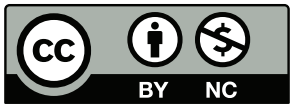
Viruses, Worms, and Trojan Horses



Viruses, Worms, and Trojan Horses

- What are they?
- How do they spread?
- What can be done about them?

Worms in Science Fiction

"Let me put it another way. You have a computer with an auto-dial phone link. You put the VIRUS program into it and it starts dialing phone numbers at random until it connects to another computer with an auto-dial. The VIRUS program then *injects* itself into the new computer. Or rather, it reprograms the new computer with a VIRUS program of its own and erases itself from the first computer. The second machine then begins to dial phone numbers at random until it connects with a third machine. . . .

"It's fun to think about, but it was hell to get out of the system. The guy who wrote it had a few little extra goodies tacked onto it – well, I won't go into any detail. I'll just tell you that he also wrote a second program, only this one would cost you – it was called VACCINE.

When Harlie Was One, David Gerrold, 1972

Viruses

- “Infected” program (or floppy)
- When program is executed, it performs its normal function
- It also infects some other programs
- It may carry an extra “payload” that performs other functions

Worms

- Similar to viruses, but they spread *between* machines
- Some are fully automatic; some require manual intervention to spread
- Some exploit bugs; others use social engineering
- Name from John Brunner's *The Shockwave Rider*, 1975

Trojan Horse

- Program pretends to be useful, but does something else as well
- Generally spread by “come and get it”
- Name from the Odyssey, the Aeneid, and other ancient myths, some from about 2700 years ago

Malware

- From French (or Latin) *mal*: “bad”
- Refers generically to any malicious software
- Etymological note: *malicious* has the same same prefix. . .

Academic History

- Informal, “midnight” experimentation with related concepts
- First formal worm built at Xerox PARC (published 1982)
  - For distributed computation
  - They still had a run-away
- Viruses formally described in the academic literature in 1984
- But the first one in the wild was in 1982!

History of Viruses

- Major plague in the PC world, back in MS-DOS days
- (More energy these days for worms)
- Many different types of viruses
- Different types of viruses spread in different ways

Executable Program Viruses

- Part of an ordinary executable
- Add itself to one end, saves the old code, inserts a branch to it
- Sometimes hidden in unused part of file—avoid changing file length

Boot Sector Viruses

- Modified *boot sector* of hard drive and floppies
- The “boot sector” is the part of the disk read in to memory and executed when you boot from that disk
- Generally spread via floppies—in MS-DOS days, you rebooted frequently, and frequently had a floppy in the drive
- You probably didn’t want to boot from that floppy, but it tended to happen a lot by accident
- Largely extinct now. (Why?)

Infected Flash Drives

- Today, some viruses spread via infected USB flash drives
- For a while, the U.S. Defense Department *banned* use of flash drives
- Stuxnet was injected and spread that way
- ☞ People share them the way they used to share floppies
- Systems aren't rebooted as frequently—but many systems honor **`autorun.inf`** files on USB drives
- For those that don't, use U2 flash drives—they emulate a flash drive and a CD-ROM. . .

A Flash Drive with a Write-Protect Switch

Autorun: A Lesson in Technology Change

- When autorun was first implemented, it was for CDs only
- Recordable CDs were rare; routine use for casual file transfers were rarer still
- Microsoft opted for convenience and a good user experience
- But—technology changed. USB drives look like hard drives, so of course they should be auto-opened
- The technology (and hence usage patterns) changed—but the interface didn't. . .
- On today's Windows machines, autorun is disabled by default

Macro Viruses

- Viruses can be written in any sufficiently-powerful language
- Microsoft Word has a powerful macro language. . .
- Word documents (also Powerpoint presentations and Excel spreadsheets) can thus spread viruses
- They usually infect **`normal.dot`**, the default template file that is read in by most documents
- They thus infect virtually all new Word documents you create

The First Virus: Elk Cloner

- Written by a 15-year-old for the Apple II in 1982
- Boot sector virus
- Annoyed the victim every 50th reboot
- Infected most of his friends' computers

Virus-Spreading Patterns

- Boot-sector viruses spread in affinity groups—floppies were a normal means of communication before networks
- Program viruses spread by people sharing software—often improper or illegal
- Word macro viruses spread by ordinary business behavior!

