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

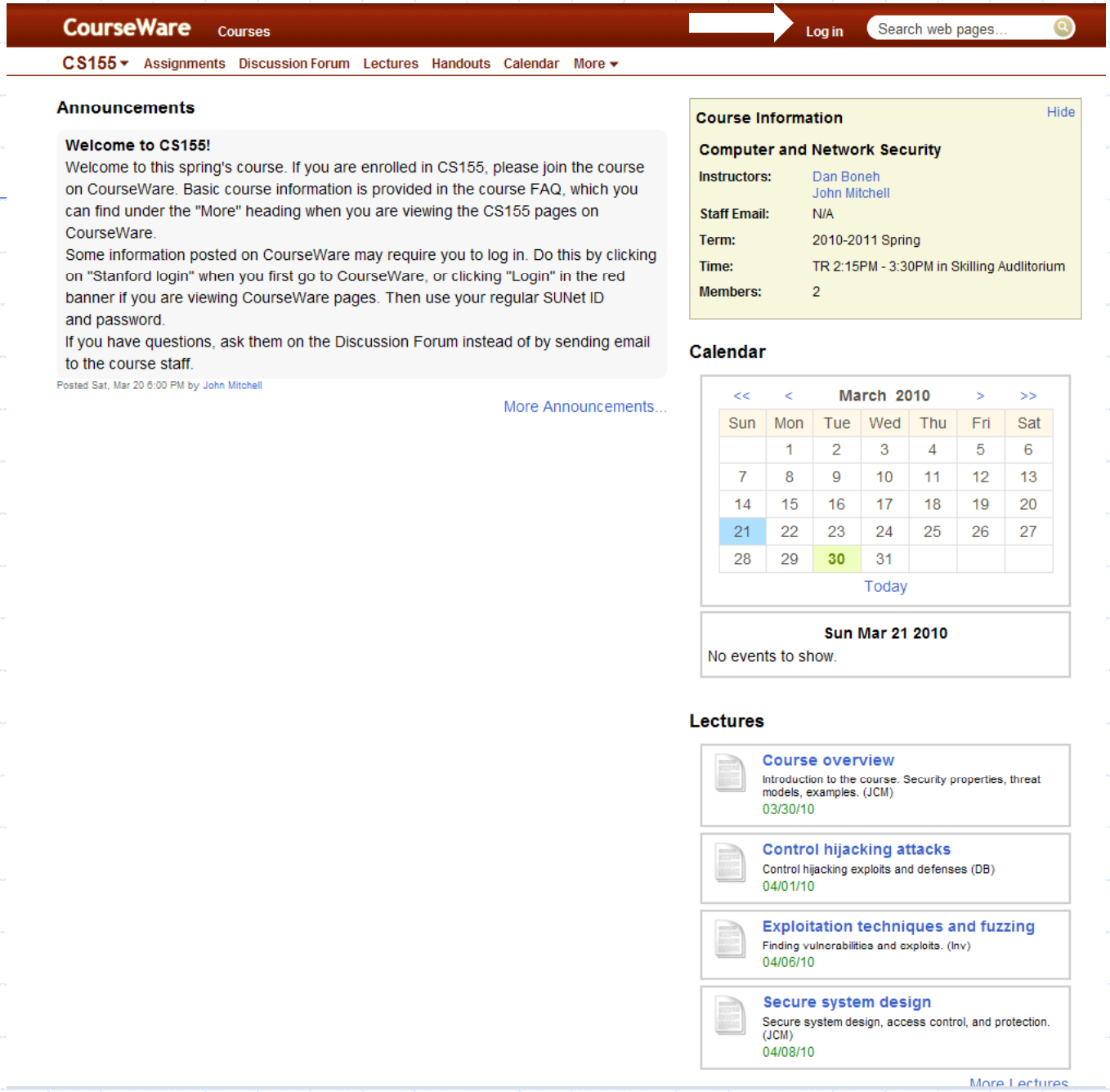
Not a course on Cryptography (take CS255)

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


CourseWare
Courses
Log in
Search web pages...
CS155
Assignments
Discussion Forum
Lectures
Handouts
Calendar
More
Announcements
Welcome to CS155!
Welcome to this spring's course. If you are enrolled in CS155, 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 6:00 PM by John Mitchell
More Announcements...
Course Information
Hide
Computer and Network Security
Instructors: Dan Boneh, John Mitchell
Staff Email: N/A
Term: 2010-2011 Spring
Time: TR 2:15PM - 3:30PM in Skilling Audlitorium
Members: 2
Calendar
<< < March 2010 > >>
Sun Mon Tue Wed Thu Fri Sat
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
Today
Sun Mar 21 2010
No events to show.
Lectures
Course overview
Introduction to the course. Security properties, threat models, examples. (JCM)
03/30/10
Control hijacking attacks
Control hijacking exploits and defenses (DB)
04/01/10
Exploitation techniques and fuzzing
Finding vulnerabilities and exploits. (Inv)
04/06/10
Secure system design
Secure system design, access control, and protection. (JCM)
04/08/10
More Lectures

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

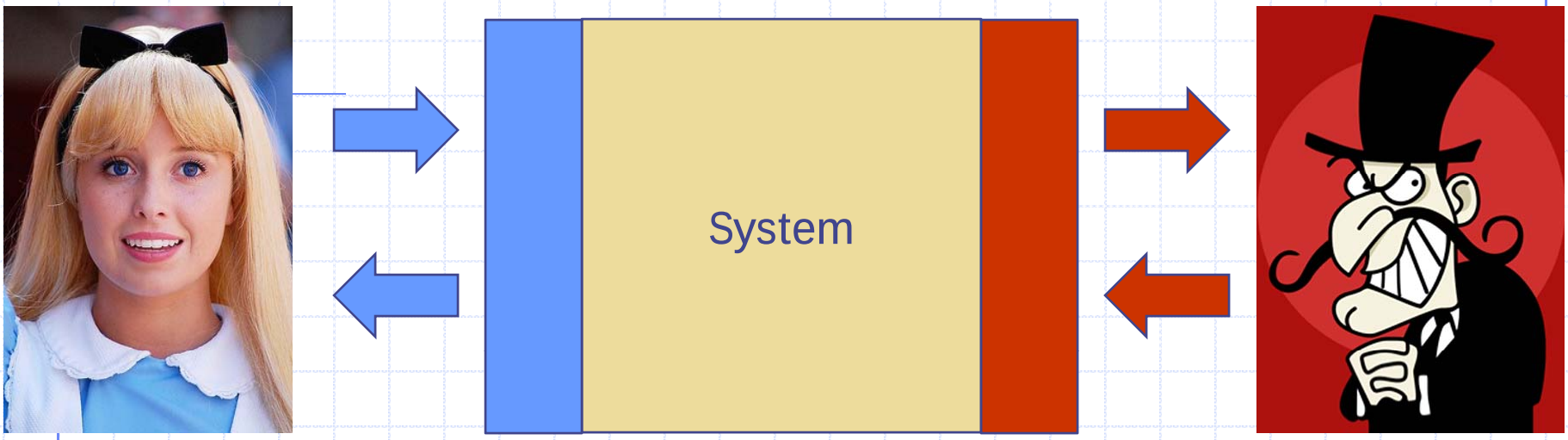
What is security?

