Malware: Malicious Code

Robert H. Sloan

Malicious Software

- Satan vs. Murphy
- In 2010, very likely to arrive via email attachment or web, 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?

Scope of problem

- Estimated 12% of computers, 25% of computer-facing computer infected
- Today huge volume and diversity of new malware; few pandemics
- For profit: Send out spams (penny buys 1–200+), SSNs worth $2-3 each, sell a DDoS, $50–2,000 for 24 hours

But, for profit = kinder?

- Typical malware today does NOT stop you from using your computer.

Taxonomy

- Malicious code, **malware**, rogue program, definitions vary slightly. One set:
  - **Trojan horse**: Appears to perform neutral/helpful stuff. Contains unexpected additional malicious effects
  - **Worm**: Spreads copies of itself (via network); host program optional
  - **Virus**: Replicates itself by embedding itself in program; subspecies of worm

Taxonomy continued

- **Rootkit:** Modern twist on virus. Piece of software that, when installed on a computer, puts it under remote control
  - Most rapidly growing problem today
  - Most common use: make machine part of botnet
  - But may also be installing spyware, adware, . . .

Worm/Virus Life cycle

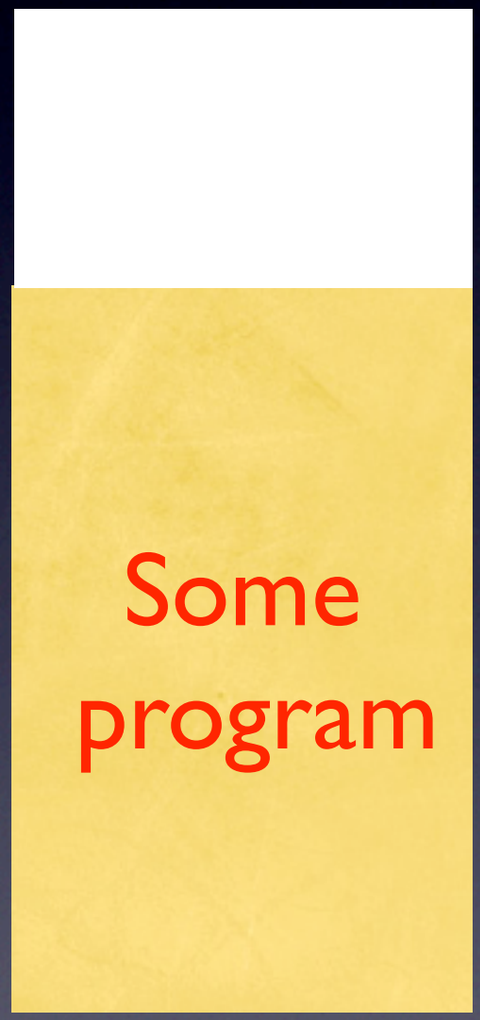
1. **Infection mechanism/vector:** Replication mechanism, e.g., break into another system, or email itself as an attachment to addresses in address book
2. **Trigger** (If not immediate)
3. **Payload**—Do whatever it does besides spreading

Infection mechanisms today (2009 and 2010)

- Top: ((Spear) Phishing email to) Get user to access corrupted file (attached to email or at web site) through one of:
  - Adobe PDF Reader, Flash, or Quicktime
  - Microsoft Office (Word, Excel, Ppt)
- Old school: Still some unpatched Windows systems getting late 2008 Conficker/ Downadup Worm

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

Malware prevention

- Patch OS, Browser, Office, Adobe apps regularly
- Windows: Run good anti-virus software that is **regularly automatically** updated. Don't run Windows 2000, earlier.
  - But much less effective than 5 years ago, because malware writers have gone pro.
- Or use Mac OS X or Linux.

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 hours of community service, 3-year suspended sentence.

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* (very probably a bug/mistake by RTM)

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.

Conficker (Downadup)

- Worm with 5 versions, first seen in wild from November 2008 (A) to April 2009 (E)
- Spread through buffer overflow vulnerability in Windows and Dictionary attacks on passwords
- Today: 2–10 million PCs in dead (or dormant??) botnet
- Never did anything but protected itself, nobody knows why

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.
- Other, MediaMax (from US company SunnComm) was also highly problematic—e.g., installed if you clicked "No, don't install."

DRM for CDs

- CDs must have music in CD Audio format for CD player to work. Set in 1980s (70s?), no notion of DRM
- 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.
- XCD 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 security holes! (Unintentional—bad design)

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, and indeed almost all Berkeley Unix
  - Conficker worm: Windows on >90% desktops/laptops.

Web sites

- Another major trend is consumer-facing commercial web sites being altered in various ways
  - Technically probably XSS or inject
  - Result to us: Medium-size trusted company put up good file, program, but site (or apparent site) now has bad file, malware