CSCI E-11

Class 5

February 24, 2026

The Building Blocks of Security

Harvard
Extension School
HARVARD DIVISION OF
CONTINUING EDUCATION

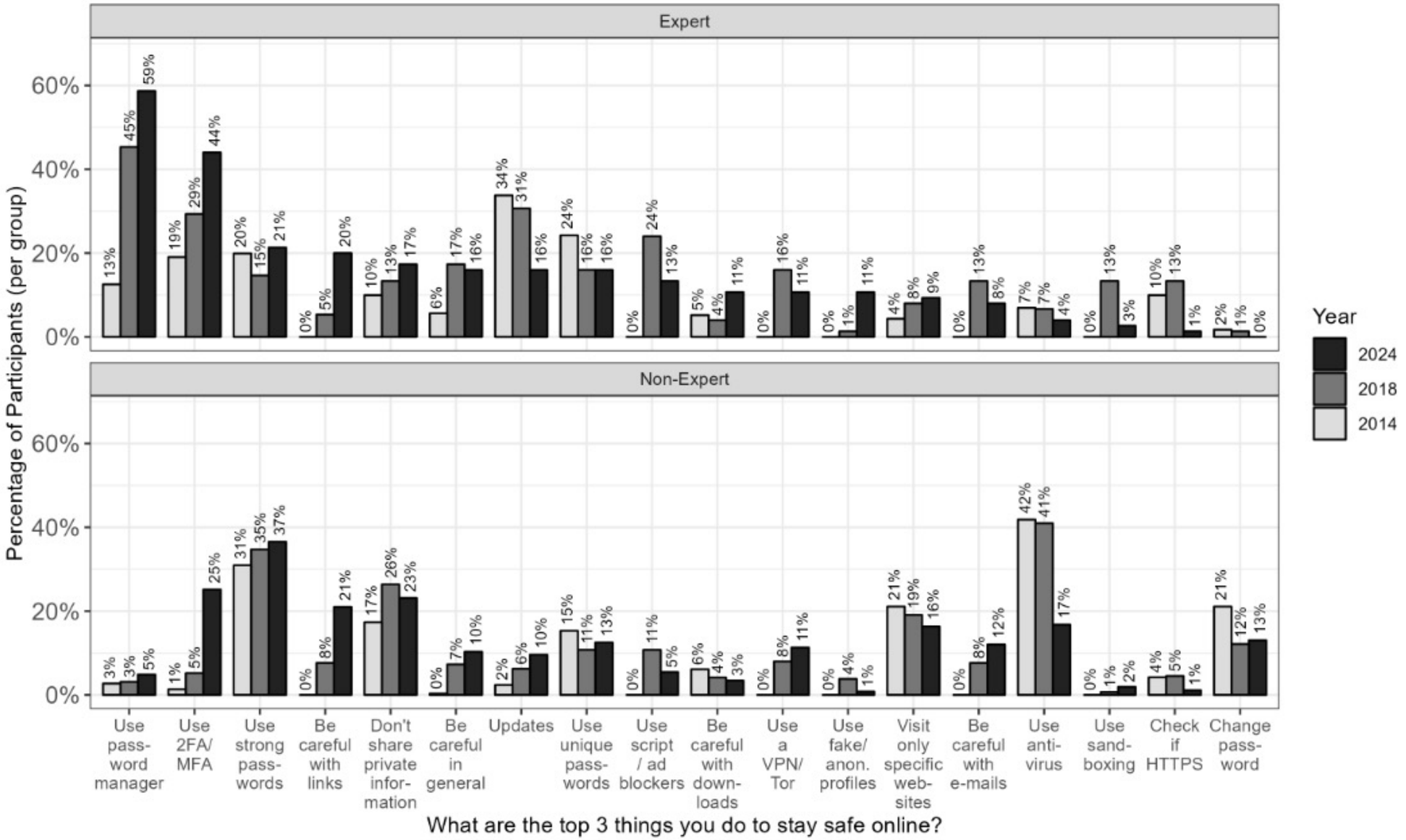
Another Replication: No One Can Hack My Mind, Part III

Replication: “No one can hack my mind” - 10 years later: An update and outlook on experts’ and non-experts’ security practices and advice

Anna-Marie Ortloff, *University of Bonn;* Jenny Tang, *Carnegie Mellon University;* Arthi Arumugam, Daniel Huschina, Lisa Geierhaas, and Florin Martius, *University of Bonn;* Luisa Jansen, *University of Bern;* Kolja von der Twer and Lilly Jungbluth, *University of Bonn;* Matthew Smith, *University of Bonn and Fraunhofer FKIE*

https://www.usenix.org/conference/soups2025/presentation/ortloff

Highlights of the 10-year replication study

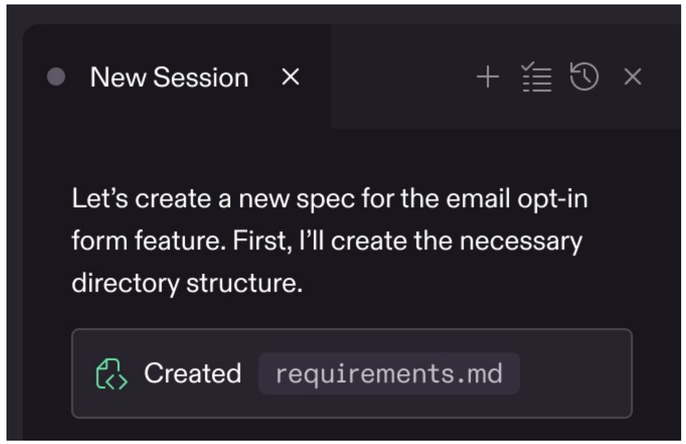


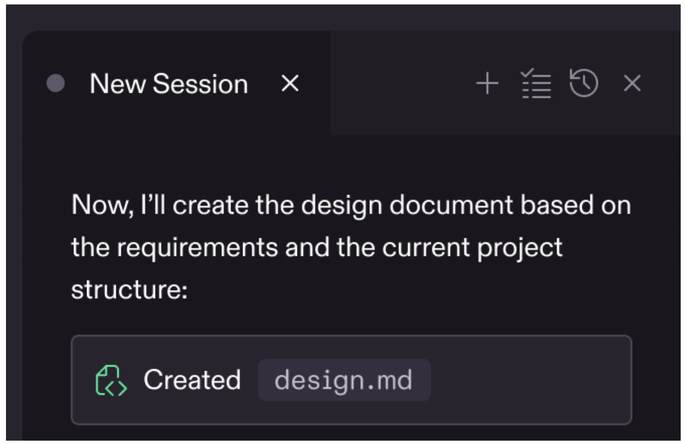
Class Updates

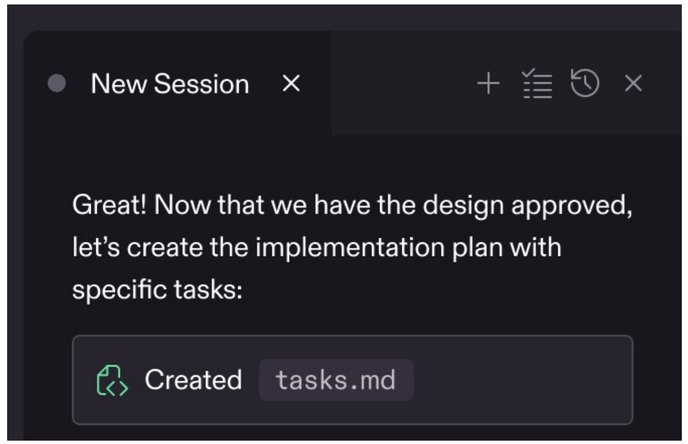
How to report an error

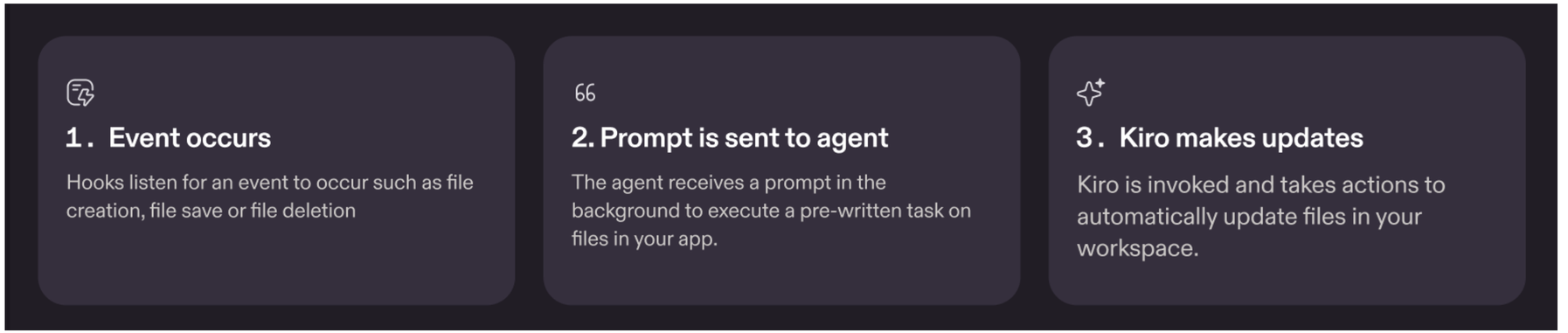
- What configuration are you running
- Your location/system
- What commands did you run (exactly, all of them)
- What error messages did you get (exactly, all of them)
- Is the problem persistent, repeatable, intermittent?
- Attach code, screenshots

Amazon uses Kiro, a “vibe-coding” tool









AI coding tool brings down Amazon services. Amazon calls it “user error.”

AWS AI coding tool decided to "delete and recreate" a customer-facing system, causing 13-hour outage, report says

Maximilian Schreiner Feb 20, 2026

- AWS suffered at least two outages in recent months after engineers let AI coding tools Kiro and Amazon Q Developer make autonomous changes without oversight, according to a report by the Financial Times.
- Amazon calls it "user error" due to misconfigured access controls, not an AI autonomy issue, and describes the incidents as extremely limited.
- However, the AI tools reportedly had operator-level permissions with no mandatory peer review — safeguards that AWS only introduced after the incidents.

Batman Hacked My Password

Batman Hacked My Password:
A Subtitle-Based Analysis of Password Depiction in Movies

Maike M. Raphael
Leibniz University Hannover

Aikaterini Kanta
University of Portsmouth

Rico Seebonn
Leibniz University Hannover

Markus Dürmuth
Leibniz University Hannover

Camille Cobb
University of Illinois Urbana-Champaign

Abstract

Password security is and will likely remain an issue that non-experts have to deal with. It is therefore important that they understand the criteria of secure passwords and the characteristics of good password behavior. Related literature indicates that people often acquire knowledge from media such as movies, which influences their perceptions about cybersecurity including their mindset about passwords. We contribute a novel approach based on subtitles and an analysis of the depiction of passwords and password behavior in movies. We scanned subtitles of 97,709 movies from 1960 to 2022 for password appearance and analyzed resulting scenes from 2,851 movies using mixed methods to show what people could learn from watching movies. Selected films were viewed for an in-depth analysis.

Among other things, we find that passwords are often portrayed as weak and easy to guess, but there are different contexts of use with very strong passwords. Password hacking is frequently depicted as unrealistically powerful, potentially leading to a sense of helplessness and futility of security efforts. In contrast, password guessing is shown as quite realistic and with a lower (but still overestimated) success rate. There appears to be a lack of best practices as password managers and multi-factor authentication are practically non-existent.

1 Introduction

Cybersecurity is a topic that virtually everyone encounters every day, from the first unlocking of the smartphone in the morning to reading emails at work or communicating with friends at night. This requires many decisions about which links to click, which websites to trust or which password to choose. Among other things, these decisions are based on knowledge and beliefs about the subject area in question that determine, for example, what is perceived as "secure" or "insecure" [20, 73]. It is therefore important that this knowledge is correct and beliefs are aligned with reality. However, studies show that for cybersecurity these are often incorrect or incomplete, leading to "bad" security practices [1, 63].

This problem becomes particularly evident in password security, which is an area where many misconceptions are found. Various studies show that people do not know the characteristics of good passwords, do not know how to handle passwords in general or do not remember the recommendation to change the password regularly, which has been proven to be bad advice in recent years [11, 24, 45]. This is a big issue because, despite their weaknesses and numerous alternatives being available, passwords are still by far the most common form of online authentication [26]. It is therefore important that the understanding of password security and good password behavior is reinforced.

One source that influences people's perception of cybersecurity is likely films [54, 73]. Literature shows that people learn from media and use it as a source of information [51, 54]. Films play a major role in this; they have been a popular entertainment medium for decades and are watched by thousands of people every day. Therefore it is hardly surprising that films influence knowledge and behavior [19] or the attitude toward technology [9] and that this may change how people handle specific topics and make decisions. Because people often cannot decide if what they see is realistic or not, they are in danger of taking fictional portrayals as realistic which may influence their thinking about certain topics including cybersecurity [20, 73]. It is therefore important to ensure that things are presented in a good and realistic way so that people potentially learn something *true* from them [12].

The use of certain technologies in movies "both reflects and influences society's use and attitudes toward the portrayed

Copyright is held by the author/owner. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee.
USENIX Symposium on Usable Privacy and Security (SOUPS) 2024.
August 11–13, 2024, Philadelphia, PA, United States.

Themes:

- Media conveys the impossibility of defense/glamorizes hacking
- Always high-value targets
- Passwords create inherent gaps between user behavior and good practice
- Too many password changes makes it impossible to remember

(Some) Building Blocks of Secure Systems

Saltzer & Schroeder — The Protection of Information in Computer Systems (1975)

related to hazard from lasers and other light sources," *Amer. J. Ophthalmol.*, vol. 66, p. 15, 1968.
[57] A. Vassiliadis, H. C. Zweng, N. A. Peppers, R. R. Peabody, and R. C. Honey, "Thresholds of laser eye hazards," *Arch. Environ. Health*, vol. 20, p. 161, 1970.
[58] P. W. Lappin, "Ocular damage thresholds for the helium-neon laser," *Arch. Environ. Health*, vol. 20, p. 177, 1970.
[59] W. T. Ham *et al.*, "Retinal burn thresholds for the He-Ne laser in the rhesus monkey," *Arch. Ophthalmol.*, to be published.
[60] T. P. Davis and W. J. Mautner, "Helium-neon laser effects on the eye," U.S. Army Med. Res. Develop. Com., Washington, D.C., Annu. Rep. Contr. DADA 17-69-C-9013, 1969.
[61] J. J. Vos, "Digital computations of temperature in retinal burn problems," Inst. Perception, Soesterberg, The Netherlands, RVO-TNO, Rep. IZF 1965016, 1963.
[62] M. A. Mainster, T. J. White, J. H. Tips, and P. W. Wilson, "Retinal-temperature increases produced by intense light sources," *J. Opt. Soc. Amer.*, vol. 60, p. 264, 1970.
[63] A. M. Clarke, W. T. Ham, W. J. Geeraets, R. C. Williams, and H. A. Mueller, "Laser effects on the eye," *Arch. Environ. Health*, vol. 18, p. 424, 1969.
[64] R. H. Stern and R. F. Sognnaes, "Laser beam on dental hard tissues," *J. Dent. Res.*, vol. 43, p. 873, 1964.
[65] R. H. Stern, "Dentistry and the laser," in *Laser Applications in Medicine and Biology*, vol. II, Dr. M. L. Wolbarsht, Ed. New York: Plenum, 1974, pp. 361–388.
[66] T. E. Gordon, Jr., and D. L. Smith, "Laser welding of prostheses—an initial report," *J. Prosth. Dent.*, vol. 24, p. 472, 1970.

The Protection of Information in Computer Systems

JEROME H. SALTZER, SENIOR MEMBER, IEEE, AND MICHAEL D. SCHROEDER, MEMBER, IEEE

Invited Paper

***Abstract*—This tutorial paper explores the mechanics of protecting computer-stored information from unauthorized use or modification. It concentrates on those architectural structures—whether hardware or software—that are necessary to support information protection. The paper develops in three main sections. Section I describes desired functions, design principles, and examples of elementary protection and authentication mechanisms. Any reader familiar with computers should find the first section to be reasonably accessible. Section II requires some familiarity with descriptor-based computer architecture. It examines in depth the principles of modern protection architectures and the relation between capability systems and access control list systems, and ends with a brief analysis of protected subsystems and protected objects. The reader who is dismayed by either the prerequisites or the level of detail in the second section may wish to skip to Section III, which reviews the state of the art and current research projects and provides suggestions for further reading.**

GLOSSARY

THE FOLLOWING glossary provides, for reference, brief definitions for several terms as used in this paper in the context of protecting information in computers.

Access	The ability to make use of information stored in a computer system. Used frequently as a verb, to the horror of grammarians.
Access control list	A list of principals that are authorized to have access to some object.
Authenticate	To verify the identity of a person (or other agent external to the protection system) making a request.
Authorize	To grant a principal access to certain information.
Capability	In a computer system, an unforgeable ticket, which when presented can be taken as incontestable proof that the presenter is authorized to have access to the object named in the ticket.
Certify	To check the accuracy, correctness, and completeness of a security or protection mechanism.
Complete isolation	A protection system that separates principals into compartments between which no flow of information or control is possible.
Confinement	Allowing a borrowed program to have access to data, while ensuring that the program cannot release the information.
Descriptor	A protected value which is (or leads to) the physical address of some protected object.
Discretionary	(In contrast with *nondiscretionary*.) Controls on access to an object that may be changed by the creator of the object.
Domain	The set of objects that currently may be directly accessed by a principal.
Encipherment	The (usually) reversible scrambling of data according to a secret transformation key, so as to make it safe for transmission or storage in a physically unprotected environment.
Grant	To authorize (*q.v.*).
Hierarchical control	Referring to ability to change authorization, a scheme in which the record of

Manuscript received October 11, 1974; revised April 17, 1975. Copyright © 1975 by J. H. Saltzer.
The authors are with Project MAC and the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Mass. 02139.

Authorized licensed use limited to: Harvard University SEAS. Downloaded on February 23,2025 at 00:42:59 UTC from IEEE Xplore. Restrictions apply.

