to send hidden messages;
- Allows groups with few resources to offset even some huge propaganda machines in advanced countries;
- Can be used to disrupt business;
- Can mobilise a group or Diaspora, or other hackers to action;
- Takes advantage of legal norms; and
- Can be used to divert attention from a real attack scenario.

In terms of increased capability, these measures highlight the value of the Internet to terrorist groups. Weimann (2006), who monitored terrorist web sites for eight years, validates this by explaining that terrorists are now clearly using the Internet for fund-raising and recruitment, training and instruction, propaganda and psychological warfare, and gathering information to plan attacks. Deeks et al (2005) provide an overview of the degree of information terrorism capability by noting that terrorist groups are now using the information means available to form an ideological platform that preconditions its audience to respond to subsequent solicitations of financial support and entreaties to undertake violent operations. In addition to conditioning the

minds of participants to accept acts of murder and destruction as politically expedient and morally acceptable, they explain that the electronic forms of communications used actively encourages participants to undertake such acts. This intimates that terrorists are even taking advantage of many organisations to directly solicit support. They are also perpetrating online crimes such as identity and credit card theft. In regard to the latter, cyber crime has increased dramatically over the past few years with several recent terrorist events funded partially through online credit card fraud (Wilson, 2005). As Alexander and Swetnam (2001) highlight, the capability of terrorist groups to conduct IW is occurring on all fronts. On the non-technical (soft) side transnational terrorist groups are becoming increasingly empowered by media outlets that present their position unaltered and unfiltered by governments or communication laws. They are using cellular phones, the Internet, and alternate news sources to do this. On the technical (hard) side, a growing danger is terrorists' slow but steady acquisition of digitally enhanced miniaturised technologies. Therefore, the 'information' or 'cyber' age has enormously increased the capabilities of terrorists (Thomas, 2005).

The whole spectrum of IW is now available to terrorist groups, as all the elements can be used and in combination with each other (Ibid). Additionally, as noted by Wilson (2005), terrorist group links with hackers and cyber criminals are adding to their skills, and finances obtained through drug trafficking may also provide terrorists with access to highly skilled computer programmers. In effect, terrorist groups are becoming technically sophisticated and years of publicity about information security weaknesses have made them more aware of the vulnerabilities of nation states. Terrorists and their symphasisers are already embedded in societies with a large information technology workforce (Wilson, 2005), and the observations made above infer that the human aspects of IW must be used when considering Information Terrorism. The human aspect includes the central part the Internet is playing in the battle for 'hearts and minds' (due to the global, easy and widespread accessibility of this form of information transmission). This is particularly apt, as terrorists attempt to influence their will upon people. Their threat to societies and states is more about the adverse psychological influence they cause rather than the physical damage they inflict (Radvanyi, 1990; Casciani, 2004), and this introduces the psychological warfare dimension again. The increasing capability of terrorists to conduct IW is evident. A real example is given by Wilson (2005) in the form of a well distributed and publicised book by Iman Samundra, who was convicted for his part in the 2002 terrorist bombings in Bali. In this book, Samundra advocates that Muslim youth actively develop hacking skills, and names several websites and chat rooms as sources for increasing these skills. He also urges Muslim youth to conduct credit card fraud to fund the cause. While it is acknowledged that Muslim youth generally will not follow this advice there are some who may be inclined to do so and this increases the capacity of this particular terrorist movement. An outlook given by Post et al (2000) adds a further dimension to this capability. They outline that what is avant garde now will become mainstream in the next 5-10 years when youth who have been socialised on computers join terrorist groups. Therefore, there is every reason to believe the reliance on information technology will become increasingly routinised and IW tactics will become incorporated in all terrorist operations. Post et al clarify this by explaining that the rapidity with which information technology is integrated will depend primarily on the group's socio-political context and the degree to which utilising it facilitates the group's cause. Additionally, Information Terrorism is likely to emerge among anti-establishment groups who perceive technological resources to be a source of vulnerability to their opponent or to be important to their opponent's efforts against them, and this creates a gambit of terrorist possibilities.

5. The threat advantages

Arquilla and Ronfeldt (2001) contest that not only has the information environment increased the types of targets and weapons for terrorist activity, but it has also provided more ways for those wanting to conduct terrorism to better operate and structure themselves, thereby placing them at an advantage. They explain that not only has the world embraced an increased access to information, so it can operate more efficiently and with greater flexibility, but terrorists have harnessed this power to enable new operational doctrines and forms of organisations. Therefore, they have become empowered to conduct IW. This ability to conduct information terrorism is elevated by groups considering terrorism who are 'disaggregating' from hierarchical bureaucracies, and moving to flatter, more decentralised webs of groups united by a common purpose. This has advantageously brought a whole new dimension and set of capabilities (Ibid). This advantage introduces the relatively untouched dimension of 'netwars', which has practice ahead of theory. The 'information' revolution favours the rise of network forms of organisations and these appear to be the next major form of organisation after tribes, hierarchies and markets. This form is redefining societies, and in doing so, the nature of conflict and cooperation. Resultantly, network-based conflict and crime is a major future phenomenon (Ibid). Furthermore, as Thomas (2005) outlines, symbolic leadership, which is instrumental to such organisations, is being magnified by using the Internet and other forms of information transmission. This appears to be giving terrorist groups conducting IW a further advantage. He uses Osama bin Laden, the

alleged leader of the al-Qaeda terrorist group, as an example. Thomas explains that bin Laden is an expert at buttressing Arab opinion with his TV and Internet messages that touch the souls and spirits of exasperated Arabs. He is a credible figure to many, as he extols al-Qaeda successes, gives directives and offers moral support. The media and international communication stations often pick up these broadcasts and replay parts of them to the public. Thomas emphasises that while bin Laden relays his message, as any leader should, the nation states are slow to respond because they are busy deciphering the message and trying to substantiate its authenticity. In the meantime, he moves onto his next planning phase.

Another advantage offered to terrorist groups is that a consequence of the prompt evolution in information terrorism is the inability of nation states to respond quickly to such, which has been the approach up till now in countering terrorism. This puts nations in a defensive mode where they are impacting heavily on their own society. It also means it has become hard for nations to communicate worldwide and influence large geographical populaces in a globalised society. The traditional law enforcement model used is also not proving an effective strategy for addressing terrorist conduct of IW and the flexibility provided by means such as the Internet is making countermeasures difficult (Deeks et al, 2005). Also, Libicki (1995) claims information terrorism has its advantages, as it allows effective terrorism by directing itself against very specific targets, even if carried out infrequently, from remote and often anonymous locations. This also introduces jurisdictional issues for law enforcement. The final advantage found is offered by Emery (2005), who notes that because terrorists cannot engage a superior force in the physical environment they conduct selected acts in the physical environment, such as bombings and small-scale attacks, to shape the information environment. These acts can help achieve objectives in the information environment and, ultimately, in the physical environment. Therefore, terrorists who lack military parity achieve their ultimate objectives by being successful in the information environment, and this highlights the effectiveness and reasoning for Information Terrorism in combination with physical terrorism.

