Malware: Malicious Code

UIC 594/Kent Law:

Computer and Network Privacy and Security: Ethical, Legal, and Technical Considerations

Malicious code: Viruses

- Most famous type of malicious code
- Malware Program that seeks out a particular program (most often part of MS Office) and embeds a copy of itself inside the program
- Infected program is called *host*; when host runs virus program attempts to duplicate itself and do other things without

Malicious Software

- Satan vs. Murphy
- In 2008, very likely to arrive via network, but does its work on one (your!) computer
- Does and does not violate access control:
  - It's got your permissions!
- What happens when you install new program?

Taxonomy

- Malicious code, malware, rogue program:
- **Virus**: Can replicate itself (typically via copied program and/or data) and pass on malicious code to other non-malicious programs by modifying them
- **Trojan horse**: Contains unexpected additional usually malicious effects

Taxonomy continued

- **Logic bomb**: Malware that starts doing its thing only when some condition is met
- **Time bomb**: Condition is a time
- **Trapdoor** or **backdoor**: built-in surreptitious "extra" way to access. (E.g., extra unlisted super-user account with name maintenance & password 99999999.)

Viruses spreading across computers

- Requires host to get to uninfected computer.
- Either user sends it (e.g., Word document via email or memory stick) or it has infected program use, e.g., Microsoft Outlook.
- Today majority worms and/or involve MS Office, Outlook, and/or Internet Explorer

Life cycle

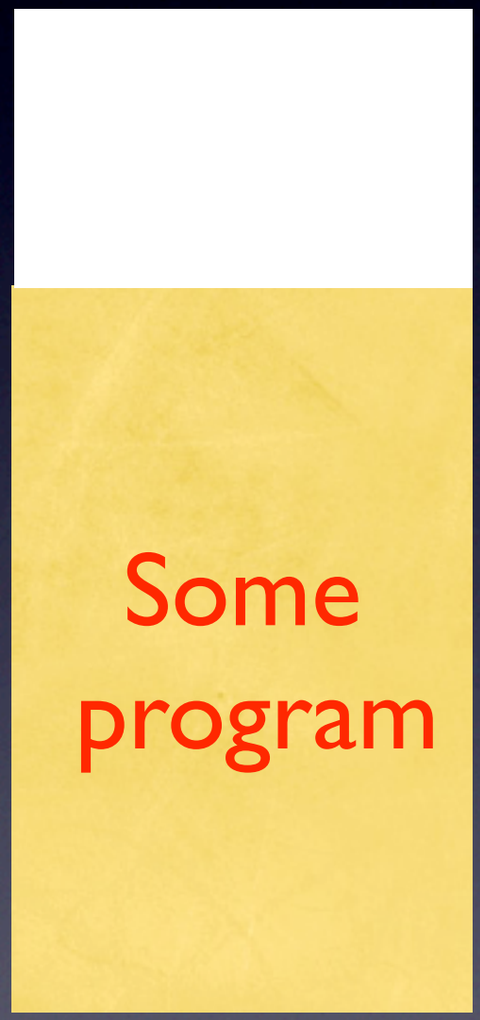
1. **Infection mechanism/vector**
2. **Trigger**
3. **Payload**—Do whatever it does besides spreading

Infection mechanisms today

- Email (92% of attacks)
- Peer-to-peer file sharing (14% of attacks)
- Remote exploitation of system/software vulnerability (13% of attacks)

Virus: Basic idea

- Simplest case are some instructions (lines of computer code) that insert themselves at the beginning of a program.
- User runs that program; doesn't even know virus code is running; control flows on to intended program.



Virus theory

- is interesting but not terribly relevant
  - Written up by Cohen in mid 1980s, though viruses existed earlier
- involves self-modifying code
- can prove that there exists virus that will defeat any particular anti-virus software

Document (macro) viruses

- # 1 form mid and late 1990s, still popular today
- Instructions in formatted document, especially MS Word
- Macro—shortcut "hotkey" programmed in, e.g., Basic
- Cross platform (but payload usually is not)

Virus prevention

- Run good anti-virus software that is **regularly automatically** updated.
- Or use Mac OS X or Linux.
  - Very few viruses "in the wild" for Linux; still none for Mac OS X (though first worm).

Worms

- Like a virus in terms of having r*eplication mechanism* and *payload*, etc.
- Distinct because it's stand-alone; no host
- Most famous was 1988 Robert Morris Internet Worm—shut down Internet for the day.
  - Intended only to spread!

