Today’s Agenda

1. Administrivia

2. Missing Readings for L09, L10

3. Threat Models: Who is the attacker? What can the attacker do?

4. Secure Coding

5. Translucent Databases

6. RFID
1. Quizes - If you are a remote student and you want it back, please email csci_e-170-staff@ex.com with a fax number and we will fax it out.

2. Midterm Projects should be in.
Final Projects

This is a research project, not a book review. You are expected to:

1. Create something and write about it.

2. Analyze something in detail.

You have until next Monday to form groups of 4 students. Email group names and your proposed topic to csci_e-170-staff@ex.com. Students who have not chosen groups will be assigned.
Why is building a *secure* system different than building a system that is:

1. Reliable

2. Safe

3. Easy-to-use
With security, there is an adversary.

No reason to defend against an unbounded adversary. Why?
The nature of the adversary determines your defenses.

Possible adversaries include:

1. Employees (good and bad)

2. High school students

3. Foreign Governments (“Titan Rain?”)

Evaluate according to *who they are* and by *what they can accomplish*. 
Remember RFC 602?

Public acknowledgment of hackers on the Internet:

1. Sites used physical security have not taken measures to secure machines accessible over the network.

2. “TIPs” allow anyone who knows a phone number access to the Internet.

3. “There is lingering affection for the challenge of breaking someone’s system. This affection lingers despite the fact that everyone knows that it’s easy to break systems, even easier to crash them.”

http://www.faqs.org/rfcs/rfc602.html
1983: War Games

“How about a nice game of Chess?”

“Later. Let’s play Global Thermonuclear War.”

All of a sudden, hacking is cool.
“75 cent accounting error”

Stoll sets up a honeypot filled with “SDINet” files.

Hacker gets traced back to Germany. Apparently sold secrets to KGB in exchange for cash and cocaine.
Emergence of the Hacker Underground

“Captain Crunch” (John Draper)
Based on the phone phreaks of the 1960s/1970s.
Magazines like “2600” and “Phrak”
Warez
Collections of attack tools (War dialers, root kits, etc.)
January 15, 1990: AT&T’s long distance network crashes

FBI starts massive investigation into “hacker phenomena;” raids 100+ hacker homes and Steve Jackson Games.

Results: EFF; computer crime laws; lots of media attention

http://www.mit.edu/hacker/hacker.html
War Dialing is scanning the *telephone* network.

1. Determine phone numbers to call

2. Call each number.

3. Identify what answers:
   (a) Carrier
   (b) Fax
   (c) Voice
   (d) Busy (repeat if necessary)

4. Repeat

5. Analyze the Results


- Department of Justice (August 17, 1996)
- Central Intelligence Agency (September 18, 1996)
- Lost World Movie (May 23, 1997)
- New York Times (February 16, 2001)

1996–Spamming for porn and pharmaceuticals.

2004—”Phishing”
Threat evolution parallels but lags the commercialization of the Internet.
Understanding the adversary

The adversary needs:

- Skills
- Motive
- Access
Understanding the adversary: Skills

Readily available online.
Many opportunities for improvement.
Online training from some hacking groups.
Understanding the adversary: Motive

Originally: fun & reputation

Increasingly: profit
Access

Physical (need to secure perimeter & control access)
Software (AIDS virus disk)
Telephone (voice & modem)
Wireless
Internet
Software Exploitation: Terminology

Computer virus

• Modifies other programs on a system to replicate itself.
• Originally transmitted by floppy disks

Computer worm

• Copies itself onto your computer
• Stand-alone
Fred Cohen invented the computer virus.

Cohen created the first computer virus while studying for his PhD at University of Southern California.

Presented research a computer security seminar on November 10, 1983.

http://news.bbc.co.uk/2/hi/technology/3257165.stm
Early software exploits in the wild

1986 — BRAIN Virus

- Written by a pair of brothers in Pakistan. Given to tourists from the US who bought pirated programs.

