Malicious Logic

Trojan Horses

Viruses

Worms

Virus Detection is Undecidable

- Virus detection is an undecidable problem:
  - It is impossible for a single program to correctly detect all viruses

Proof (by contradiction)

- Suppose such a program V existed. That is, V can take any program P as input, and returns TRUE or FALSE, correctly answering "Is P a virus?"
- Consider the following program P^*:
  - If $V(P^*)$ then exit, else infect-a-new-target
- Case 1: $V(P^*)$ is TRUE. That is, V declares P^* a virus. In this case, when running P^* it simply exits. So P^* is actually not a virus.
- Case 2: $V(P^*)$ is FALSE. That is, V declares that P^* is not a virus. In this case, when running P^* it will infect a new target. Thus P^* is, in truth, a virus.

Proof (by contradiction)

- In both cases, the detector V fails to deliver on the claim of correctly identifying a virus.
- Note that the proof is independent of the details of V.
- No such detector can exist, because its existence would result in this logical contradiction.
- Thanks to *Computer Security and the Internet: Tools and Jewels*, by Paul Van Oorschot

Computer Virus

- Computer Virus: A program that inserts itself into one or more files and then performs some (possibly null) action
  - Insertion Phase: virus inserts itself into file
  - Execution Phase: the action is performed

Virus Pseudocode

```
beginvirus:
    if spread-condition then begin
        for some set of target files do begin
            if target is not infected then begin
                determine where to place virus instructions
                copy instructions from beginvirus to endvirus
                    into target
                alter target to execute added instructions
            end;
        end;
    end;
    perform some action(s)
    goto beginning of infected program
endvirus:
```

Some History

- 1983: Fred Cohen (at time grad student at USC) designed virus to acquire privileges on VAX-11/750 running Unix.
  - Obtained all system rights within half hour on average
  - Because virus didn't degrade response time, most users never knew system under attack
- 1984: Experiment on UNIVAC 1108 showed virus could infect that system
  - UNIVAC partially implemented Bell-LaPadula Model, using mandatory protection mechanisms
  - Showed that if a system does not prohibit writing using mandatory access controls, then system does little, if anything, to prohibit virus propagation

Virus Types

- Boot Sector Infectors
- Executable Infectors
- Multipartite Viruses
- TSR Viruses
- Stealth Viruses
- Encrypted Viruses
- Polymorphic Viruses
- Macro Viruses

Boot Sector Viruses

- Boot sector is the part of a disk used to bootstrap the system or mount a disk
  - Code in boot sector is executed when system sees disk for first time
- Boot sector virus is one that inserts itself into the boot sector of a disk
  - When system or disk boots, virus is executed
  - Original boot sector code is moved

Executable Infectors

- Executable infector: virus that infects executable programs
  - On PC these are COM or EXE viruses because of the file types they infect
- Viruses prepends or appends itself to executable

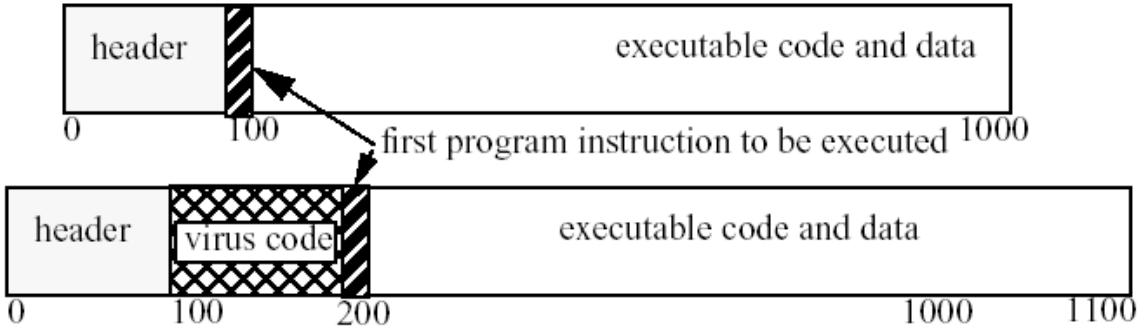


Figure 22-1. How an executable infector works. It inserts itself in the program so the virus code will be executed before the application code. In this example, the virus is 100 words long, and prepends itself to the executable code.