6. Positives from the threat

Worth considering from a positive aspect of national security is that factors that can be used to counter terrorism have evolved but are not yet generally publicised. An example is given by Weimann (2006), who argues that, despite the multiplicity and diversity of terrorist websites, there are nonetheless core and common characteristics those terrorist groups with a presence on the Internet share. As Weimann explains, most terrorist sites are particularly notable for their colourful, well-designed and visually arresting graphic content, selective presentation of information, and effective message. Other common elements include descriptions of the given terrorist group's history, aims and objectives. Much other information such as biographies of leaders, maps, photos and communiqués are also on the sites. Regardless of the example above being confined solely to the Internet, this observation highlights that messages for improving the awareness of information terrorism and how to counter it exist when examining the positive indicators. Additionally, as Deeks et al (2005) and Smith (2005) imply, the approach taken can be changed by exploiting the increased information ability that has been created by nations, thereby placing nations at an advantage, and by winning the debate on aspects such as the Internet rather than by trying to suppress it. These are all worthwhile considerations for national security and enhance the importance of the intelligence function. Another positive consideration for national security is given by Post et al (2000). They identify that, in contrast to the powerful group dynamics of traditional terrorist groups, those groups relying upon networked organisational structures and computer-mediated communications are subject to virtual group dynamics, which significantly affects their decision-making and risk-taking. Also, it implies that groups conducting information terrorism have serious security implications within themselves. Irrespective of these potential positives and vide Emery (2005), it is clear that terrorists compared to nation states have now adopted a much different strategy to achieve victory for their cause. They are now using a complex information operations strategy that, considering their circumstances, is more effective, efficacious and efficient than previous forms of operations. This has become an enabler for IW and, "as information systems increasingly form the underpinnings of modern society, terrorist attacks using tools of the information revolution and targeting information systems will become prevalent" (Post et al, p.119).

7. Concluding remarks

As this paper highlights, the topic of information terrorism is diverse and relatively new in its nature. However, after reviewing the applicable literature, the conduct of information terrorism is interpreted as: A non-state actor's premeditated and asymmetrical warlike conduct of information activities to fulfil their ethos, foster mass acts of terror and/or affect and disrupt the security and/or well-being of a nation or series of nations. This is done to:

- Appropriately effect and manage a change in a target audience's perception;
- Market philosophical propaganda so a target public's governance,
- Livelihood and will is influenced through fear;
- Operate advantageously, efficiently, effectively and efficaciously; and
- Preserve themselves from activities by allies, competitors and adversaries.

This definition has been composed to maximise the possible coverage of all the relevant and salient points derived from reviewing the literature. It has also been composed to ensure the right context is used when referring to the term. This paper also implies that the threat of information terrorism is relatively new and its impending future use is frightening. A plethora of commentators reviewed in the literature, many of which are given in this paper, espouse a whole range of solutions to intervene and counter this threat. However, this observation intimates the complexity and nature of the overall problem of information terrorism, which is a major determinant for the new security environment. Despite this, it is noticeable that a large number of authors argue that, irrespective of what viewpoint they hold, solutions such as a humanitarian convention need to be considered. This infers a preference to just plain aggression and force, which many authors contend is the current approach. It also suggests that, due to the information capacity of the word today, IW itself be employed to counter Information terrorism, as it not only addresses the threat holistically but can be done immediately and expediently considering the capacity of nation states countering terrorism. However, both Buzan et al (1998) and Rogers (2000) explain that there are two main obstacles to this challenge. The first, and by far the most substantial in their eyes, is that the necessary response will involve considerable limits being placed on wealth and power of the elite global minority, thereby requiring radical economic and political changes that are substantially greater than anything previously experienced. The second is that most thinking and writing on international security is deeply ethnocentric and conservative. Information terrorism in the new security environment has provided a new dimension and understanding of terrorism generally. It is further complicated by whether nation states use information operations or not to counter terrorism. This highlights a dilemma facing the world.

References

Alexander, Y. & Swetnam, M.S. (2001). Cyber Terrorism and Information Warfare: Threats and Responses. New York: Transnational.

Anderson, K. (1999). Intelligence Based Threat Assessments for Information Networks and Infrastructures. Global Technology Research Inc. [On line] Available: http://www.aracnet.com/~kea/Papers/threat_white_paper.shtml [23 March 2004].

Armstrong, H. L. (2001). Denial of Service and Protection of Critical Infrastructure. Journal of Information Warfare 1(2): 23-34.

Arquilla, J. & Ronfeldt, D. (2001). Networks and Netwars: The Future of Terror, Crime, and Militancy. Santa Monica: RAND.

Bayliss, J., Wirtz, J., Cohen, E. & Gray, C.S. (2002). Strategy in the Contemporary World. New York: Oxford University Press.

Buzan, B., Waever, O. & de Wilde, J. (1998). Security: a new framework for analysis. Boulder, Colorado:Lynne Rienner Publishers.

Casciani, D. (2004). An online war for hearts and minds. BBC Magazine 2(1): 4.

Clarke, R. (2003). Interview: Vulnerability - what are Al Qaeda's capabilities? CYBERWAR!. WGBH/Frontline. [On line] Available: http://www.pbs.org/wgbh/pages/frontline/shows/cyberwar/interviews/clarke.html [23 March 2004].

Conway, M. (2003). Cyberterrorism: The Story So Far. Journal of Information Warfare 2(2): 38-47.

Dearth, D. H. (2001). Critical Infrastructures and the Human Target in Information Operations. Journal of Information Warfare 1(2): 62-67.

Deeks, A.S., Berman, B., Brenner, S.W. & Lewis, J.A. (2005). Combating Terrorist Uses of the Internet. American Society of International Law. Proceedings of the Annual General Meeting 2005. pp.103-115.

Devost M.G., Houghton B.K. & Pollard N.A. (2002). Information Terrorism: Can You Trust Your Toaster? Sun Tzu Art of War in Information Warfare Research Competition. Washington DC: Institute for National Strategic Studies, National Defense University.

Emery, N.E. (2005).Fighting Terrorism and Insurgency: Shaping the Information Environment. Military Review 85(1): 32-39.

Emery, N.E., Earl, R.S. & Buettner, R. (2004). Terrorist Use of Information Operations. Journal of Information Operations 3(2): 34-45.

Henych, M., Holmes, S. & Mesloh, C. (2003). Cyber Terrorism: An Examination of the Critical Issues. Journal of Information Warfare 2(2): 1-14.

Hinde, S. (2001). Incalculable Potential for Damage by Cyber-Terrorism. Computers & Security, 20: 568-572.
Jones, A., Kovacich, G. L. & Luzwick, P. G. (2002). Global Information Warfare: How Businesses, Government, and Others Achieve Objectives and Attain Competitive Advantages. Florida: Auerbach Publications.
Libicki, M.C. (1995). What is Information Warfare? Washington DC: US Government Printing Office.
Post, J.M., Ruby, K.G. & Shaw, E.D. (2000). From Car Bombs to Logic Bombs: The Growing Threat from Information Terrorism. Terrorism and Political Violence, 12(2): 97-122.
Radvanyi, J. (1990). Psychological Operations and Political Warfare in Long Term Planning. New York: Praeger Publishers.
Rogers, P. (2000). Losing Control. Global Security in the 21st Century. London: Pluto Press.
Schwartau, W. (2001). Asymmetrical Adversarialsm in National Defense Policy, The Marketplace and Personal Privacy. Journal of Information Warfare 1(2): 1-11.
Sloan, S. (2006). Terrorism: The Present Threat in Context. Oxford: Berg
Smith, R. (2005). The Utility of Force: The Art of War in the Modern World. London: Allen Lane.
Thomas, T.L. (2004). Dragon Bytes. Chinese Information – War Theory and Practice. Fort Leavenworth, Kansas: Foreign Military Studies Office (FMSO).
Thomas, T.L. (2005). Cyber Silhouettes. Shadows over Information Operations. Fort Leavenworth, Kansas: Foreign Military Studies Office (FMSO).
Weimann, G. (2006). Terror on the Internet: The New Arena, the New Challenges. Washington DC: United States Institute of Peace Press.
Wilson, C. (2005). Emerging Terrorist Capabilities for Cyber Conflict against the US Homeland. US Congressional Research Service Report, November 1. pp.1-18.

