Applied Cryptography and Network Security

Adam J. Lee

adamlee@cs.pitt.edu

6111 Sennott Square

Lecture #25: Viruses and Worms

April 8, 2014

Announcements

Please sign up for a Project 4 demo---check the web for available slots

Project 5 is now posted. It’s due Thursday 4/17.

Overview

Definitions

Case studies:

- The Brain virus
- The Morris worm
- Code Red

Into the future

Malicious logic is a set of instructions that cause an organization's security policy to be violated

One common type of malicious logic is the Trojan horse, which is a program that has a well known overt effect, and an unknown covert effect.

Example: NetBus

- Allows an attacker to remotely administer a Windows NT box
- Remote admin code was bundled with games/"fun" programs

Not all Trojan horses are so easy to spot…

Example: Thompson's login program

- Modification to UNIX login program that accepted a default password
- Modify compiler to insert default case when compiling the login program
  - ➢ Backdoor is not visible in login source code
  - ➢ Backdoor code is visible in compiler code
- Modify compiler to insert above code if the compiler is being compiled
- Install malicious compiler binary, and benign compiler source
  - ➢ Backdoor invisible in all source code

The problem with a Trojan horse is that you need to convince the victim to install the host program

A computer virus is a program that inserts itself into one or more files and then performs some action

The term computer virus was coined by Fred Cohen and his advisor Len Adelman at USC in 1983

- Cohen's thesis is one of the few theoretical results regarding viruses
- Key result: No algorithm can detect computer viruses precisely
- This is why virus scanners are largely heuristic!

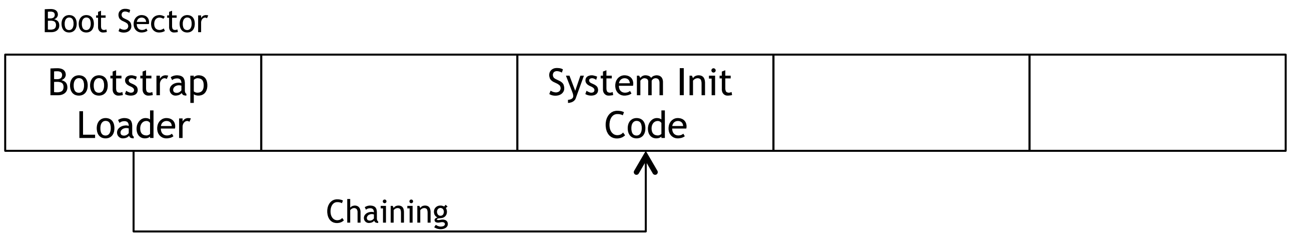
Why are viruses called viruses? Because they self-replicate by attaching themselves to host programs!

In the days before widespread Internet availability, virus writers had to get creative to enable the spread of a virus beyond one system!

Some viruses spread by infecting a drive's boot sector

How does the boot process work?

- System firmware reads a specific disk sector (the boot sector) into memory
- The system then jumps to that memory location
- The boot program then begins loading the OS