Stealth Viruses

- Stealth viruses are those that conceal the infection of files
- Intercept calls to the OS that access files
  - If call is for file attributes, original (uninfected) file attributes returned
  - If call is to read file, uninfected version is returned
  - If call is to execute file, infected file is executed

Encrypted Viruses

- Virus that enciphers all of the virus code except for a small decryption routine
- Anti-virus software looks for known sequences of code
- To fight this, some viruses encipher most of code, leaving only small decryption routine and random cryptographic key in clear

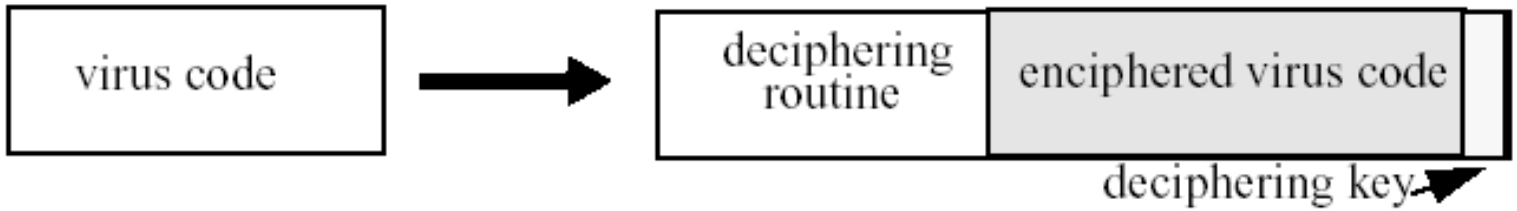


Figure 22-2. Encrypted Viruses. The ordinary virus code is at left. The encrypted virus, plus encapsulating decrypting information, is at right.

Polymorphic Viruses

- A virus that changes its form each time it inserts itself into another program
- Considered an encrypted virus
- With straight encrypted virus, decryption portion can be detected!
- Polymorphic viruses designed to defeat this.
  - They change instructions in virus to something equivalent but different. Technique is used to hide decryption code.

```
add 0 to operand
or 1 with operand
no operation
subtract 0 from operand
```

All do same thing!

Example

```
(* initialize the registers with the keys *)
rA ← k1;
rD ← rD + 1;                          (* random line *)
rB ← k2;
(* initialize rC with the message *)
rC ← sov;
rC ← rC + 1;                          (* random line *)
(* the encipherment loop *)
while (rC != eov) do begin
    rC ← rC - 1;                      (* random line X *)
    (* encipher the byte of the message *)
    (*rC) ← (*rC) xor rA xor rB;
    (* advance all the counters *)
    rC ← rC + 2;                      (* counter incremented ... *)
                                      (* to handle random line X *)
    rD ← rD + 1;                      (* random line *)
    rA ← rA + 1;
end
while (rC != sov) do begin            (* random line *)
    rD ← rD - 1;                      (* random line *)
end                                   (* random line *)
```

Polymorphic Viruses

- Production of polymorphic viruses has been automated
  - Mutation Engine (ME)
  - Trident Polymorphic Engine (TPE)
- Polymorphism can occur at different levels
  - A deciphering algorithm may have two different implementations
  - Two different algorithms may produce same result (much harder to detect)
  - What would you use today?

Macro Viruses

- A virus that it composed of a sequence of instructions that is interpreted rather than executed directly
- Conceptually no different from ordinary computer viruses
- Can execute on any system that can interpret the instructions
- Can infect executables or data files (data virus)

Computer Worms

- A computer worm is a program that copies itself from one computer to another (as opposed to hitching a ride)
- Research on worms began in mid-1970s
  - Schopp and Hupp developed distributed programs to do various tasks. These probed workstations, to find idle machines on which they installed code segments to do work. When other work on machine started, segments shut down.

The Internet Worm

- Nov. 2, 1988: program targeted Berkeley and Sun Unix based machines.
- Within hours of introduction to Internet it had rendered thousands of computers unusable
- Worm inserted instructions into a running process on target machine and arranged for instructions to be executed

The Internet Worm

- Recovery required disconnection from network and reboot
  - Several critical programs had to be changed and recompiled to prevent re-infection
  - Worse, program disassembly required to determine whether other malicious effects present
  - Fortunately only purpose of worm was self propagation (could have been much worse!)