Inter-Network Operations Center Dial-By-ASN (INOC-DBA), a Hotline for Critical Internet Infrastructure Management

Bill Woodcock[1] and Ross Stapleton-Gray[2]
[1]Packet Clearing House, Berkeley, CA, USA
[2]Internet Awareness, Inc., Berkeley, CA, USA
woody@pch.net
ross@internetawareness.net

Abstract: The paper describes the nature and operation of INOC-DBA, a global VoIP-based hotline, a practical and effective means for inter-NOC coordination in response to Internet security and stability incidents. INOC-DBA is a response to threats to Internet operations, which has emerged organically from within the Internet operations community.

Keywords: NOC, VoIP, incident response, infrastructure protection

1. Introduction

INOC-DBA ("Inter-Network Operations Center Dial-By-ASN") is a Voice-over-IP hotline telephone system linking the operators of critical Internet infrastructure worldwide. The system is used to carry both routine operational and emergency-response traffic between Network Operations Centers, or NOCs, where operations engineers monitor the status of data networks and coordinate response to natural disasters, hacker attacks, or terrorist threats. Autonomous System Numbers, or ASNs, uniquely identify administratively-distinct regions of the Internet, and are a preexisting well-understood numbering system used in this context. Callers on the INOC-DBA system can reach the operator of a network simply by dialing the ASN, as though it were a phone number. For the pre-qualified users of the system, this allows rapid disintermediated communications with responsible and authoritative parties, bypassing directories, receptionists, and voicemail.

Packet Clearing House (PCH) established INOC-DBA in 2002 and manages its operations and ongoing development. Packet Clearing House is a non-profit institute which provides operational support for critical Internet infrastructure. Founded in 1994, the organization directs projects to optimize the routing of Internet traffic by, among other things, educating and facilitating communication between Internet service providers worldwide. Based in San Francisco, California, PCH offers education, equipment, and operational support to organizations and individuals seeking to improve the quality, reliability, and accessibility of the Internet.

Over the past four years, as more users come to depend upon it as a principal platform for emergency coordination, INOC-DBA has been gradually transitioning from an experimental system to a production network with full-time, dedicated support staff and controlled development cycle.

2. History of INOC-DBA

INOC-DBA began as the result of discussions between Sean Donelan, a network security and stability expert, then at AT&T, and PCH in 2000 and 2001. At that time, Voice-over-IP was principally still used for point-to-point tunneling of E&M interconnections between PBXes across the Internet, using the venerable H.323 protocol. But as Internet-native VoIP became standardized, the possibility of direct many-to-many communications between Internet voice terminal devices became a possibility, and the first SIP-based version of INOC-DBA was put into limited trials in May of 2002, followed by an announcement of general availability at the end of September of that year (PCH 2002).

Because the need for network operators to more readily communicate with their peers to resolve problems is an inherent one, and the INOC-DBA communications system arose organically from within the community which faced the problem, it has been enthusiastically adopted within the community, facing none of the ideological or "not invented here syndrome" problems that have stunted similar emergency communications systems that were built outside the community.

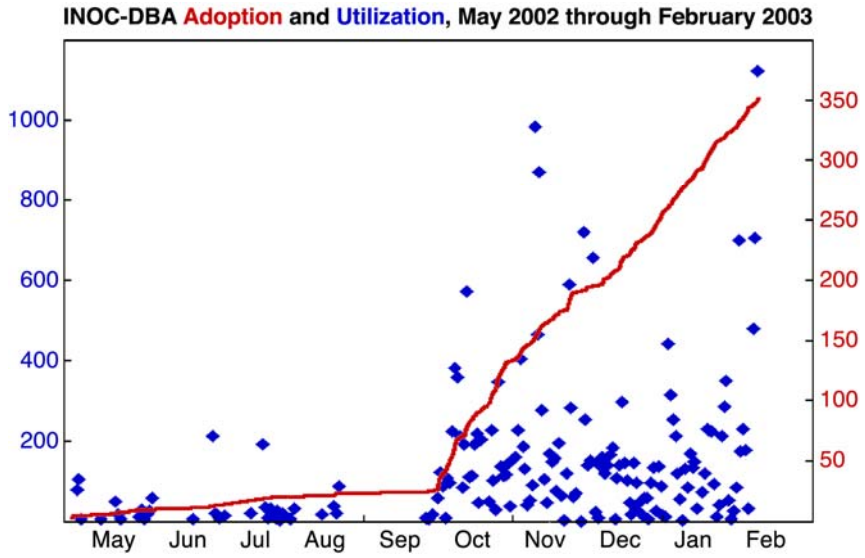


Figure 1: INOC-DBA adoption and utilization

The Internet's origins as an experimental network, and a growth characterized by "rough consensus and running code," are mirrored in its management. There is no central authority to direct its operations, though the thousands of administrators responsible for its myriad pieces generally work well as a collaborative community. Inter-NOC coordination was and is often an *ad hoc* exercise, with NOC operators maintaining personal contact lists, or posting contact numbers on web sites.

The crux of the INOC-DBA system is the binding of ASNs to the individual or individuals with actual responsibility for and authority over the Internet's many constituent networks. As such, it is little more than a logical overlay atop the Internet, employing "vanilla" VoIP technologies. That said, such a system can be a powerful tool in rapidly coordinating the parties responsible for an increasingly critical, global infrastructure.

PCH launched INOC-DBA with little more authority than the various contact compilation authors; on the other hand, a good many successful efforts at establishing resources of common concern for the Internet have had similar pedigrees.

The system utilizes the IETF-standard SIP protocol, which is non-propietary and enjoys functionally universal support among Voice-over-IP telephone vendors. Thus any SIP-compliant phone will work on the INOC-DBA network. The most commonly used terminal devices on the system are SIP "hard phones" such as those manufactured by Cisco, Grandstream and Linksys, and the "soft-phone" from Xten, but many others are available and many dozens are being successfully used on the system. (PCH would like to acknowledge in particular the role of Cisco in enabling construction of INOC-DBA, through contributions of many of its VoIP 7960 phones, particularly to encourage network operators from developing countries to participate.)

PCH provides registry, proxy and directory infrastructure. The ASNs of participating networks are registered with the service; it is also possible to register role accounts as two-digit, and individuals as three-digit extensions.

A listing of the INOC-DBA participants who have opted in to a public directory is visible at:

https://www.pch.net/inoc-dba/console.cgi?op=show_pubdir&list=org
In order to ring all of the phones within an organization from an INOC-DBA phone, one dials the organization's ASN. To reach a specific person's phone, one dials the ASN, the star or asterisk key (*), and their extension number. For instance, to reach anyone within PCH, one can simply dial 3856. In order to reach Bill Woodcock specifically, one would dial 3856*WEW (3856*939). Using native SIP addresses rather than dialing numbers, this last would be 3856*WEW@inoc-dba.pch.net.

3. INOC-DBA today

3.1 Operations and services

INOC-DBA is an operational, albeit still experimental, system, spanning all seven continents, and serving more than 1300 network operators in the NOCs of more than 600 telecommunications and Internet carriers, responsible for more than 900 unique networks worldwide.

As an Internet-routed VoIP service, INOC-DBA relies on the same technologies as its users already manage; there is obviously a concern that such a service would be vulnerable to the exact things it is supposed to be available to address. Some participants provision separate network resources for INOC-DBA within their own infrastructure, or lease outside infrastructure to achieve the appropriate logical and physical separation. And as one observer noted, "You could also argue that trying to contact someone about a major fiber cut using the PSTN is also doomed to failure, at least for some fiber cuts. That's no reason to never try using the PSTN." (Abley 2006) INOC-DBA is also not, as yet, a critical resource; as it becomes more widely adopted we anticipate needing to direct greater attention to risk and availability issues.

