CS 155 Spring 2009

Computer Security

Dan Boneh and John Mitchell

http://crypto.stanford.edu/cs155

What's this course about?

- Some challenging fun projects
  - Learn about attacks
  - Learn about preventing attacks
- Lectures on many topics
  - Application security
  - Operating system security
  - Network security
  - Web security
    - Some overlap with CS142; redesign for next year

not a course on Cryptography (take CS255)

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
  - Hristo Bojinov, Indrajit Khare, Gary Luu
- Occasional optional section
  - Fridays, 4:15 - 5:05, Gates B03

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)*

Washington Post: *Invasion of the Computer Snatchers*

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 in 2007-08
  - 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
- Gozi trojan steals data from SSL streams
  - Undetected for 50 days
- Vint Cerf estimates ¼ of PCs on Internet are bots

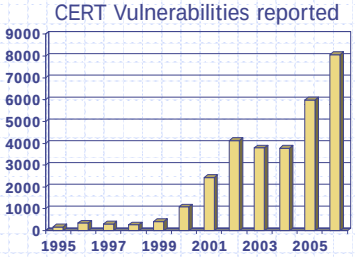
Trends

- Malicious software levels consistently rising
  - More malicious software in '08 than all previous years combined
  - By all accounts, '09 will see increasing rise
  - Good vs. bad software inflection point
- Underground economy and supply chain
  - Lowers bar for who can participate in cybercrime
- Web will continue as an attack vector
  - Popular medium, rich content, remote access to your home/office
- Targeted attacks
  - Necessitate defense-in-depth protection
- Attackers starting at the supply chain
  - Infected digital picture frames

Credit: Zulfikar Ramzan

How big is the security problem?

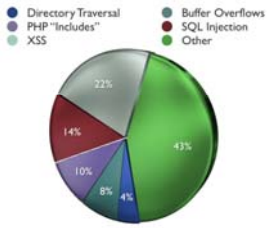
CERT Vulnerabilities reported



http://www.cert.org/stats/

Most-common attacks on systems

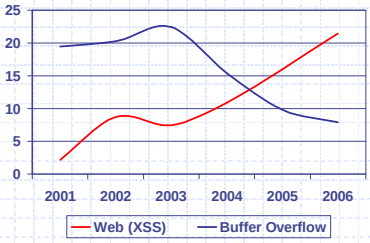
- 2006 MITRE CVE stats:
  - 21.5 % of CVEs were XSS
  - 14 percent SQL injection
  - 9.5 percent php "includes"
  - 7.9 buffer overflow



2005 was the first year that XSS jumped ahead of buffer overflows ...

Vulnerability Stats: web is "winning"

Majority of vulnerabilities now found in web software



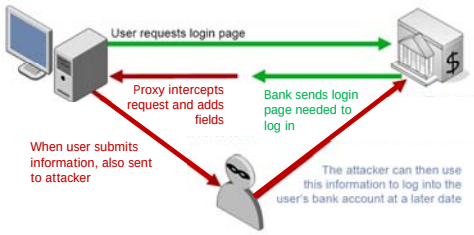
Source: MITRE CVE trends

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



Credit: Zulfikar Ramzan

Steal cars with a laptop

- NEW YORK - Security technology created to protect luxury vehicles may now make it easier for tech-savy 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 ...

The New York Times — Technology

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

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

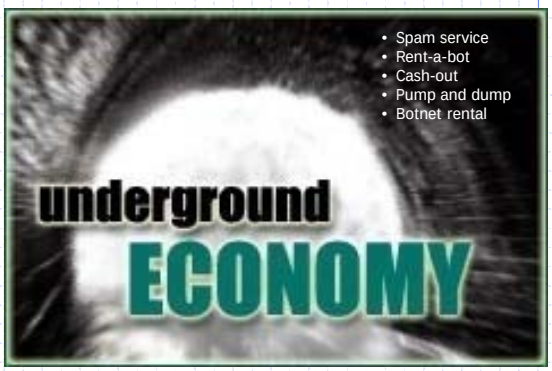
Analysis methods

- Extract and statically analyze binaries
  - Using jailbreak and iPhoneInterface,
- 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 deos 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



Underground goods and services

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

Credit: Zulfikar Ramzan

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 anyplace 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()

```
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*

Ken Thompson

- 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

```
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