Saltzer & Schroeder — The Protection of Information in Computer Systems (1975)

Design principles

The following design principles are laid out in the paper:

- Economy of mechanism: Keep the design as simple and small as possible.
- Fail-safe defaults: Base access decisions on permission rather than exclusion.
- Complete mediation: Every access to every object must be checked for authority.
- Open design: The design should not be secret.
- Separation of privilege: Where feasible, a protection mechanism that requires two keys to unlock it is more robust and flexible than one that allows access to the presenter of only a single key.
- Least privilege: Every program and every user of the system should operate using the least set of privileges necessary to complete the job.
- Least common mechanism: Minimize the amount of mechanism common to more than one user and depended on by all users.
- Psychological acceptability: It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly.
- Work factor: Compare the cost of circumventing the mechanism with the resources of a potential attacker.
- Compromise recording: It is sometimes suggested that mechanisms that reliably record that a compromise of information has occurred can be used in place of more elaborate mechanisms that completely prevent loss.

https://en.wikipedia.org/wiki/The_Protection_of_Information_in_Computer_Systems

OWASP Top Ten 2025

1. A01:2025 - Broken Access Control
2. A02:2025 - Security Misconfiguration
3. A03:2025 - Software Supply Chain Failures
4. A04:2025 - Cryptographic Failures
5. A05:2025 - Injection
6. A06:2025 - Insecure Design
7. A07:2025 - Authentication Failures
8. A08:2025 - Software or Data Integrity Failures
9. A09:2025 - Security Logging and Alerting Failures
10. A10:2025 - Mishandling of Exceptional Conditions

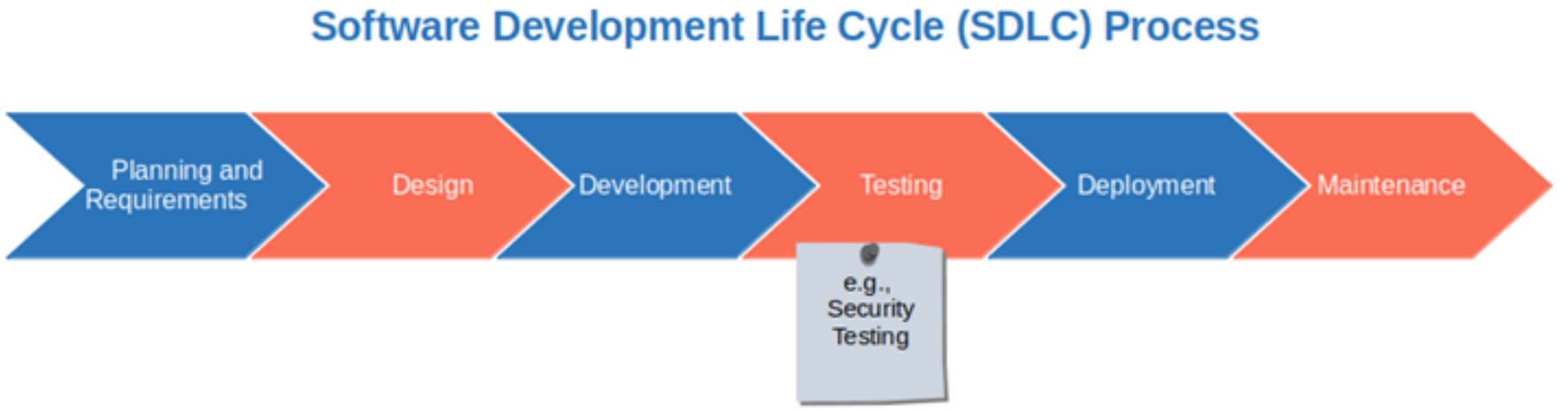
https://owasp.org/Top10/2025/

Insecure design vs. insecure implementation

"Note that there is a difference between *insecure design* and *insecure implementation*. We differentiate between design flaws and implementation defects for a reason, they have different root causes, take place at different times in the development process, and have different remediations. A secure design can still have implementation defects leading to vulnerabilities that may be exploited. **An insecure design cannot be fixed by a perfect implementation as needed security controls were never created to defend against specific attacks.**"

https://owasp.org/Top10/2025/A06_2025-Insecure_Design/

Security by Design



Secure By Design, Red Hat’s Version

Defense in depth

Secure by default

Least privilege

Separation of duties

Minimize attack surface

Complete mediation

Open design

Isolated compartments

Evidence production

Application coding best practices

Secure By Design, CISA

1. **Take ownership of customer security outcomes** and evolve products accordingly. The burden of security should not fall solely on the customer.

2. **Embrace radical transparency and accountability.**
Software manufacturers should pride themselves in delivering safe and secure products, as well as differentiating themselves from the rest of the manufacturer community based on their ability to do so. This may include sharing information they learn from their customer deployments, such as the uptake of strong authentication mechanisms by default. It also includes a strong commitment to ensure vulnerability advisories and associated common vulnerability and exposure (CVE) records are complete and accurate. However, beware of the temptation to count CVEs as a negative metric, since such numbers are also a sign of a healthy code analysis and testing community.

3. **Build organizational structure and leadership to achieve these goals.**
While technical subject matter expertise is critical to product security, senior executives are the primary decision makers for implementing change in an organization. Executives need to prioritize security as a critical element of product development across the organization, and in partnership with customers.

Secure By Design, UK Government Security Group

1. Create responsibility for cyber security risk
2. Source secure technology products
3. Adopt a risk-driven approach
4. Design usable security controls
5. Build in detect and respond security
6. Design flexible architectures
7. Minimise the attack surface
8. Defend in depth
9. Embed continuous assurance
10. Make changes securely

https://www.security.gov.uk/policy-and-guidance/secure-by-design/principles/

Open Security vs. Security through Obscurity

Minimize Attack Surface

https://www.paloaltonetworks.com/cyberpedia/attack-surface-assessment

Microsoft adds exciting new AI-powered features to Notepad!

Enhance Your Writing with AI in Notepad

Notepad in Windows includes Rewrite, Summarize, and Write—three AI-powered features that help you refine, shorten, and generate text with the assistance of GPT. Whether you want to fine-tune your writing or distill complex information, these AI features help you work more efficiently in Notepad.

- Rewrite allows you to improve clarity, adjust tone, or modify the length of your content, making it easy to enhance your writing seamlessly.
- Summarize quickly shortens long-form text into high-level overviews, extracting key points from documents, notes, or articles to streamline your workflow.
- Write enables you to draft new content quickly and effortlessly by entering a prompt, with results tailored to your needs. It can start from a blank page or use selected text as a reference, generating relevant content based on your instructions and surrounding context.

You won’t believe what happened next!

Windows Notepad App Remote Code Execution Vulnerability New Recently updated

CVE-2026-20841
Security Vulnerability

Released: Feb 10, 2026

Last updated: Feb 12, 2026

Assigning CNA: Microsoft

CVE.org link: CVE-2026-20841

Impact: Remote Code Execution **Max Severity:** Important

Weakness: CWE-77: Improper Neutralization of Special Elements used in a Command ('Command Injection')

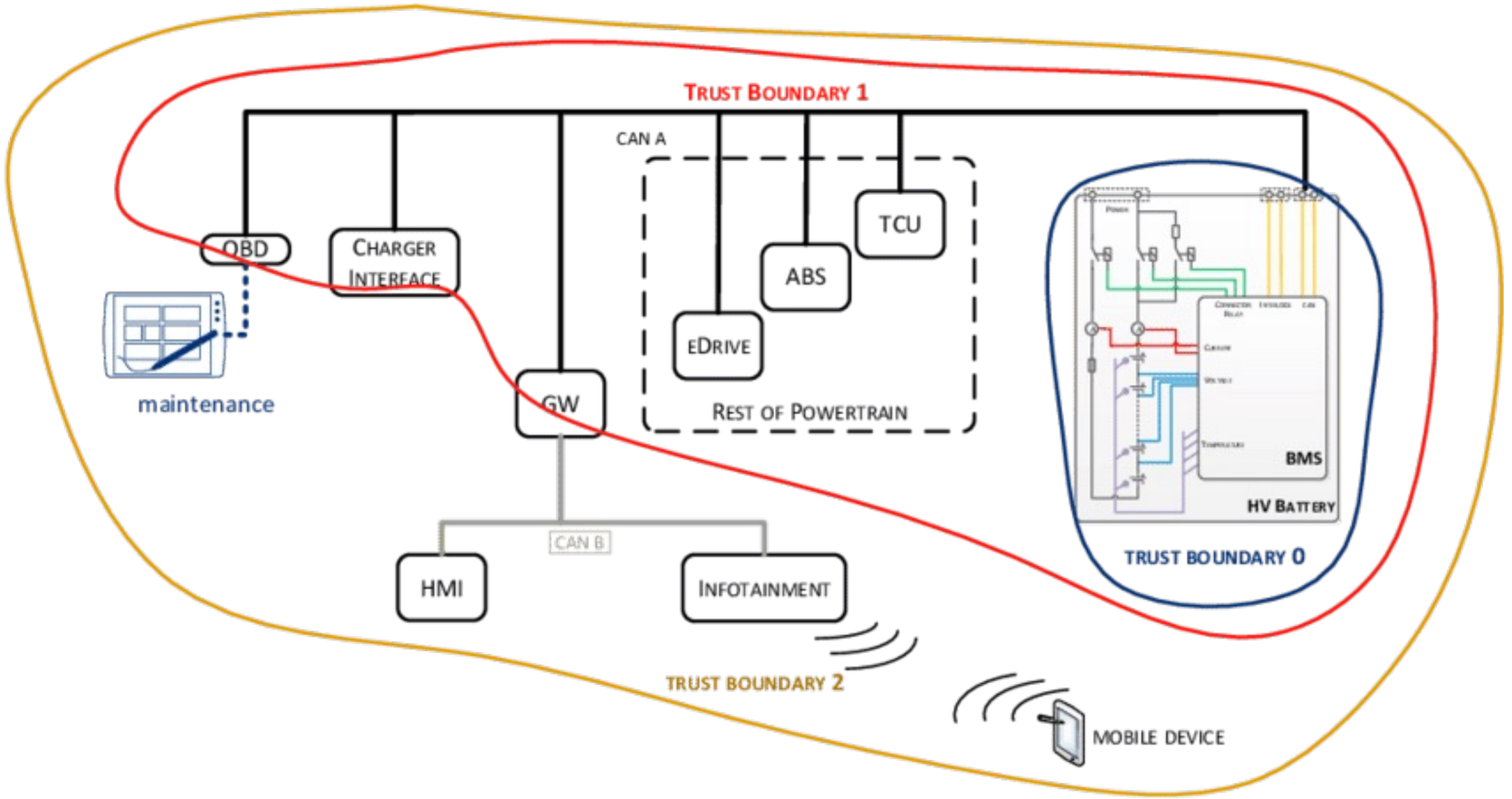
CVSS Source: Microsoft

Vector String: `CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H/E:U/RL:O/RC:C`

Metrics: CVSS:3.1 7.8 / 6.8

https://msrc.microsoft.com/update-guide/vulnerability/CVE-2026-20841

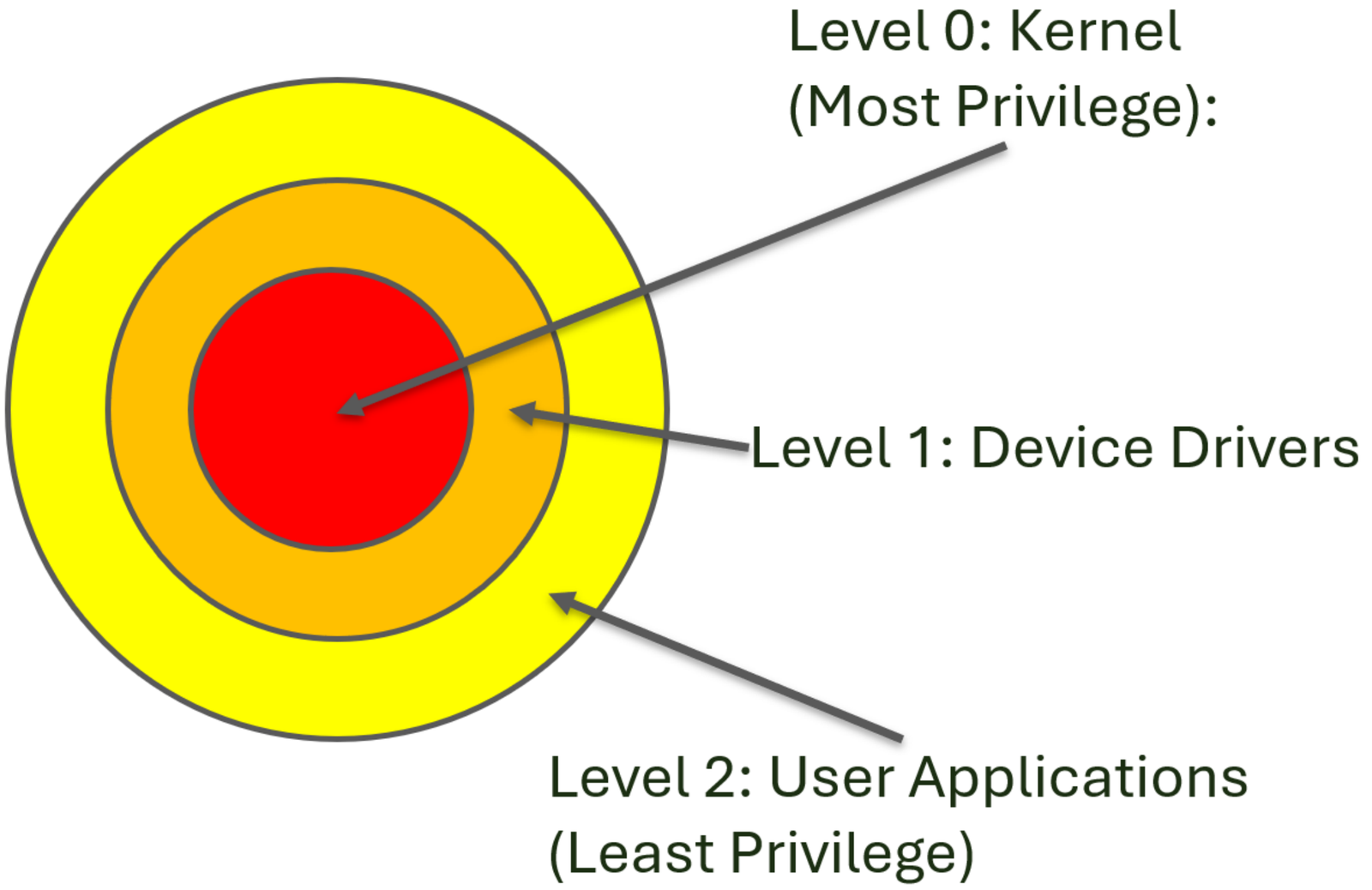
Trust Boundaries



Macher, Georg & Sporer, Harald & Brenner, Eugen & Kreiner, Christian. (2017). An Automotive Signal-Layer Security and Trust-Boundary Identification Approach. Procedia Computer Science. 109. 490-497. 10.1016/j.procs.2017.05.317.

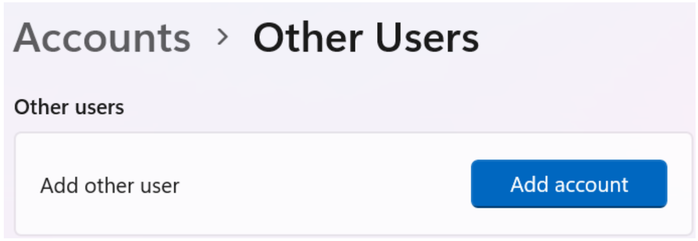
https://www.researchgate.net/publication/317546519_An_Automotive_Signal-Layer_Security_and_Trust-Boundary_Identification_Approach

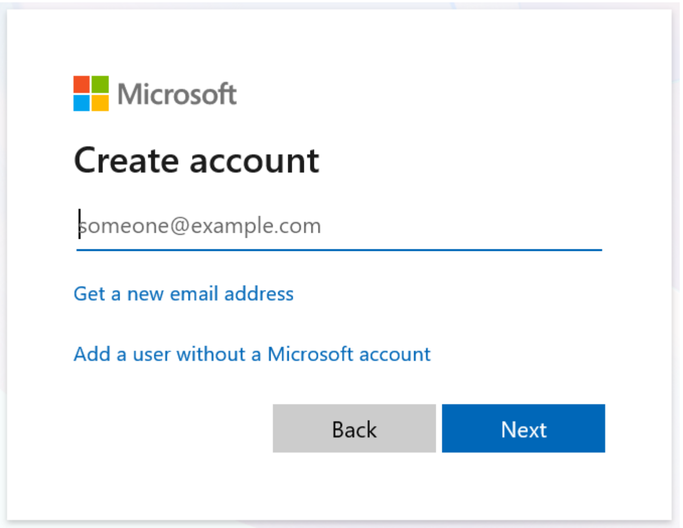
Each outward layer of the system has fewer privileges.