INOC-DBA was established as a special-purpose network, separate from the public switched telephone network (PSTN). This separation is purposeful, for two reasons. First, INOC-DBA is truly global, and in many countries, the regulatory regime prohibits "toll bypass" or the routing of voice communications between originators on foreign PSTN terminals, and domestic call recipients, without passing through a licensed (and taxed) international PSTN carrier. Thus INOC-DBA needs to remain officially disconnected from PSTN-originated calls in order to qualify as a private network, exempt in all countries. Second, INOC-DBA calls being answered promptly by people in positions of power requires that those people be assured that the call they're receiving over the hotline system is indeed important, and not just some end-user. INOC-DBA pre-qualifies participants, and authenticates callers; neither of these screening mechanisms would be possible if it were possible to originate calls on the PSTN for recipients on the INOC-DBA network. In practice, some INOC-DBA participants have implemented outbound gateways which allow them to receive INOC-DBA calls on their personal PSTN-connected mobile phones, and this has caused no problems.

INOC-DBA provides minimal core services, e.g., there is no provision for voicemail services (though participants could direct INOC-DBA calls into other systems, or to the PSTN, and a variety of services there). Secure and authenticated conference bridging is an often requested service, and is currently under development, along with a mean a means of positively identifying all participants in a bridge.

3.2 INOC-DBA membership

INOC-DBA membership has increased steadily, in a fairly linear fashion, with about twenty new organizations joining each month.

U.S. Federal adoption of INOC-DBA *has* accelerated recently. US-CERT, a part of the Department of Homeland Security's National Cyber Security Division (NCSD), has recently connected their emergency call-center to the INOC-DBA network, joining other divisions of the Department of Homeland Security, NASA, the U.S. Treasury, and the Federal Communications Commission among the U.S. governmental users of the system.

In general, users of the system are specialists, with responsibility for specific areas like network intrusion detection, BGP routing, peering, or inter-agency liaison. Large carriers are disproportionately represented, across the board, and carriers based in Europe and North and South America are represented more heavily than those from Asia. There's good representation among African ISPs, particularly the larger of them, though they're small organizations by comparison with the big multi-national carriers based in other regions. The vast majority of INOC-DBA users speak English, 64% as a first or principle language, and 12% as a second or subsidiary language, with Portuguese, German, French, Swedish, and Dutch being the next most common languages, out of fifty-two spoken by INOC-DBA participants overall. INOC-DBA participants tend to be nearer the core of the Internet than the periphery, accounting for a quarter of all transit networks globally so even if a network isn't one of the more than 900 directly participating in INOC-DBA, it's nearly certain that most or all of their upstream providers are, which is often sufficient for emergency-response needs.

3.3 INOC-DBA data for research

Data of many forms are provided to Packet Clearing House on a voluntary basis by PCH's research partners, peers, and project participants. PCH publishes this information to the public, including some statistics on INOC-DBA, and a portion of INOC-DBA's funding has been underwritten by the Department of Homeland Security's PREDICT (Protected Repository for the Defense of Infrastructure Against Cyber Threats) program to provide anonymized end-to-end call-quality statistics derived from INOC-DBA usage as a data-source for analysis by federally-funded network researchers.

PCH's current operating procedure with respect to statistics from the INOC-DBA telephone system is to not retain either phone numbers or individual IP addresses, and immediately anonymize all statistics by aggregating them up to the smallest BGP-advertised enclosing netblock. INOC-DBA participants opt-in to the statistics-collection by electing to participate through the online provisioning system, and by configuring their telephone device to send SIP call-teardown messages to the proxy at the conclusion of each call. Although this sounds difficult, in practice more than three-quarters of INOC-DBA participants also participate in the statistics-gathering project. Because of the opt-in nature, INOC-DBA statistics should be regarded as generally representative rather than complete or necessarily a statistically-representative subset, since they do not include data representing use by parties who have opted out.

The dataset provided under the PREDICT program contains data which characterize the quality of the paths which VOIP calls take across the Internet, worldwide. It consists of anonymized Session Initiation Protocol teardown messages collected from both ends of INOC-DBA conversations, and includes call duration; volume of data sent; number of packets sent, received and lost; and the number delivered out of order. The endpoints of the call are identified by country, Autonomous System Number (ASN), and subnet. We anticipate that the VOIP End-to-End Quality data will be used by researchers who wish to compare differential quality of service in similar and dissimilar regions of the Internet, such as across different backbone carriers which utilize different technology or capacity-planning methodologies. The data could also be used by researchers who are interested in correlations between the quality of service underlying voice communications and callers' patterns of utilization (Maughan 2005).

4. INOC-DBA into the future

4.1 Increasing use, and usefulness

We would like to think of INOC-DBA as an element of management infrastructure critical to a fast and efficient recovery from Internet disruptions. The Government Accountability Office (GAO) performed an analysis of existing government and private sector means and authorities to carry out such operations in the aftermath of a large scale cyber attack or natural disaster, and concluded that more work was needed (GAO 2006, GAO 2006a). That same analysis noted INOC-DBA as one of the very few collaborative tools that the private sector has produced to address these threats.

As a network, INOC-DBA is subject to traditional network effects, and as such adheres to Metcalfe's Law ("The value of a network increases exponentially with the number of nodes"). Thus seed populations within specific geographies or interest-groups tend to grow to include other members of those groups, as we've seen in the examples of particularly fast and complete uptake within the ISP communities in Brazil and Australia, for example. The accession of several federal agencies as subscribers is likely to attract more, and Packet Clearing House has submitted INOC-DBA as a concept in the recent Air Force Information Operations Battlelab solicitation, as a means to allow Air Force and other DoD NOC operators to better coordinate information assurance activities with federal, commercial and other NOCs worldwide. Similarly, within the Internet service provider community, a number of industry associations have settled upon INOC-DBA deployment as a service they can provide to their members, which has caused growth spurts and regions of particularly dense deployment. We're also seeing a trend toward more stand-alone Computer Emergency Response Teams and Computer Incident Response Centers (CERTs and CIRCs) joining the network, and they are very much welcome.

4.2 Extending the virtual network

Not only would increased membership increase that the chance that any particular network will be reachable within INOC-DBA, members might leverage each other's influence over networks that are *not* participants, as has been seen in other similar groups, like the Network Service Provider Security (NSP-Sec) mailing list, with which INOC-DBA has a very high degree of overlap. We have only conducted minimal analysis of INOC-

DBA network metadata to date, but expect that it should reveal interesting information on "pressure points" to influence Internet performance, e.g., nonsubscriber ASNs that are recipients of a high number of call attempts by subscribers (which would suggest, among other things, that other contact information for the networks of concern is unavailable or unreliable, and INOC-DBA is being tried as a last resort).

4.3 Authentication

As previously mentioned, features in the area of security are among our most-requested services, and are undergoing constant improvement. INOC-DBA is very much like the Internet in being a network of networks, a network composed of its participants and their willingness to interoperate; that said, like the Internet, as it grows it would also support defining smaller, closed communities within the whole. The tools necessary to support these closed communities, like "instant on" predefined emergency conference bridges that place outgoing calls to all members of a group when triggered, or voting tools which allow participants to make public or anonymous verifiable decisions in real-time, for instance on whether to disconnect from a network that's sourcing a large attack. PCH's focus will remain global, but INOC-DBA could serve as a platform for coordinating activities among subsets of its subscribers, e.g., as a secure VoIP network among U.S. federal agencies' NOCs.

