Computer and Network Security

Dan Boneh and John Mitchell

https://courseware.stanford.edu/pg/courses/CS155
What’s this course about?

- Intro to computer and network security
- Some challenging fun projects
  - Learn about attacks
  - Learn about preventing attacks
- Lectures on related topics
  - Application and operating system security
  - Web security
  - Network security

Some overlap with CS241, Web Security
Not a course on Cryptography (take CS255)
Organization

- Application and OS security (5 lectures)
  - Buffer overflow project
  - Vulnerabilities: control hijacking attacks, fuzzing
  - Prevention: System design, robust coding, isolation

- Web security (4 lectures)
  - Web site attack and defenses project
  - Browser policies, session mgmt, user authentication
  - HTTPS and web application security

- Network security (6 lectures)
  - Network traceroute and packet filtering project
  - Protocol designs, vulnerabilities, prevention
  - Malware, botnets, DDoS, network security testing

- A few other topics
  - Cryptography (user perspective), digital rights management, final guest lecture, ...
General course info (see web)

- Prerequisite: Operating systems (CS140)
- Textbook: none – reading online
- Coursework
  - 3 projects, 2 homeworks, final exam
  - grade: 0.25 H + 0.5 P + 0.25 F
- Teaching assistants
  - Hariny Murli, Hristo Bojinov
- Occasional optional section
  - Experiment this year: Live Meeting
Announcements

Welcome to CS 155!

Welcome to this spring’s course. If you are enrolled in CS 155, please join the course on CourseWare. Basic course information is provided in the course FAQ, which you can find under the “More” heading when you are viewing the CS155 pages on CourseWare.

Some information posted on CourseWare may require you to log in. Do this by clicking on “stanford login” when you first go to Courseware, or clicking “Login” in the red banner if you are viewing CourseWare pages. Then use your regular SUNet ID and password.

If you have questions, ask them on the Discussion Forum instead of by sending email to the course staff.

Posted Sat, Mar 20 0:00 PM by John Mitchell
What is security?

- **System correctness**
  - If user supplies expected input, system generates desired output

- **Security**
  - If attacker supplies unexpected input, system does not fail in certain ways
What is security?

- **System correctness**
  - Good input $\Rightarrow$ Good output

- **Security**
  - Bad input $\not\Rightarrow$ Bad output
What is security?

- **System correctness**
  - More features: better

- **Security**
  - More features: can be worse
Security properties

- **Confidentiality**
  - Information about system or its users cannot be learned by an attacker

- **Integrity**
  - The system continues to operate properly, only reaching states that would occur if there were no attacker

- **Availability**
  - Actions by an attacker do not prevent users from having access to use of the system
Security is about

- Honest user (e.g., Alice, Bob, ...)
- Dishonest Attacker
- How the Attacker
  - Disrupts honest user’s use of the system (Integrity, Availability)
  - Learns information intended for Alice only (Confidentiality)
Network security

Network Attacker

Intercepts and controls network communication

Alice
Web security

Web Attacker
Sets up malicious site visited by victim; no control of network

Alice
Operating system security

OS Attacker
Controls malicious files and applications

Alice
Confidentiality: Attacker does not learn Alice's secrets
Integrity: Attacker does not undetectably corrupt system's function for Alice
Availability: Attacker does not keep system from being useful to Alice
Current Trends
Historical hackers (prior to 2000)

Profile:
- Male
- Between 14 and 34 years of age
- Computer addicted
- No permanent girlfriend

No Commercial Interest !!!

Source: Raimund Genes
Typical Botherder: 0x80” (pronounced X-eighty)

High school dropout
- “…most of these people I infect are so stupid they really ain't got no business being on the Internet in the first place.”

Working hours: approx. 2 minutes/day to manage Botnet

Monthly earnings: $6,800 on average

Daily Activities:
- Chatting with people while his bots make him money
- Recently paid $800 for an hour alone in a VIP room with several dancers

Job Description:
- Controls 13,000+ computers in more than 20 countries
- Infected Bot PCs download Adware then search for new victim PCs
- Adware displays ads and mines data on victim's online browsing habits.
- Bots collect password, e-mail address, SS#, credit and banking data
- Gets paid by companies like TopConverting.com, GammaCash.com, Loudcash, or 180Solutions.
Some things in the news