Writing Viruses

- Scanner to find new places to infect
- Replicator: copy virus text to new place
- Payload (optional)

Payloads

"You can tap into any computer you want, raid it for any information you want, and do it all without any possibility of being detected. *Or*, you could set the VIRUS program to alter information in another computer, falsify it according to your direction, or just scramble it at random...."

When Harlie Was One, David Gerrold, 1972

Virus Payloads Found in the Wild

- Corrupt files
- Delete files
- Encrypt files and hold them for ransom (payment these days is commonly via Bitcoin)
- Change BIOS settings, thus requiring hardware fixes
- Erase flash BIOS
- Steal information
- Hide

Rootkits

- Mechanisms by which malware hides
- Block the "**`ps`**" command, "**`netstat`**", etc.
- Subvert "**`ls`**" so it doesn't show up on disk
- Analogous mechanisms in the Windows world
- Used *after* malware has penetrated the system

Anti-Virus Software

- There is no way to recognize all possible viruses
- There can't be; it runs afoul of the halting problem
- Anti-virus programs look for patterns of *known* viruses

Virus Defenses

- Encrypted viruses: most of the text is encrypted
- Polymorphic viruses: uses variant byte patterns to foil detectors. Sometimes combined with encryption using various algorithms and/or keys
- Defense: A-V programs simulate the execution and watch for certain known behavior patterns
- Defense: *anomaly detection*—look for behavior patterns that seem abnormal or don't fit a predetermined norm