- 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

General picture



- 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

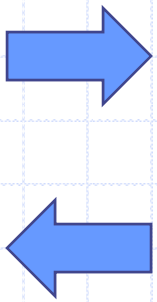
Alice

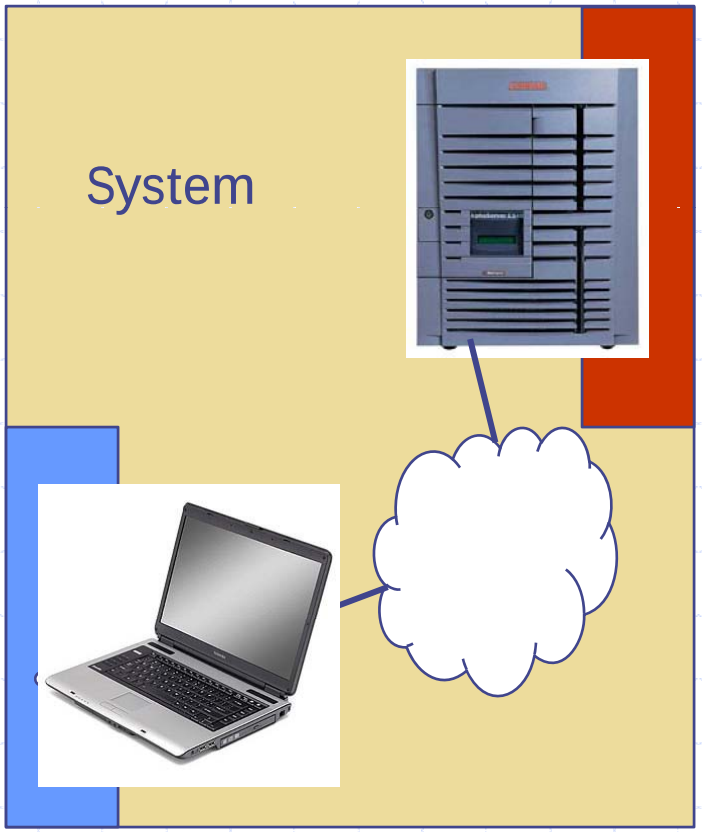
Network Attacker

Intercepts and controls network communication

Web security

Alice



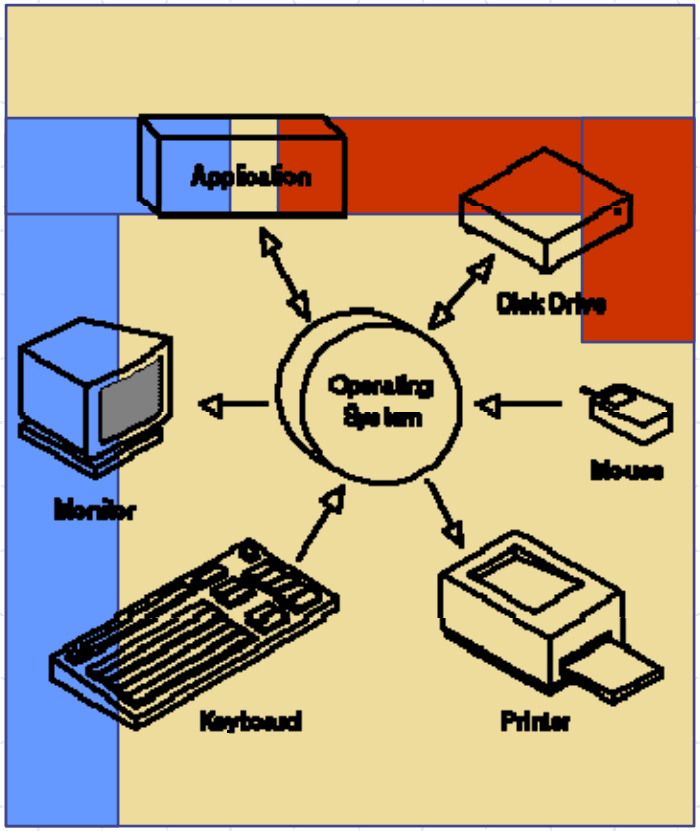


Web Attacker

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

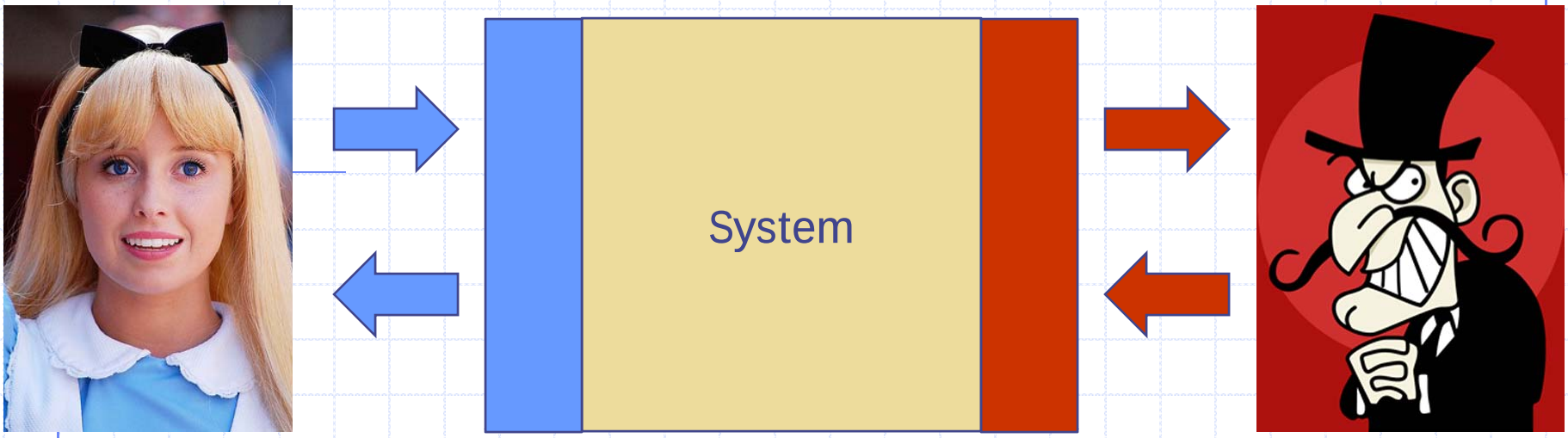
Operating system security

Alice



OS Attacker

Controls malicious files and applications



Alice

Attacker

Confidentiality: Attacker does not learn Alice's secrets

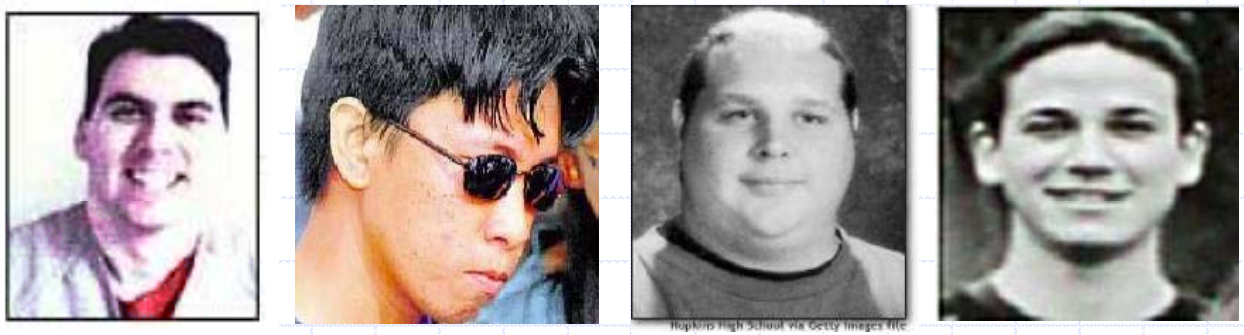
Integrity: Attacker does not undetectably corrupt system's function for Alice