4.4 Learning dot-com and web 2.0 lessons

Although much can be said about the follies of the all-too-often-overextended private-sector Internet applications industry, there have been a few lessons of general applicability which can benefit INOC-DBA users. Foremost among these is the success of reputation systems and the building of trust through "friend-of-a-friend" social networks. But whereas "social networking" web sites like LinkedIn and Friendster have had very little to build upon, the INOC-DBA network provides a predefined underlying framework, a reason-for-being. And whereas social networking sites use vague definitions of "friendship" as the link between participants, INOC-DBA has the very concrete PGP web-of-trust, with well-established protocols for establishing and attesting to the *bona fides* of other parties, published on multiple redundant preexisting public-key servers. INOC-DBA's developers intend to adhere as much to open standards and extenal verifiability in the authentication protocols and the network of trust, as they do in the voice protocols at the system's core.

5. Conclusions

INOC-DBA is in the process of evolving from an experimental exercise in means for inter-NOC coordination, toward a more robust, production service, suitable for use in ensuring continuity of operations and disaster recovery across the global Internet. While the concept is a simple one—using ASN in a one-to-many mapping to those responsible for its management—and its constituent parts all off-the-shelf components and protocols, it can provide an effective means of making the appropriate connections for NOC collaboration, linking colleagues and peers. Some data on its operation to date are available, and PCH will collect and analyze more, to understand how INOC-DBA might be expanded to respond to the demand for more means to ensure Internet operation in the face of both man-made and natural disruptions.

References

Abley, J. (2006) "Re: Anyone heard of INOC-DBA?", [online], http://www.mail-archive.com/nanog@merit.edu/msg40270.html

GAO (2006) *Internet Infrastructure: DHS Faces Challenges in Developing a Joint Public/Private Recovery Plan*, Report to Congressional Requesters, U.S. Government Accountability Office, GAO-06-672. http://www.gao.gov/new.items/d06672.pdf

GAO (2006) *Internet Infrastructure: Challenges in Developing a Public/Private Recovery Plan*, testimony before the Subcommittee on Telecommunications and the Internet, House Committee on Energy and Commerce, U.S. Government Accountability Office, GAO-06-1100T. http://www.gao.gov/new.items/d061100t.pdf

Maughan, D. (2005) "PREDICT Overview," PREDICT Workshop, Newport Beach, CA, Sept. 27, 2005, http://www.cyber.st.dhs.gov/public/PREDICT/PREDICT%20-%20Workshop%20-%20Sep2005%20-%20Maughan.pdf

PCH (2002) "Packet Clearing House demonstrates worldwide Internet infrastructure protection hotline", press release, October 28, 2002, http://www.pch.net/about/news/pressreleases/pressrelease10.28.02.html

SME Security in the Digital Age

Don Milne[1], John McCarthy[2], and Bryan Mills[2]
[1]Buckinghamshire Chiltern University College, High Wycombe, England
[2]ServiceTec Global Services Ltd, UK
don.milne@bcuc.ac.uk
john.mccarthy@bcuc.ac.uk
bryan.mills@servicetec.com

Abstract: The past few years has shown an increasing level of concern by governments, policy makers and the ICT Industry over SME's reluctance to engage in the digital economy. This will have an impact on the economy as a whole. Current data shows the uptake in the use of ICT technology has not been expanding at the rate expected. The security issues and concerns of SME's owner managers have been shown to contribute to the lack of uptake of the more complex ICT required for e-business engagement. Many small organisations, in both the commercial and the not for profit sector have a different approach to prioritising and decision making. This is reflected in their use of ICT systems and as a consequence has an impact upon ICT security. Whilst some of this under performance in the digital economy may be related to the management style and lack of ICT expertise of the companies the part played by the ICT service suppliers must also be investigated. Evaluation of the traditional security threats to organisations and individuals is undertaken to recommend suitable responses to these threats. By evaluating, through a series of case studies continued from a previous work, how the ICT service industry currently provides support for such organisations and how the products they offer satisfy or fail to match the needs of the SME's. From this it is possible to take a new look at how the provision of security support for the smaller end of the market should be undertaken in the future. From this study security service providers can relate their products and services to the functional needs of the SME giving them the ability to respond better to the needs of the small organisation.

Keywords: SME's; security; distributed systems; digital economy;

1. Introduction

Small businesses present a particular concern in the digital age. They commonly have neither ICT, security expertise, time or funds to devote to securing their systems. The adoption of e-commerce has slowed in these sectors (DTI 2004) causing concern to governments, policy makers and industry. Reports in the media on the scale and consequences of security breaches coupled with the lack of planning when implementing ICT (Dixon et al 2002) have compounded the problem. Despite initiatives by the UK government and policy makers (EURIM 2004, 2005, NHTCU 2004, NCC 2000,) e-business engagement in SME's is still causing concerns. Some academics (Wyer 2005, Milne and McCarthy 2006, Fillis and Wagner 2006) suggest that this failure is due to a lack of understanding of the small firm, its needs and how it operates.

2. SME e-business research

Much of the research undertaken in e-commerce has tended to focus on the large organisation Fillis and Wagner (2005), Drew (2002). The DTI Benchmarking 2003 study in this area reveals a clear divide between the use of ICT in medium and large businesses and ICT for micro and small businesses. There is a digital divide amongst the SME groups. Smaller and micro businesses are struggling with ICT. They do not possess the more sophisticated levels of ICT seen at corporate level. Arbore and Ordanini (2006) and Daniel and Grimshaw (2002) suggest the digital divide between large organisations and SME's is a fairly well-known phenomenon, which points out the asymmetry in information and communication technologies (ICT) endowment and use. In other words SME's are adopting Internet connections but are not utilising the technologies that it can foster. Another factor for the digital divide is the lack of environmental resources. This not only refers to the availability of physical infrastructure, but also that of service providers and partners with expertise Corrocher and Ordanini (2002)).

SME's are much less formalised or structured in their thinking about ICT.They have a more opportunistic approach to ICT investment. A report by the Small Business Service Research Programme (2002), agrees with these findings. Small firms have been slower to adopt e-business and research into this area has been slow to develop Fillis and Wagner (2005). Some e-business research has been undertaken which includes, benchmarking Internet use Webb and Sayer (1998), innovation and learning Chaston (2001), the micro enterprise and Internet usage Dandridge (2000), entrepreneurship and the Internet Colombo (2001) and barriers to adoption Walczuch et al.(2000). An important issue in understanding the needs of SME's is that they are not a homogeneous group Fillis and Wagner (2005). Their needs will vary by sector. Naturally, high-technology SME's are more likely to use e-commerce than others. There are also differences between

industry sectors in terms of e-commerce use and strategy development. The Small Business Research Team SBS (2006) gives data on ICT investment and e-business. The report shows that the business service sector invests more than the manufacturing sector and the retail sector invests the least of the three sectors they studied. These sectoral differences add to the problematic nature of e-business research in SME's.

3. SME characteristics

The characteristics of a small firm can be very diverse. This paper examines Micro and Small SME's in relation to ICT adoption. It also examines SME's and ICT use in SME relationship strategies and the relationship of the ICT provider with the SME. The DTI benchmarking study (2003) shows that over 98% of all SME's are either Micro or Small. This group of SME's account for over 35% of GDP in the UK, and are comprised of approximately 3.7 million enterprises and employ 36% of the UK workforce. An understanding of SME's is an important consideration to policy makers, governments and academic bodies and others. Due to the nature of the classification of SME's problems can occur when examining SME's as a whole group.

Arbore and Ordanini (2006) state:

> *Considering SME's as a monolithic reality, however, might be misleading. This group of firms can be extremely heterogeneous and its average values may hide important internal variances."*

SME's are a very disparate group and we have the issue of overarching SME classifications. The Bolton report of 1971 first defined SME's and the broad base it used often causes problems when attempting to research a particular issue. An SME with 240 employees will be far more like a large organisation that a micro SME with 5 employees. Cohen and Levinthal, (1990) suggest that the ICT absorptive capacity of an SME, namely its capability to exploit ICT potentials differ according to the size of the firm, even within the boundaries of the SME classifications. An understanding of these issues could assist ICT providers and others to facilitate better ICT provision and therefore ICT security for the Micro and Small SME's. Government, policy makers and academia have often been prescriptive in terms of the adoption strategic planning and learning models in SME's. There is an assumption that an SME is in fact a mini corporate company. Therefore a tailored cut-down model of the traditional planning and learning models would be suited to SME's. This has not been proven to be effective Wyer and Barret(2004).