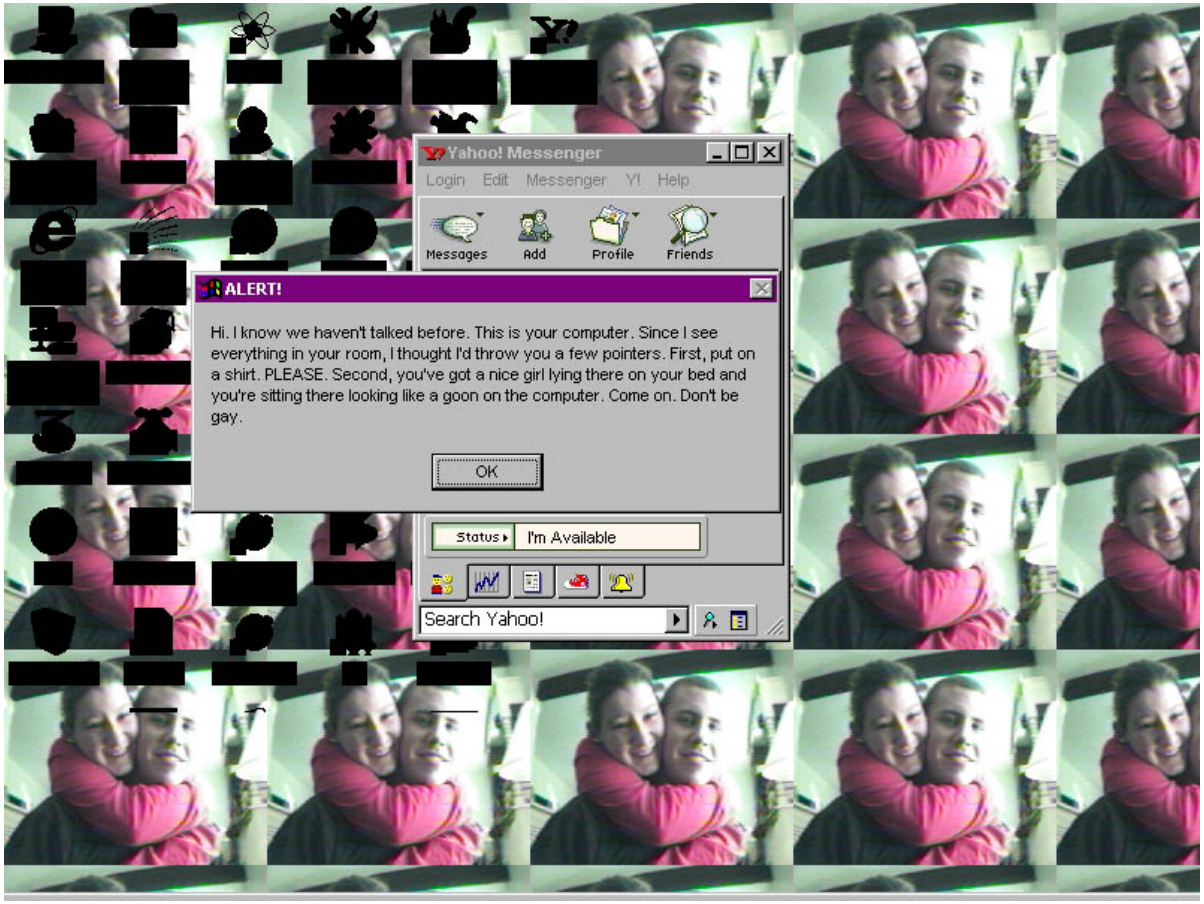
A Unix Virus

- Tom Duff wrote a Unix virus
- In fact, he wrote a shell version as well
- ☞ See the reading
- *No* system is immune

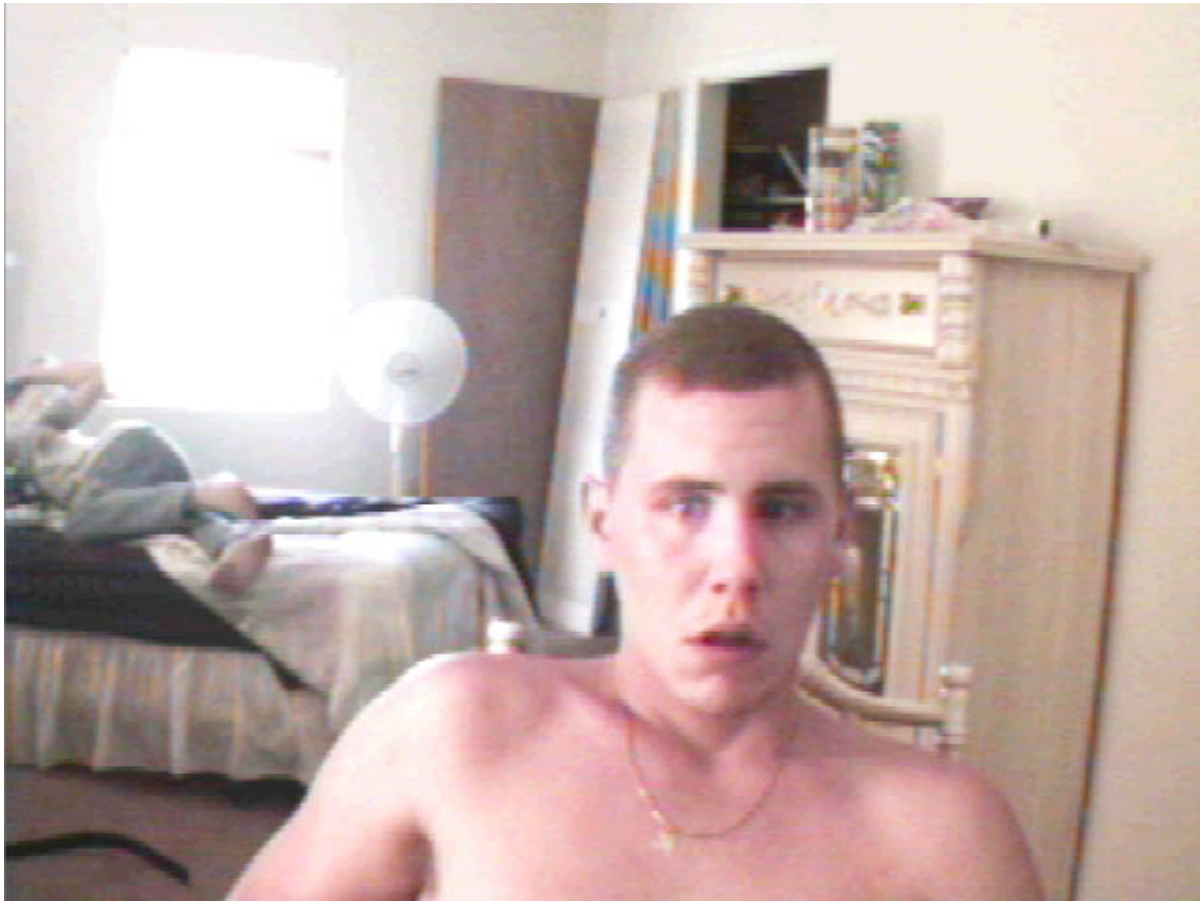
Trojan Horses

- Many types of Trojans!
- Back doors
- Remote camera access—spy on people
- Keystroke loggers—steal passwords
- Web clickers—run up advertising costs
- Proxies—allow others to use your machine to launder connections
- Spam engines
- DDoS engines