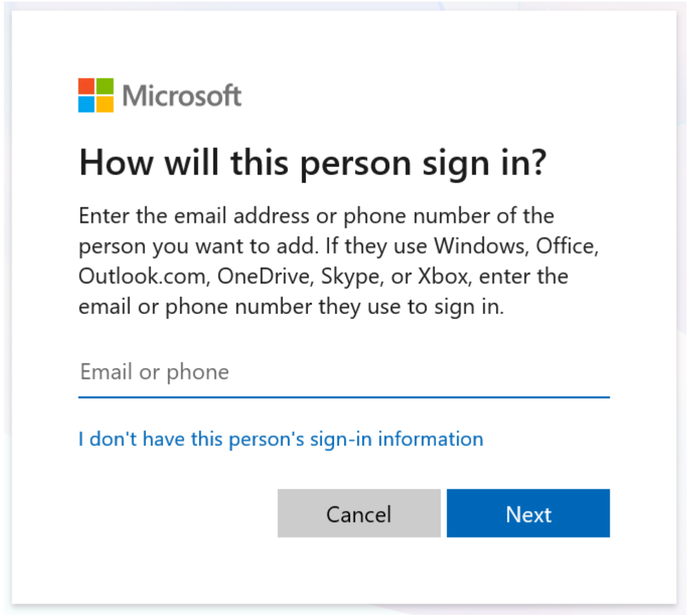


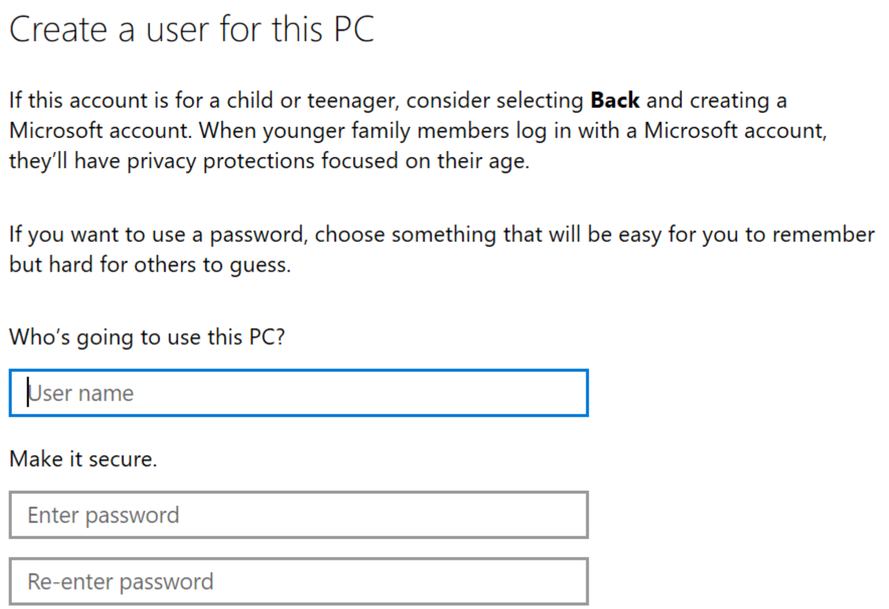
Least Privilege

After the user is authenticated, give them the least necessary access.



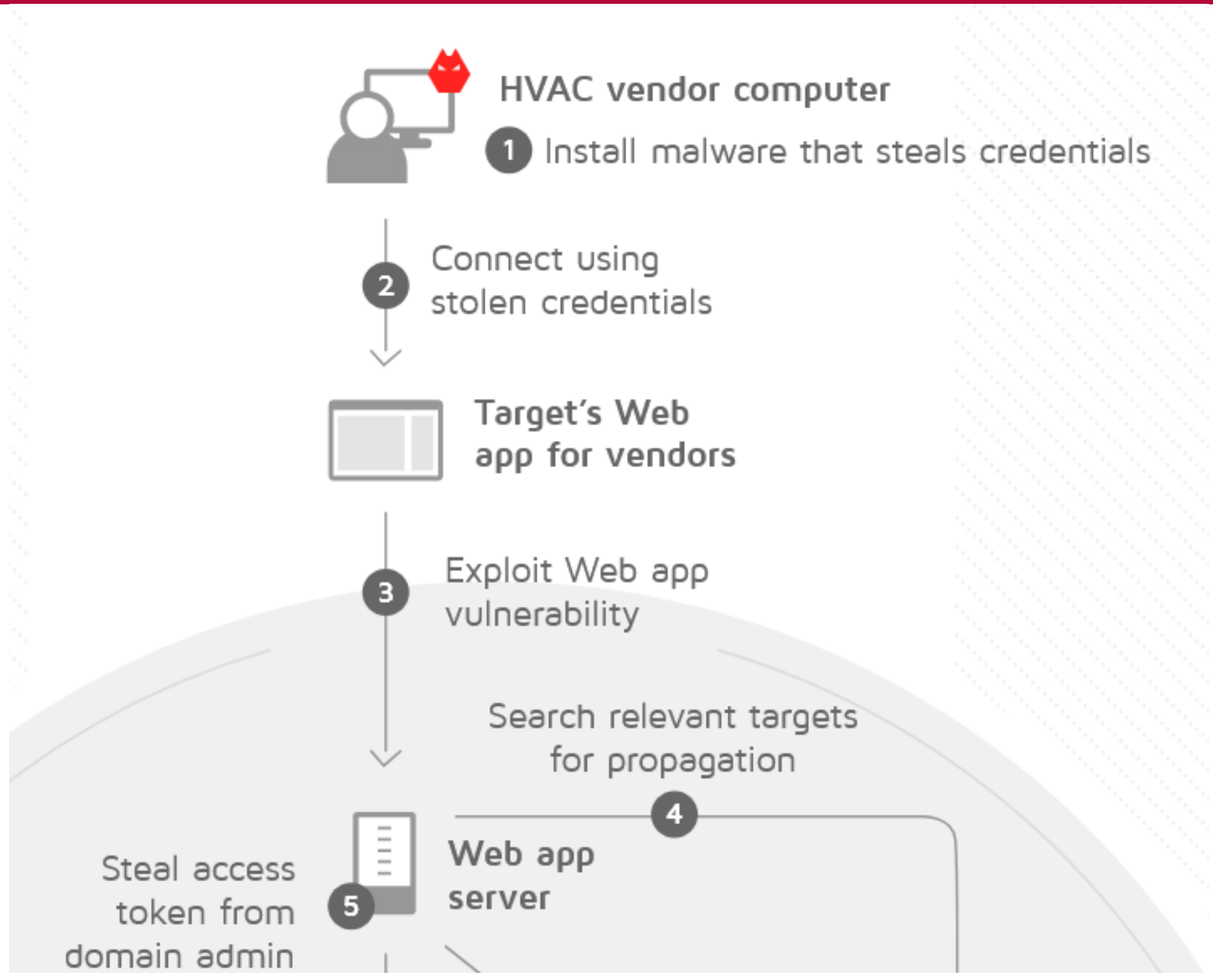






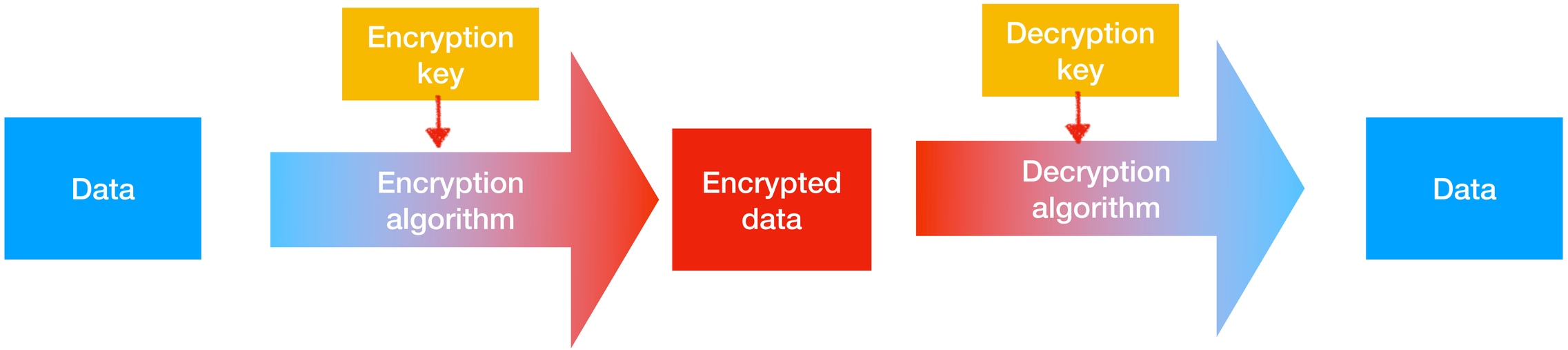
Target hack, 2013

Target attackers' methodology (partial)



https://aroundcyber.wordpress.com/wp-content/uploads/2014/09/aorato-target-report.pdf

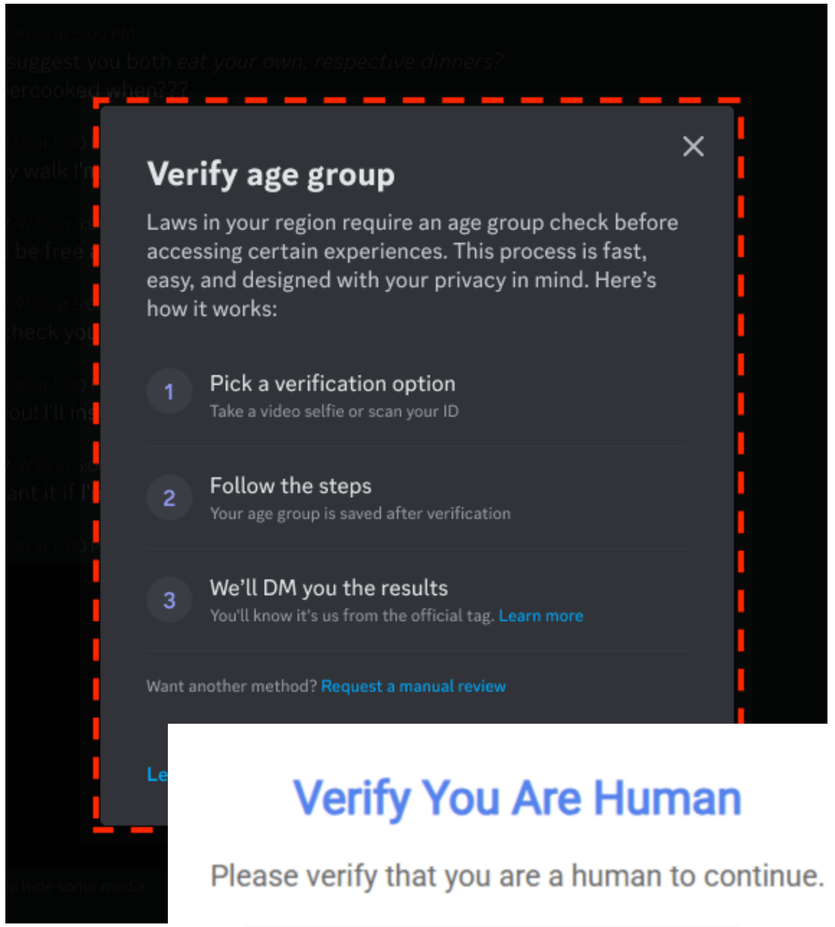
Encryption

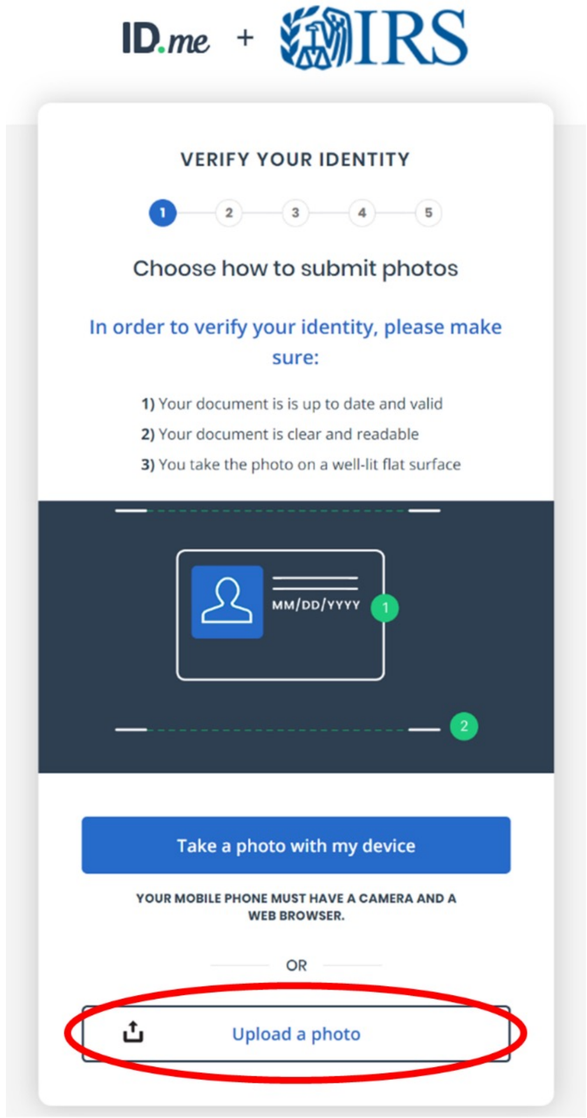


Symmetric encryption - *secret key cryptography* - Encryption & Decryption keys are the *same*

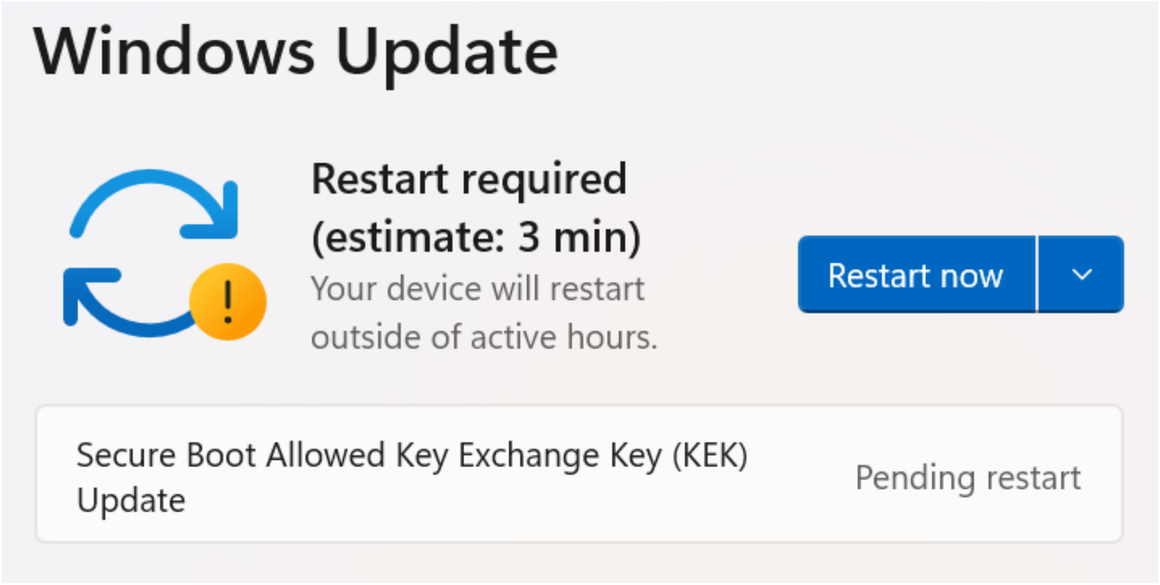
Asymmetric encryption - *public key cryptography* - Encryption & Decryption keys are *different*

Proving (something about) your identity.