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)*

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

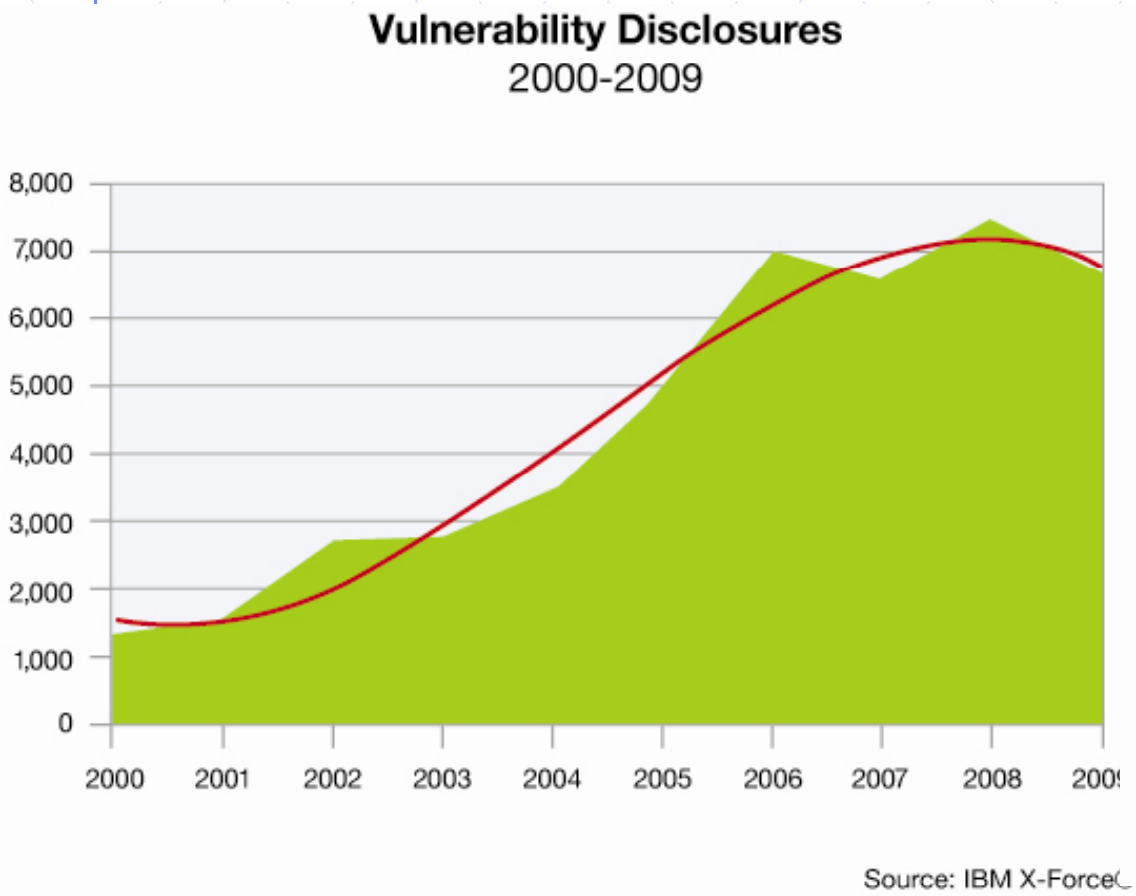
- Nigerian letter (419 Scams) still works:
  - Michigan Treasurer Sends 1.2MUSD of State Funds !!!
- Many zero-day attacks
  - Google, Excel, Word, Powerpoint, Office ...
- Criminal access to important devices
  - Numerous lost, stolen laptops, storage media, containing customer information
  - Second-hand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

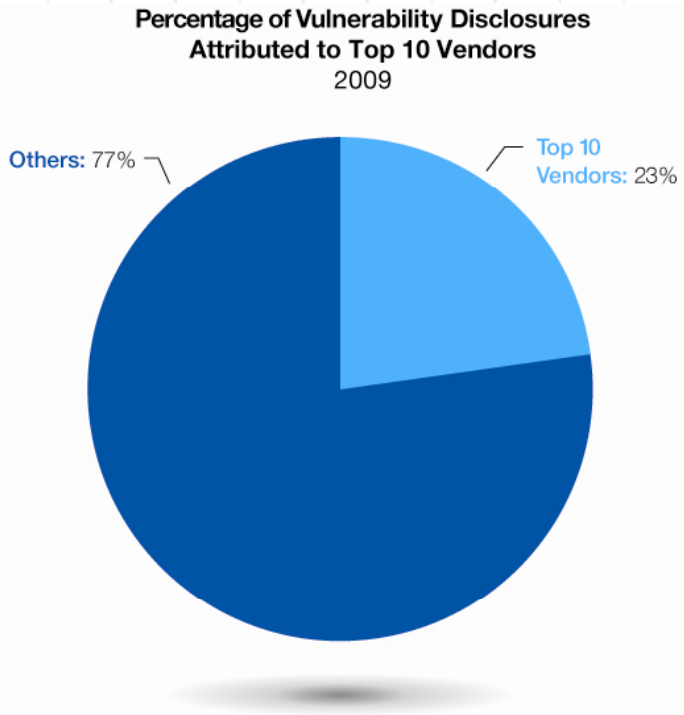
Trends for 2010

- 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.

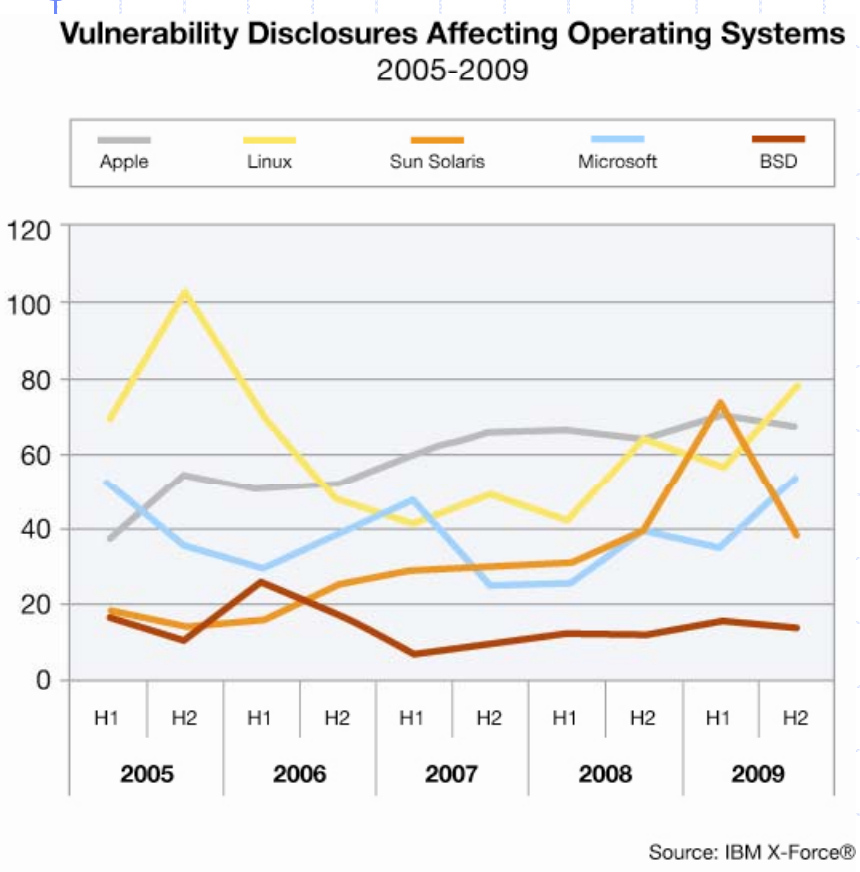
Same list in Oklahoma Monthly Security Tips Newsletter

Trends

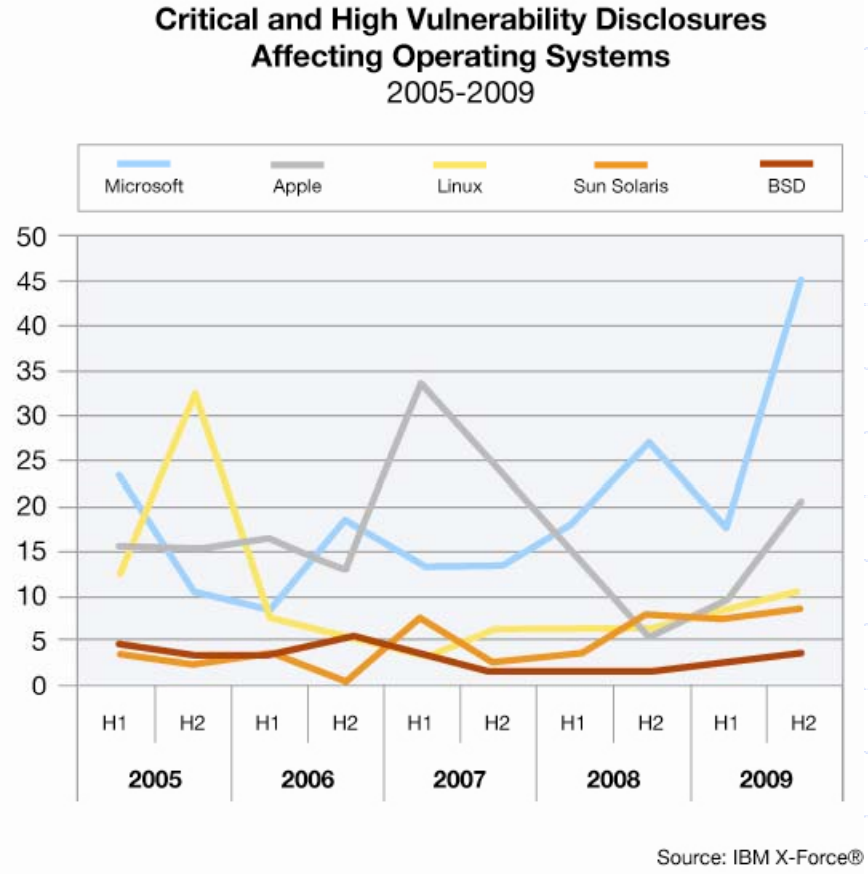

Vulnerability Disclosures
2000-2009
8,000
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
2000
2001
2002
2003
2004
2005
2006
2007
2008
Source: IBM X-Force