Harassing a Remote Camera Victim



His Reaction



Come and Get It

- Many Trojans are inadvertently installed by the user
- Users are lured to “useful” software
- Recent trend: bury Trojan horse in purported hacking tool. . .

Malicious Attacks

- If you agree to a Trojan being installed, is it legal?
- Is it legal if the license agreement is deliberately confusing?
- How carefully do *you* read license agreements?

Ken Thompson's C Compiler Hack

- Write a self-reproducing code fragment
- Modify the C compiler to detect that it's compiling login, and if so to insert a back door
- Modify the C compiler to detect that it's compiling itself, and if so to insert both the login back door and the C compiler modifications
- Delete the source of the the trap
- The back door persists in the executable!
- Rumor has it that this version was shipped to NSA. This rumor has been denied by Ken. . .

A Deliberate Back Door

- Eric Allman, the author of **`sendmail`**, wanted continued development access on a production system
- The system manager wouldn't let him
- He installed a password-protected back door in the next release
- Due to a bug, this back door was generally unprotected
- It was (ab)used by the Internet Worm of 1988

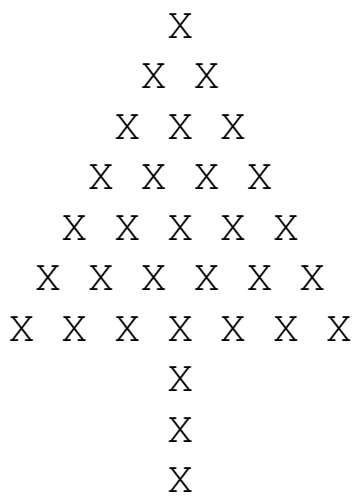
Early Worms

- IBM Christmas Card “Virus”, December 1987
- Morris Internet Worm, November 1988
- Most worms since then have emulated one or both of those

Christmas Card Virus

- Infected EARN, BITNET, and IBM's VNET
- (Old, pre-TCP/IP network for IBM mainframes)
- Spread by *social engineering*

What Users Saw



```
A very happy Christmas and my best wishes for the
next year.  Let this run and enjoy yourself.
Browsing this file is no fun at all.
Just type Christmas.
```

What Happened

- A file transfer mechanism (not quite email, though it could have been) delivered a short script to users
- It was written in REXX, a shell script-like language for IBM's VM/CMS system
- The script displayed the Christmas card; it also looked through the (equivalent of) the user's email alias file and the file transfer log
- It transmitted a copy of itself to any usernames it found
- People trusted it, because it was coming from a regular correspondent. . .

Essential Elements

- Self-replicating executable
- Apparently from a trusted source
- Request that the recipient execute the program
- Using the email alias file to find new victims
- These characterize most current email worms

The Damage

- The worm itself wasn't malicious
- However, it had exponential growth patterns
- It clogged servers, communication paths, spool directories, etc.
- In other words, it was an unintentional denial of service attack

The Internet Worm

- We recently passed the 30th anniversary of the first Internet worm
- Launched by Robert T. Morris, then a grad student at Cornell (and now a professor at MIT and member of the National Academy of Engineering)
- Disabled many of the hosts on the Internet
- Exploited: password guessing, buffer overflow, back door, patterns of trust
- Multi-protocol, multi-platform

Modern Worms

- Most resemble either the Christmas card worm or the Internet worm
- Today's email worms try to trick the user with tempting **`Subject:`** lines — nude pictures, software "updates", etc.
- A notable one: "Osama bin Laden Captured", with an attached "video"
- Some pose as anti-virus software updates. . .
- Can get through many firewalls

Stealthiness

- Deceptive filenames for the attachments
- Add a phony extension before the real one: **`kournikova.jpg.exe`**
- Hide in a **`.zip`** file
- Hide in an encrypted **`.zip`** file, with the password in the body of the email
- Many strategies for hiding on hosts, including strange filenames, tinkering with the registry, etc.