- Nigerian letter (419 Scams) still works:
  - Michigan Treasurer Sends 1.2MUSD of State Funds !!!
- Many zero-day attacks
  - Google, Excel, Word, Powerpoint, Office …
- Criminal access to important devices
  - Numerous lost, stolen laptops, storage media, containing customer information
  - Second-hand computers (hard drives) pose risk
- Vint Cerf estimates ¼ of PCs on Internet are bots
Trends for 2010

- Malware, worms, and Trojan horses
  - spread by email, instant messaging, malicious or infected websites
- Botnets and zombies
  - improving their encryption capabilities, more difficult to detect
- Scareware – fake/rogue security software
- Attacks on client-side software
  - browsers, media players, PDF readers, etc.
- Ransom attacks
  - malware encrypts hard drives, or DDOS attack
- Social network attacks
  - Users’ trust in online friends makes these networks a prime target.
- Cloud Computing - growing use will make this a prime target for attack.
- Web Applications - developed with inadequate security controls
- Budget cuts - problem for security personnel and a boon to cyber criminals.

Same list in Oklahoma Monthly Security Tips Newsletter
Trends

Vulnerability Disclosures
2000-2009

Percentage of Vulnerability Disclosures
Attributed to Top 10 Vendors
2009

Others: 77%
Top 10 Vendors: 23%

Source: IBM X-Force®
Operating system vulnerabilities

Vulnerability Disclosures Affecting Operating Systems
2005-2009

Critical and High Vulnerability Disclosures
Affecting Operating Systems
2005-2009

Source: IBM X-Force®
Reported Web Vulnerabilities "In the Wild"

Data from aggregator and validator of NVD-reported vulnerabilities
Web vs System vulnerabilities

Evolution of the number of vulnerabilities by years

- Web
- System

XSS peak
Botnet Lifecycle

**Propagation**
- Compromised host activity
- Network probe and other activity
- Recognizable activity on newly infected host
Recent work on malware distribution

- Blogs are widely used
  - 184 Million blogs world-wide
  - 73% of internet users have read a blog
  - 50% post comments
- Blogs have automated Linkbacks
  - Facilitate cross-referencing
  - Exploited by spammers
- We carried out a 1-year study
  - Analyzed 10 million spam samples
  - Gained insight on attacker’s method of operation and resources
  - Propose a defense against blog spams
How big is the problem?

Source: Akismet.com

One blog spam can reach thousands of users.

Total spam: 13,275,940,950
Total ham: 2,701,440,026
Honeyblog Experiment

Blog acting as potential target for spamming

- Hosted a real blog (dotclear) with a modified TrackBack mechanism
- Record TrackBacks
- Passive fingerprinting
- Sample the lure site
Malware installation

- Servers submit Trackback spam
- Spam points to Social network site exploited as relay site
- Relay site links to lure sites with purported adult content
- Lure site badgers user to download fake video plugins hosted on malware site

- TrojanDownloader:Win32/Zlob.gen!dll
- Trojan.Popuper.origin
- Downloader/Zlob.LI
Trackback spam example

Apparent Bayesian poisoning against spam filters:

[tile] => Please teacher hentai pics
[excerpt] => pics Please teacher hentai pics
...
[blog_name] => Please teacher hentai pics
Number of notifications detected

Trackback Spam

Mar-Apr 2007

May-Jun 2007

July 2007-Apr 2008

Number of Spams

0

25000

50000

75000

100000
Number of IP Addresses

Origin

IP Geolocation Distribution


Russia | USA | Germany | UK
User agents reported to honeyblog
Web attack toolkit: MPack

- **Basic setup**
  - Toolkit hosted on web server
  - Infects pages on that server
  - Page visitors get infected

- **Features**
  - Customized: determines exploit on the fly, based on user’s OS, browser, etc
  - Easy to use: management console provides stats on infection rates
  - Customer care toolkit can be purchased with one-year support contract!
SilentBanker

Advanced Information Stealing

User requests login page

Proxy intercepts request and adds fields

Bank sends login page needed to log in

When user submits information, also sent to attacker

The attacker can then use this information to log into the user’s bank account at a later date

Credit: Zulfikar Ramzan
Estonia: network attack

Jaak Aaviksoo, Minister of Defence
Steal cars with a laptop

NEW YORK - Security technology created to protect luxury vehicles may now make it easier for tech-savvy thieves to drive away with them.