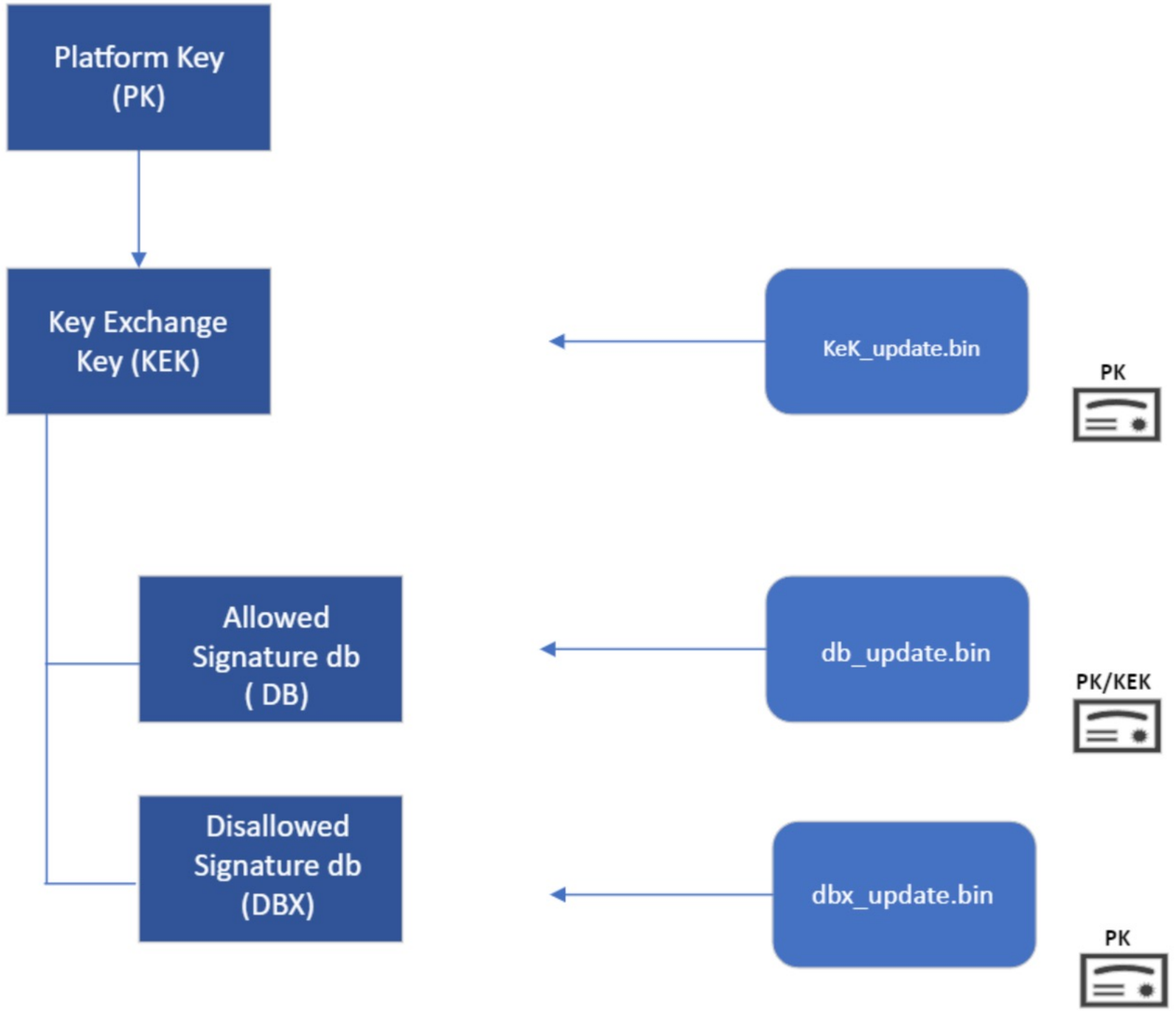




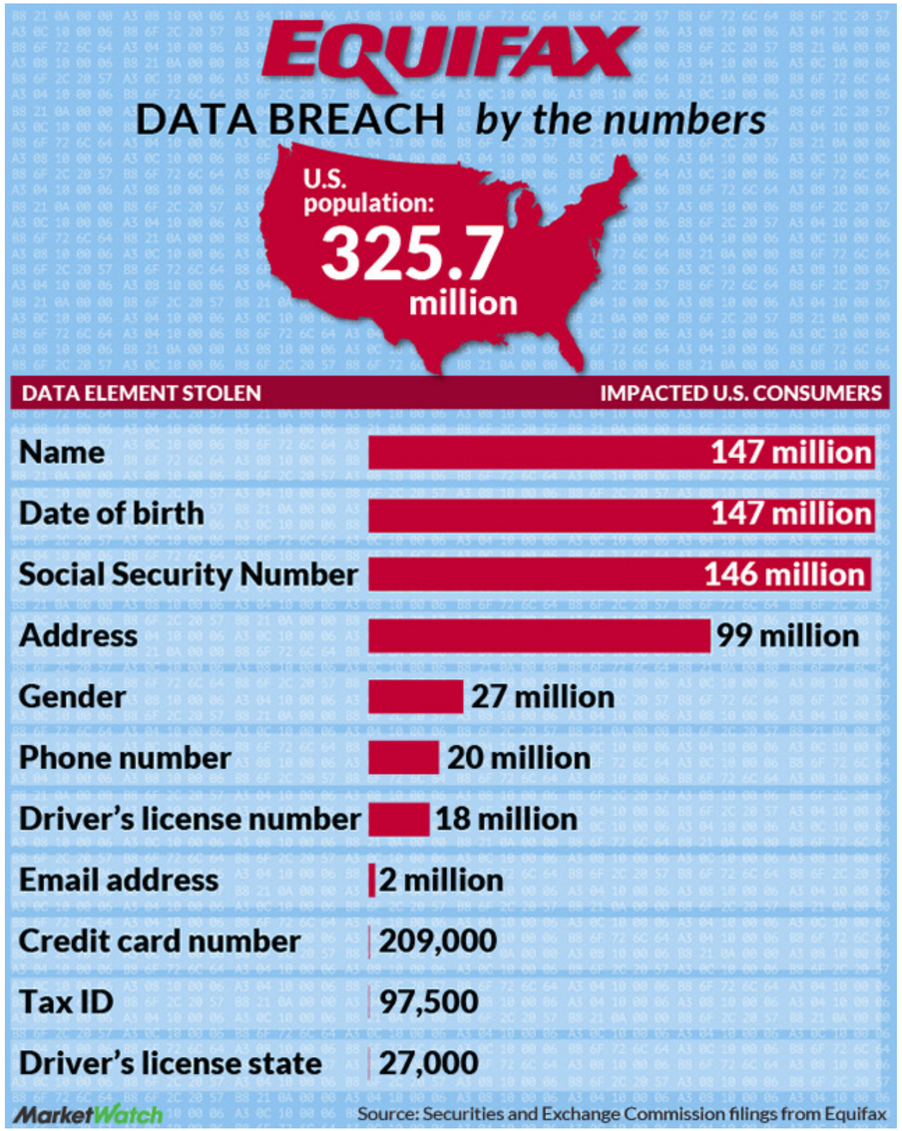
Keep Software Updated



The what now?


Platform Key (PK)
Key Exchange Key (KEK)
Allowed Signature db (DB)
Disallowed Signature db (DBX)
KeK_update.bin
db_update.bin
dbx_update.bin
PK
PK/KEK
PK

Equifax Data Breach, 2017



https://www.marketwatch.com/story/the-equifax-data-breach-in-one-chart-2018-09-07

Formal Methods apply mathematical techniques to system analysis.

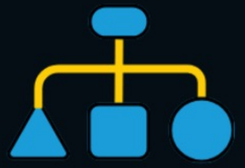
	TRADITIONAL SOFTWARE DEVELOPMENT	FORMAL METHODS	MATHEMATICS
TARGET	Nuclear Reactor	Nuclear Reactor	Right Triangles
OBJECTIVE	Prove Some Functionality (E.g., AutoShutdown if Temp is High)	Prove Some Functionality (E.g., Auto Shutdown if Temp is High)	Prove Pythagorean Theorem Always Holds
		Applied FM Begins	
METHOD	Testing	Formal Proof (Tools & Technology)	Formal Proof (Pen and Paper)
RESULTS	Weak	Strong	Strong

Translating Code to Math

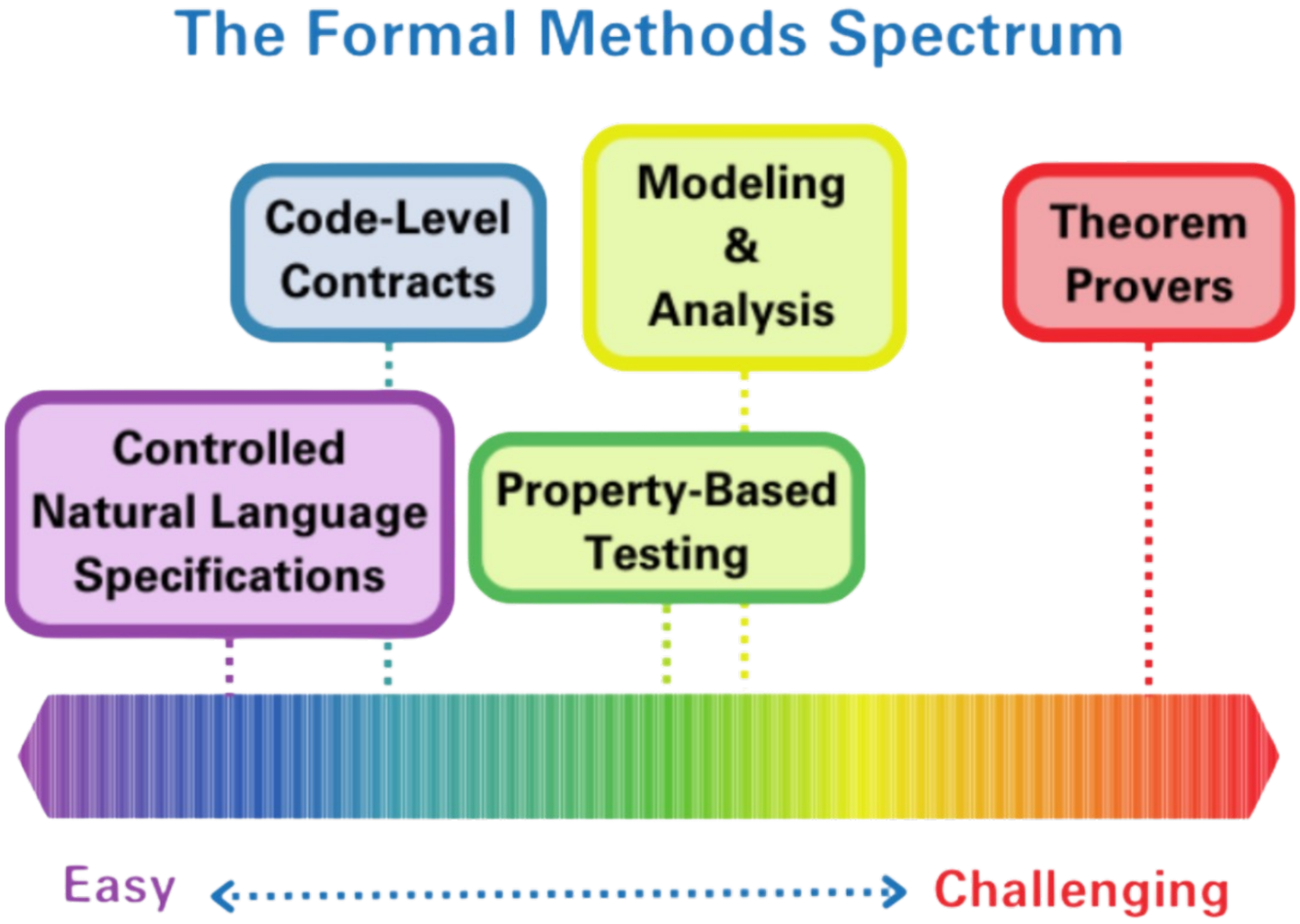
=

Applied Formal Methods

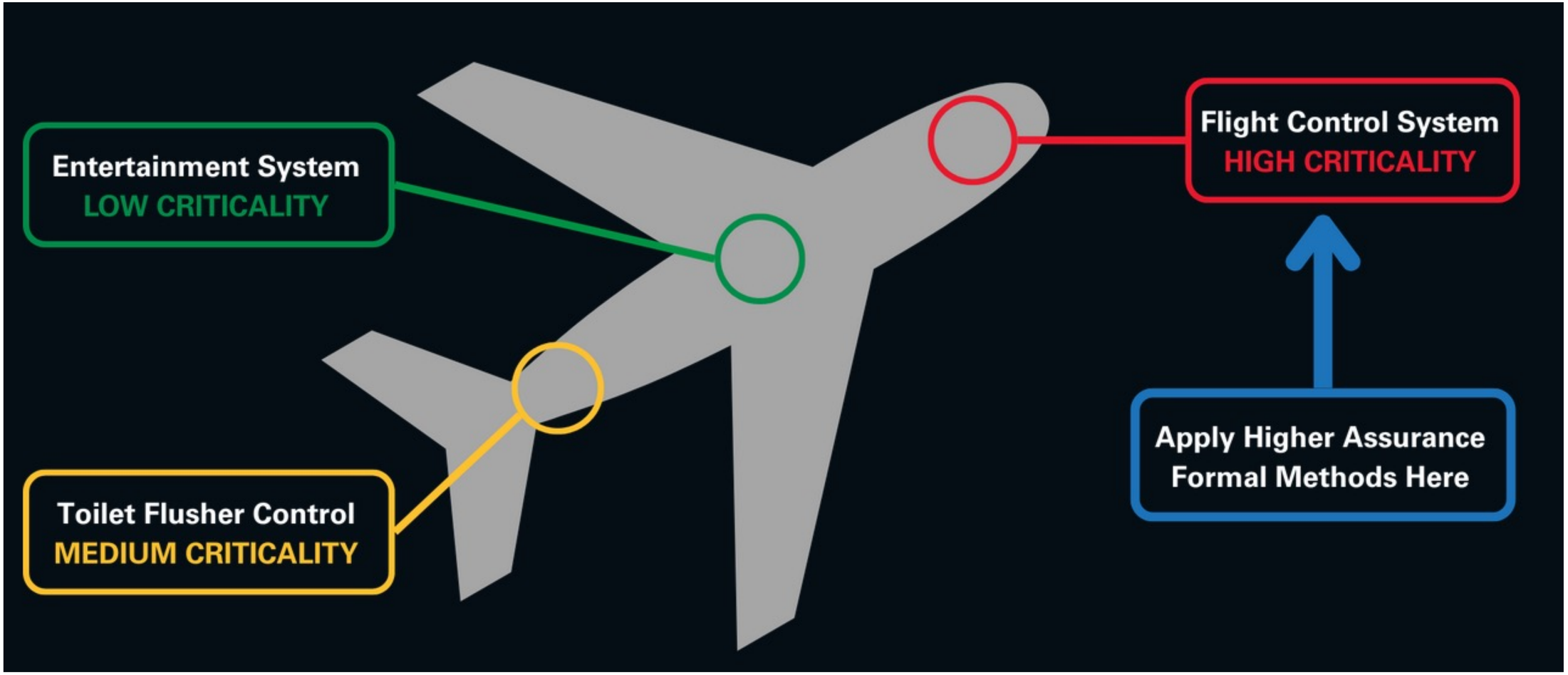
Code / System is translated into a detailed mathemetical model, which can be reasoned about with formal proofs



There’s a range of formal methods of different complexity and difficulty.



Formal methods are expensive, slow, and difficult.



Case study: Amazon Web Services use of formal methods.

Applying TLA+ to some of Amazon's more complex systems.

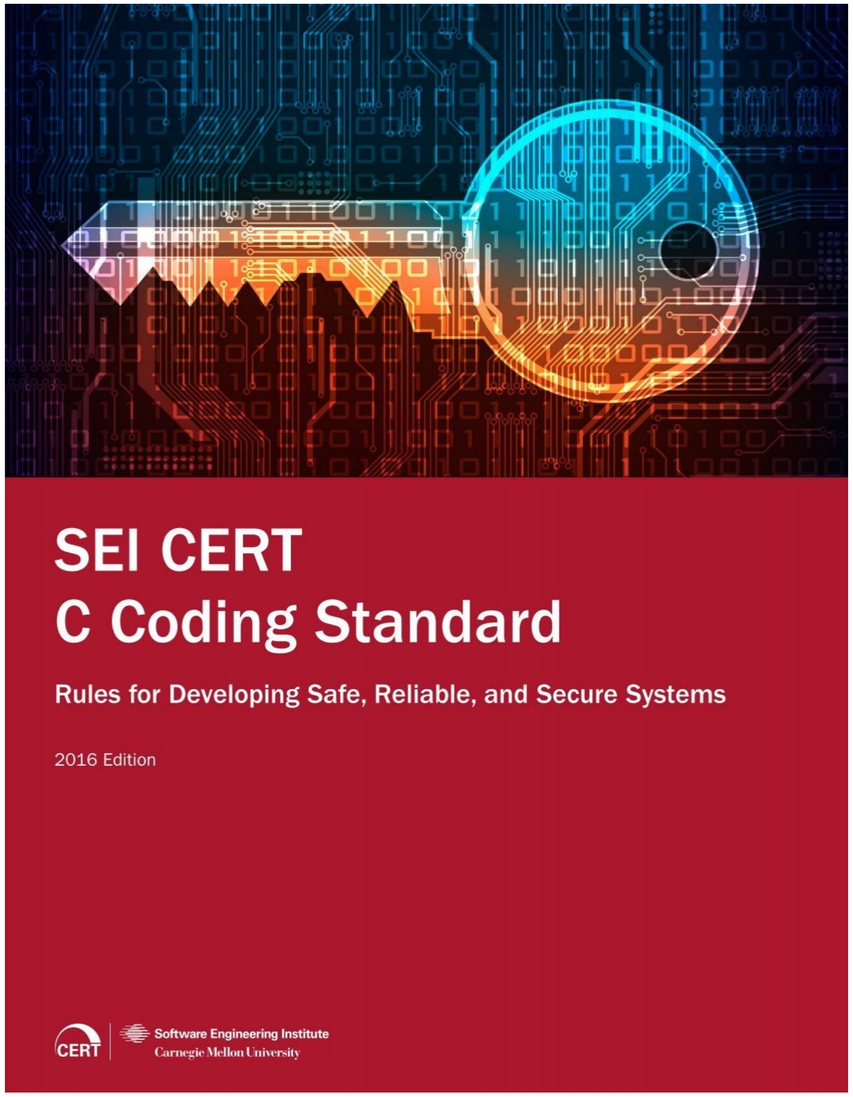
System	Components	Line Count (Excluding Comments)	Benefit
S3	Fault-tolerant, low-level network algorithm	804 PlusCal	Found two bugs, then others in proposed optimizations
	Background redistribution of data	645 PlusCal	Found one bug, then another in the first proposed fix
DynamoDB	Replication and group-membership system	939 TLA+	Found three bugs requiring traces of up to 35 steps
EBS	Volume management	102 PlusCal	Found three bugs
Internal distributed lock manager	Lock-free data structure	223 PlusCal	Improved confidence though failed to find a liveness bug, as liveness not checked
	Fault-tolerant replication-and-reconfiguration algorithm	318 TLA+	Found one bug and verified an aggressive optimization

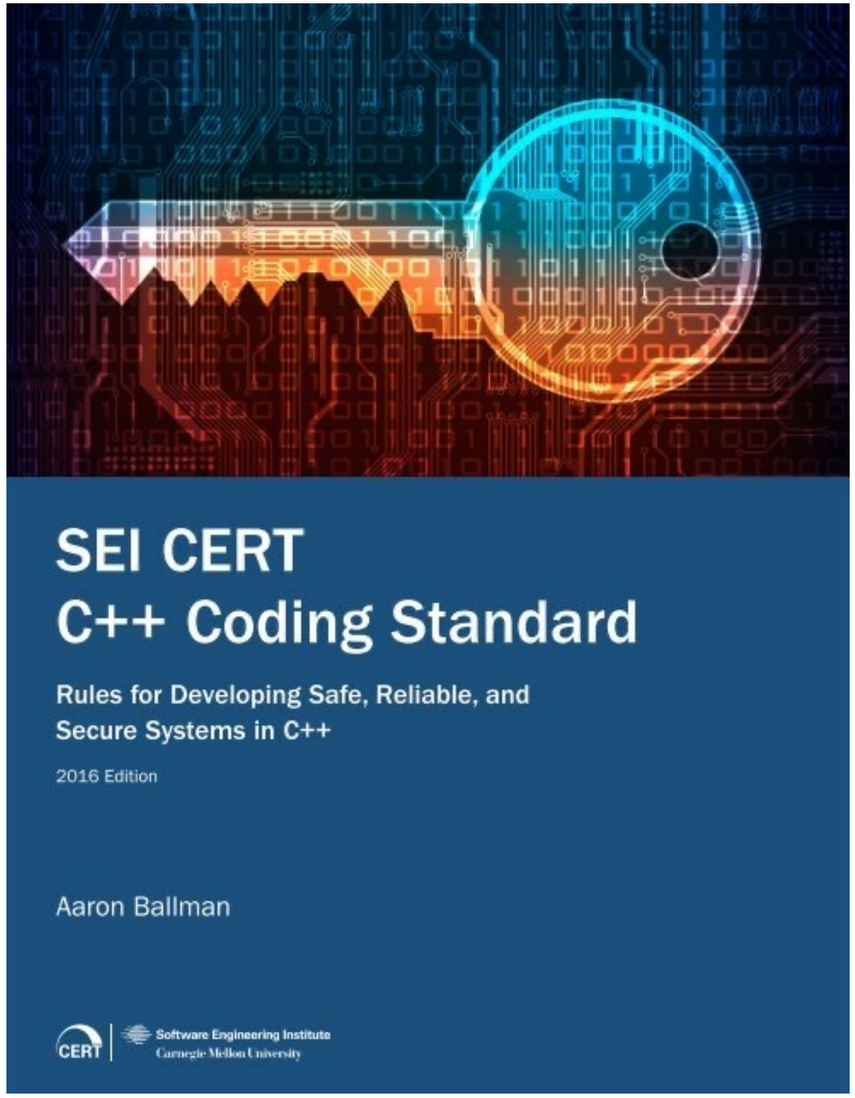
Secure Design vs. Implementation

Expectations

Reality

Secure Coding Standards





Memory Safety and Undefined Behavior

```
#include <limits.h>
#define LEN 42
int a[LEN] = {1,2,3};

int f(int base, int off) {
   if (base < 0 || off < 0) return -1;
   int idx = base + off;
   if (idx < 0) return -2;
   if (idx >= LEN) return -3;
   return a[idx];
}

int main() {
   return f(INT_MAX,2);
}
```