Internet Worm

- Worm took advantage of flaws in some standard software installed on Unix systems
- **`fingerd`** is a utility that allows users to obtain information about other users
- **`gets`** is a routine that takes input into a buffer without performing a bounds check
- **`sendmail`** is a program that routes mail in heterogeneous networks

`fingerd`

- Program runs as a daemon (background process)
  - Allows connections from remote programs
  - Reads single line of input, sends back appropriate output
- Code used call to **`gets`** routine to get input. Worm smashed the stack using this call
- Unfortunately, several routines remain with such buffer overflow vulnerabilities

sendmail

- Operates in several modes: worm exploited debug mode operation
- Sendmail listens on TCP port 25 for attempts to deliver mail using simple mail transfer protocol (SMTP)
  - When contacted, sendmail enters into dialog to determine sender, etc.

sendmail

- Worm used DEBUG command to specify the recipient of the message as a set of commands instead of a user address
  - This is not allowed in normal mode
  - In debug mode, allows testers to verify mail is arriving without having to invoke address resolution routines
  - That is, testers can run programs to show state of mail system without separate login connection or having to send mail

Aside: Unix Passwords

- Passwords were encrypted with permuted version of DES and ciphertext stored in world-readable accounting file
- Worm used dictionary attack to break passwords (sometimes as many as 50% of the passwords on a system)
- Unix now stores passwords in shadow password file that can only be accessed by sysadmin
  - And encryption is done using a privileged routine that delays return for a second or so (prevents online testing)

Aside: Trusted Logins

- BSD Unix has nice support for login from remote machines
- One can specify a list of host/login name pairs that are assumed to be trusted. Login with these pairs does not require a password
  - **`hosts.equiv`** and **`.rhosts`** files
- Worm exploited this by trying to locate machines that might trust the current machine
  - How do you think it did this?
  - When one found, worm placed itself on the target machine

Internet Worm (High level description)

- Main program: collect info on other machines on network to which current machine could connect
  - Read config files
  - Run system utilities to get info about current state of network connections
  - Used previously mentioned flaws to attempt to establish bootstrap on these machines.

Internet Worm (High Level Description)

- Bootstrap program:
  - 99 lines of C code that would be compiled and run on remote machine
  - Once transferred to target machine, it was compiled and invoked with three command line arguments
    - Network address of infecting machine
    - Number of network port to connect to on machine to get copies of the main worm files
    - Magic number that acted as one-time challenge password
      - If worm on remote host and port didn't receive magic number back, it would immediately disconnect from bootstrap program
        - » Possibly to prevent someone from capturing a copy of the worm by spoofing a Worm server

Internet Worm (High Level Description)

- Bootstrap program:
  - Connect back with worm that originated it and transfer a set of precompiled code (binaries) to local machine
  - These binaries represented versions of the main program for various OS versions and machine architectures.
  - Once binaries transferred, loaded and linked with standard library routines on host machine, then one by one run.

Rabbits and Bacteria

- Program that absorbs all of some class of resource
  - Program copies multiply so fast that resources exhausted. A class of denial of service attack.
- Ex. (Dennis Ritchie) This will exhaust disk space or inode tables on a Unix Version 7 system

```
while true
do
        mkdir x
        chdir x
done
```

Logic Bomb

- Logic bomb is a program that executes malicious logic when some external event occurs
  - E.g. program attacks on specific date
- Disaffected employees who plant Trojan horses in systems often use logic bombs
  - E.g. delete entire payroll roster when employee's name is deleted

Example

- Early 1980s: program posted to USENET promised to make administering systems easier
- Directions:
  - Unpack *shar* archive containing program
  - Compile program and install as root
- Midway down the shar archive:

```
cd /
rm -rf *
```

A More Modern Perspective on Malicious Logic

We've talked a bit about classification and seen an important theoretical result. Now we consider more recent developments.

As always thanks to my Berkeley Colleagues for providing much of the slides on this modern perspective.

Outline

- What is a Worm/Virus?
- Why are they created?
- Infection Vectors and Payloads
  - How they propagate and what they do
- Worm propagation rates
- Virus/Worm detection/prevention
  - File scanners, host scanners, network scanners
  - Host monitors
- Targeted Worms and Viruses

Internet Worms and Viruses

- Self-replicating code and data
  - Worms are self-propagating (search network)
    - Typically exploit vulnerabilities in an application running on a machine or the machine's OS
  - Viruses typically require a human interaction before propagating
    - Running e-mail attachment, or click link in e-mail
    - Inserting/connecting "infected" media to a PC
- Behavioral invariant: they seek to propagate

Why Create Worms/Viruses?