In April ‘07, high-tech criminals made international headlines when they used a laptop and transmitter to open the locks and start the ignition of an armor-plated BMW X5 belonging to soccer player David Beckham, the second X5 stolen from him using this technology within six months.

… Beckham's BMW X5s were stolen by thieves who hacked into the codes for the vehicles' RFID chips …
IPhone Flaw Lets Hackers Take Over, Security Firm Says

A team of computer security consultants say they have found a flaw in Apple’s wildly popular iPhone that allows them to take control of the device.

Charles Miller, shown on his iPhone, said that after finding a hole in security, “you were in complete control.”

by John Schwartz
Published: July 23, 2007

Next Article in Technology (4 of 17) »
iPhone attack  (summer 2007)

- iPhone Safari downloads malicious web page
  - Arbitrary code is run with administrative privileges
  - Can read SMS log, address book, call history, other data
  - Can perform physical actions on the phone.
    - system sound and vibrate the phone for a second
    - could dial phone numbers, send text messages, or record audio (as a bugging device)
  - Transmit collected data over network to attacker

See http://www.securityevaluators.com/iphone/
iPhone security measures

“Reduced attack surface”
- Stripped down and customized version of Mac OS X
  - does not have common binaries such as bash, ssh, or even ls.
- MobileSafari - many features of Safari have been removed
  - No Flash plug-in, many file types cannot be downloaded

Some internal protection
- If USB syncing with iTunes, file system cannot be mounted
- File system accessible to iTunes is chroot’ed

Weak security architecture
- All processes of interest run with administrative privileges
- iPhone does not utilize some widely accepted practices
  - Address randomization
    - Each time a process runs, the stack, heap, and executable code located at precisely the same spot in memory
  - Non-executable heaps
    - Buffer overflow on heap can write executable instructions
Analysis methods

- Extract and statically analyze binaries
  - Using jailbreak and iPhone Interface,
- Audit related open-source code
  - MobileSafari and MobileMail applications are based on the open source WebKit project
- Dynamic analysis, or “fuzzing”
  - Sending malformed data to cause a fault or crash
  - Look at error messages, memory dump, etc.
- MobileSafari attack discovered using fuzzing
  - What kind of vulnerability do you think it was?
Suggestions for improvement

- Run applications as an unprivileged user
  - This would result in a successful attacker only gaining the rights of this unprivileged user.

- chroot apps to prevent access to unrelated data
  - MobileSafari does not need access to email or SMS msgs
  - MobileMail does not need access to browsing history

- Add heap and stack address randomization
  - This will serve to make the development of exploits for vulnerabilities more difficult

- Memory protection: no pages both writable and executable

See http://www.securityevaluators.com/iphone/exploitingiphone.pdf
- Spam service
- Rent-a-bot
- Cash-out
- Pump and dump
- Botnet rental
# Underground Goods and Services

<table>
<thead>
<tr>
<th>Rank</th>
<th>Last</th>
<th>Goods and Services</th>
<th>Current</th>
<th>Previous</th>
<th>Prices</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>Bank accounts</td>
<td>22%</td>
<td>21%</td>
<td>$10-1000</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Credit cards</td>
<td>13%</td>
<td>22%</td>
<td>$0.40-$20</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>Full identity</td>
<td>9%</td>
<td>6%</td>
<td>$1-15</td>
</tr>
<tr>
<td>4</td>
<td>N/R</td>
<td>Online auction site accounts</td>
<td>7%</td>
<td>N/A</td>
<td>$1-8</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>Scams</td>
<td>7%</td>
<td>6%</td>
<td>$2.50/wk - $50/wk (hosting); $25 design</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>Mailers</td>
<td>6%</td>
<td>8%</td>
<td>$1-10</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>Email Addresses</td>
<td>5%</td>
<td>6%</td>
<td>$0.83-$10/MB</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>Email Passwords</td>
<td>5%</td>
<td>8%</td>
<td>$4-30</td>
</tr>
<tr>
<td>9</td>
<td>N/R</td>
<td>Drop (request or offer)</td>
<td>5%</td>
<td>N/A</td>
<td>10-50% of drop amount</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>Proxies</td>
<td>5%</td>
<td>6%</td>
<td>$1.50-$30</td>
</tr>
</tbody>
</table>
Why are there security vulnerabilities?

Lots of buggy software...