1987 — Jerusalem Virus

- Discovered in Israel. Some thought written by the PLO as a way of punishing Israel. (Unlikely.)
- Rapidly “mutated.” (Used as a template for other viruses)

1989 — AIDS Trojan

- Sent out by “PC Cyborg” in Panama City to health care providers.

1992 - Michelangelo Virus
• Timed to go off on March 6, 1992. Massive public information campaign either prevented epidemic or overstated it.
Second Generation: Word Macro Viruses

“Concept” written by a Microsoft employee to demonstrate the problem.
Microsoft released this by accident at a developer’s conference
A very happy Christmas and my best wishes for the next year. Let this run and enjoy yourself. Browsing this file is no fun at all. Just type Christmas.
Self-propagating worms

“The Internet Worm” (November 1988) Written by Robert T. Morris

• Now a professor at MIT; father was famous security expert at NSA

Infected 2000 Unix systems

• 5 different attack vectors
• Attacked both DEC and Sun computers
• Anatomy was worrisome: included “DES” implementation.

Shut down the Internet

• First time the word “Internet” appears on front page of the New York Times.

Other examples include NIMDA, Code Red, Slammer
User-assisted worms

Melissa (March 1999)
ILOVEYOU (2000)
HAPPY99
Numerous screen savers
Understanding software exploitation

Three phases:

1. Identify vulnerable systems
2. Infect
3. Payload
How fast can a virus propagate?

Code Red propagation statistics

- Most hosts infected within 12 hours
- Source: CAIDA (Cooperative Association for Internet Data Analysis)
Sapphire / Slammer

Doubled every 8.5 seconds
Infected 90% of vulnerable hosts in 30 minutes.

- 74,855 hosts
- Reasons:
  1 packet infection

Aggregate Scans/Second in the 12 Hours After the Initial Outbreak.
Theoretical Minimum: 30 seconds?

Flash Worm Paper
- “Flash Worms: Thirty Seconds to Infect the Internet”
- Stuart Staniford, Gary Grim, Roelof Jonkman
- http://www.silicondefense.com/flash/
- August 16, 2001

Warhol Worms
- “How to Own the Internet in your Spare Time”
- Stuart Staniford, Vern Paxson, Nicholas Weaver
- http://www.cs.berkeley.edu/~nweaver/cdc.web/
- August 2002
Typical payloads

None

SPAM proxy

Hardware Destruction

CHI/Chernobyl Virus

April 26, 1999: One million computers destroyed

Cost: Korea $300M; China $291M
Access through telephones:
SF Bay War Dialing Survey [Garfinkel & Shipley, ’01]

<table>
<thead>
<tr>
<th>Time period</th>
<th>April 1997 — January 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialed Phone Numbers</td>
<td>5.7 million</td>
</tr>
<tr>
<td>Area codes</td>
<td>408, 415, 510, 650</td>
</tr>
<tr>
<td>Carriers Found</td>
<td>46,192</td>
</tr>
</tbody>
</table>

Phone Survey Finding 1:
Business & Residential exchanges look different.

Business

Lots of structure

Residential

Random distribution
Finding 2: Modems are friendly

94 modems per exchange, on average

- $\simeq 1\%$
- $\simeq 4.0\% - 6.1\%$ in the “top 10” exchanges (U.C. Berkeley and others)

87% of modems responded with a banner

- 335,412 lines of banners!
- Microsoft RAS gives no banner.
- Less than 2% had warning banners.

Friendly banners make it easier for an attacker to compromise the system.
Finding 3: Many modems are vulnerable

3% of all Shiva LAN Rover had no password on “root” account

• Shiva had documented “admin” account but not “root account.”

30% of Ascend concentrators gave “ascend%” prompt

Majority of Cisco routers gave command prompt.