Memory Safety

TLP:CLEAR

The Case for Memory Safe Roadmaps

Why Both C-Suite Executives and Technical Experts Need to Take Memory Safe Coding Seriously

https://media.defense.gov/2023/Dec/06/2003352724/-1/-1/0/THE-CASE-FOR-MEMORY-SAFE-ROADMAPS-TLP-CLEAR.PDF

Safe Programming Languages

MIT Technology Review

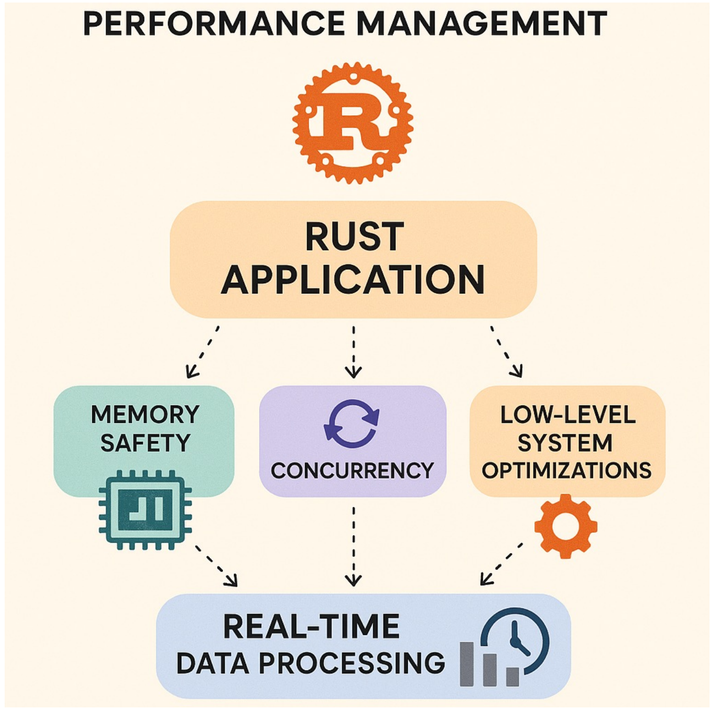
COMPUTING

How Rust went from a side project to the world's most-loved programming language

For decades, coders wrote critical systems in C and C++. Now they turn to Rust.

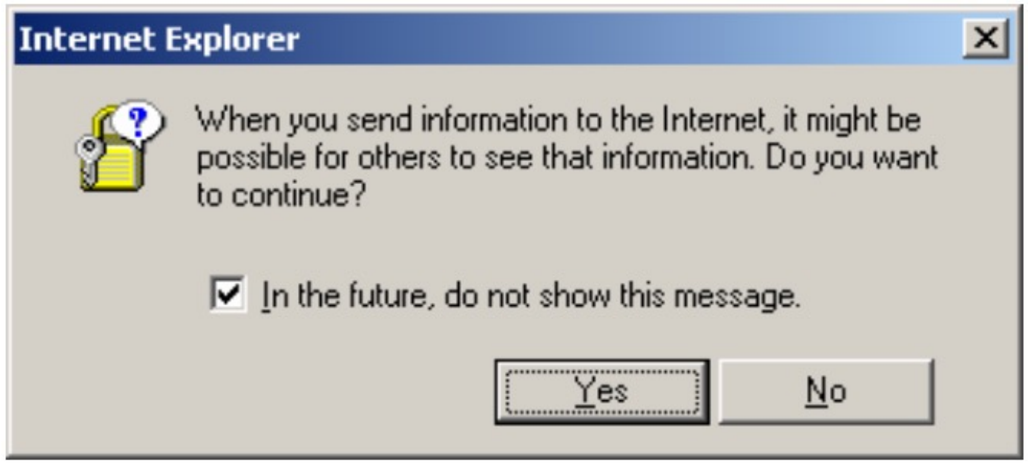
By Clive Thompson February 14, 2023

https://www.technologyreview.com/2023/02/14/1067869/rust-worlds-fastest-growing-programming-language/



https://medium.com/@kaly.salas.7/rust-performance-optimizations-compared-to-other-programming-languages-c2e3685163e2

Usability


Internet Explorer
When you send information to the Internet, it might be possible for others to see that information. Do you want to continue?
In the future, do not show this message.
Yes
No


Internet Explorer
When you see a dialog box like this, click 'Yes' to make it go away. If available, click the checkbox first to avoid being bothered by it again.
In the future, do not show this message.
Yes
No

Usability of Security

Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0

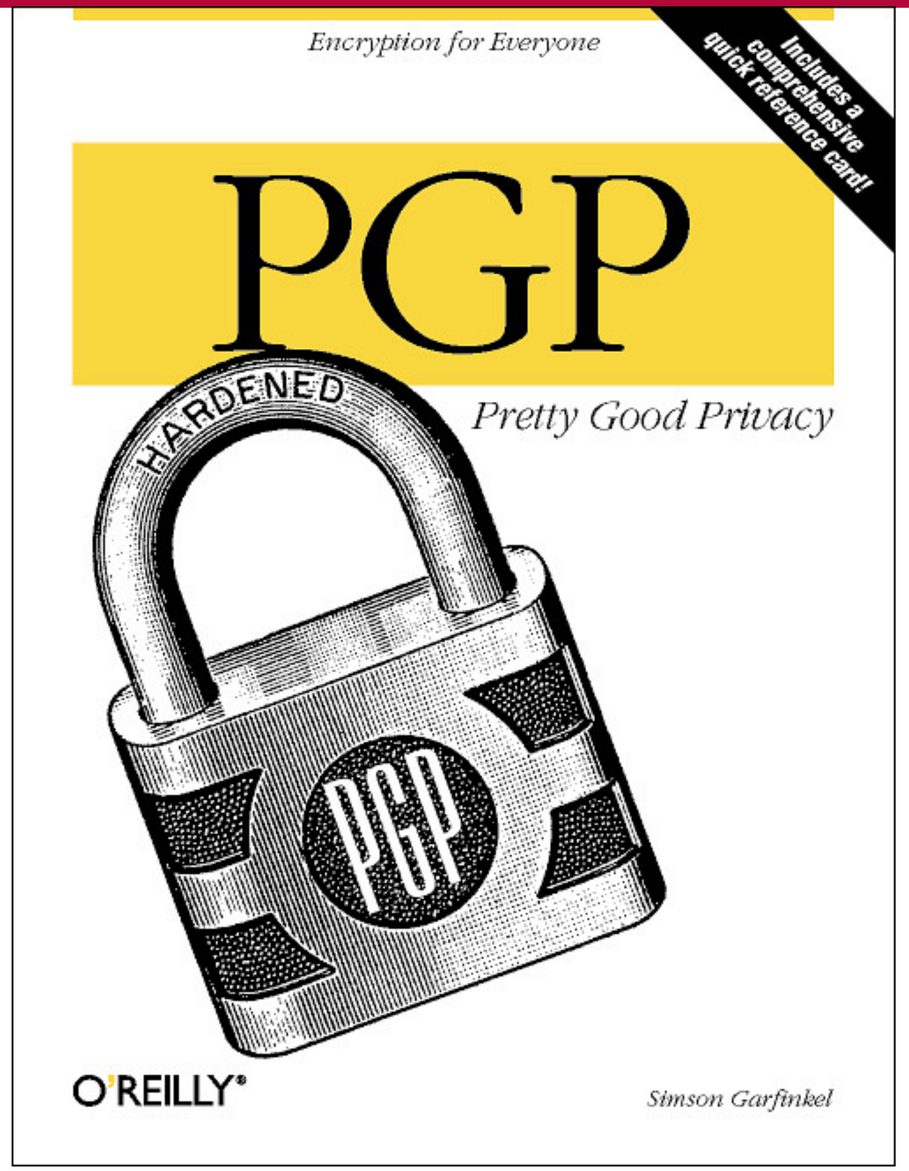
Alma Whitten and J.D. Tygar

Usenix Sec'99

Berkeley SCHOOL OF INFORMATION ABOUT PROGRAMS COURSES PEOPLE RESEARCH CAREERS

Why Johnny Can't Encrypt: Doug Tygar's Landmark Paper Stands the Test of Time

Aug 18, 2015



PGP: Pretty Good Privacy, Garfinkel, O'Reilly, 1994

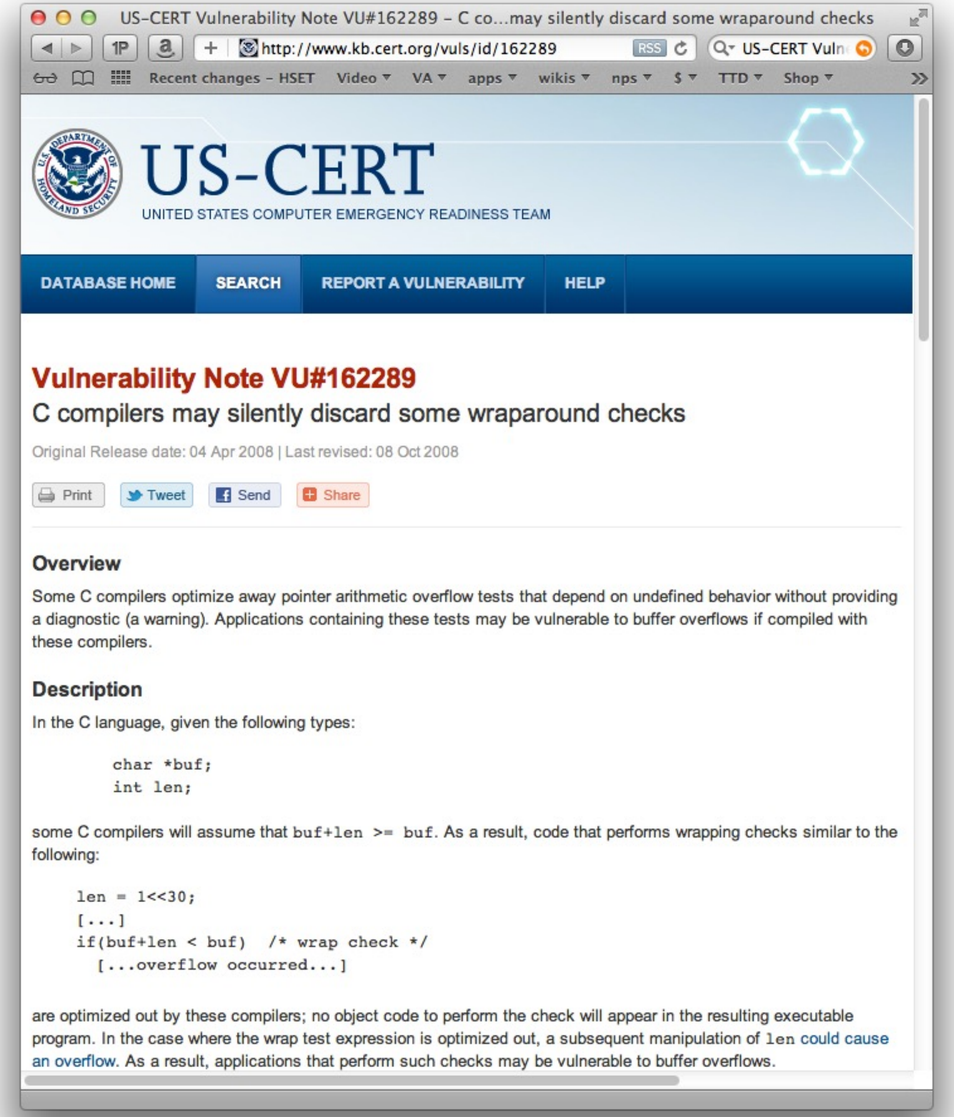
Whitten & Tygar analyzed the 1998 Macintosh PGP program.

Definition: Security software is usable if the people who are expected to use it:

- **Are reliably made aware of the security tasks they need to perform**
- **Are able to figure out how to successfully perform those tasks**
- **Don't make dangerous errors**
- **Are sufficiently comfortable with the interface to continue using it**

—Whitten & Tygar, 1999

With active adversaries, *all software is security software*, and *all programmers are security programmers.*

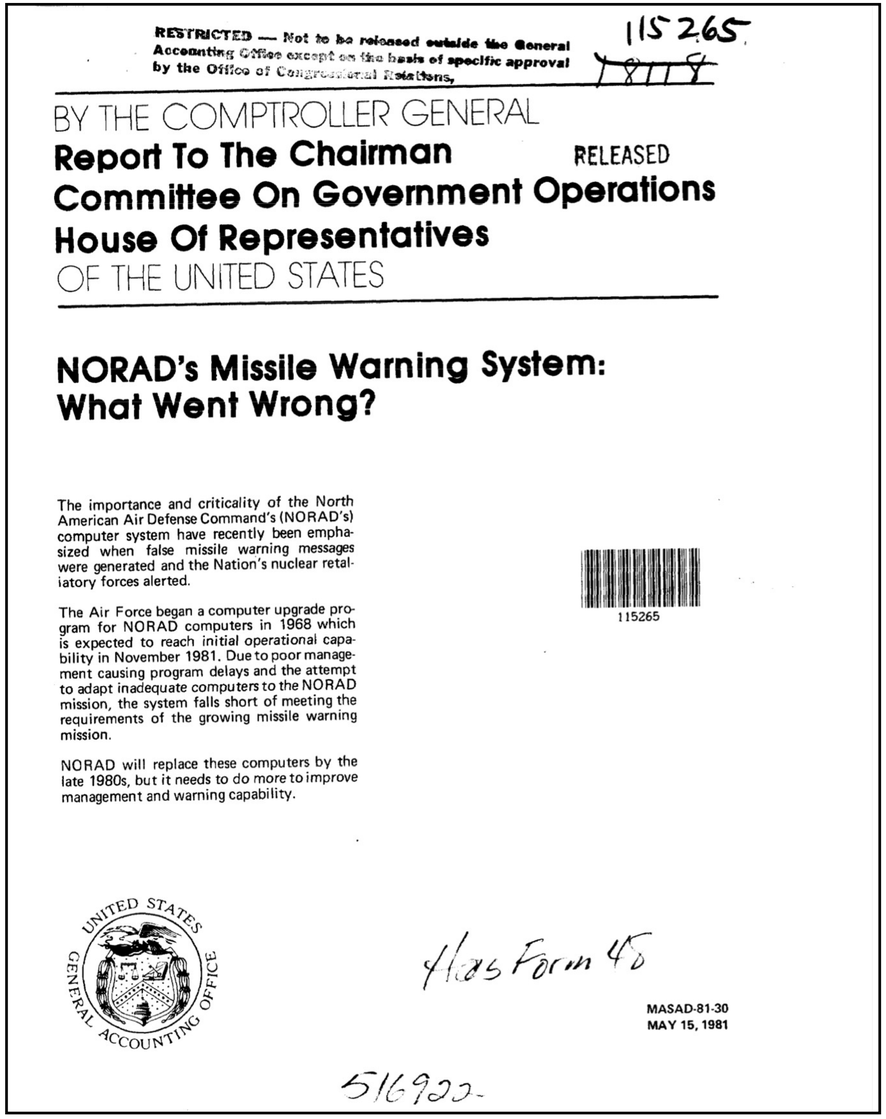


A (Brief) History of Information Security Failures

Guglielmo Marconi

Nevil Maskelyne

The film “WarGames” (1983) (Loosely inspired by actual events at NORAD)



RESTRICTED — Not to be released outside the General Accounting Office except on the basis of specific approval by the Office of Congressional Relations.

115265

BY THE COMPTROLLER GENERAL

Report To The Chairman
Committee On Government Operations
House Of Representatives

RELEASED

OF THE UNITED STATES

NORAD's Missile Warning System: What Went Wrong?

The importance and criticality of the North American Air Defense Command's (NORAD's) computer system have recently been emphasized when false missile warning messages were generated and the Nation's nuclear retaliatory forces alerted.

The Air Force began a computer upgrade program for NORAD computers in 1968 which is expected to reach initial operational capability in November 1981. Due to poor management causing program delays and the attempt to adapt inadequate computers to the NORAD mission, the system falls short of meeting the requirements of the growing missile warning mission.

NORAD will replace these computers by the late 1980s, but it needs to do more to improve management and warning capability.