Trust Patterns

- Preferentially attack within the same network — may be on the inside of a firewall
- Exploit shared disks
- Mass-mailing worms rely on apparent trustworthy source

Spreading Via Buggy Code

- Exploit many different (Windows) bugs
- Can spread much more quickly
- Slammer spread about as far is it could in just 15 minutes, and clogged much of the Internet

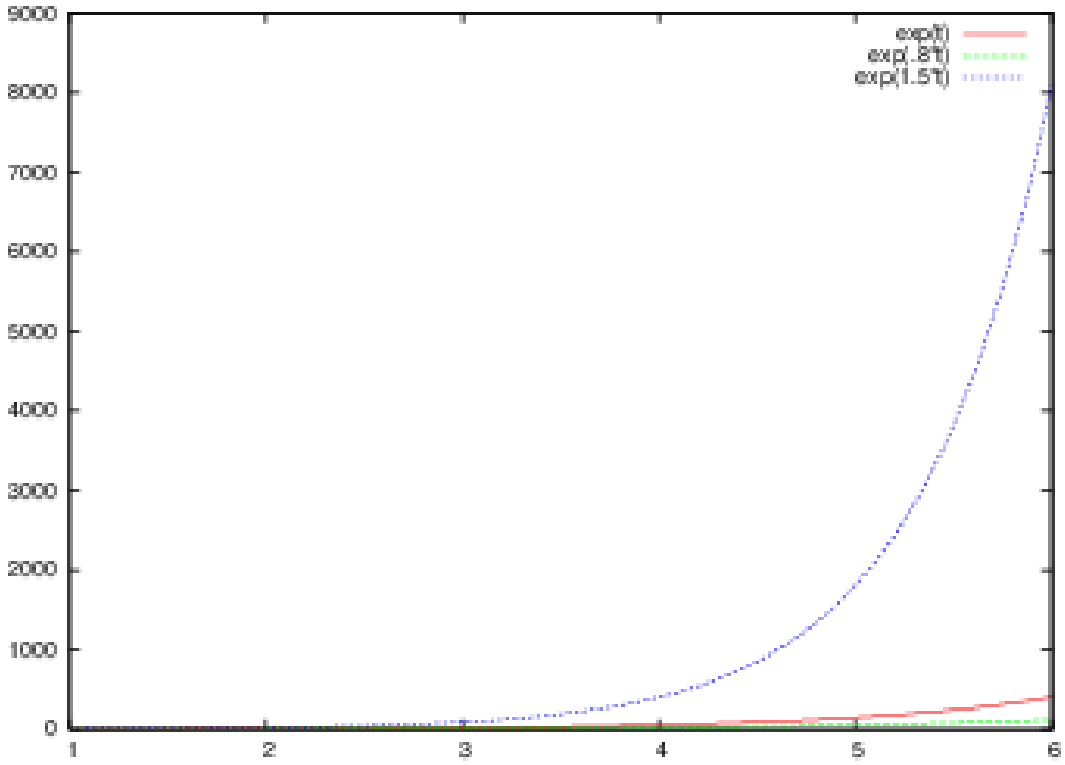
The Slammer Worm

- Exploited a bug in Microsoft's SQL server
- Used UDP, not TCP — a single 376-byte packet to UDP port 1434 could infect a machine!
- Use of UDP instead of TCP let it spread much faster — one packet, from a forged source address, instead of a three-way handshake, payload transmission, and a three-packet `close()` sequence
- No direct damage, but it clogged network links very quickly