Percentage of Vulnerability Disclosures
Attributed to Top 10 Vendors
2009
Others: 77%
Top 10
Vendors: 23%
Source: IBM X-Force®

Operating system vulnerabilities

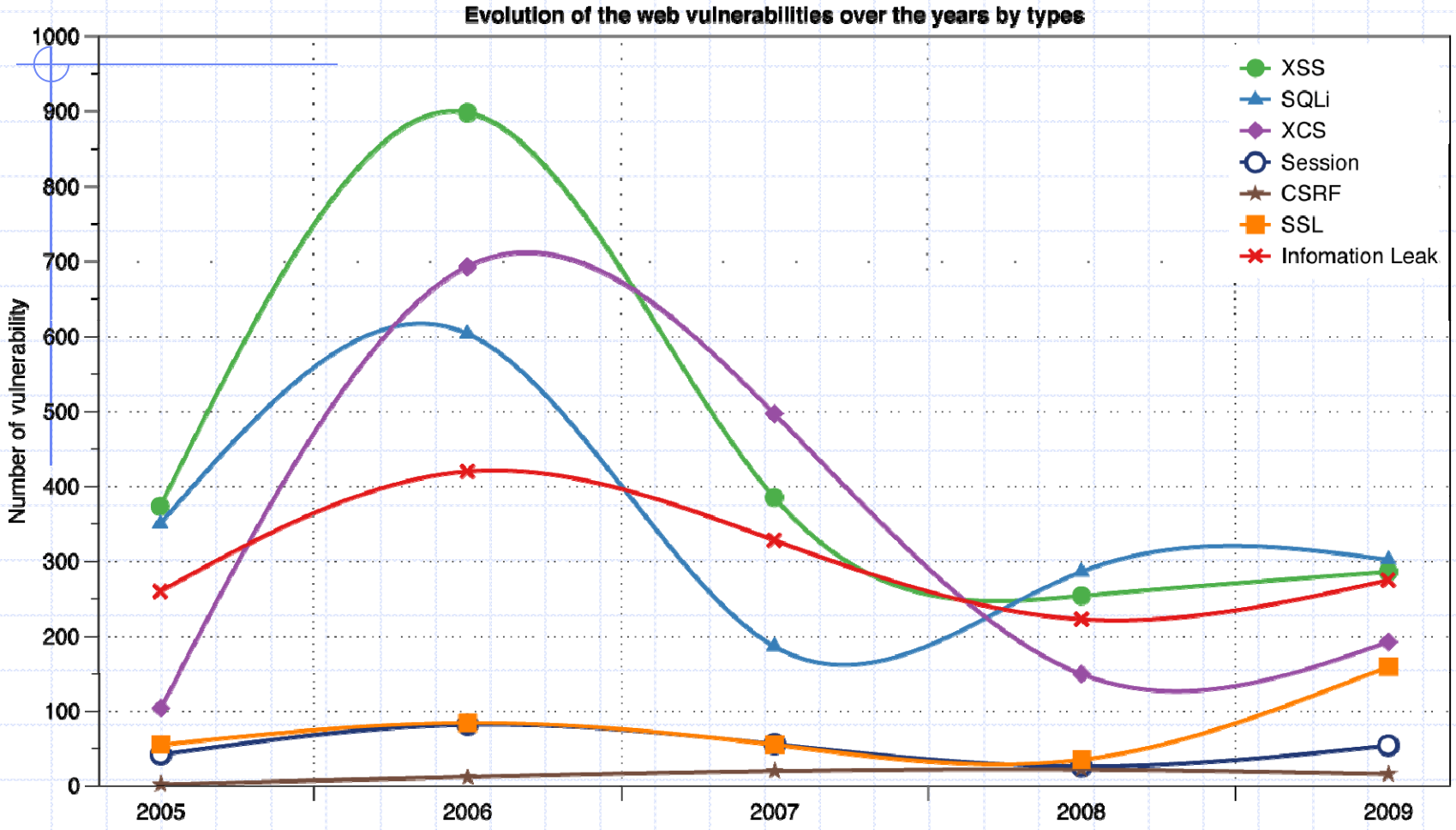


Source: IBM X-Force®



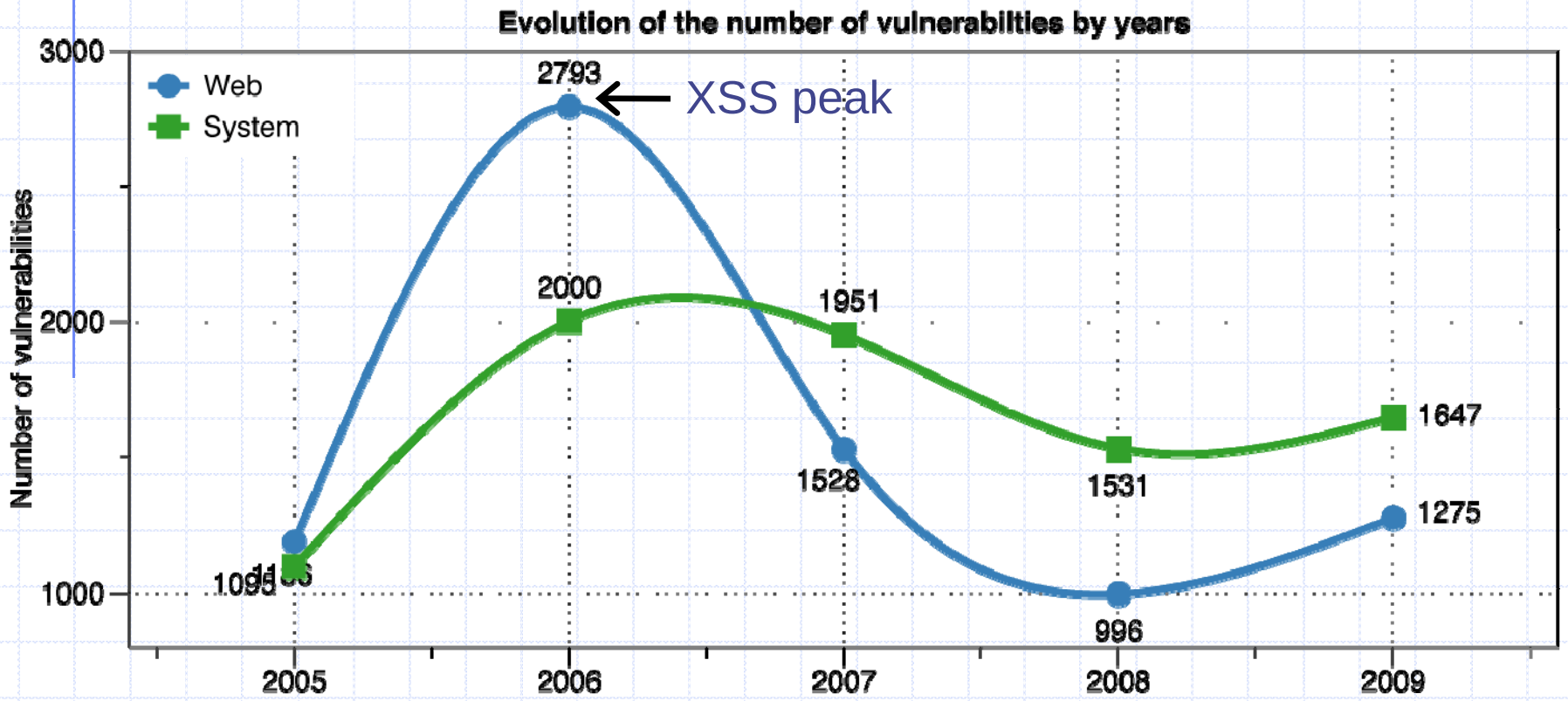
Source: IBM X-Force®

Reported Web Vulnerabilities "In the Wild"