115265

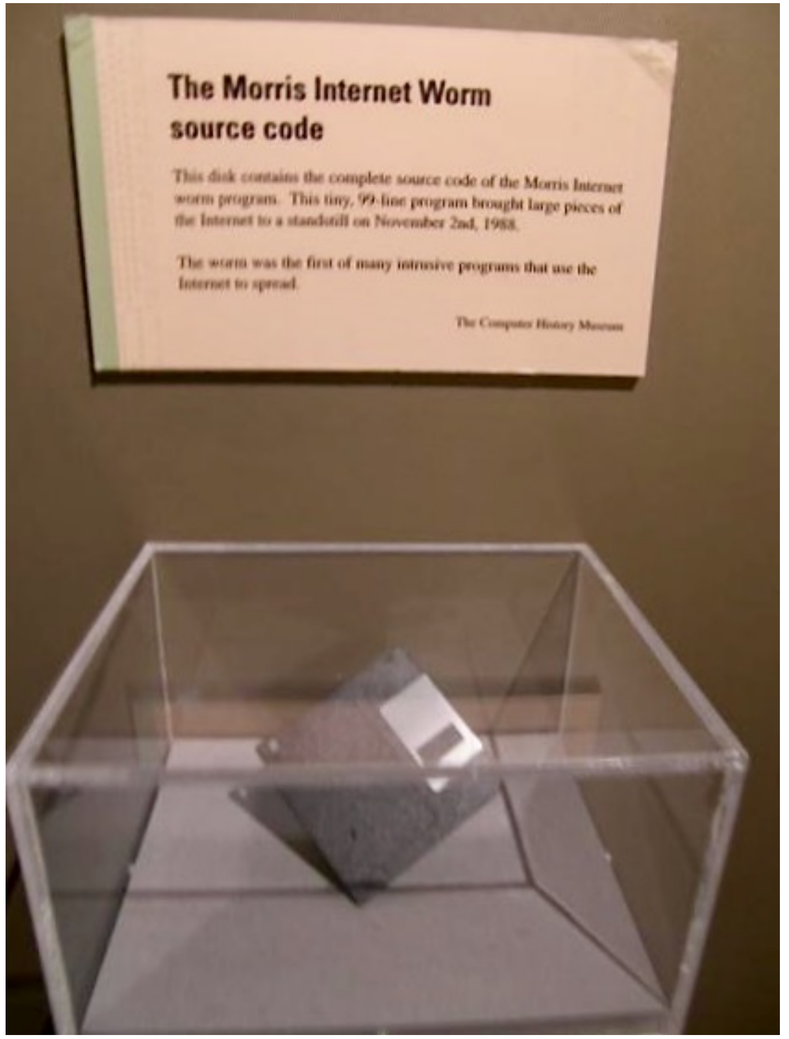
UNITED STATES GENERAL ACCOUNTING OFFICE

Has Form 45

MASAD-81-30
MAY 15, 1981

516922

The Morris Worm (1988)



2010 IoT attack — Stuxnet shows that there are no “air gapped” networks.

Iranian President Mahmoud Ahmadinejad inspects nuclear centrifuges
March 8, 2007

October 2018: GAO-19-128

United States Government Accountability Office

Report to the Committee on Armed Services, U.S. Senate

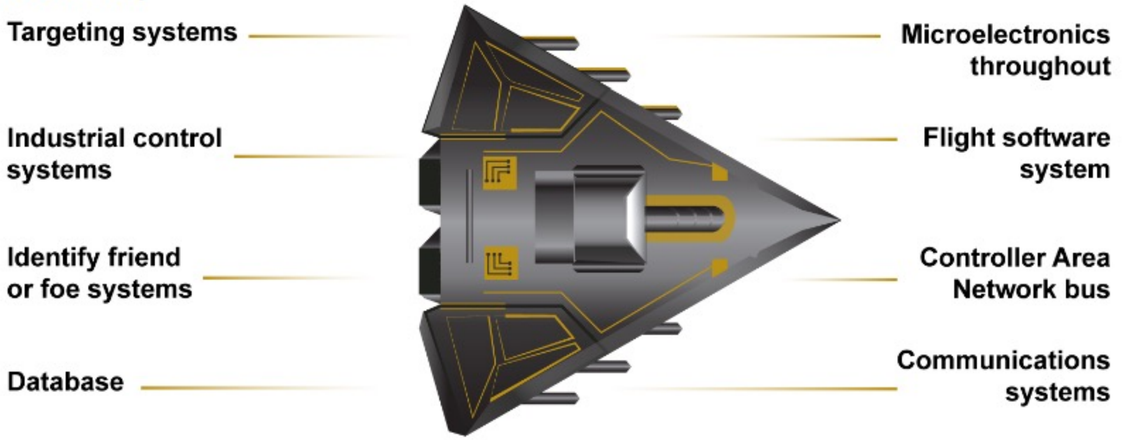
October 2018

WEAPON SYSTEMS CYBERSECURITY

DOD Just Beginning to Grapple with Scale of Vulnerabilities

GAO-19-128

Embedded Software and Information Technology Systems Are Pervasive in Weapon Systems (Represented via Fictitious Weapon System for Classification Reasons)



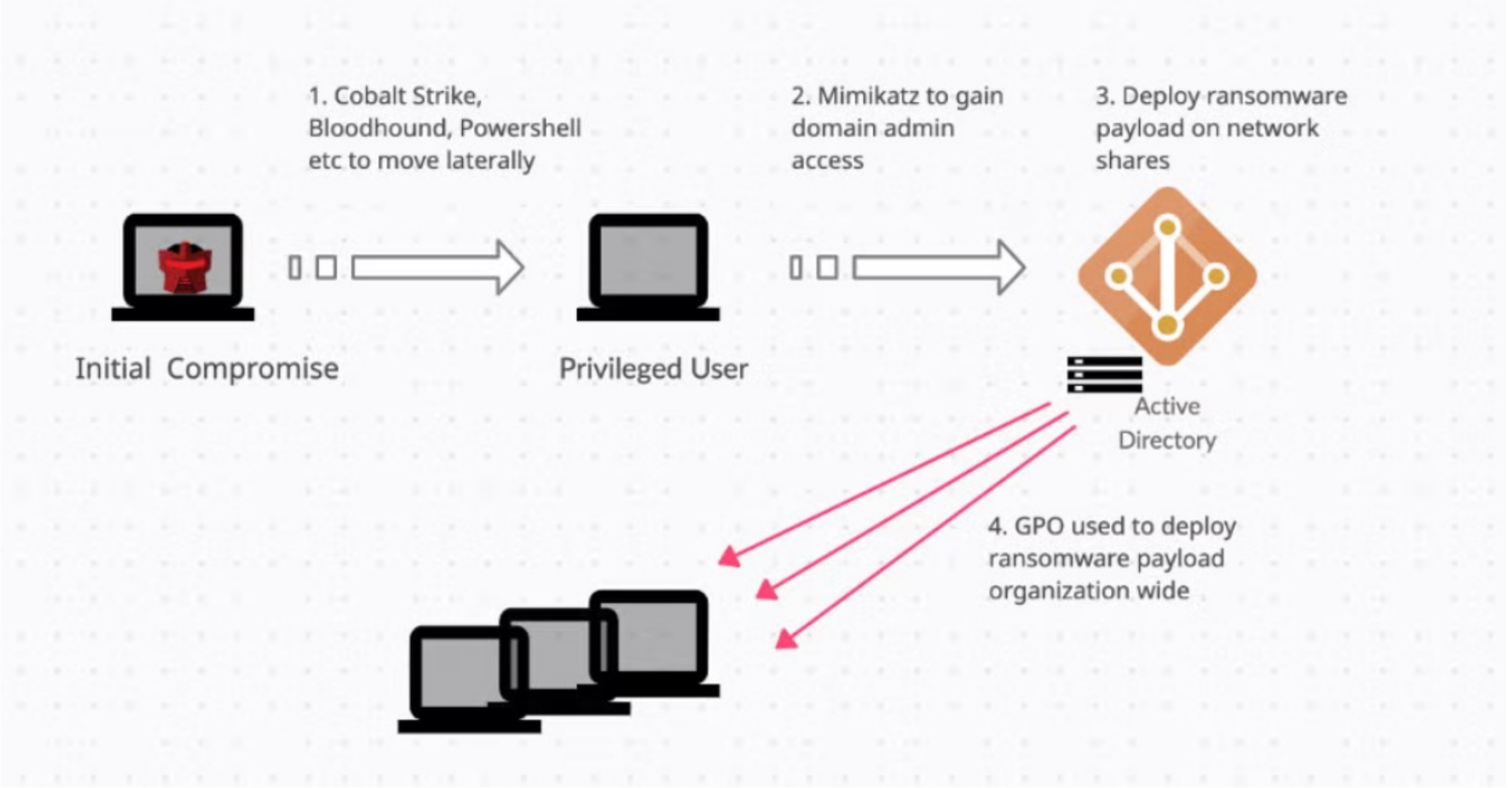
Source: GAO analysis of Department of Defense information. | GAO-19-128

GAO found systemic security shortcomings.

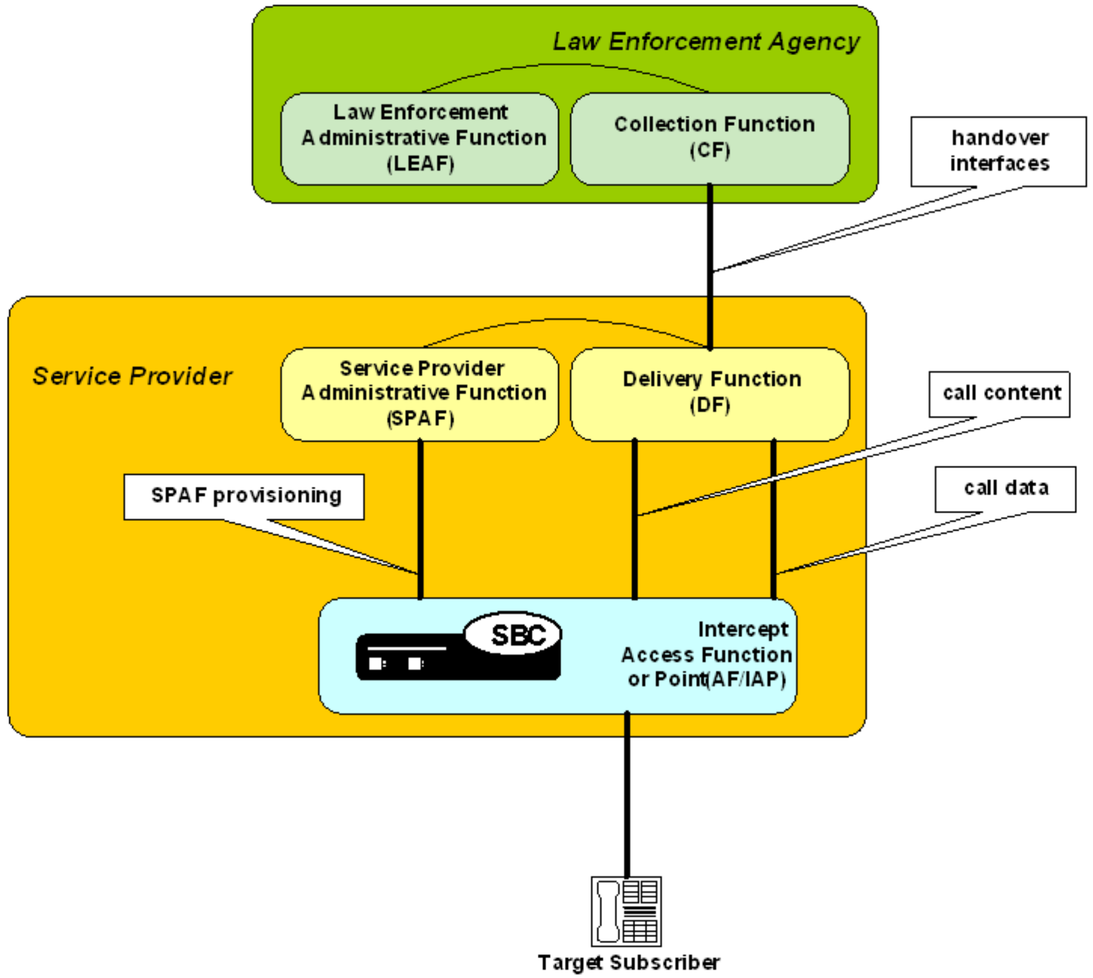
- “DOD routinely found mission-critical cyber vulnerabilities in systems that were under development, yet program officials GAO met with believed their systems were secure.”
- “Using relatively simple tools and techniques, testers were able to take control of systems and largely operate undetected, due in part to basic issues such as poor password management and unencrypted communications.”
- “Vulnerabilities that DOD is aware of likely represent a fraction of total vulnerabilities due to testing limitations.”

Colonial Pipeline Ransomware Attack, 2021

Colonial Pipeline Company

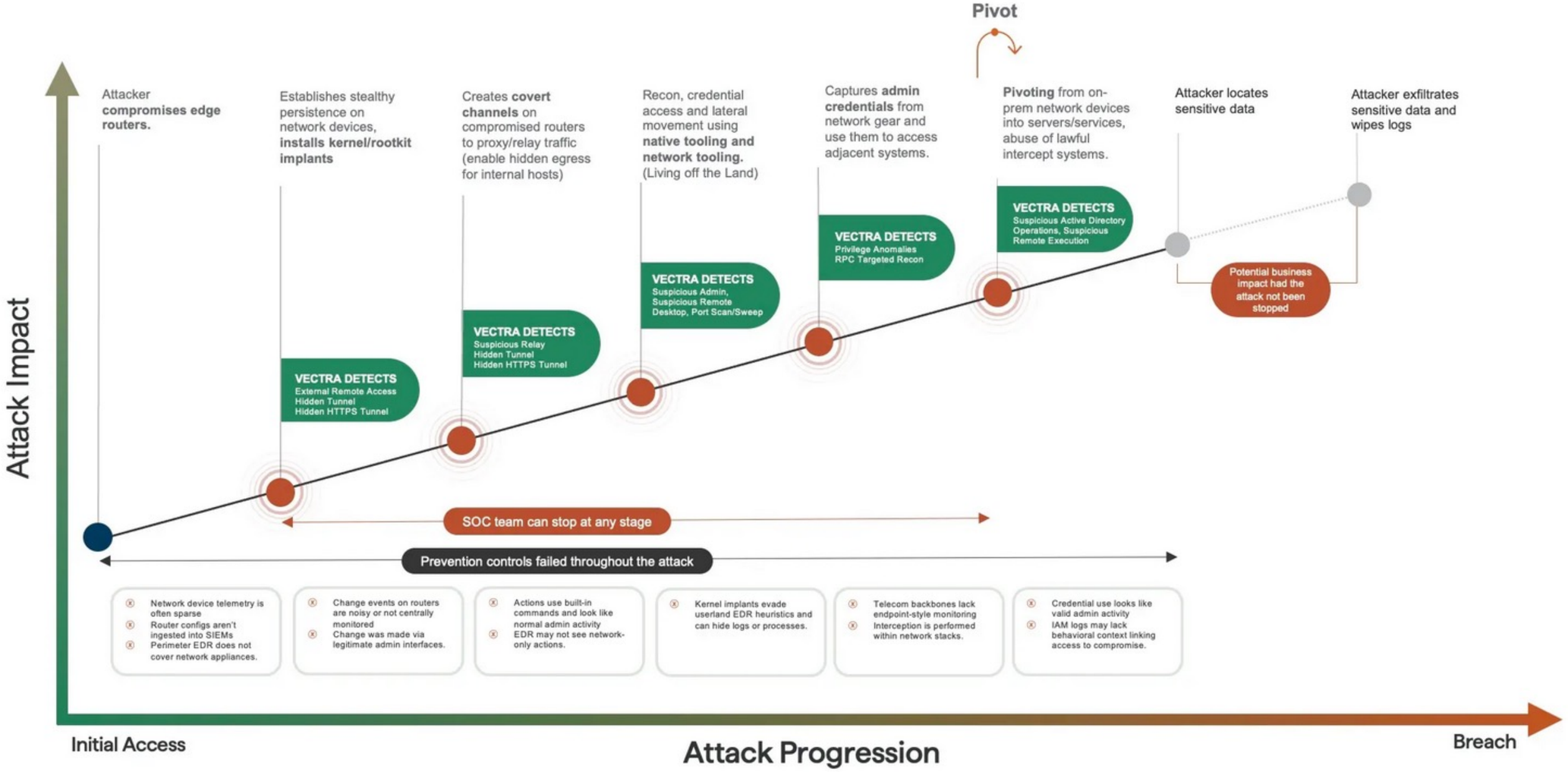


Salt Typhoon, 2024



https://docs.oracle.com/en/industries/communications/session-border-controller/9.1.0/configuration/lawful-intercept.html