• 25% were in “enable” mode!
Finding 4: Some significant systems were vulnerable

Oakland Fire Dispatch:

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHC</td>
<td>Display adjacent hazards caution</td>
</tr>
<tr>
<td>CN</td>
<td>Display caution notes for loc</td>
</tr>
<tr>
<td>CQ</td>
<td>Display coverages and quarters</td>
</tr>
<tr>
<td>CYC</td>
<td>Cycle Through Moveup Maps</td>
</tr>
<tr>
<td>DA</td>
<td>Display CJ days activity</td>
</tr>
<tr>
<td>EC</td>
<td>Emergency contact information</td>
</tr>
<tr>
<td>F</td>
<td>Display fire actsives</td>
</tr>
<tr>
<td>H</td>
<td>Hazardous materials research</td>
</tr>
<tr>
<td>INF</td>
<td>General info. file inquiry</td>
</tr>
<tr>
<td>M</td>
<td>Display recommended moveups</td>
</tr>
<tr>
<td>MED</td>
<td>Display medical notes for addr</td>
</tr>
<tr>
<td>MO</td>
<td>Memo system access</td>
</tr>
<tr>
<td>PC</td>
<td>Display prior calls</td>
</tr>
<tr>
<td>PI</td>
<td>Display prior incidents at loc</td>
</tr>
<tr>
<td>RUN</td>
<td>Display unit times and notes</td>
</tr>
<tr>
<td>SOP</td>
<td>Standard operating procedures</td>
</tr>
<tr>
<td>SR</td>
<td>Display shift roster/schedule</td>
</tr>
<tr>
<td>T</td>
<td>Display truck status screen #1</td>
</tr>
<tr>
<td>TIM</td>
<td>Display and reset timers</td>
</tr>
<tr>
<td>TSP</td>
<td>Test station printer</td>
</tr>
<tr>
<td>UP</td>
<td>Menu of user-written programs</td>
</tr>
<tr>
<td>US</td>
<td>Display Unit Status</td>
</tr>
<tr>
<td>UT</td>
<td>Display unit times</td>
</tr>
<tr>
<td>@</td>
<td>Log off</td>
</tr>
<tr>
<td>#</td>
<td>Telephone / pager directory</td>
</tr>
<tr>
<td>#T</td>
<td>Truck status screen (1-9)</td>
</tr>
<tr>
<td>?</td>
<td>Display this help screen</td>
</tr>
</tbody>
</table>
Other notable vulnerables:

Leased line control system

- Similar dialup shut down Worcester, MA airport in March 1997

Cody’s Bookstore order system

- Customer names & credit card numbers

Berkeley Pediatrics

- Concurrent DOS prompt

Numerous LAN Rovers at financial institutions

- Behind firewalls

Dialup for a high-voltage transmission line system
Unauthorized and unsecured modems are *still* a problem today.

- Legacy modems (frequently unknown)
- HVAC systems
- Elevators

http://www.heat-timer.com/?page=products
War dialing: Conclusions

War dialing is a *technique*.

The Shipley/Garfinkel study established that there is a vulnerability.

Dial-up modems continue to represent a vulnerability for many organizations.

Telephone scanning large areas finds more than scanning known blocks.

The most vulnerable dialups were not part of PBX exchanges.

**But who would exploit this?**
Road Island Teenager shuts down airport in Worcester, MA (March 10, 1997)

Airport operations disrupted.

600 homes left without telephone services.

Teenager discovered fiber-optic controller with a war dialer; types “shutdown” command.

http://www.cnn.com/TECH/computing/9803/18/juvenile.hacker/
Two weeks of unfettered access, through unsecured dialup.

Apparently a former employee
War Dialing Conclusions

Dial-up modems continue to represent a vulnerability for many organizations. Many organizations are not even aware that they have these modems operating. Telephone scanning large areas finds more than scanning known blocks.

- Many vulnerable dialups were not part of PBX exchanges.
“War Driving” (Shipley et. al.)