The problem lies in the understanding of how an SME performs its strategic planning and learning. They operate using the discovery, choice and action model but in small experimental steps. However this is difficult for an SME adopting an e-business strategy often due to the expense, training, hardware resources and other issues.

4. SME owner managers

In many SME's the owner manager is a pivotal figure who has a major influence on the decisions and actions of the small firm. Brown (2005) suggests that an understanding the attitudes and philosophies of the owner-manager assists in the construction of a more complete picture of SME e-business development .Fillis and Wagner (2006) suggest that there are a number of owner-manager orientation implications that relate directly to e-business development. Being able to successfully combine the hard technical skills alongside the often softer entrepreneurial marketing competencies was identified as a key factor in small firm e-business development. They also found a range of attitudes in the SME's they studied to e-business adoption. They ranges from a "wait and see" approach" to a strategic integration of e-business as part of ongoing business development. The first principal fork in the adoption chain is that of positive and negative attitudes to e-business. This in many SME's is the attitudes of the owner/manager. For Fillis and Wagner (2006):

- "A range of attitudes have been identified that relate to the orientation of the
- owner-manager of the firm. Entrepreneurial thinking is more likely to be linked
- to positive attitudes towards e-business compared to conservative, risk-averse
- behaviour. These latter types will be less likely to take the lead in e-business
- development and will either be followers or non-adopters. Linked to entrepreneurial/
- conservative orientation is the ability to overcome both physical and
- perceptual barriers to e-business. Entrepreneurially oriented small firms perceive
- barriers in a positive fashion whereas the conservative firm may exhibit highly

- negative attitudes."

We see that the mind set of the owner/manager is crucial. In the SME's used in this work this is particularly true due to their Micro and Small size.

SME's may adopt e-business for several reasons. They may adopt ICT to gain increased revenue and stay ahead in the market place. There may however be other factors involved, which mean the reason to change or not is dependent on factors outside of pure profit and revenue. Some studies show that pure profit generation is not the main motivating factor in some SME's Fuller (2000). Fuller further states that other factors such as increased service to customers or remote management of the business may play a part. The owner may just have an interest in this area. Ramsey et al.(2003) believe that e-business decision making in the small firm is mostly directed by the competencies of the owner-manager.

5. ICT evolution and trusted agents

Milne and McCarthy (2006) showed that SME's are at their most vulnerable whilst in a state of ICT transition. As an SME evolves its ICT services it is often faced with the problems of a lack of knowledge and skills, and concerns relating to ICT security. Milne and McCarthy (2005, 2006) demonstrated that this transition could be achieved and managed through the location of the services provided. These services could be locally based within the SME or provided by an ASP (application service provider). Case studies undertaken demonstrated that the relationship between the ICT provider and the SME is important to the success of the ICT services as well as addressing security. Developing this work we now look at security in terms of the functionality of the ICT systems. This functionality could determine the security needs of the SME and the relationship strategies that could be adopted by ICT providers. Our case studies were continued in the development of this work and give a deeper and richer insight into ICT security.

6. The business system carries the risk

Businesses benefit from using ICT only if the functionality provided by the technology gives direct or indirect advantages over alternative methods. Thus individual applications combine to satisfy functional needs of the business. Most security policies examine parts of the security problem. They often look at software and consider how it can be exploited, but software forms part of the businesses system and it is the system that may carry the risk. Thus ICT providers could address security polices not in terms of each individual applications but business functionality. An ICT provider in the role of a trusted agent has the perspective within an SME to facilitate ICT security across a range of applications and services. This is often not possible due to ICT providers competing with each other at the expense of the SME.

An SSP (system service provider) has an overall perspective which allows for business functionality to be addressed as a primary concern in terms of ICT security. As an SSP it is where these functions sit and hence the location of the applications which we can use to design suitable security models for each business. These functional needs also have different levels of exposure to users of the systems (employees, customers and suppliers) and thus create policies according to the SME's relationship with them. From these models appropriate relationship strategies can be developed by the ICT provider to enhance ICT security.

Taking this approach when looking at the business system we can consider that all systems provide a degree of functionality and we should consider who controls this functionality. Clearly the code of the operating system provides functionality and with it the onus for the writer/vendor to provide a suitable level of security. The same can be said of the developer/vendor of the web servers through to the database server as well as the end user applications. If we look on the ICT solution as a single system/entity which is often how it is perceived by the SME owner/manager we will see that for security purposes many people have responsibility for individual components but we need to know who is responsible for the whole system? The obvious answer would seem to be the manager/owner of the business. In many SME's this is simply not practical as it is beyond the scope of many owner/managers. This is where the role of the ICT provider can be of benefit and could utilise an appropriate relationship strategy with the SME owner/manager. In order to achieve this and to develop a relevant relationship strategy we need to consider where the systems functionality sits and who is responsible for each functional element. In simple terms we can consider it as in Figure 1where we see that any security solution will depend not just upon the risk but the functionality within the systems and the location of the service. This will indicate the responsibility of the ICT provider and the SME and model their relationship.

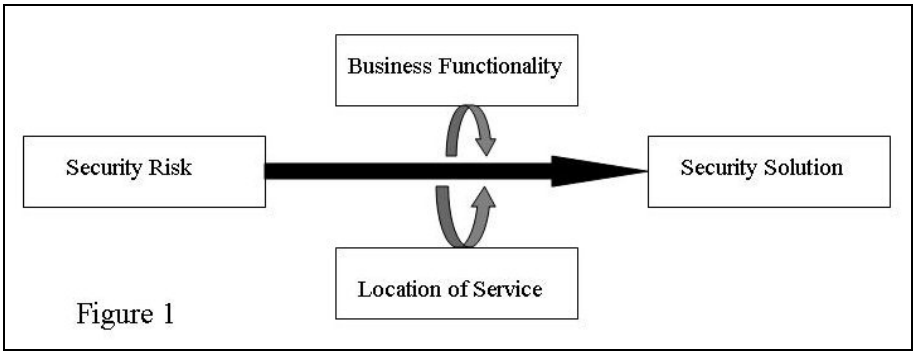


Figure1: Relationship of functionality and location of service to risk

Each application may be required in a number of functions and whilst we need to protect the individual application it is the use of that application in providing the required functionality of the system which we need to protect. We now need to model the functionality of the business in particular the use of ICT to create e-business solutions.

The academic literature contains a number of models that have been developed to describe the use of technology within organisations based upon the functions required. For the purpose of this and previous studies, a framework based upon the IBM model is utilised which gives focus to strategic partnerships for e-business adoption in SME's Stone(2004). It involves six states, each of which relies upon the previous existing state before adoption of the next. The parameters of this model are as follows:

1. **Access:** Individuals have access to the Internet, usually just for e-mail. A very simple Web site may be maintained.
2. **Publish:** "Static" multi-page Web site for information purposes only.
3. **Transact**: The Web site supports one- or two-way transactions with the end user, but does also support fully integrated processing of transactions.
4. **Integrate internally:** Use of e-business to transform internal organization and processes, radically reducing costs and optimizing the business.
5. **Integrate externally:** Use of e-business to create a seamless, transparent process across the entire value chain, from customer through intermediary through all supplier tiers.
6. **Adapt dynamically:** Use of Internet technologies as the foundation for operating in a digital, virtual community. For example, a fully e-integrated institution can seamlessly in-source or out-source processing functions as demanded by market conditions and changing company structures.