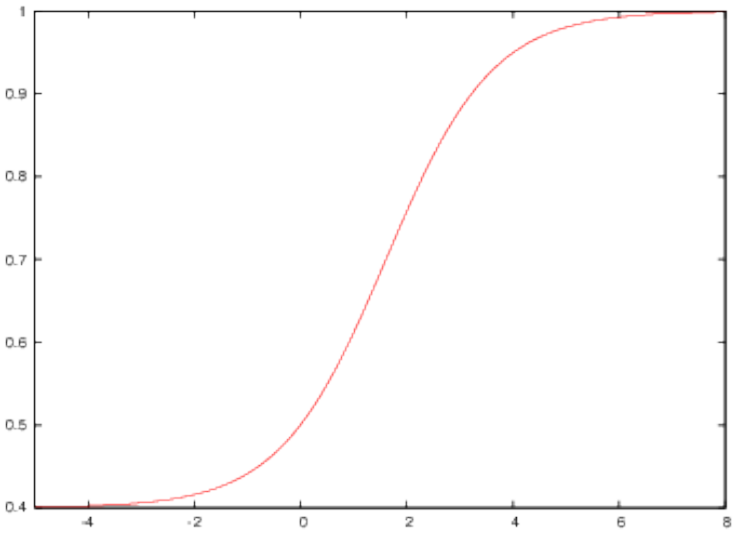
Spread Patterns

- Worms tend to exhibit *exponential growth* patterns
- They start slow, but get very big very quickly
- Equation: $y = e^{kt}$, where t is time
- If k is small, it spreads more slowly — but it still grows

Exponential Growth



There's a Ceiling



- Worms run out of vulnerable hosts
- Doesn't matter much if a machine is infected twice (and worms often prevent that)
- Actual graph is a *logistic curve*:

$$y = a\frac{1 + me^{-t/\tau}}{1 + ne^{-t/\tau}}$$

Source Repositories

- Hackers have often gone after popular source code repositories
- They then plant Trojan horses (generally back doors) in popular packages
- Old example:
  `http://www.cert.org/advisories/CA-1994-07.html`
- This form of attack continues—but we don't *think* anything was changed when the main GNU repository was hit (`http://seclists.org/cert/2003/23`) or when PHP.net was compromised (`http://php.net/archive/2013.php#id2013-10-24-2`).

Viruses Today

- Few pure viruses; worms are more interesting
- Macro viruses have made a comeback
- But—the payloads are much worse today
- Most install back doors, keystroke loggers, etc.
- The motive? Money
- Taking down the Internet is bad for business—the bad guys' business, too. . .

Current Trends

- Trojans (71%), Viruses (11%), Worms (6%), Adware (4%), other (8%)
- High infection rates in China, Turkey, Peru; low in Sweden, Norway, Japan, much of Europe
- New focus on mobile device malware, especially for Android
  ☞ New find: many mobile apps—including some in the Apple app store—steal personal information

Current Actors

- Criminals—they want money
- Commercial espionage, including by governments (China and Russia widely blamed)
- Traditional espionage (many governments, inclduing US)
- “Preparing the battlefield”—plant stuff for future use

Malware and the Military

- Malware is a major part of most military's cyberarsenals
- Used for espionage, attack, and "preparing the battlefield"
- Why viruses and worms?
- Because they *spread*—and can get to the target when it can't be reached directly

Stuxnet

- One of the most sophisticated known cyberattacks
- Attacked the Iranian nuclear centrifuge plant
- Generally attributed to the US and Israel
- Used four different “0-day” attacks—a sign that it was (a) created by a sophisticated party, and (b) used against a very valuable target
- Spread via USB sticks and on-LAN attacks
- Attacked the *Programmable Logic Controllers* that controlled the centrifuge motors
- Altered the displays to make it seem like all was well

Theories about the Initial Stuxnet Infection

- Original infection was on an outside vendor's machine
- Flash drives in the parking lot—it's worked on penetration tests
- Infected USB stick inserted by a corrupted insider
- Infected USB stick inserted by a spy
- Use some other technique to attack the outside machine; let it infect USB drives
- Warning: these are only theories—there is no hard evidence

Fundamental Issues

- The execution environment of infected applications is too large
- Why should (most) executables be able to overwrite (most) other files?
- Classic operating system designs used userid as the scope of file protection—but that's not right for virus protection
- All modern operating systems support easy-to-use sandboxes

☞ Sandboxing is mandatory for apps in the Apple and Microsoft app stores

- By definition, sandboxes limit the execution environment
- Worms are a harder call—so much stuff today is network-enabled

Is Antivirus a Good Idea?

- Antivirus software has a large execution environment: it can touch more or less any part of the system
- All file creation and (often) file open requests go through the antivirus software
- What if it's buggy?
- Today's operating systems have built-in defenses, e.g., Windows Defender
- Is traditional AV software still a good idea?
- Many security experts are wondering