Materials:
- 802.11(b) card
- 8db antenna
- GPS
- Acquisition Software

Started by Shipley in 2000; now a popular geek pastime.
802.11(b) Security

2.4Ghz transmission; 11 Mbps
Access Points (APs) provide wireless connectivity.
SSID – Service Set Identifier --- Like an “SNMP” community
- A password transmitted in the clear
- 802.11 vendors initially claimed that SSID provided security.
- In 2000, WaveLAN drivers allowed “Any” SSID to associate with any observed AP
WEP – Wired Equivalent Privacy encryption algorithm.
- Poor encryption algorithm
- Poor key setup
- Nevertheless, provides limited security against people who follow the rules.
Latest Berkeley Findings (as of 6/21/2002)

Totals: 173 APs
SSID:
- 53 default SSIDs,
- 105 unique SSIDs
- 30.6% default SSIDs
WEP:
- 60 with WEP
- 113 without WEP (34.7%)
SSID:
- 45 Default without WEP (26%)
- 8 Default with WEP (4.6)

RED = NoWep & default SSID
Orange = NoWep
Green = Wep
Netstumbler: War driving for the masses
Stumbler Nation
Long Distance?

Some security officers feel that if AP is distanced from the street or on a high floor of a building they will be safe from network trespassers.

Shipley’s experiments show that it is possible to successfully make a network connection twenty-five (25) miles away from hilltops and high-rise buildings.
Hardware

Connecting to WLANs networks from across the bay.

24db dish
500mw amplifier
The view from a hilltop in Berkeley.
Why does 802.11 security matter?

Home Network
- Primary threats are unauthorized, anonymous access:
  - Spamming
  - Hacking
  - Anonymous threats
- Violations can result in loss of service

Corporate Networks
- Primary threat is theft of corporate information

Accidental Trespass
- Individuals may think they are associating with café, but actually be associating with nearby business
Typical Case (Mass)

MA business: attacker sat on a park bench and stole username & password of CEO and senior management using 802.11(b) sniffer. Attacker then logged into Exchange server and downloaded corporate email archives. Email was published on a website, resulting in $10M in damage to the company (lost contracts, renegotiated contracts, etc.)
802.11 solutions

Place APs
- Outside corporate LANs
- in DMZs
- On separate Internet connections

“arpwatch” to detect unknown/unauthorized users.

IPsec

802.1x (support is not uniform)

Enterprise solutions from Cisco, Newberry Networks
Today

Hackers have grown up

Most hacking seems to be criminal-related. (Make money fast.)

International scope.
Cyberwar and Cyberterrorism
“first cyberwar.”

Unsolicited e-mail hits targets in America in first cyberwar

April 8 — Think of it as the first cyberwar. While missiles explode over Belgrade, refugees from Kosovo pour into Albania and politics play themselves out on a global scale, some Serbs are fighting for American support, using laptops as their weapon.

IN RECENT DAYS, electronic mail attacking the NATO bombing campaign has been lobbed by at least 25 computers in Yugoslavia, clogging the in-boxes of well more than 10,000 Internet users, mostly in the U.S. Many people on the receiving end are annoyed by this unwanted Serbian “spam,” which at the very least is a pain to delete.

BOOMERANG EFFECT
For many recipients, there’s an added, irksome twist. Hundreds have sent reply e-mail messages demanding to be taken off the Yugoslav mailing lists. In many cases, copies of the requests are then circulated to everyone who received the message in the first place and that engenders new messages from new sources. That’s a lot of e-mail.

There are, for instance, 6,500 names on the mailing list of the Belgrade Academic Association for Equal Rights in the World, an organization whose mail is boomeranging all over the world.
This is was not cyberwar
Wired Magazine: “The Great Cyberwar of 2002”

10 July 2002
PFW Announcement appears on websites
CNN
USA Today
The Guardian
DISNEY.COM

http://www.wired.com/wired/archive/6.02/cyberwar.html
Wired Magazine...