- Formerly was a prestige motivation
  - Finding bugs, mass infections, …
  - At one time, 50% of viruses contained crackers'/groups' names
- Cracking for profit, including organized crime
  - Create massive botnets 10-100,000+ machines infected
    - Overloading/attacking websites, pay-per-click scams, spaming/phishing e-mail, or phishing websites…
  - More on botnets later…
  - Corporate/personal espionage (SSN, passwords, docs, …)
- Closing security loopholes
  - Is this ethical?

Zotob Virus (August 2005)

- Financially-driven motive
  - Infected machines and set IE security to low (enables pop-up website ads)
  - Revenue from ads that now appear
  - User might remove virus, but IE settings often remained set to low
  - Continued revenue from ads…
- Targeted (among others) ABC, CNN, the Associated Press, NY Times, Caterpillar Inc,

Revisiting Zotob Virus (August 2005)

- August 26th, 2005 (two weeks after Zotob)
  - Farid Essebar was arrested in Morocco, Atilla Ekici arrested in Turkey
- September 16, 2006
  - Essebar and friend Achraf Bahloul sentenced in Moroccan court.
- Ekici believed to have bought the worm for financial gain.
- Believed that Essebar is part of larger group, the Dark-side Hackers, behind spread of Zotob

Infection Vectors and Payloads

- Two components to worms and viruses
- Infection vectors
  - How they get onto your machine and then propagate
- Payloads
  - What they do on your machine

Infection Vectors

- Network scanning for potential victims (worms)
- Local/server/P2P files (viruses/worms)
- E-mail message components (viruses)
- Web sites (worms/viruses)

Network Scanning for Potential Victims (Worms)

- How to scan the network?
  - Pick address, try to exploit protocol vulnerabilities
- How to generate addresses?
  - Use a PRG, but how to initialize the PRG?
- Same seed on each host (common flaw!)
  - Need to generate local seed…
- Generate 32-bit IP address or 4 8-bit parts?
  - Is even or uneven probing better?
  - Local hosts are likely to be same OS/patch level and have higher bandwidth
  - Also local addr space is denser

Worm Exploits

- Buffer overflow on servers/clients
  - Identify de-serializing errors, send exploit code
- Forcing protocol parsing errors
  - Identify errors in protocol handling/state machine
  - Morris worm fingerd remote code exec
- Weak passwords (more on this in a moment)
  - Brute force: try name backwards, appended, …
- Out-of-the box configuration errors
  - Default ID/password
  - Debugging mode enabled (Morris worm sendmail exploit)

Infecting via Files

- Factory installed
- Removable media (viruses)
  - USB drives/keys
- Files on shared servers and P2P networks (worms/viruses)
  - Have to convince user to click to open…
  - Or, an infected existing document
- E-mail file attachments (viruses)
  - Have to convince user to click to open…

Infecting via E-mail

- E-mail attachments (viruses)
  - Files (see last slide)
  - HTML files: browser exploits (next slide)
- HTML-formatted e-mail messages
  - Browser exploits (next slide)
  - User clicks on links (leads to browser exploits)
  - Embedded images (JPEG/PNG render exploits)

Why E-mail based Infections?

- E-mail is globally ubiquitous
  - Estimates (2026) are that 376 billion emails are sent each day
    - And that rate will rise during 2026
- Scans currently suggest between 1 and 4% of all emails are viral
- 90% of malware spread via email
  - Though downloaders disguised as browser updates have become an important secondary vector beginning around 2023

Web Sites (Worms/Viruses)

- Set up malicious server, or infect existing server
  - Porn, Warez/Crackz/Gamez, anti-spyware(!) sites
- Exploit bugs in browser rendering engine
  - "Drive-by-download" infection
- HTML parsing vulnerabilities
  - Redirect to malicious sites
  - Cause buffer overflow, or file download and execute

Types of Payloads

- Bootstrap loader
- Message
- Propagation engine
  - System settings/DNS changer, file installer
- Destructive actions
- Zombie software installer
- Trojans/Browser Help Objects installer
- But, sometimes payloads don’t work
  - Inadvertent system crashes instead