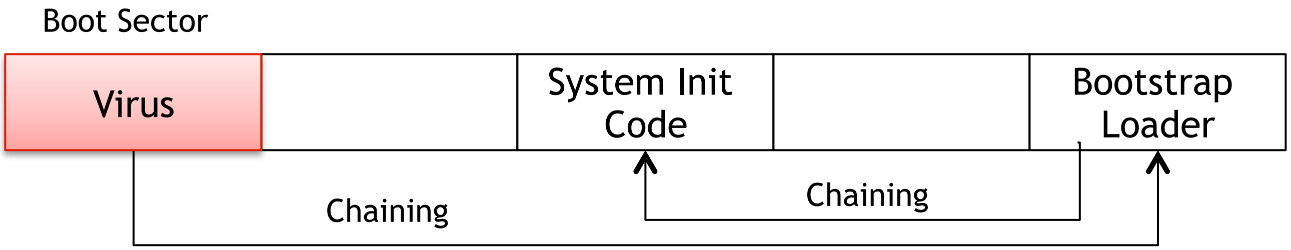
Boot Sector

Bootstrap Loader

System Init Code

Chaining

A boot sector virus hijacks this process to facilitate its spread



Boot Sector

Virus

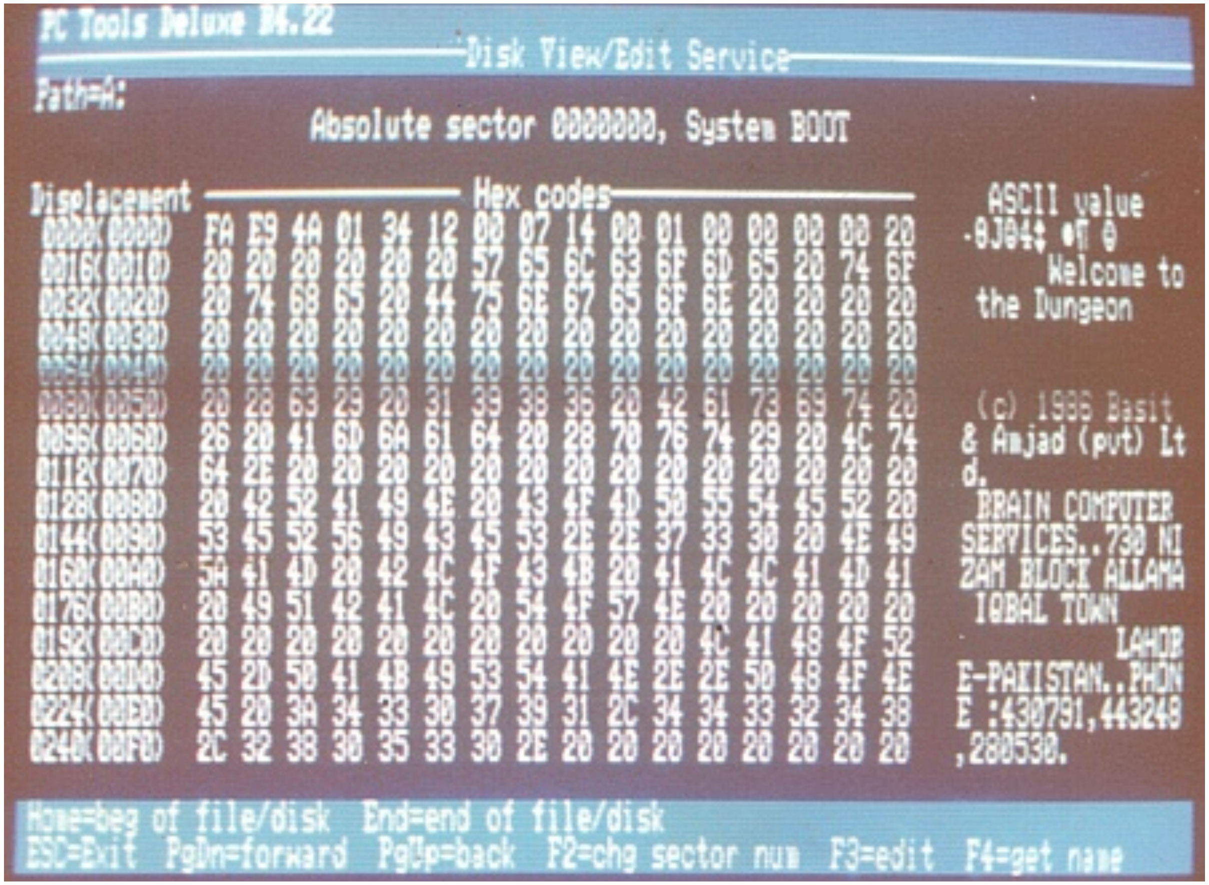
System Init Code

Bootstrap Loader

Chaining

Chaining

Case Study: The Brain virus



```
PC Tools Deluxe R4.22
                                  ─Disk View/Edit Service─────────────────
Path=A:
                     Absolute sector 0000000, System BOOT

Displacement ──────────── Hex codes─────────────────────────     ASCII value
 0000(0000)  FA E9 4A 01 34 12 00 07 14 00 01 00 00 00 00 20   ·θJ04‡ ●¶ θ
 0016(0010)  20 20 20 20 20 20 57 65 6C 63 6F 6D 65 20 74 6F         Welcome to
 0032(0020)  20 74 68 65 20 44 75 6E 67 65 6F 6E 20 20 20 20    the Dungeon
 0048(0030)  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
 0064(0040)  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
 0080(0050)  20 28 63 29 20 31 39 38 36 20 42 61 73 69 74 20    (c) 1986 Basit
 0096(0060)  26 20 41 6D 6A 61 64 20 28 70 76 74 29 20 4C 74   & Amjad (pvt) Lt
 0112(0070)  64 2E 20 20 20 20 20 20 20 20 20 20 20 20 20 20   d.
 0128(0080)  20 42 52 41 49 4E 20 43 4F 4D 50 55 54 45 52 20    BRAIN COMPUTER
 0144(0090)  53 45 52 56 49 43 45 53 2E 2E 37 33 30 20 4E 49   SERVICES..730 NI
 0160(00A0)  5A 41 4D 20 42 4C 4F 43 4B 20 41 4C 4C 41 4D 41   ZAM BLOCK ALLAMA
 0176(00B0)  20 49 51 42 41 4C 20 54 4F 57 4E 20 20 20 20 20    IQBAL TOWN
 0192(00C0)  20 20 20 20 20 20 20 20 20 20 20 4C 41 48 4F 52              LAHOR
 0208(00D0)  45 2D 50 41 4B 49 53 54 41 4E 2E 2E 50 48 4F 4E   E-PAKISTAN..PHON
 0224(00E0)  45 20 3A 34 33 30 37 39 31 2C 34 34 33 32 34 38   E :430791,443248
 0240(00F0)  2C 32 38 30 35 33 30 2E 20 20 20 20 20 20 20 20   ,280530.

Home=beg of file/disk  End=end of file/disk
ESC=Exit  PgDn=forward  PgUp=back  F2=chg sector num  F3=edit  F4=get name
```