Anatomy of a Salt Typhoon Attack

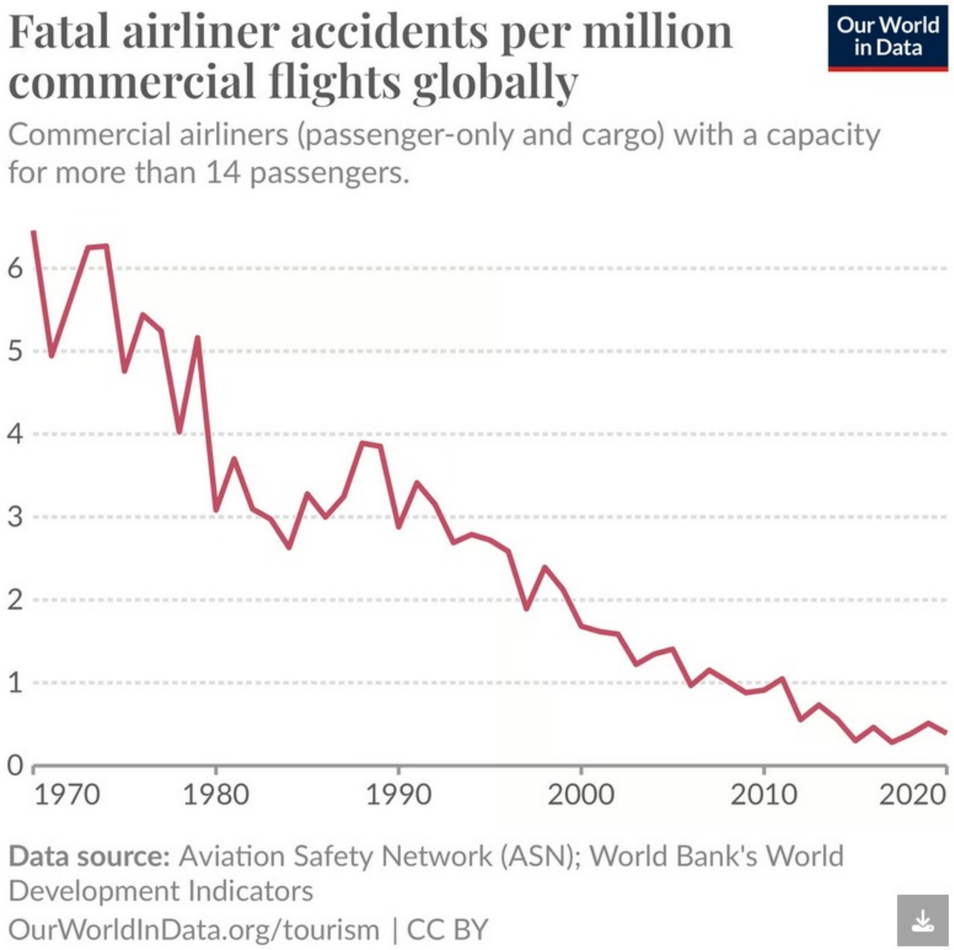


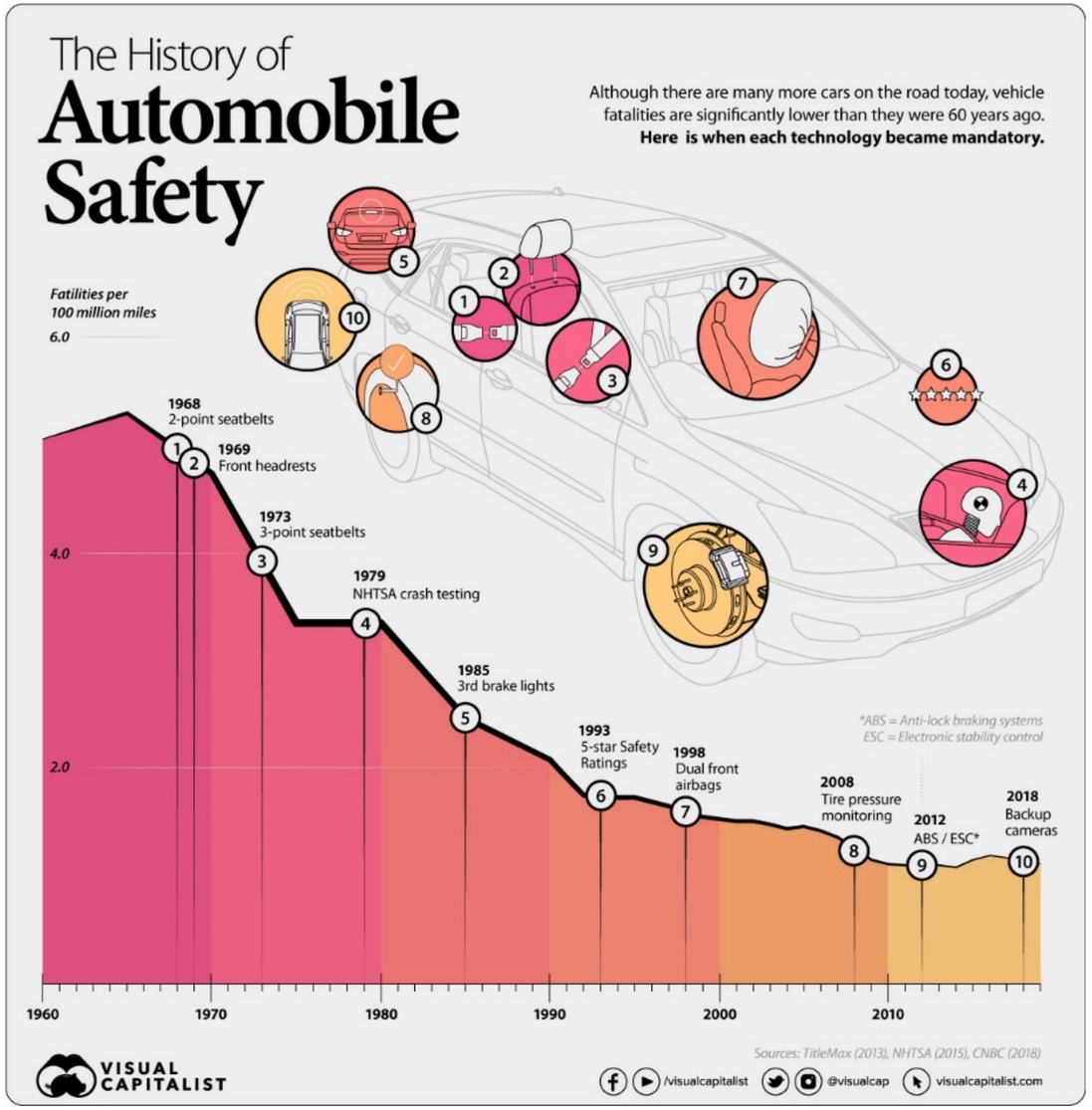
https://www.vectra.ai/blog/how-typhoon-apts-infiltrate-infrastructure-without-leaving-a-trace

(Ignore advertisements)

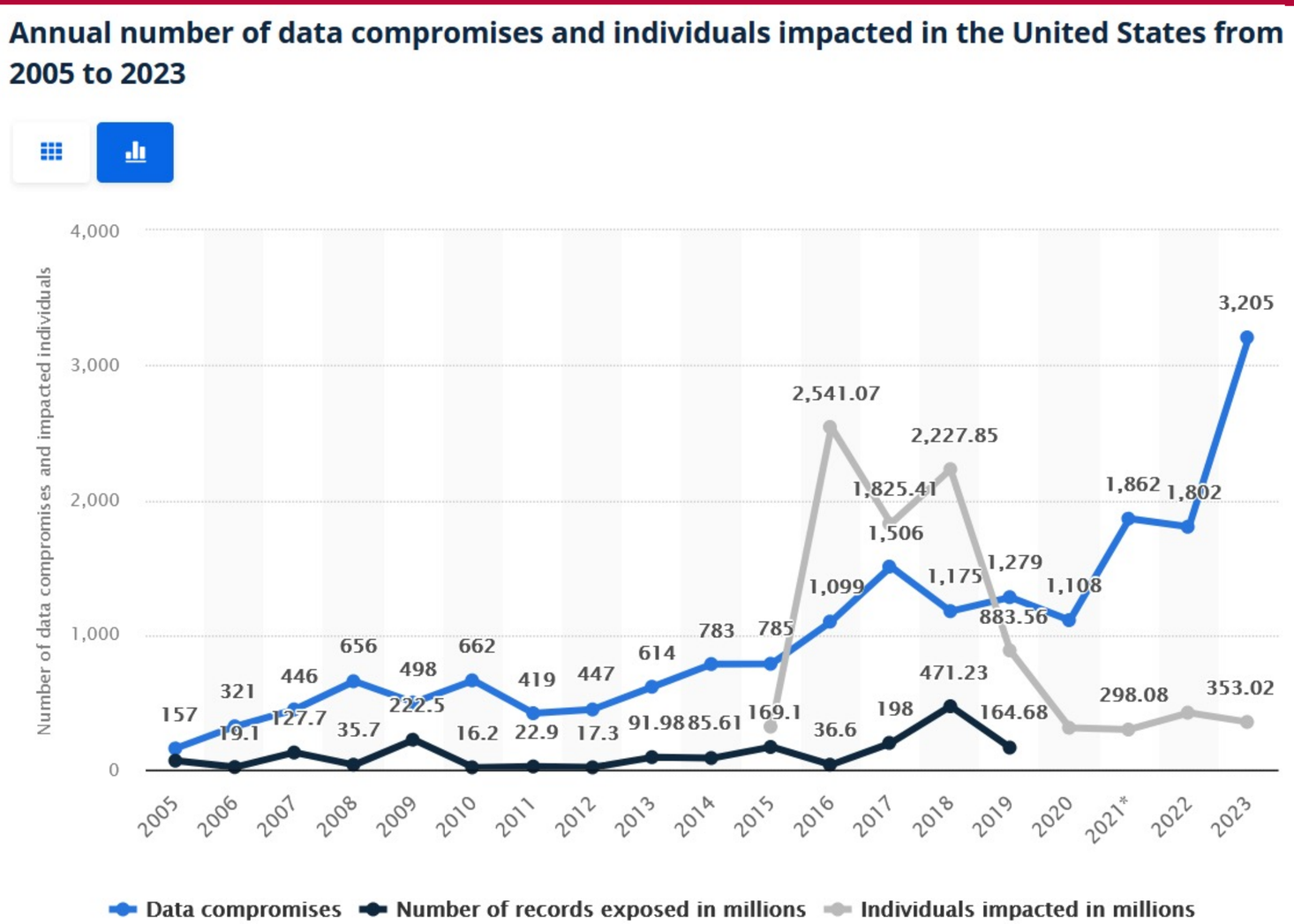
Why is Security So Hard?

We’ve made incredible progress improving safety in air and automobile travel.





But for cybersecurity, things seem to be getting worse.



https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/

Security is a technical problem.

SNARK: a Succinct ARgument of Knowledge

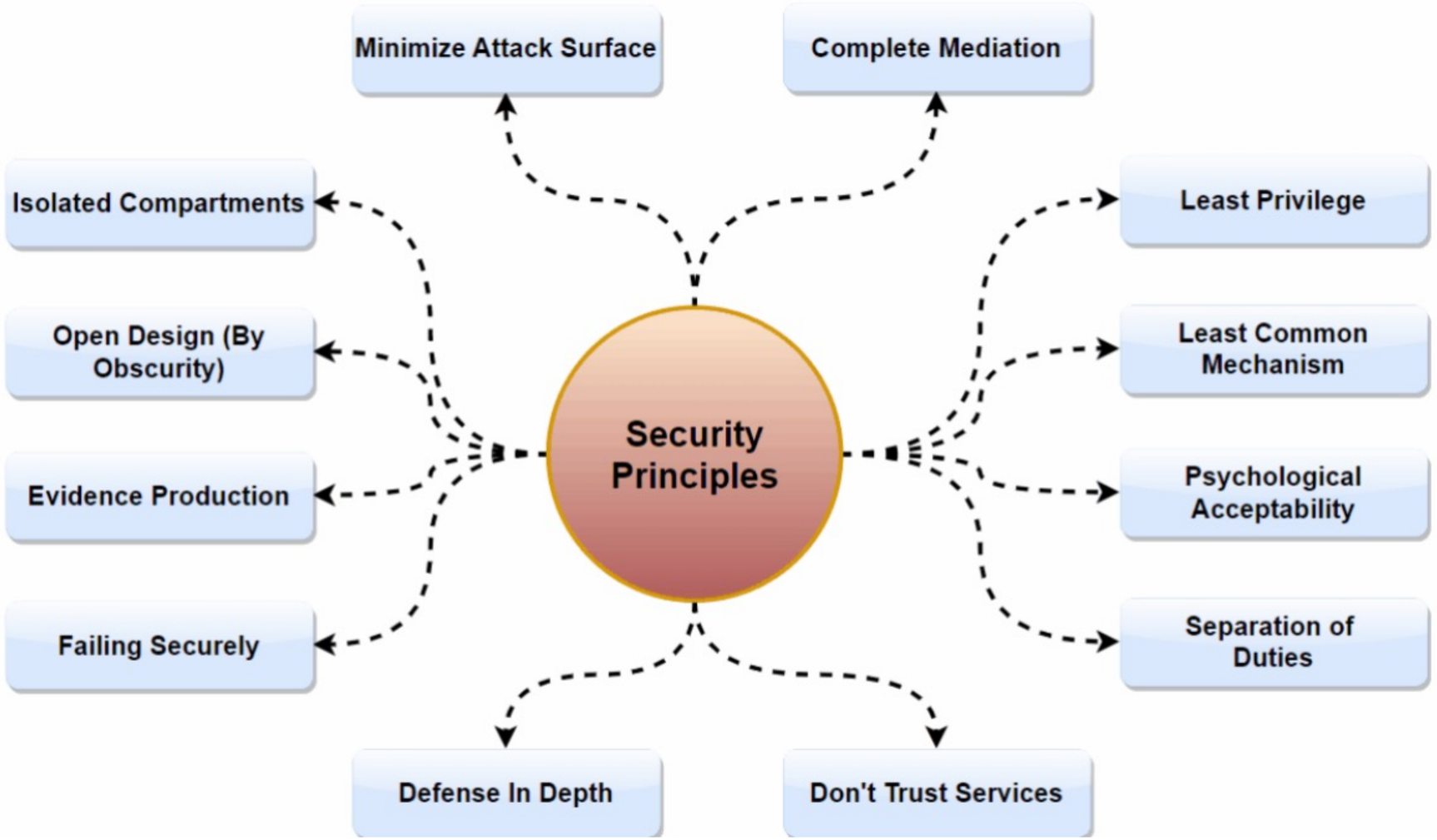
A **succinct preprocessing argument system** is a triple (S, P, V):

- **S**$(C) \rightarrow$ public parameters (S_p, S_v) for prover and verifier
- **P**$(S_p, \boldsymbol{x}, \boldsymbol{w}) \rightarrow$ **short** proof π ; $|\pi| = O(\mathbf{log}(|\boldsymbol{C}|), \lambda)$
- **V**$(S_v, \boldsymbol{x}, \boldsymbol{\pi})$ **fast to verify** ; time(V) = $O(|x|, \mathbf{log}(|\boldsymbol{C}|), \lambda)$

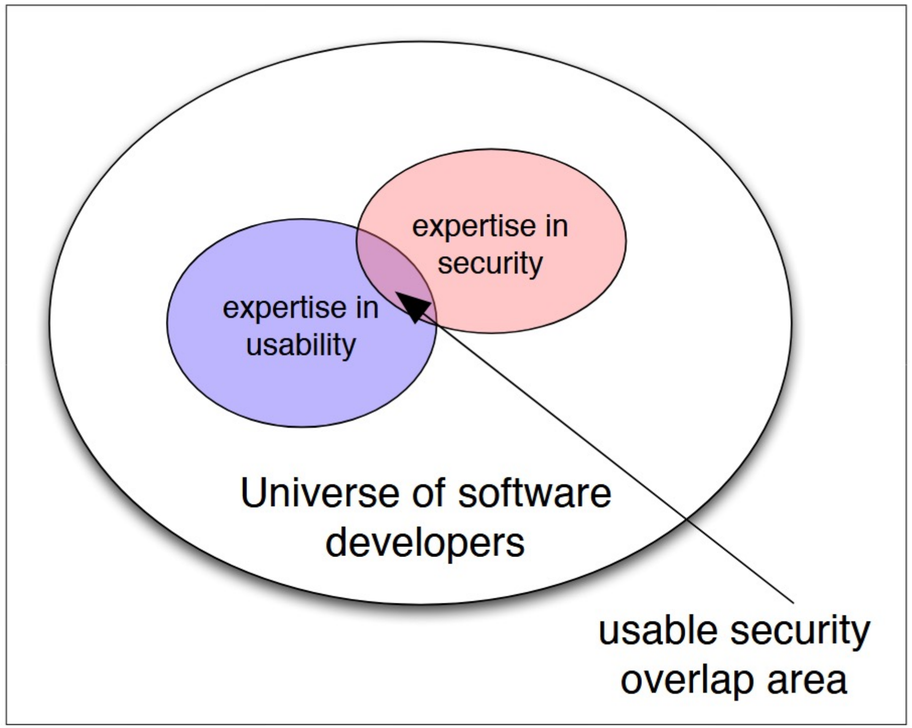
SNARK: (S, P, V) is **complete**, **knowledge sound**, and **succinct**

zk-SNARK: (S, P, V) is a SNARK and is **zero knowledge**

Security is a design problem.



Security is a usability problem.



"Design principles and patterns for computer systems that are simultaneously secure and usable," Simson Garfinkel, 2005. https://dspace.mit.edu/handle/1721.1/33204

Security is an economic problem, but companies that have security breaches are not penalized in the long-term.

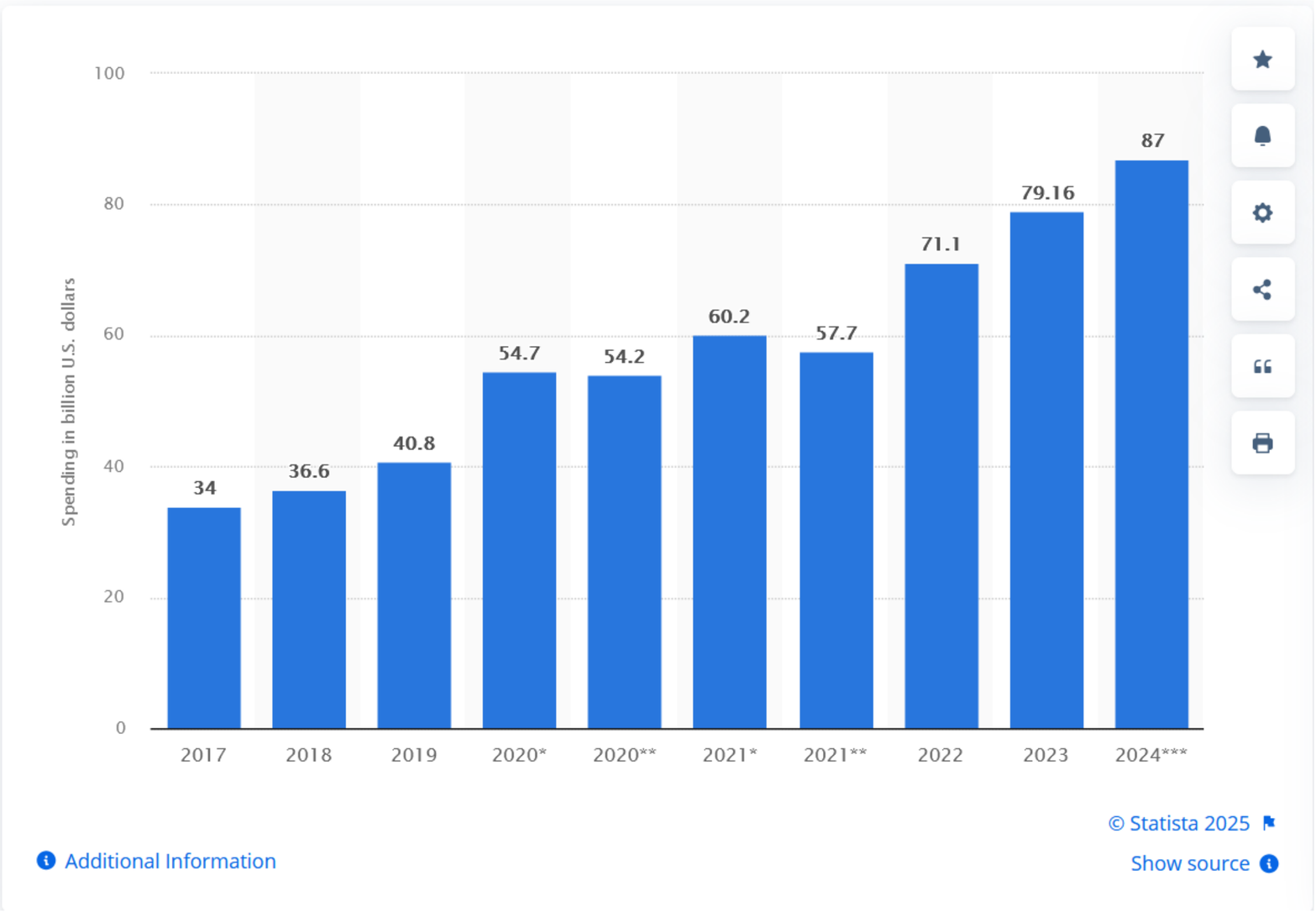
Yahoo breach:
2013-2014: 3 billion accounts,
revealed Sept. 2016

eBay breach:
May 2014: 145 million users,

Equifax breach:
July 2017: 143 million consumers

The three largest breaches in history.