7. e-Business architecture

E-business in many of its forms requires specialist computer architecture. This is coupled with an appropriate operating system and other requirements enable e-business. These are:

- A strategic network architecture
- An Internet connection
- A server with the required hardware capabilities
- An operating system on the server that is capable of collating and managing requests from the internal and external network.
- An RDMS (relational database management system).
- A series of protocols and computer languages capable of processing the requests made to the operating system. Examples of these protocols are SSH, FTP RSYNC. Examples of the computer languages required are SQL, HTML, and PHP.
- An operating system capable of managing users' levels in the system in order to establish an effective hierarchy for management and control of users and processes.
- An effective set of security protocols to enable the safe and secure transfer and management of data. These processes and infrastructure once in place allow the owner to participate in hosting and control of B2B (business to business) and B2C (business to customer) applications.

8. SME ICT services and associated security risks

Managing some ICT security risks are beyond the skills of the SME as a single ICT provider. One example is the operating system, being common to many users across the internet and thus having common areas of risk of exploits for all users. The solution is for auto update direct from the OS vendor, which should remove the onus for security of this level from the inexpert end user. Likewise generic applications, with exploitable code errors, should be the responsibility of the provider. Bespoke applications will be more difficult to update depending upon the contract/relationship with the developer but as such applications are specific to individual companies or may be subgroups within a single industry they are less of a target

Table 1: Responsibility analysis

	Network provider (ISP)	Systems provider(SSP)	Applications vendor	In house support/developer
Web site operating system	X			
Workstation operating Systems				X
Web server email server	X			
Generic applications	X	X		
Anti-virus	X	X		X
Industry specific applications			X	
Data backup	X	X		X
Organisation specific applications			X	X

Table 1 shows the spread of responsibility for exploitable risks within the software. The problem is co-ordination of the response and the problems of responsibility for configuration issues. Whilst this is just a guide, all organisations should review their own position in relation to their requirements. It is clearly of benefit to the micro SME if they are able to buy in a generic package of core systems services complete with security; such a package could be provided by an ISP with expanded functionality, a Systems Service Provider (SSP). This would leave any industry specific software to be provide by a specialist Applications vendor or developed in-house from the generic application tools provide by the SSP.

If we look at the six internet states and examine what services we expect to run on them and allocate responsibility between the Systems Service Provider (SSP) and the local management of the SME. There is clear responsibility for some functions, such as email provision, where the SSP provides multiple email accounts for the user, which we would expect in most cases and certainly for Internet States 1-4 to be the responsibility of the SSP. On the other hand, for a dynamic website hosted on the SSP, it would be expected that backups and protection from Denial of Service attacks would be in the hands of the SSP, whilst public and user access rights to contents would be the responsibility of the SME. Clearly the SME would be advised to have their own additional backup of all data and scripts.

Table 2: Responsibility related to the six internet states

		Location of Services				
States	Applications	Email	Web browsing	Web site	RDMS	Email
1	Email, Hosted basic website	SSP	Local	SSP		SSP
2	Email, Complex static website	SSP	Local	SSP		SSP
3	Email, dynamic transaction processing website with shop/sales functionality	SSP	Local	SSP	SSP	SSP
4	Email, web shop presence linked to internal systems	SSP	Local	SSP	SSP/ Local	SSP
5	Email, website linked to business suppliers	SSP	Local	SSP	SSP/ Local	SSP
6	Email, website linked forward to customers and backward to suppliers	SSP	Local	SSP	SSP/ Local	SSP

From table 2 we can see that by moving the position of the services the responsibility for security will change. The location of services in any distributed system is flexible and thus we could consider concentrating services in the most secure areas according the needs of the individual SME. This gives us the possibility of providing security as part of the service provider's package. This will also mean that some key generic functionality may also be provided so that our SSP provides a range of additional services allowing the SME some flexibility in the selection and design of the final application. This can be offered by using generic Content Management Systems and web-shop applications which the end user can then

configure to their own needs. In this way they provide a set of services which gives the user a system which they can then manage to suit their own needs. This empowers the SME to develop its own systems by removing complex installation and configuration and allowing it to manage the flow and storage of data within its system.

Figure 2 shows details of an SME at one end of a spectrum where all the services required are provided by an SSP. In this scenario it is viable for an SME to reach at least Internet State 5 with very basic in house ICT provision. This confirms the work of Daniel et al.(2002) who suggests that e-business growth, like SME growth generally, tends to be non linear and does not conform to a step-by-step approach. Some SME's are now behaving in an 'instant international' or 'born global' sense, where the stage-by-stage progression is now redundant Knight and Cavusgil (1996); McAuley (1999). An SSP could greatly assist this process in those SME's without the appropriate skills and ICT infrastructure.

The SSP could provide a range of services including:

- e-commerce,
- credit card transacting,
- use of delivery agents for goods
- CRM (Customer Relationship Management) systems that use newsletters and SMS messaging.
- Integrated advertising systems with its sponsors
- virtual offices for document storage and management

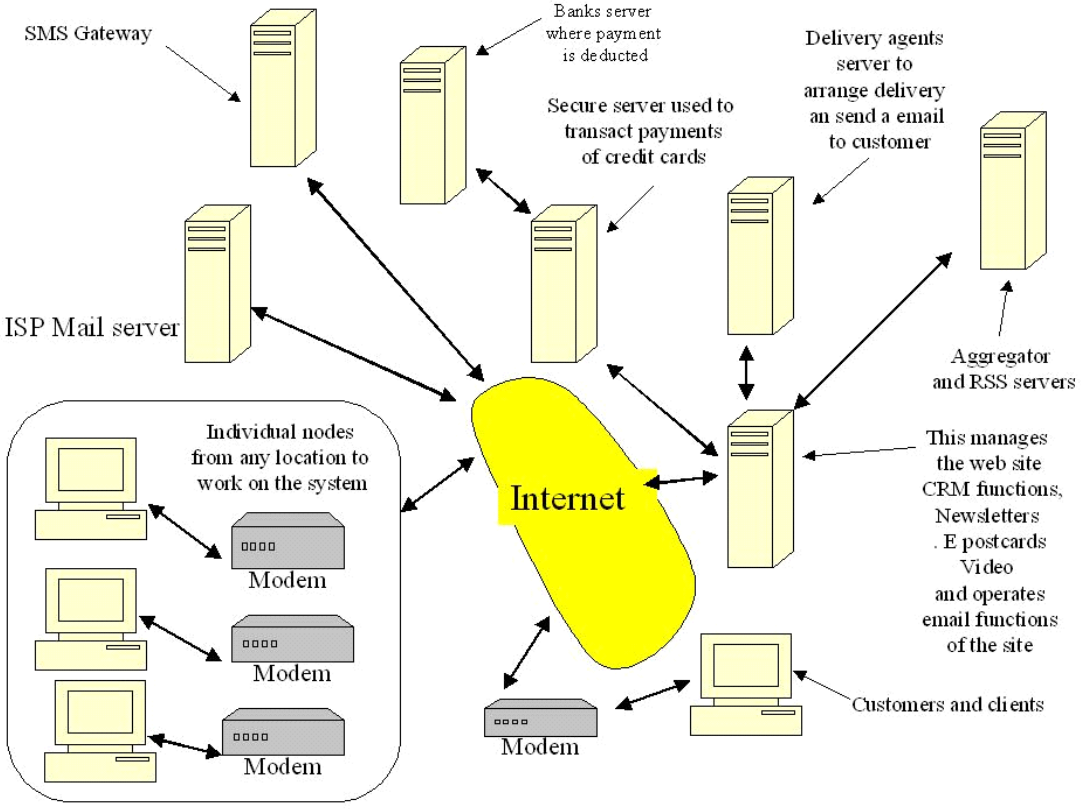


Figure 2: Showing internet state 5 capability

8.1 Security and the internet states model

The six internet states will each require 2 security models from the security specifications at each level for both the SME and the ISP/SSP. Each model will reflect the business functionality, supported by the applications used and the location of these services between the SME and the provider. Case studies undertaken in this work and previous studies McCarthy, Milne (2005) & Milne, McCarthy (2006) demonstrate

that the location of services is unimportant to the business's functionality so long as the SME has access to the data and the ability to process it as and when required.