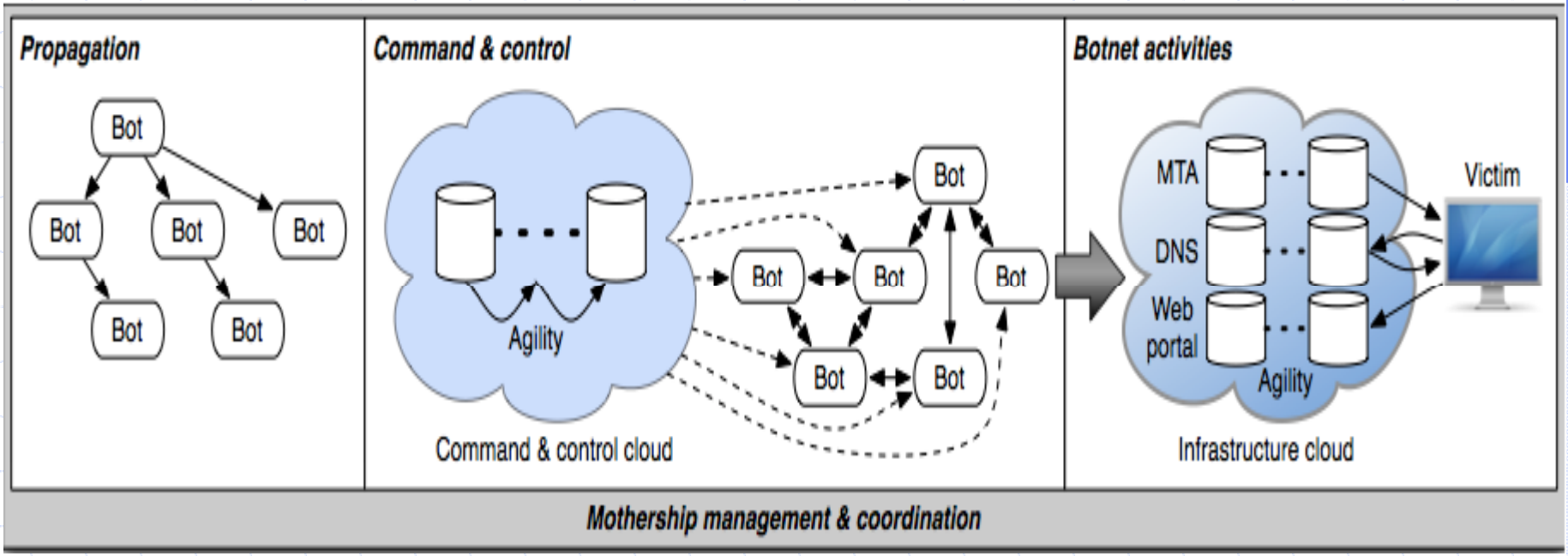


Data from aggregator and validator of NVD-reported vulnerabilities

Web vs System vulnerabilities



Botnet Lifecycle



- Propagation
  - Compromised host activity
  - Network probe and other activity
  - Recognizable activity on newly infected host

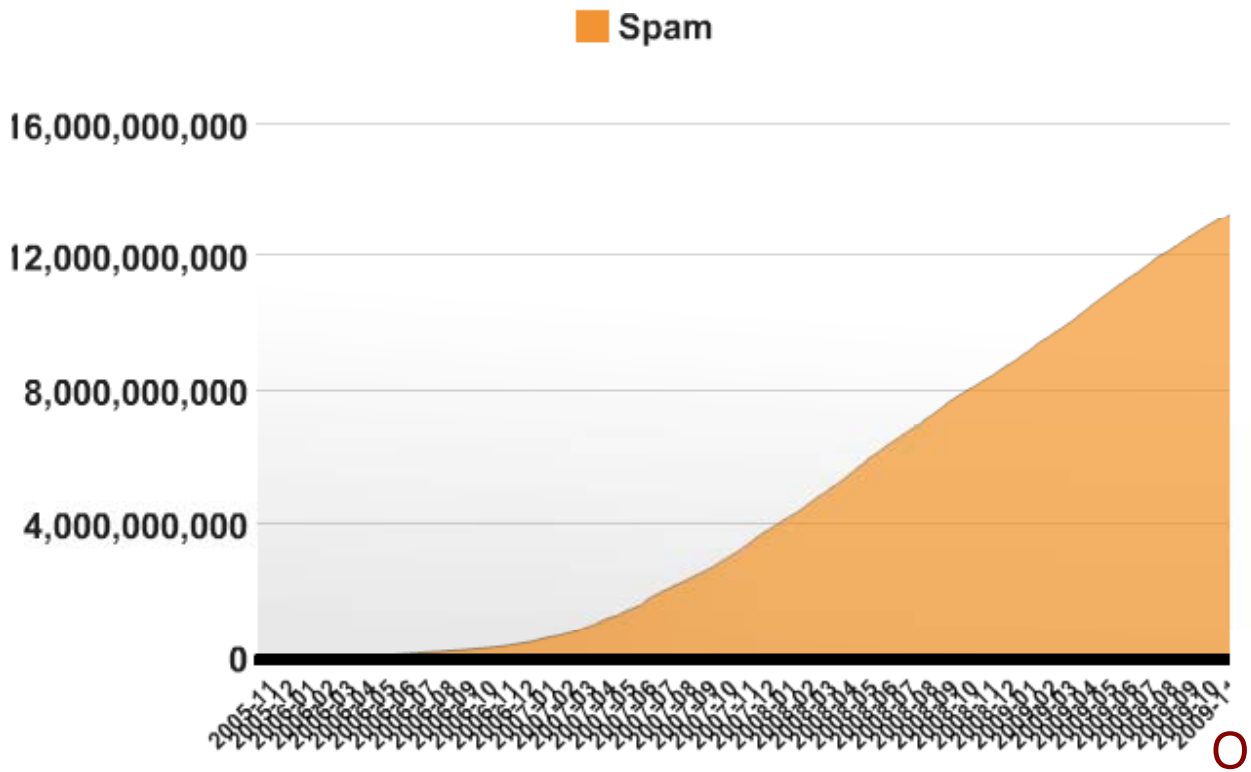
Recent work on malware distribution

- 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?