There is an OS X worm in the wild

- Leap.A; discovered February 2006
- Targets OS X version 10.4 and spreads via Apple's IM program iChat
- Known # of sites infected worldwide, as of Feb. 2007, according to Symantec: 1–2

Hot new thing: XSS

- Newest member of virus/worm family is cross-site scripting (XSS) virus/worm.
- Uses XSS scripting vulnerabilities to propagate
  - Roughly, web site uses user-provided data to generate a page (e.g., Google search result page) without checking that the user-provided data was "okay."
- Most notable attacks were on MySpace and

Famous Malware: Morris Worm

- Nov. 3, 1988, I got a day off from grad school….
- Robert T. Morris, Jr., then Cornell CS Grad Student, created and released Internet Worm
- Convicted in 1990 of violating 1986 Computer Fraud and Abuse Act, fined $10,000, 400 h0urs of community service,

Morris Worm

- Exploited 3 long known, well known flaws in Berkeley Unix v. 4 systems:
  1. Passwords: people pick bad passwords (coffee, aaa) and encrypted password file world readable
  2. Bug in UNIX finger
  3. Trapdoor in UNIX sendmail
- No harmful payload, but *resource exhaustion*

Worm's effects

- Shut down roughly 6,000 hosts on the 1988 Internet, typically for 1 day; some longer
- Robert T. Morris Sr. never became head of the NSA.
- Internet academic community woke up to the danger; CERT formed.

Example 2: Code Red

- First modern worm, July 2001

  Exploited (known) security hole (buffer overflow) in Microsoft Internet Information Server on servers:

```
HELLO!
Welcome to
http://www.worm.com !
Hacked by Chinese!
```

Code Red (continued)

- Came in various versions to avoid fixed signature that anti-virus software could detect; more than a worm, installed back doors, etc.
- First version spread 1st 19 days of month, launched DDoS attack on www.whitehouse.gov next 9 days of month, took vacation 28th-1st.

Recent well-known worms

- SQL Slammer (2003) Microsoft SQL server
- Mydoom (2004), mass-mailing email worm
- Warezov family

Virus + Worm prevention

- If on Windows, run good anti-virus software
- On any OS, apply security patches on a very regular basis.
- Don’t have ridiculously weak passwords. Many worms Morris-onward have brute-force password crackers.

Trojan (horses)

- Software that user runs on purpose that also does something malicious
- These days often an IM software, media player, or add on to one of those two

Rootkits

- Recall top administrative user on Unix systems = “root” (or “superuser”), so **root access** = total administrator access.
- **Rootkit** = set of programs installed on a system to maintain root access to it
- Typically changes the system to hide its own existence

Hiding oneself

- When user issues command that would show rootkit's presence, e.g., listing files or processes, rootkit intercepts call and returns edited results to user
  - E.g., file listing not listing rootkit file
- Normally also has parts to reestablish itself if discovered and removed.

Rootkit installation

1. Get initial access (password cracking, malware, esp. trojan, system vulnerability)
2. Attacker uploads rootkit to user machine (plus optional extra virus, etc.)
3. Attacker runs rootkit's installation script
4. Rootkit replaces files, system commands, binaries, etc., to hide its presence
5. Rootkit payload activities

Rootkit revealers

- Idea: Program that displays files the usual way, and that examines disk directly and displays that way, and compares.
- Computer security expert Mark Russinovich developed one, which he ran on his own system.
- Surprised to find he had a rootkit on it.

Sony XCP

- Rootkit was installed when he had played a music CD.
- Sony XCP (extended copy protection)!
- Rootkit that prevents user from copying CD while allowing it to be played
- Installed its own music player that is allowed to play the CD.

Sony DRM debacle

- Problem with music CDs is music is in easily readable format.
- Sony tried 2 different essentially malware DRM approaches; one, XCD, was a rootkit.
- Need to stop user from getting raw music as soon as CD is put in PC.
- “Helpful” Windows feature: autorun

Installation: Windows Autorun

- Windows runs program called autorun.exe on CD insertion. (Suggest you disable it.)
- No user—nor, with music CD, expectation.
- This one installed rootkit program that stopped music from being accessed by anything but itself, a music player. (So no iPod, etc.)

Story gets worse

- Program hid itself by hiding all program names starting sys
  - Thus making user vulnerable to *any* malware with name beginning sys….
- Sony XCD phoned home to Sony with info each time CD was inserted—Spyware!
- Uninstaller (of both programs) opened new

Sony XCD Analysis

- At least 500,000 installs; maybe 100s of millions.
- Felten & Halderman analysis: DRM has similar design specs to malware: get user to install something that gives him no benefit, and get him to leave it installed.

Additional Defense to malware

- Limiting what can be on your network!
- This is one very big why University/ Company/Etc don’t want unauthorized wireless access points, machines, etc., on their networks.

Threat of monoculture

- Many (all?) famous massively successful were aided by *lack of software "genetic diversity."* E.g.,
  - Morris Worm–in 1988, machines on Internet were all running Unix
  - Code Red: Significant fraction of all servers in 2001 were running Windows.