In all internet states the functionality required can be provided by the network provider (ISP/SSP). It then leaves the growth path and manageability to be a decision by the SME owner manager who can select the location of services and to move through the internet states. Our case studies have shown that in smaller SME's the location of services is often preferred to be in the realm of the SSP. Also by removing the location of the business functionality from the SME's we also remove much of the security needs that are often alien and difficult for the SME to manage for reasons suggested in previous work Milne, McCarthy (2006).

We have also shown that the success of managing the introduction and expansion of ICT within SME's depends upon the relationship between the SME owner/manager and the IT provider. This relationship can in part be measured by the trust between the two parties and hence the degree of responsibility which the SME owner/manager will hand over to the IT provider. With sufficient trust between the two parties the IT provider can deliver a range of generic configurable applications (e.g. content management systems) for the SME to manage the functional use, without concern of the configuration and installation and hence the security. This empowerment of the SME by the IT provider shows how the trust relationship creates a new position for the ICT provider within the SME's extended enterprise. This can only happen if the IT provider is willing to and sees the potential of the relationship in a different way to existing ISP's who simply sell connectivity and ASP's who wish to sell a business solution. The new organisation provides a modifiable 'systems solution' where the two parties work together, hence the need for a new descriptive title of Systems Service Provider.

Table 3: Proposed model showing system service providers provision of applications and support.

Role at outset of studies	Role at the end of the studies	State	Applications
ISP	SSP	1, 2 access, publish	Static web sites , email telephone
ASP	SSP and ASP Specialist applications were used by one study integrated with the SSP)	3- 4 transact – integrate internally	CMS , e-commerce Wan stock control systems, WAN accounting packages
NA	SSP	5 Integrate externally	Use of SSP's to manage interactions with large organisations for vertical integration
NA	SSP	6 Digital institution	Fully integrated in and outsourcing

An SSP can and will provide solutions at all levels. Table 3 shows where the initial location/responsibility was at the start of the case studies and where this relocated to by the end of the studies, allowing the companies to migrate through the internet state levels. Our case studies have shown that the most critical time for an SME in terms of ICT security is the transition from one internet state to the next. This is when SSP's hold real value in the relationship they have with the SME. Often the ICT industry limits itself with a set of packages that perform functions, what is needed is an understanding of the evolution and growth of an SME through the levels. This understanding allows for the SSP and SME relationship to develop along side growth in ICT benefiting both partners.

9. Conclusion

This paper has attempted to highlight the relationship between an SME and any organisation providing it with IT services. We have shown that this relationship is not the same as that which traditionally exists in the corporate sector. A corporate entity has a clear sense of how to plan and manage each stage of deployment of ICT within its operation and has individual specialists to undertake this, SME's often do not have this degree of specialisation.

By providing the installation management and hence security of the system, the SSP leaves the SME with the web based equivalent of the desktop office applications package. They only need to configure the systems to meet their business management needs whilst the functional systems needs are handled by the SSP. We have shown in Milne and McCarthy (2005) that this can be achieved with successful results to internet state 4 even with very small and micro SME's. Further development work in this area will give consideration to the relationship at each internet state between the SSP and SME and go on to further model the services required for each of the internet states and how the responsibility for security is managed.

References

Arbore and Ordanini Broadband Divide Among SMEs The Role of Size, Location and Outsourcing Strategies Bocconi University, Italy 2006

Bolton, J. E. (1971) *Report of a Committee of Enquiry on Small Firms*. London: HMSO.

Chaston, I. (2001) 'The Internet and E-Commerce: An Opportunity to Examine Organisational

Learning in Progress in Small Manufacturing Firms?', International Small *Business Journal* 19(2): 13–30.

Cohen,W. M. and Levinthal, D. A. (1990) 'Absorptive Capacity: A New Perspective on Learning and Innovation', Administrative Science Quarterly 35(1): 128–52.

Colombo, M. G. (2001) 'Technology-Based Entrepreneurs: Does Internet Make a Difference?', *Small Business Economics* 16(3): 177–90.

Corrocher, N. and Ordanini, A. (2002) 'Measuring the Digital Divide: A Framework for the Analysis of Cross-countries Differences', Journal of Information Technology 17(1): 9–19.

Dandridge, T. (2000) 'High-Tech Potential? An Exploratory Study of Very Small Firms' Usage of the Internet', *International Small Business Journal* 18(2): 81–91.

Daniel, E. M. and Grimshaw, D. J. (2002) 'An Exploratory Comparison of Electronic Commerce Adoption in Large and Small Enterprises', Journal of Information Technology 17(3): 133–47.

Dixon, T., Thompson, B., and McAllister, P. 2003, The value of ICT for SME's in the UK: A Critical Literature Review. Report for Small Business Service Research Programme, The College of Estate Management.

Drew, S. (2002) 'E-Business Research Practice: Towards an Agenda', Electronic Journal on

Business Research Methods 1(1): 18–26.

DTI Benchmarking Study 2004, 2005

EURIM Policy Document 2005 The scale and nature of computer assisted crime

EURIM 2004 -IPRE E-Crime Study "Addressing the needs of Small Firms and others without in-house ICT or security skills"

Fillis I and Wagner B University of Stirling, UK International Small Business Journal Copyright © 2005 SAGE Publications (London, Thousand Oaks and New Delhi) http://isb.sagepub.com [DOI: 10.1177/0266242605057655] Vol 23(6): 604–634

Fuller, T and Lewis, J.2003 Putting the "R" in E-Business; Relationships matter to technology choices. ICSB World Conference, Belfast 2003

Knight, G. and Cavusgil, S. T. (1996) 'The Born Global Firm: A Challenge to Traditional Internationalisation Theory', in S. T. Cavusgil and T. K. Madsen (eds) Advances in International *Marketing, 8*, pp. 11–26. London: JAI Press.

McAuley, A. (1999) 'Entrepreneurial Instant Exporters in the Scottish Arts and Crafts Sector', *Journal of International Marketing* 7(4): 67–82.

Milne, D. and McCarthy, J. (2006) 'Security in an Uncertain World SME's and a Level Playing Field', Proceedings of the International Conference on Information Warfare and Security 2006

McCarthy, J. and Milne, D. (2005) 'ASP's and SME's The new Trusted Agents', Proceedings of the Institute for Small Business and Entrepreneurship Conference 2005

NCC (2000) The businesses information security survey 2000. national computing centre. http://www.ncc.co.uk/ncc

NHTCU 2004 The Risk of Computer Crime to the SME. What your business really needs to know. An Exploratory Investigation of the Small Firm

Ramsey, E., Ibbotson, P., Bell, J. and Gray, B. (2003) 'E-Opportunities of Service Sector SMEs: An Irish Cross-Border Study', Journal of Small Business and Enterprise Development 10(3): 250–64.

Small Business Service Research Programme (2002) http://www.sbs.gov.uk/sbsgov/action/layer?topicId=7000000237 accesed january 2004

Stone Merlin, 2004, SME e-business and customer supplier relations, Journal of Small Business and Enterprise Development Volume 10 Number 3 2003 pp. 345-353

Webb, B. and Sayer, R. (1998) 'Benchmarking Small Companies on the Internet', Long *Range Planning* 31(6): 815–27.