Cybersecurity is expensive.



Security is a legal problem.

FBI seizes website used by notorious ransomware gang

By Sean Lyngaas, CNN

3 minute read · Updated 11:58 AM EST, Thu January 26, 2023

Security is a skill problem.

Cybersecurity Workforce Supply and Demand Report

May 2024
Final Report

https://ncses.nsf.gov/760/assets/0/files/ncses-cwdi-supply-demand-report.pdf

NSF Report: Key Findings

Current cybersecurity workforce

- 1,180,000 to 1,340,000 (source: Bureau of Labor Statistics)
- 164,000 to 3,492,000 (source: BLS, US Census Bureau, and others)
- Unemployment rate: 2,500 to 74,000 individuals

Unfilled positions:

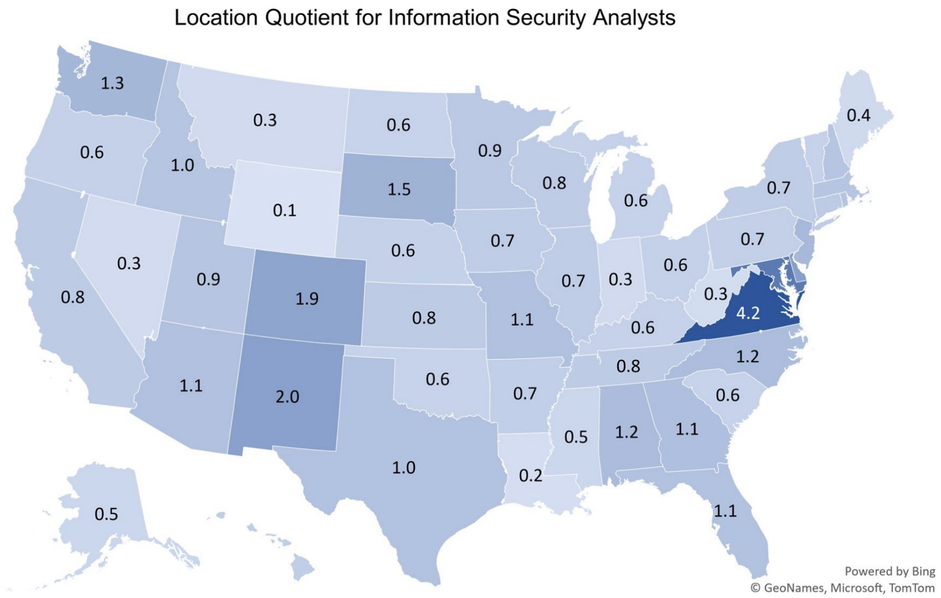
- 480,000 (source: ICS2) to 570,000 (Source: CyberSeek)

Highest concentration of jobs in the DC area.

Degrees are mostly being awarded to men.

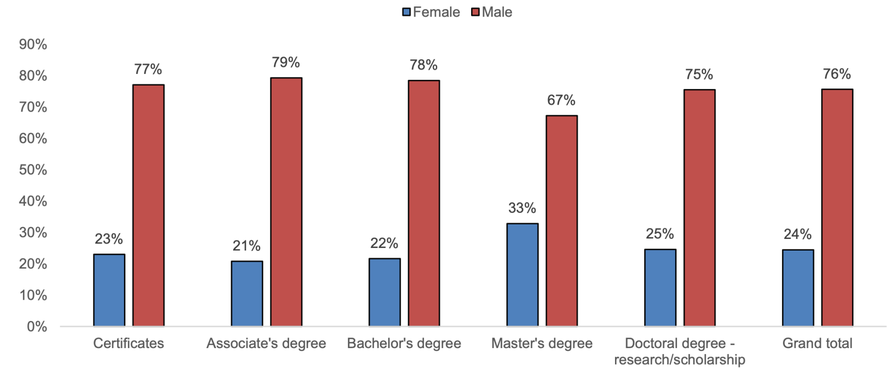
Out of 56,000 open jobs on ClearanceJobs, only 4,000 are for information technology professionals.

Figure 1
Employment concentration for information security analysts, by state: 2022
(Location quotient)



Source(s):
Bureau of Labor Statistics, Occupational Employment and Wage Statistics (2022).

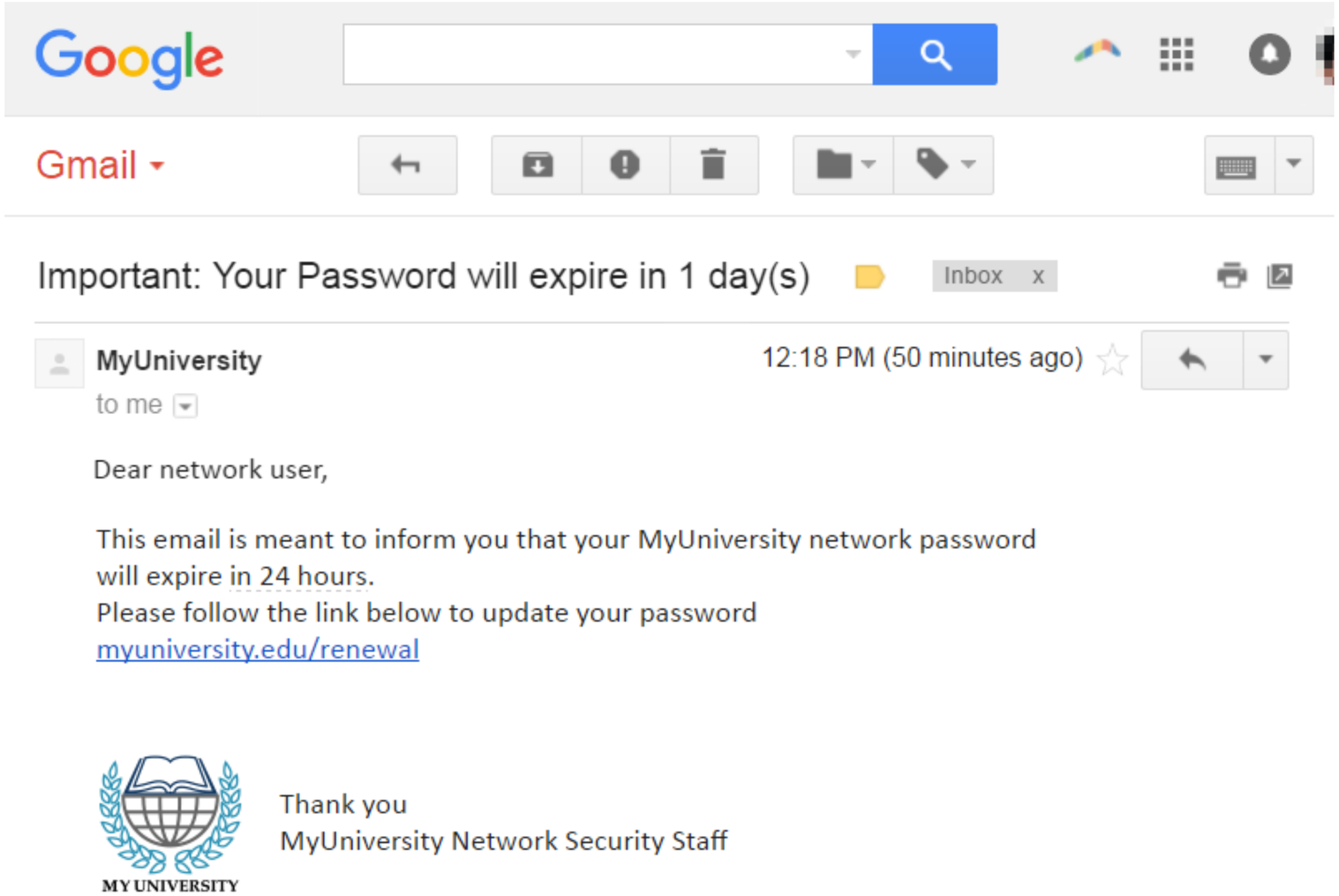
Figure 4
Degrees awarded in cybersecurity, by sex: 2022
(Percent)



Source(s):
National Center for Science and Engineering Statistics, Integrated Postsecondary Education Data System (IPEDS), Degrees and Certificates Conferred, available at https://ncsesdata.nsf.gov/builder/ipeds_c, accessed 22 February 2024.

Security is an adversarial problem.

Security is an asymmetric problem.



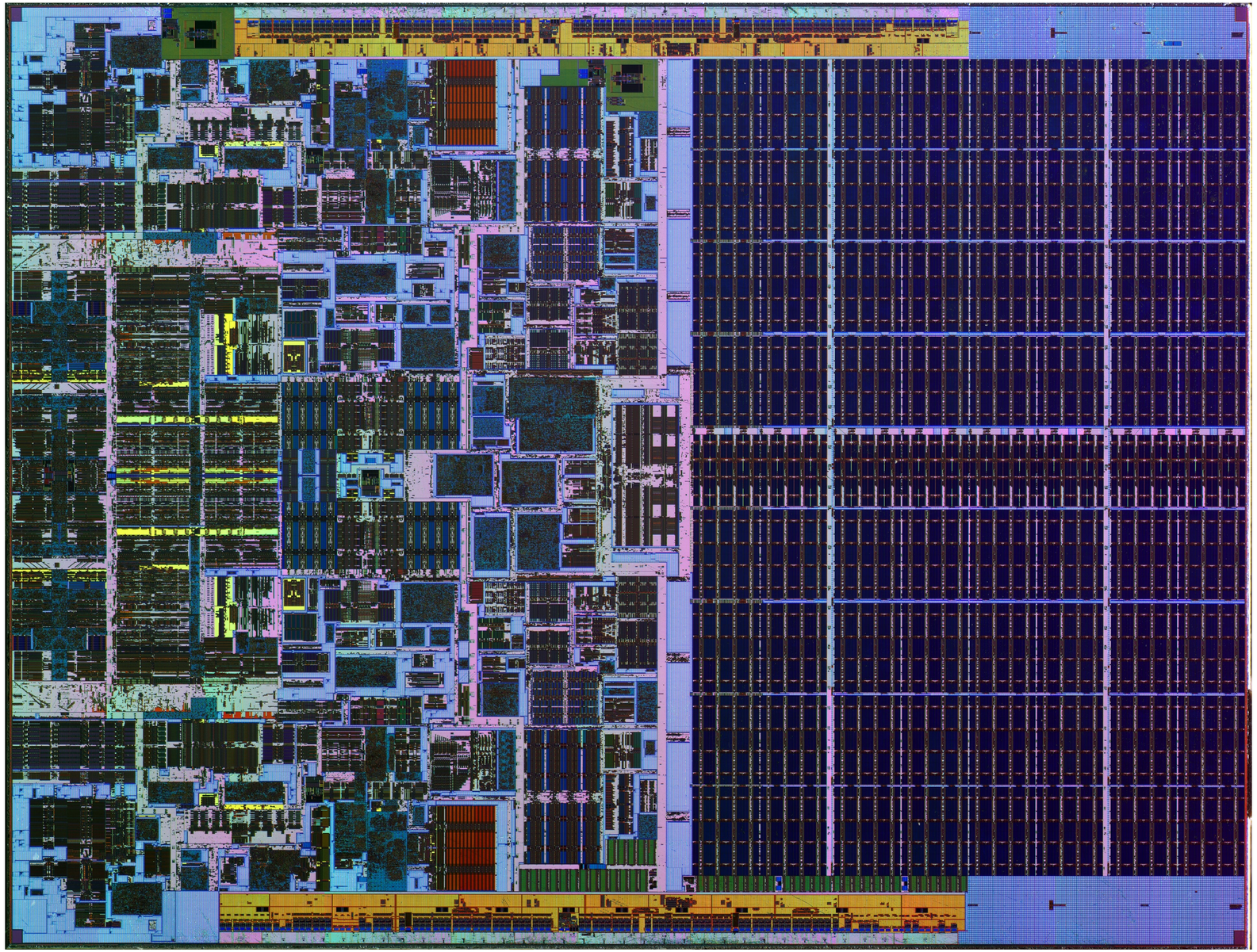
Cybersecurity is a “wicked problem”

- **Wicked Problems: Rittel and Webber, “Dilemmas in a General Theory of Planning,” 1973**

- **No clear definition**
  - *You don’t understand the problem until you have a solution.*
- **No “stopping rule”**
  - *The problem can never be solved.*
- **Solutions not right or wrong**
  - *Benefits to one player hurt another — Information security vs. Free speech*
- **Solutions are “one-shot” — no learning by trial and error**
  - *No two systems are the same. The game keeps changing.*
- **Every wicked problem is a symptom of another problem**
  - *Dave Clement, “Cyber Security as a Wicked Problem,” Chatham House, 2011*

Cybersecurity is too hard for both users *and* experts!

Labs



Xenon 3060 (2006)

Labs 1-3 – If you didn’t get a “5,” you were regraded multiple times

Originally we said late labs would not be accepted.

Then we said late labs would have a max score of “4”

We re-graded all labs 1, 2 and 3 that were under “5.”

lab4 builds on lab3

Objectives for this lab:

- Learn about password hashing, an authentication method that combines an identifier and a shared secret.
- Deploy a web-based message board that is built on an API that uses password hashing for authentication and an SQLite database for data storage.
- Build in computer security from the start rather than add it as an after-thought.

You create a system that uses an API_KEY and a API_SECRET_KEY

API_KEY is like a username – it identifies the account

API_SECRET_KEY is like a password — it authenticates access to the account.

The password-hashing approach we introduce in Lab4 is the similar one that Amazon uses for AWS_ACCESS_KEY and AWS_SECRET_ACCESS_KEY

Reminders…

"smashedemail" is the name we use in our examples

- You need to replace this with parts of your email address, smashed together
- You will find your smashedemail at https://csci-e-11.org/

Once you run **e11 check lab4** you must run **e11 grade lab4**

After you get your 5/5, you can stop your instance

You do not need to stop it!

If you are at the free tier, you have free credits to leave it running.

Do not **terminate —** you will need to reinstall everything and re-register your VM.

Please…

Start lab 4 now!

This Week's Reading

A Comparative Long-Term Study of Fallback Authentication Schemes

Leona Lassak
Ruhr University Bochum
Germany
leona.lassak@rub.de

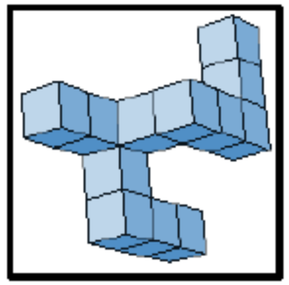
Philipp Markert
Ruhr University Bochum
Germany
philipp.markert@rub.de

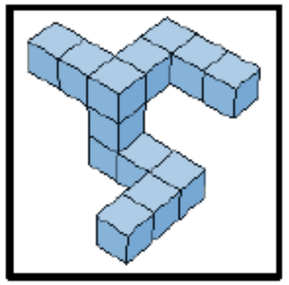
Maximilian Golla
CISPA Helmholtz Center for Information Security
Germany
golla@cispa.de

Elizabeth Stobert
Carleton University
Germany
elizabeth.stobert@carleton.ca

Markus Dürmuth
Leibniz University Hannover
Germany
markus.duermuth@itsec.uni-hannover.de

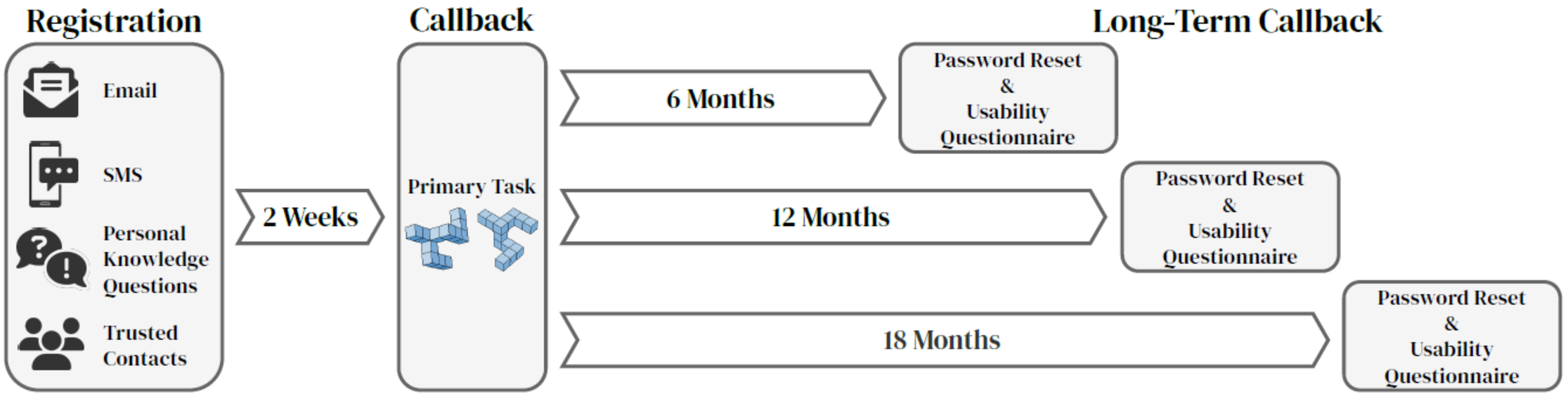
Please decide for each pair whether the two drawings portray objects with the **same** shape and size, i.e., are congruent with respect to three-dimensional shape, or depict objects of **different** three-dimensional shapes.





Same | Different

Fallback Testing



For Next Week

- Read “A Comparative Long-Term Study”
  - Comment on Perusall (your own and a response)
  - Take the reading quiz on Canvas
- Start Lab 4 (Due Mar. 9)
- Dig out from snow (if you’re in the Northeast)!