Totals

Source: Akismet.com



Total spam: 13,275,940,950

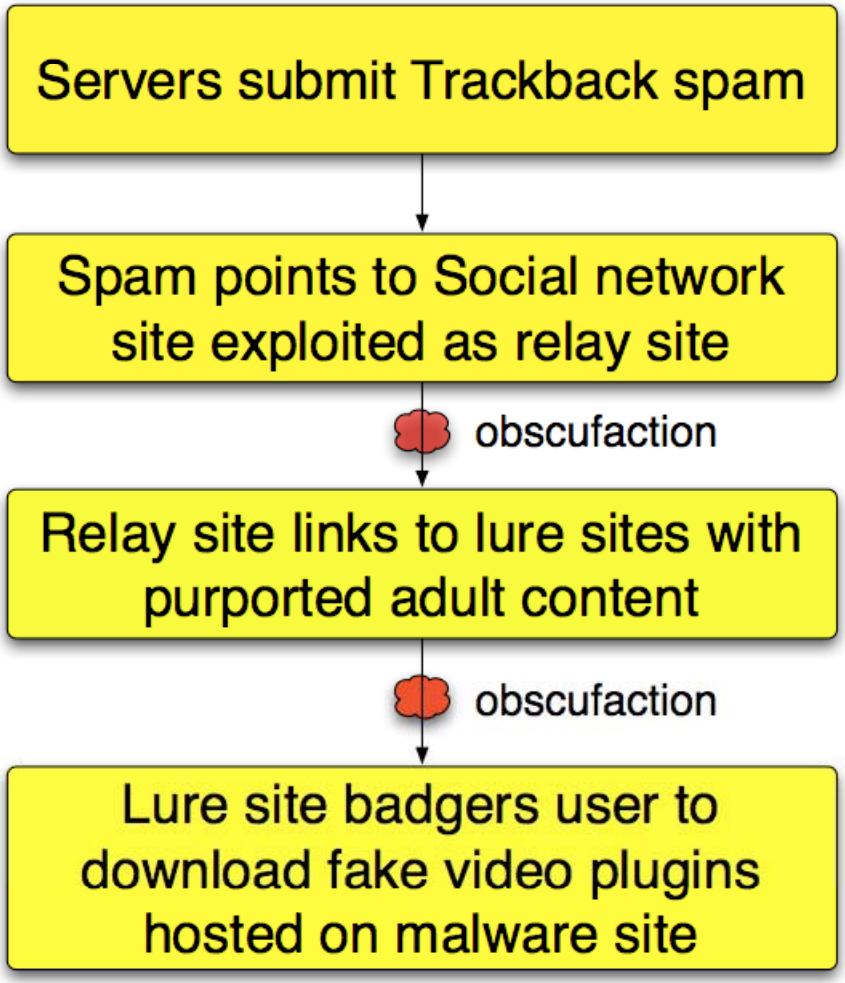
Total ham: 2,701,440,026

One blog spam can reach thousand of users

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

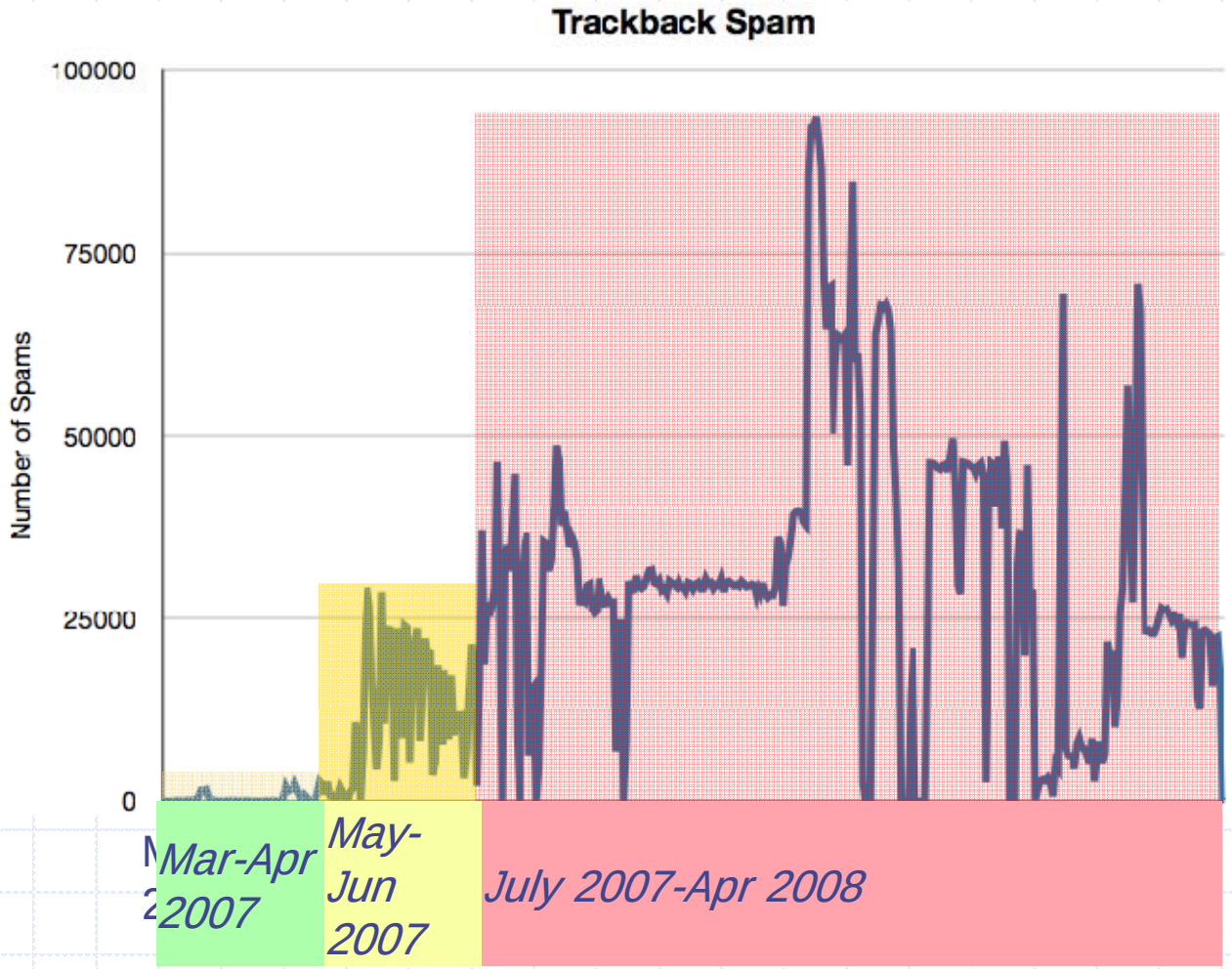


- TrojanDownloader:Win32/Zlob.gen!dll
- Trojan.Popuper.origin
- Downloader.Zlob.LI