14 July
- Western US States Suffer Blackout
- 500KV Transmission line shut down by hackers
- 35 deaths

15 July
- Second Ultimatum Issued
16 July
- Midair collision of 2 jets
- 463 dead
- All US commercial aviation grounded
Wired Magazine

21 July
- Computer-controlled Chemical factory blows up in Detroit, taking 1/2 the city with it

22 July
- Trans Alaska pipeline burst near Valdez

2 August
- Microwave bomb attack on Pentagon
National Strategy to Secure Cyberspace

Mostly a bust

- [http://www.whitehouse.gov/pcipb/](http://www.whitehouse.gov/pcipb/)
- Largely recommended antivirus and firewalls
FBI’s InfraGard

Started in 2001 by FBI; now incorporated as a non-profit
Local chapters.
24x7 system to communicate cyberthreats.
Off-the-record discussions of cybersecurity issues.
High-level meetings between government and industry
Key interest is leveraging of cyber structure by “terrorists.”

Phyllis Schneck, InfraGard’s National Chair
Members must pass FBI background check

Small and medium business to
Fortune 500

Interview in SC Magazine, March 2004
US Department of Homeland Security’s National Cyber Security Division (NCSD)

- US Computer Emergency Readiness Team (US-CERT)
- Chief Information Security Officers Forum (for federal CISOs)
- Forum of Incident Response and Security Teams (FIRST; exchanges information about incidents)
- Cyber Interagency Incident Management Group
- Critical Infrastructure Warning Information Network (a private, secure, and survivable network for use in the event of an information outage)
What the government isn’t doing for private industry:

No tax credits
No cost sharing
No real regulations
Do these worms actually cause problems?

Number of infected messages blocked by MessageLabs over 12 months

- SoBig.F: 33.3m
- Klez.h: 8.3m
- MyDoom.A: 54.1 m
Regulatory approaches:

Health Insurance Portability and Accountability Act (HIPAA)
- Businesses must secure health care information.

Sarbanese-Oxley Act (SEC Rule 17a)
- Financial reporting regulation; businesses must document their risks
References:

“Who’s Driving the Security Train,” Investigative report, pp. 6, 7, 8, 22, Computerworld, March 8, 2004
Cyber Report Cards


2003 A grades:
- Nuclear Regulatory Commission C->A
- National Science Foundation: D- -> A-

2003 B grades:
- Social Security Administration: B- -> B+
- Department of Labor: C+ -> B

2003 C grades:
- Department of Education: D -> C+  
- Department of Veteran’s affairs: F -> C  
- Environmental Protection Agency: D- -> C  
- Small Business Administration: F -> C-  
- Agency for International Devt.: F -> C-

2003 D grades:
- Department of Defense: F -> D  
- General Services Administration: D -> D  
- Department of the Treasury: F -> D  
- Office of Personnel Mgt: F -> D-  
- NASA: D+ -> D-  
- Department of Health and Human Services: F -> D-

2003 F grades:
- Department of Energy: F -> F  
- Department of Justice: F -> F  
- Department of the Interior: F -> F  
- Department of Agriculture: F -> F  
- Department of Housing and Urban Development: F -> F  
- Department of State: F -> F  
- Department of Homeland Security: F
Secure Coding
Saltzer & Schroeder
Seven Design Principles

Least Privilege
Economy of Mechanism
Complete Mediation
Open design
Separation of privilege
Least Common Mechanism
Psychological acceptability
1988: Morris Internet Worm

fingerd.c:

```c
char line[512];
...
line[0] = '\0';
gets(line);
```

Results in 6,000 computers being infected.
Fingerd bug fix

```c
line[0] = '\0';
gets(line);
```

Becomes

```c
memset(line, 0, sizeof(line));
fgets(line, sizeof(line), stdin);
```
Miller, Fredrickson & So

1995, “Fuzz Revisited”
1990 Fuzz Findings

Between 25% and 33% of Unix utilities crashed or hung by supplying them with unexpected inputs

- End-of-file in the middle of an input line
- Extra-long input
- Letters for numbers, etc.