- Why do programmers write insecure code?
- Awareness is the main issue

Some contributing factors

- Few courses in computer security
- Programming text books do not emphasize security
- Few security audits
- C is an unsafe language
- Programmers have many other things to worry about
- Legacy software (some solutions, e.g. Sandboxing)
- Consumers do not care about security
- Security is expensive and takes time
If you remember only one thing from this course:

A vulnerability that is “too complicated for anyone to ever find” will be found!

We hope you remember more than one thing
Ethical use of security information

We discuss vulnerabilities and attacks
- Most vulnerabilities have been fixed
- Some attacks may still cause harm
- Do not try these at home or anywhere else

Purpose of this class
- Learn to prevent malicious attacks
- Use knowledge for good purposes
Law enforcement

Sean Smith
- Melissa virus: 5 years in prison, $150K fine

Ehud Tenenbaum (“The Analyzer”)
- Broke into US DoD computers
- 6 mos service, suspended prison, $18K fine

Dmitry Sklyarov
- Broke Adobe ebooks
- Prosecuted under DMCA
Difficult problem: insider threat

- Easy to hide code in large software packages
  - Virtually impossible to detect back doors
  - Skill level needed to hide malicious code is much lower than needed to find it
  - Anyone with access to development environment is capable

slides: Avi Rubin
Example insider attack

Hidden trap door in Linux, Nov 2003
- Allows attacker to take over a computer
- Practically undetectable change
- Uncovered by anomaly in CVS usage

Inserted line in wait4()

```c
if ((options == (__WCLONE|__WALL)) && (current->uid = 0))
    retval = -EINVAL;
```

- Looks like a standard error check
- Anyone see the problem?

See: http://lwn.net/Articles/57135/
Example #2

- Rob Harris case - slot machines
  - an insider: worked for Gaming Control Board
- Malicious code in testing unit
  - when testers checked slot machines
    - downloaded malicious code to slot machine
  - was never detected
  - special sequence of coins activated “winning mode”
- Caught when greed sparked investigation
  - $100,000 jackpot
Example #3

Breeder’s cup race

- Upgrade of software to phone betting system
- Insider, Christopher Harn, rigged software
- Allowed him and accomplices to call in
  - change the bets that were placed
  - undetectable
- Caught when got greedy
  - won $3 million

http://horseracing.about.com/library/weekly/aa110102a.htm
Software dangers

- Software is complex
  - top metric for measuring # of flaws is lines of code
- Windows Operating System
  - tens of millions of lines of code
  - new “critical” security bug announced every week
- Unintended security flaws *unavoidable*
- Intentional security flaws *undetectable*
What code can we trust?
- Consider "login" or "su" in Unix
- Is RedHat binary reliable?
- Does it send your passwd to someone?

Can't trust binary so check source, recompile
- Read source code or write your own
- Does this solve problem?

Reflections on Trusting Trust, http://www.acm.org/classics/sep95/
Compiler backdoor

- This is the basis of Thompson's attack
  - Compiler looks for source code that looks like login program
  - If found, insert login backdoor (allow special user to log in)

- How do we solve this?
  - Inspect the compiler source
C compiler is written in C

Change compiler source S

```c
compiler(S) {
    if (match(S, "login-pattern")) {
        compile (login-backdoor)
        return
    }
    if (match(S, "compiler-pattern")) {
        compile (compiler-backdoor)
        return
    }
    .... /* compile as usual */
}
```
Clever trick to avoid detection

- Compile this compiler and delete backdoor tests from source
  - Someone can compile standard compiler source to get new compiler, then compile login, and get login with backdoor
- Simplest approach will only work once
  - Compiling the compiler twice might lose the backdoor
  - But can making code for compiler backdoor output itself
    - (Can you write a program that prints itself? Recursion thm)
- Read Thompson's article
  - Short, but requires thought
Social engineering

Many attacks don't use computers
- Call system administrator
- Dive in the dumpster

Online versions
- send trojan in email
- picture or movie with malicious code
Organization

Application and OS security (5 lectures)
- Buffer overflow project
- Vulnerabilities: control hijacking attacks, fuzzing
- Prevention: System design, robust coding, isolation

Web security (4 lectures)
- Web site attack and defenses project
- Browser policies, session mgmt, user authentication
- HTTPS and web application security

Network security (6 lectures)
- Network traceroute and packet filtering project
- Protocol designs, vulnerabilities, prevention
- Malware, botnets, DDoS, network security testing

A few other topics
- Cryptography (user perspective), digital rights management, final guest lecture, ...