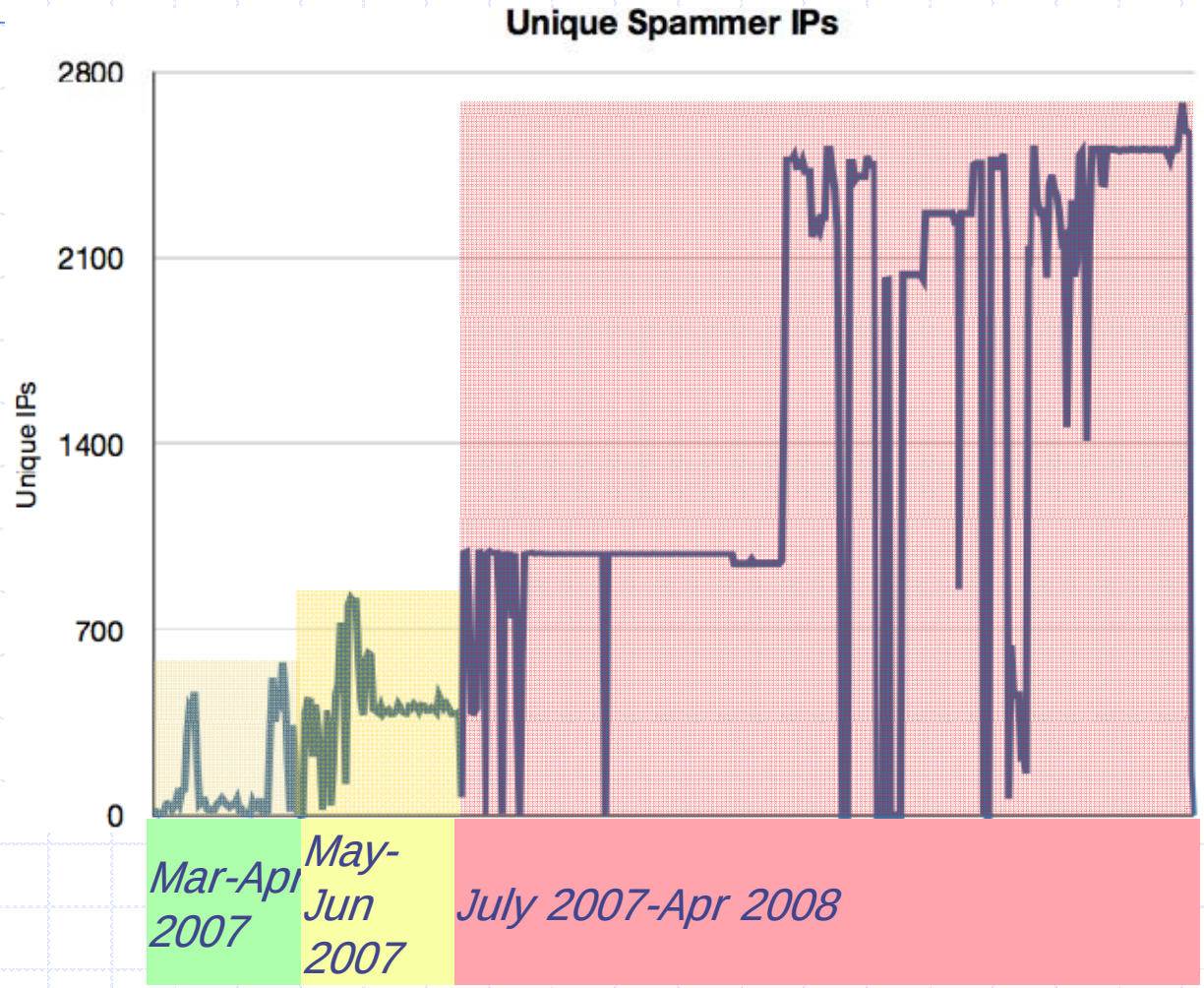
Trackback spam example

- Apparent Bayesian poisoning against spam filters:
- [**title**] => Please teacher hentai pics
- [**url**] =>http://please-teacher-hentai-pics.howdsl.nx.cn/index.html
- [**excerpt**] => pics Please teacher hentai pics ...
- [**blog_name**] =>Please teacher hentai pics