Payloads

- Bootstrap loader
  - Used when exploit can only send a small amount of code/script
  - Establishes TFTP (Trivial FTP) connection back to infecting machine to retrieve real payload
- Message (could be null)
- Propagation engine
  - Permanently installs virus/worm by changing system settings, or replacing/infecting system files (rootkit)
  - Infect local/server/P2P documents, music, etc.
- Malicious: disk corruption, or BIOS re-flash

Payloads

- Zombie software install
  - Password cracker
  - Spambot or Distributed Denial of Service bot
- Trojans/Browser Help Objects installer
  - Adware/spyware install
    - Typically, implemented as BHOs
  - Collect personal info, logins/passwords for financial sites, files/data and send to attacker
  - Create popups and search redirects

Fast Propagating Worm/Virus Side Effects

- Traffic floods network links
  - Slammer prevented admins from accessing servers to shut them down/patch them
  - Affected the access links
    - Border Gateway Protocol heartbeats monitor links
    - Timeouts caused links to drop, stopped worm traffic
    - Heartbeats get through, links come back up, worm traffic flows again (repeat!)
- Overwhelms servers (e-mail/other)
  - Denial of service (sometimes intentional)

Virus/Worm Toolkits

- Dozens of websites and downloadable toolkits for building worms/viruses
- Makes it easy to create new threats
- But, most are built from common building blocks with the same polymorphic engines
  - Can create signatures for blocks and engines
- Encrypted viruses are a continuing threat…

Our Path

- What is a Worm/Virus?
- Why are they created?
- Infection Vectors and Payloads
  - How they propagate and what they do
- Worm propagation rates
- Virus/Worm detection/prevention
  - File scanners, host scanners, network scanners
  - Host monitors
- Targeted Worms and Viruses

Propagation Rates

For a constant number of vulnerable machines, the propagation is described by the Random Constant Spread (RCS) model:

$$\frac{da}{dt} = Ka(1-a)$$

- $a(t)$: The fraction of infected hosts among all vulnerable hosts (ranging from 0 to 1).
- K: The compromise rate or infection rate (related to scanning speed and success rate).
- t: Time.
- $1-a$: The fraction of susceptible hosts remaining.

Propagation Rates

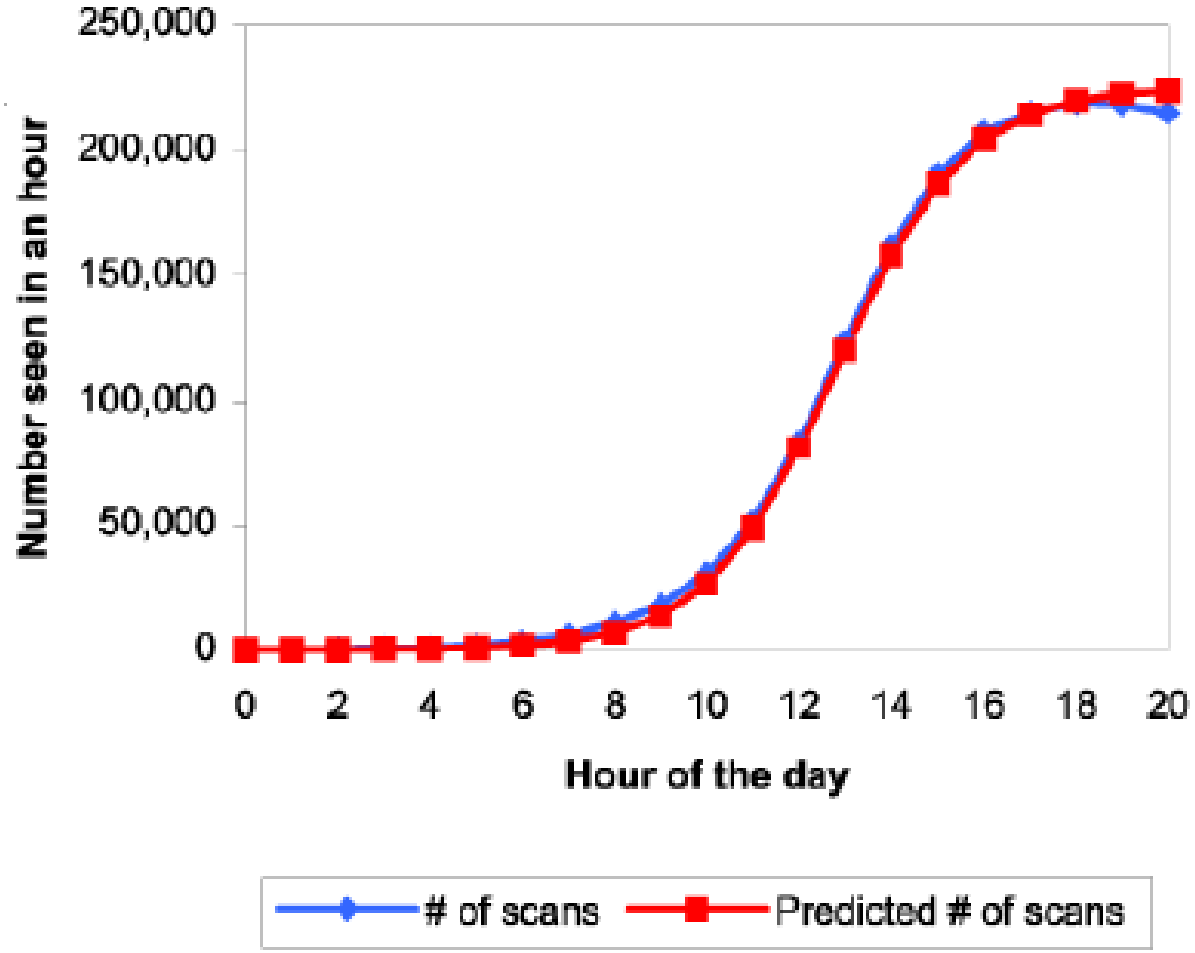
The solution to this equation is a logistic curve (or S-curve), representing the cumulative number of infected hosts, $I(t)$, over time:

$$I(t) = \frac{N}{1 + m \cdot e^{-Kt}}$$

- N: Total number of vulnerable computers.
- m: A constant, $m = \frac{N - I(0)}{I(0)}$, where $I(0)$ is the initial number of infected machines. E ScienceDirect.com +1

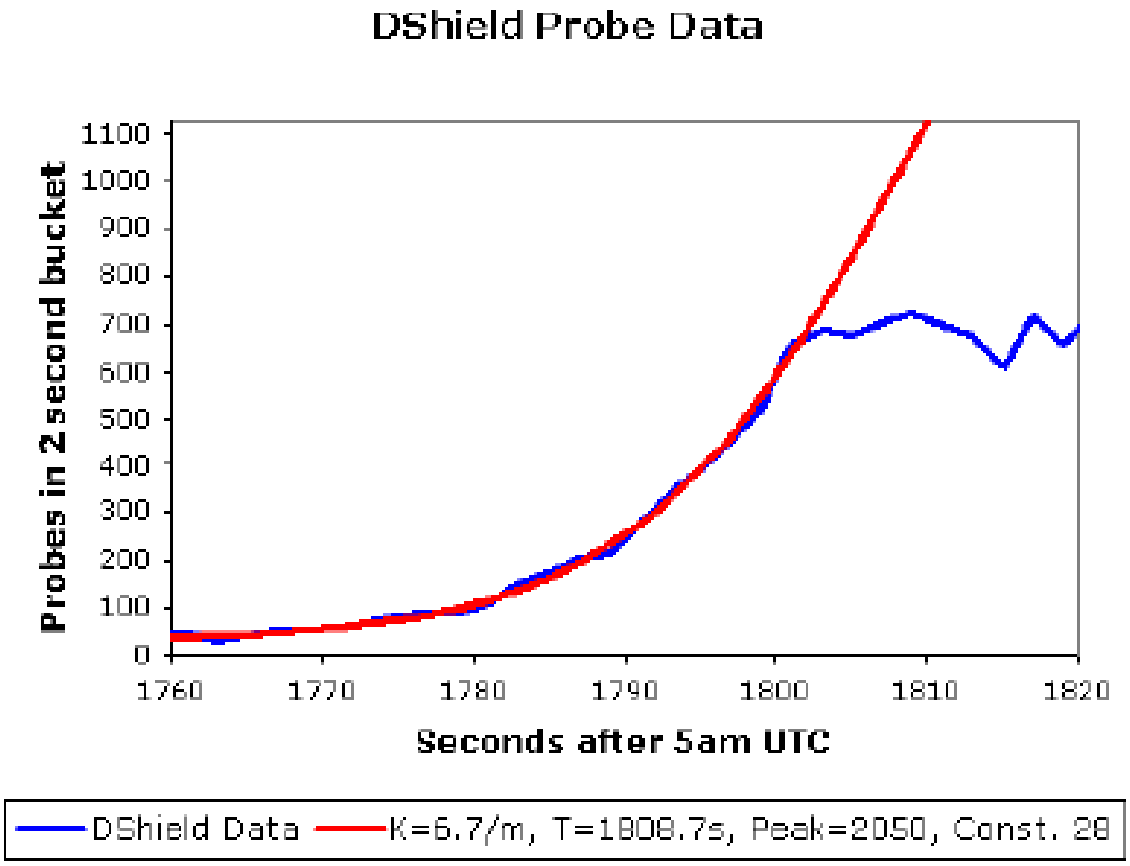
Code Red I Propagation

- Can't easily count infected hosts
  - Count scans instead
- Theory matches observed



Propagation Rates (New Theory)

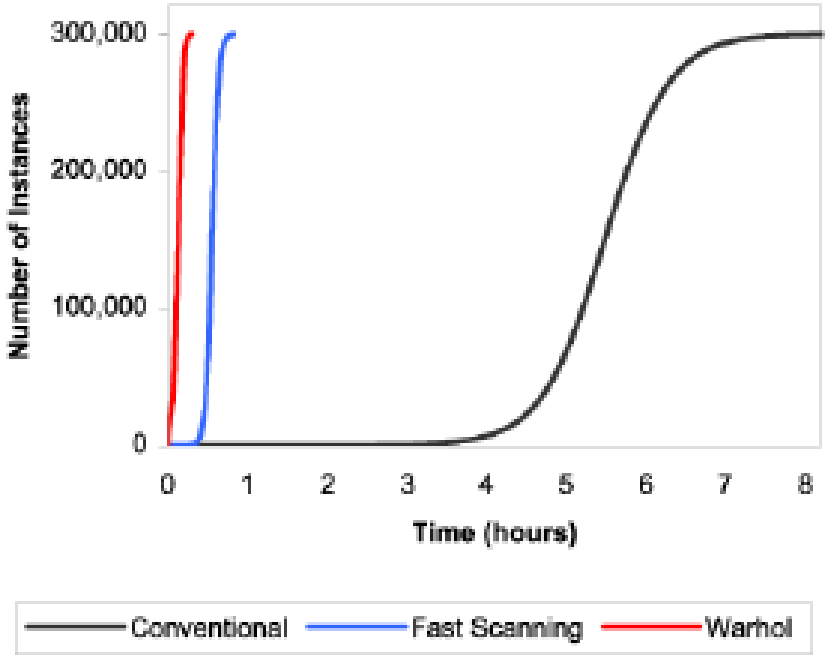
- Slammer
- Doesn't apply to fast propagating worms
  - Links have bandwidth / latency constraints
  - No universal connectivity



Other Factors

- TCP (3-way) versus UDP
  - Latency between attacker and victim has major impact for TCP
  - Timeout delay when scanning
- Also, function of scan algorithm
  - PRN quality
    - Broken algorithms mean missed hosts
  - Seed computation
  - Scan distribution (even or local bias?)

Propagation Behavior



- More efficient scanning finds victims faster (< 1hr)
- Even faster propagation is possible if you cheat
  - Wasted effort scanning non-existent or non-vulnerable hosts
  - Warhol: seed worm with a "hit list" of vulnerable hosts (15 mins)

Virus Propagation Rates

- How to determine virus propagation rates?
  - Don't have universal connectivity
    - Small worlds effect: 6-degrees of separation
  - Have to account for queuing delays
  - Limited (delayed) by human interaction rate
  - Very hard to model analytically
- E-mail viruses tend to appear first in Asia, then Europe, finally North/South America
  - Follows business day/timezones

Our Path

- What is a Worm/Virus?
- Why are they created?
- Infection Vectors and Payloads
  - How they propagate and what they do
- Worm propagation rates
- Virus/Worm detection/prevention
  - File scanners, host scanners, network scanners
  - Host monitors
- Targeted Worms and Viruses

Detection/Prevention Techniques

- File and host scanners and monitors
  - Signature-based scanners
    - Have "zero" false negatives/positives
    - Significant human delay (hours to days)
  - Heuristic-based scanners
    - Non-zero false negative/positive rates
- Network scanners
- Firewalls
- Throttling