In one case, the entire computer crashed.
1995: Fuzz Revisited

Vendors not overly concerned about bugs in their programs

“Many of the bugs discovered (approximately 40%) and reported in 1990 are still present in their exact form in 1995.

- Code was made freely available via anonymous FTP
- Exact random data streams used in testing were made available
- 2000 copies of the tools were downloaded from FTP

“It is difficult to understand why a vendor would not partake of a free and easy source of reliability improvements”
1995 Fuzz Revisited, cont.

Lowest failure rates were for the Free Software Foundation’s GNU utilities (7%)

- FSF had strict coding rules that forbid the use of fixed-length buffers.

Many X clients would readily crash when fed random streams of data.
2000 Fuzz against NT

45% of all programs expecting user input could be crashed
100% of Win32 programs could be crashed with Win32 messages

LRESULT CALLBACK
w32_wnd_proc (hwnd, msg, wParam, lParam)
{
    . . .
    POINT *pos;
pos = (POINT *)lParam;
    . . .
    if (TrackPopupMenu((HMENU)wParam,
        flags, pos->x, pos->y, 0, hwnd,
        NULL))
        . . .
}
Fuzz Today

eEye Digital Security does network fuzz testing

- http://www.eeye.com/

Most remote crashes can be turned into remote exploits

Retina Vulnerability Scanner
Morris Worm II

Exploited Sendmail’s WIZ and DEBUG commands
Cracked passwords
Caused havoc by hyper-replication (common problem)
Avoiding Security-Related Bugs

Avoid bugs in general
Test with non-standard input
Look for back doors
  - (theoretically impossible to do perfectly)
Design Principles

Carefully design the program before you start.
  - Remember: you will either design it before you start writing it, or while you are writing it. But you will design it.

Document your program before writing the code.
Make critical portions of the program as small as possible.

Resist adding new features. The less code you write, the less likely you are to introduce new bugs.
Design Principles 2

Resist rewriting standard functions. (Even when standard libraries have bugs.)

Be aware of race conditions:

- Deadlock conditions: More than one copy of your program may be running at the same time!
- Sequence conditions: Your code does not execute automatically!
  
  Do not stat() then open()
  Do not use access()

Write for clarity and correctness before optimizing.
Coding Standards

Check all input arguments. Always.
Check arguments you pass to system calls
Return Codes

Check *all system call returns.*

- \( \text{fd} = \text{open}(\text{filename}, \text{O_RDONLY}) \) *can fail!*
- \( \text{read}(\text{fd}, \text{buf}, \text{sizeof}(\text{buf})) \) *can fail*
- \( \text{close}(\text{fd}) \) *can fail!*

Use `perror("open")` or `err(1,"open failed:")` to tell the user *why something failed.*

Log important failures with `syslog()`
File Names

Always use full pathnames
Check all user-supplied input (filenames) for shell metacharacters
If you are expecting to create a new file, open with `O_EXCL|O_CREAT` to fail if the file exists. If you are expecting an old file, open with `O_EXCL` to fail if it does not exist.
Temporary Files

Use `tmpfile()` or `mkstemp()` to create temporary files.

```
FILE *f = tmpfile(void);
int fd = mkstemps(char *template, int suffixlen);
```

Never use `mktemp()` or `tmpnam()`.
## Functions to avoid

<table>
<thead>
<tr>
<th>Avoid</th>
<th>Use instead</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>gets()</code></td>
<td><code>fgets()</code></td>
</tr>
<tr>
<td><code>strcpy()</code></td>
<td><code>strncpy()</code></td>
</tr>
<tr>
<td><code>strcat()</code></td>
<td><code>strncat()</code></td>
</tr>
<tr>
<td><code>sprintf()</code></td>
<td><code>snprintf()</code></td>
</tr>
<tr>
<td><code>v sprintf()</code></td>
<td><code>vsnprintf()</code></td>
</tr>
</tbody>
</table>
Coding Standards 2

Check arguments passed to program via environment variables
  - e.g., HOME, PAGER, etc.