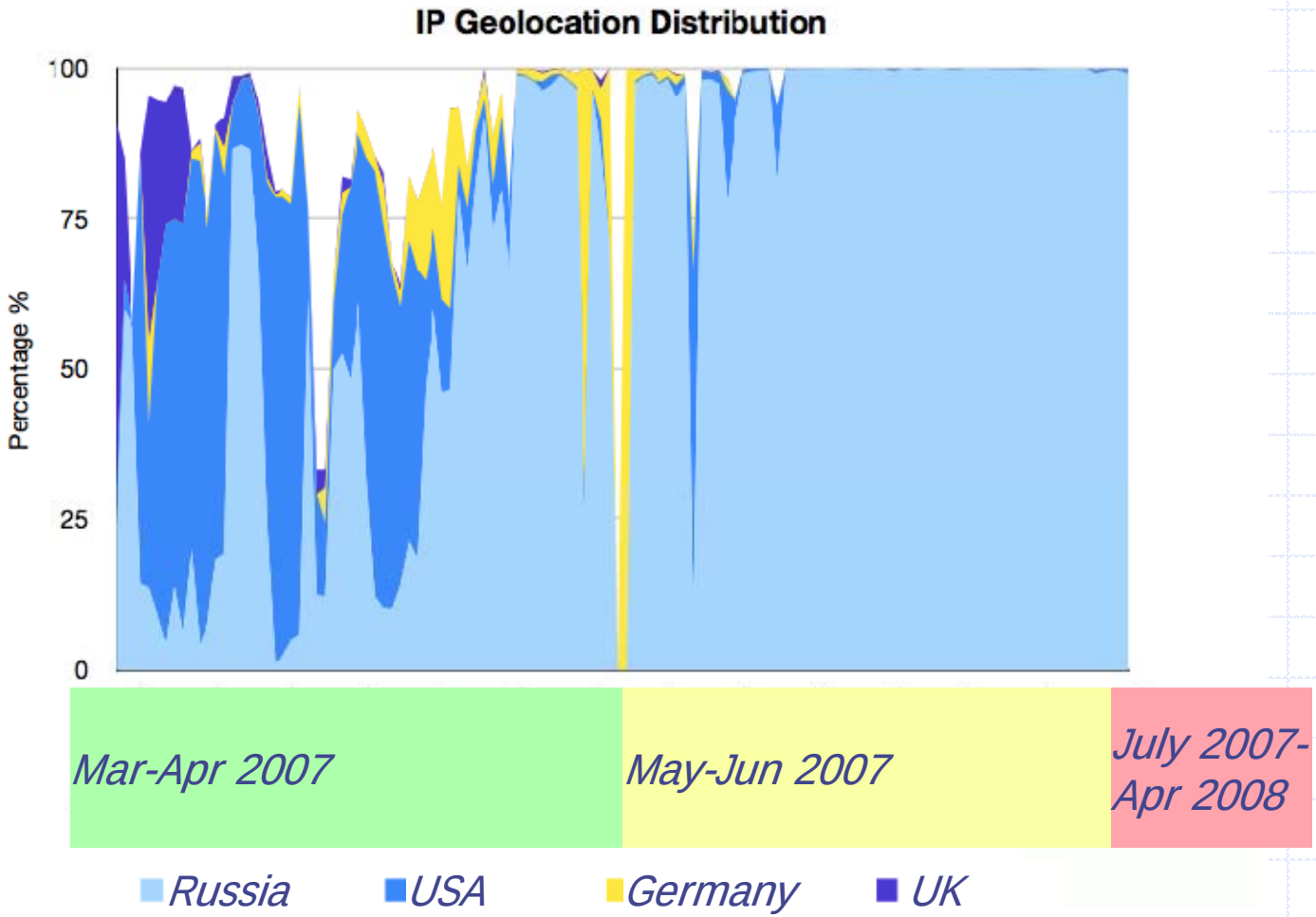
Number of notifications detected



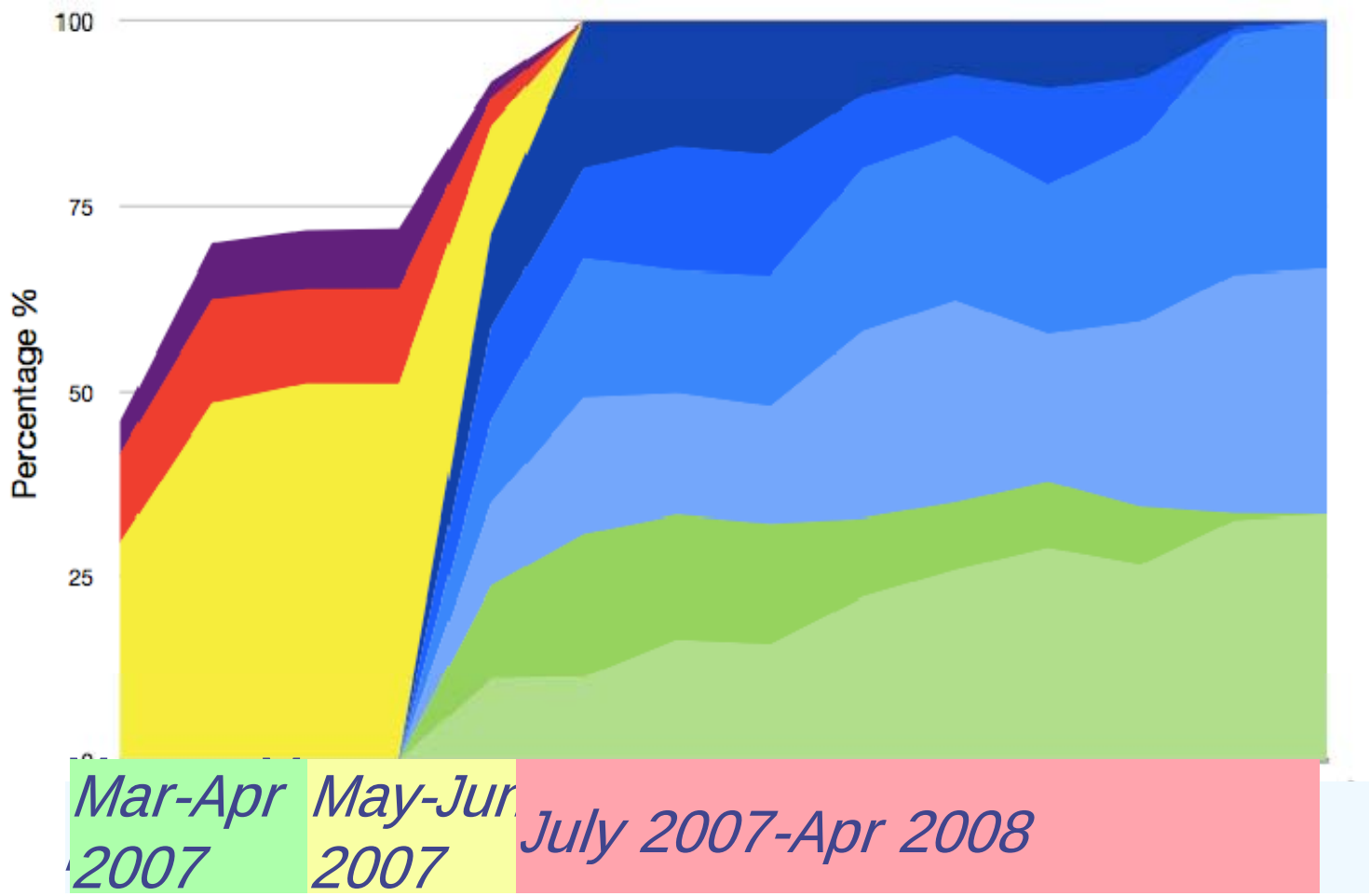
Number of IP Addresses



Origin


IP Geolocation Distribution
100
75
50
25
0
Percentage %
Mar-Apr 2007
May-Jun 2007
July 2007-
Apr 2008
Russia
USA
Germany
UK

User agents reported to honeyblog

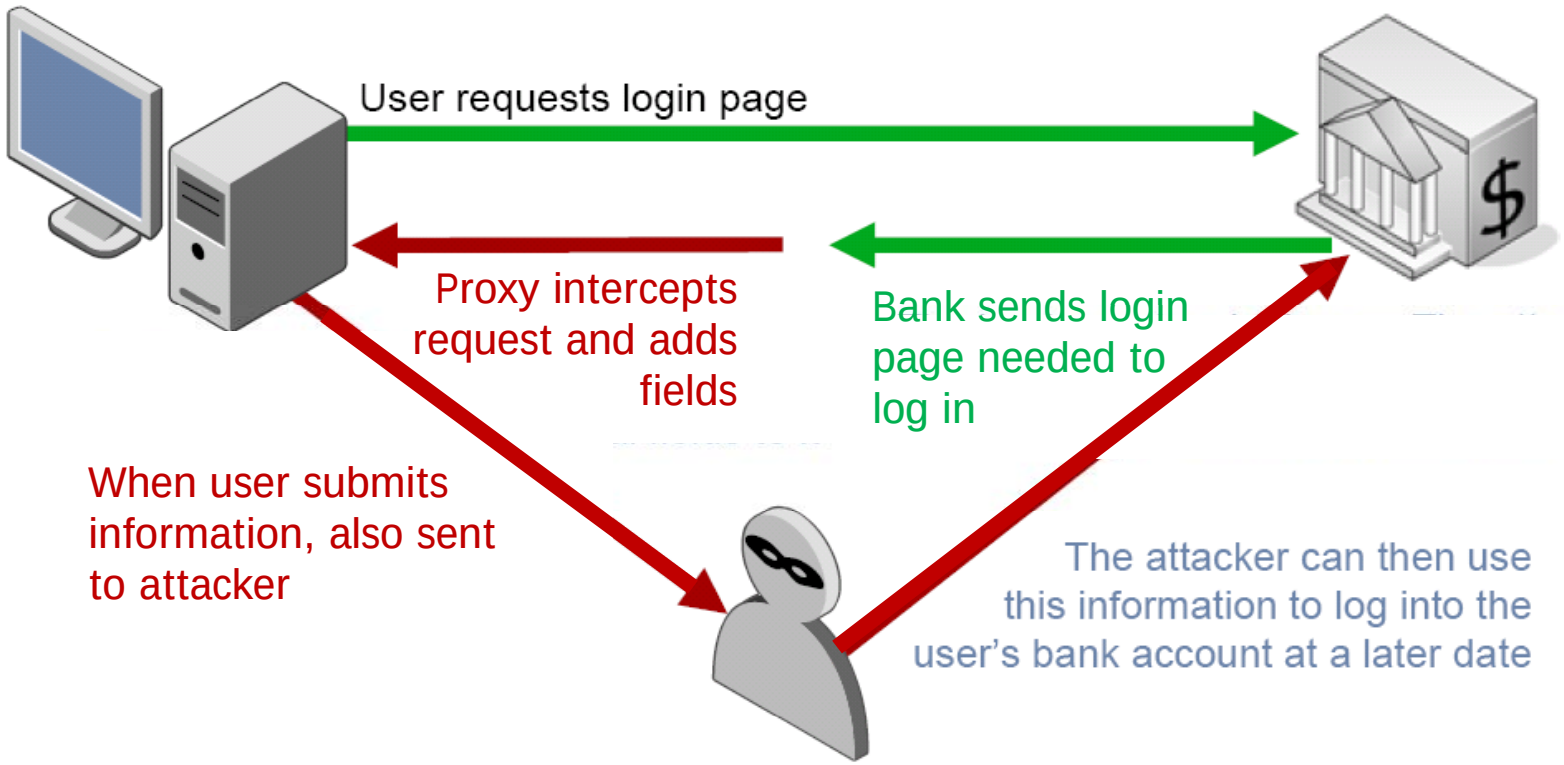


Web attack toolkit: MPack

- 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



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-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 ...

IPhone Flaw Lets Hackers Take Over, Security Firm Says

Kitra Cahana/The New York Times

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

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.

SIGN IN TO E-MAIL OR SAVE THIS
PRINT
REPRINTS

Next Article in Technology (4 of 17) »

MOST POPULAR - TECHNOLOGY

E-MAILED | BLOGGED

1. IPhone Flaw Lets Hackers Take Over, Security Firm Says
2. Google Pushes for Rules to Aid Wireless Plans
3. SunRocket Leaves Void for Callers on Internet
4. IPhone-Free Cellphone News
5. Swedish Woman Gets Superfast Internet
6. Computer Support, Can You Rock to This?
7. Headphones to Shut Out the World
8. Cute Friends to Collect, and Plug in to the Internet

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

- Spam service
- Rent-a-bot
- Cash-out
- Pump and dump
- Botnet rental

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

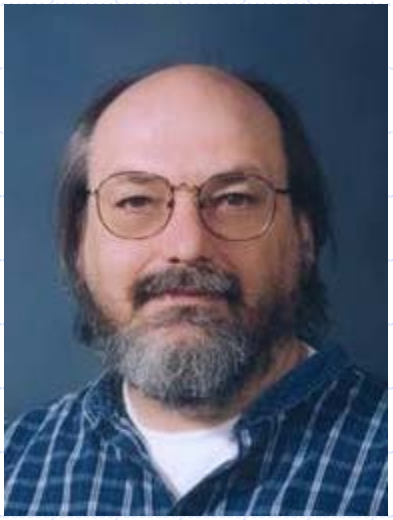
- Breeder's cup race
  - Upgrade of software to phone betting system
  - Insider, Christopher Harn, rigged software
  - Allowed him and accomplices to call in
    - change the bets that were placed
    - undetectable
  - Caught when got greedy
    - won $3 million

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

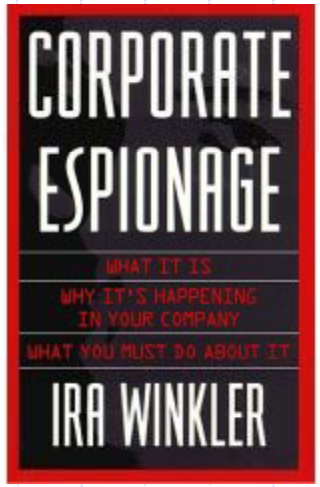
```
compiler(S) {
  if (match(S, "login-pattern")) {
        compile (login-backdoor)
        return
  }
  if (match(S, "compiler-pattern")) {
        compile (compiler-backdoor)
        return
  }
  .... /* compile as usual */
}
```

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



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, ...