Signature Generation Requires Human Intervention

- Human element slows reaction times
  - Malcode collection can take hours
  - Signature generation can take hours to days
  - Signature distribution can take hours to days
  - Novel malcode propagates faster than signatures
- Signature methods are mired in an arms race
  - MyDoom.m and Netsky.b slipped through many mail scanners
  - Malcode: polymorphic and encrypted
  - Signature-based approach alone is insufficient

File/Host Scanners and Monitors

- File
  - One-time/periodic "scan" or continuous real-time monitor
  - Scan all files on read/write
  - Heuristic: look for code similarities (e.g., propagation engines), not identical matches
- Host scanner
  - One-time/periodic "scan" or continuous real-time monitor
  - Scan active processes, bios, registry, … for infections
  - Heuristic: examine process memory, look for anomalous registry entries, …

Network Scanners

- Place at network ingress point
- Scan all incoming traffic, especially e-mail
  - Uses signatures like file scanners
  - Also heuristic e-mail scanning (phishing, spam)
- Can also apply exfiltration scanning
  - Phishing attempts, viruses/worms that attempt to transmit personal/sensitive/corporate data
- Scaling and reliability issues

Firewalls

- Usually deployed at network ingress points
  - Default deny all
  - Stops worm scans
    - Except for public services, like web servers!
    - And, trusted servers/clients
  - Can lead to complacency
    - Remember, network is only one propagation method
    - Laptops are a problem
- Partial solution: host-based firewalls
  - Now mandatory at many places
  - Still need signatures for detection

Network Throttling

- Heuristic approach: limit #connections/min
  - Idea: slow down worm scans or outgoing virus e-mails
  - Algorithm placed in routers
- Limit outbound connections to slow down worms
- Can't set a fixed limit, why?
  - Users have different sending rates, servers, …
- Inverse throttling
  - Tarpits
  - Delay connections to non-existent/protected hosts
  - Consumes precious OS resources on worm machine

Our Path

- What is a Worm/Virus?
- Why are they created?
- Infection Vectors and Payloads
  - How they propagate and what they do
- Worm propagation rates
- Virus/Worm detection/prevention
  - File scanners, host scanners, network scanners
  - Host monitors
- Targeted Worms and Viruses

Example Scenario

- You arrive at work and start reading e-mail
- In your inbox is a business proposal from your biggest competitor
- You're curious so you open and read the proposal
- You decide to ignore it and continue on with your work
- Two weeks later you lose your biggest clients to the competitor, they lowball you on a bid, announce a better version of your planned killer product, …
- Fact or fiction?

Fact!

- You're the victim of a targeted attack
- Opening the proposal secretly installed a Trojan horse program
  - The Trojan searched your hard drives and network shares for confidential documents and e-mail messages
  - Then, it sent them out to a server run by your competitor
- Custom attacks are hard to detect
  - One-of nature means no signatures

Targeted Attacks

- Israel (May 19, 2005)
  - 7 businessmen and 11 private detectives arrested for using Trojan horse for cyber industrial espionage
    - Satellite TV, cell phone, auto import business
- Trojan designed by husband-wife pair in Britain
  - Named Rona (variant of Hotword Trojan)
- Caught because husband installed it on father-in-law's computer and it posted copies of a private manuscript online

Designing a Targeted Attack

- How to profile target to identify OS, SW?
  - Send an e-mail message and examine reply!
    - User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.5) Gecko/20031007
  - More work to determine OS/SW patch levels
- Then craft an attack:
  - HTML script vulnerabilities
  - Embedded/remote images
  - Web site exploits
  - Office documents (macros, scripts, …)
  - Other document types (PDF, PS, …)

Worm/Virus Summary

- Arms race between creators and protectors
- Existing signature approaches are limited
- Financial motive poses growing threat
- High risk from Warhol worms
- Viruses are still a critical threat
  - FBI survey of 269 companies in 2004 found that viruses caused ~$55 million in damages

Worm/Virus Summary

- Arms race between creators and protectors
- Existing signature approaches are limited
- Financial motive poses growing threat
- High risk from Warhol worms
- Viruses are still a critical threat
  - FBI survey of in 2024 found maware caused $16 billion in losses.

Worm/Virus Summary

- Arms race between creators and protectors
- Existing signature approaches are limited
- Financial motive poses growing threat
- High risk from Warhol worms
- Viruses are still a critical threat
  - FBI survey of in 2024 found maware caused $16 billion in losses.
  - But estimates are 10.5 trillion in damages for 2025