Do bounds checking on every variable.
  - If a variable should be 0..5, make sure it is not -5 or 32767
  - Check lengths before you copy.
Coding Standards...

Use `assert()` within your program.

```c
j = index(buf,,);  
assert(j>0);
```
Coding Standards

Avoid C functions that use statically-allocated buffers
- These are the rules for multi-threaded coding as well!

don’t use:

```c
struct tm *
localtime(const time_t *clock);
```

Use:

```c
struct tm *
localtime_r(const time_t *clock,
struct tm *result);
```
Logging

Design your logs to be parsed by a computer
Using syslog() if possible.
Include a heartbeat log
RFC 1750:
Randomness Recommendations

Keep seeds for RNGs secret!
Don’t seed with:
- Time of day
- Serial number
- Ethernet address

Beware using:
- Network timing
- “Random selections” from databases

Use:
- Analog input devices (/dev/audio)

Never use rand()
Passwords

Store the hash of passwords and a salt, not the passwords themselves

Also store:

- Date password was changed
- # of invalid password attempts
- Location of invalid password attempt

Don’t restrict password character set

Try flipping password case (just to be nice)
Limit Privilege

Limit access to the file system
- `chroot()` and `jail()` under Unix
- Restrict use of C compiler
Programs that need privilege
(SUID/SGID/Admin)

“Don’t do it. Most of the time, it’s not necessary” (Wood & Kochan, *Unix System Security*, 1985)

Don’t use **root** or **Administrator** privs when you can create a specialty group.

Use permissions as early as possible to open files, etc., then **give up the privs**.

Avoid embedding general-purpose command languages, interfaces, etc., in programs that require privilege

Erase execution environment (PATH, etc.) and build from scratch

Use full path names
Tips for Network Programs

Do reverse lookups on all connections
Include load shedding or load limiting
Include reasonable timeouts
Make no assumptions about content of input data
Make no assumption about the amount of input
Call authd if possible — but don’t trust the results
More Network Tips

Use SSL if at all possible.
Include support for using a proxy
Build in graceful shutdown:
  - From signals
  - From closed network pipes
Include “self recognition” so that more than one copy of the server doesn’t run at the same time.
Try not to create a new network protocol
Don’t hard-code port numbers
Don’t trust “privileged” ports, IP source addresses
Don’t send passwords in clear text.
Web-based Applications

Validate all information from the client
  - Don’t trust the content of HIDDEN fields
  - Verify Cookies
  - Digitally sign or MAC all information

Use prepared SQL statements
  - Never: sprintf(\%s,"select * where username=\%s",username)
  - Always: “select * where username=?”
Programming Languages

Avoid C, C++ if possible
Use perl’s tainting feature (-T)
Be careful with Java’s class loader
Be careful with eval():
  - perl
  - python
  - shell `
Things to avoid

Don’t provide shell escapes in interactive programs
Be very careful with system() and popen() calls
Do not create files in world-writable directories
Use setrlimit() to avoid dumping core
Before you Finish

Read though your code
  - How would you attack your own code?
  - What happens if it gets unexpected inputs?
  - What happens if you place a delay between system calls?

Test your assumptions:
  - Run by *root*. Run by *nobody*
  - Run in a different directory
  - What is `/tmp` or `/tmp/root` doesn’t exist?
Testing

Test with a testing tool:
- tcov (SVR4)
- gcov (GNU)

Commercial Testing tools:
- CodeCenter
- PurifyPlus
More testing

Stress Test:
- Low memory
- Filled disk

Test Missing DLLs
- Internet Explorer fails open if msrating.dll is not installed

Monitor all reads & writes
- Holodeck (Windows)
- dtrace (Solaris)
Code Review

Walk through your code with another competent programmer
Simply putting your code on the Internet is not the same as having it reviewed!
Famous Open-Source Problems

Kerberos random number generator
Sendmail – DEBUG and WIZ
fingerd

Less famous, but affecting me personally:
  - Hylafax program
  - NNTPcache