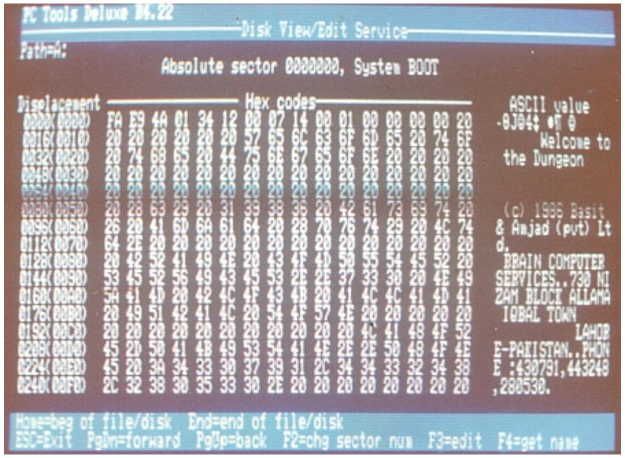
http://www.noulakaz.net/weblog/2007/02/24/welcome-to-the-dungeon/

The Brain virus was one of the first, and most carefully studied, computer viruses

Brain was a boot sector virus that originated in Pakistan in 1986

Two brothers, Basit and Amjad Farooq Alvi, supposedly wrote Brain to protect medical software that they wrote from copyright infringement

However, the Brain virus contained no malicious payload and actually advertised the brothers' contact information!



```
PC Tools Deluxe R4.22
                         Disk View/Edit Service
Path=A:
                 Absolute sector 0000000, System BOOT

Displacement  ───────────── Hex codes ─────────────────   ASCII value
 0000(0000)  FA E9 4A 01 34 12 00 07 14 00 01 00 00 00 00 20  ·ΘJ☺4↕ •¶ ☺
 0016(0010)  20 20 20 20 20 20 57 65 6C 63 6F 6D 65 20 74 6F        Welcome to
 0032(0020)  20 74 68 65 20 44 75 6E 67 65 6F 6E 20 20 20 20   the Dungeon
 0048(0030)  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
 0064(0040)  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
 0080(0050)  20 28 63 29 20 31 39 38 36 20 42 61 73 69 74 20   (c) 1986 Basit
 0096(0060)  26 20 41 6D 6A 61 64 20 28 70 76 74 29 20 4C 74  & Amjad (pvt) Lt
 0112(0070)  64 2E 20 20 20 20 20 20 20 20 20 20 20 20 20 20  d.
 0128(0080)  20 42 52 41 49 4E 20 43 4F 4D 50 55 54 45 52 20   BRAIN COMPUTER
 0144(0090)  53 45 52 56 49 43 45 53 2E 2E 37 33 30 20 4E 49  SERVICES..730 NI
 0160(00A0)  5A 41 4D 20 42 4C 4F 43 4B 20 41 4C 4C 41 4D 41  ZAM BLOCK ALLAMA
 0176(00B0)  20 49 51 42 41 4C 20 54 4F 57 4E 20 20 20 20 20   IQBAL TOWN
 0192(00C0)  20 20 20 20 20 20 20 20 20 20 20 4C 41 48 4F 52             LAHOR
 0208(00D0)  45 2D 50 41 4B 49 53 54 41 4E 2E 2E 50 48 4F 4E  E-PAKISTAN..PHON
 0224(00E0)  45 20 3A 34 33 30 37 39 31 2C 34 34 33 32 34 38  E :430791,443248
 0240(00F0)  2C 32 38 30 35 33 30 2E 20 20 20 20 20 20 20 20  ,280530.

Home=beg of file/disk  End=end of file/disk
ESC=Exit  PgDn=forward  PgUp=back  F2=chg sector num  F3=edit  F4=get name
```

It is suspected that Brain was really a cute gimmick to draw attention to their business!

How did the Brain virus work?

When a system boots from an infected disk:

1. The virus loads itself in upper memory
2. It then resets the memory boundary below itself
3. Brain then mangles the system interrupt table
   - Interrupt 19 (disk read) is reset to point to the Brain code in memory
   - Interrupt 6 (unused) is then set to point to the old code for interrupt 19

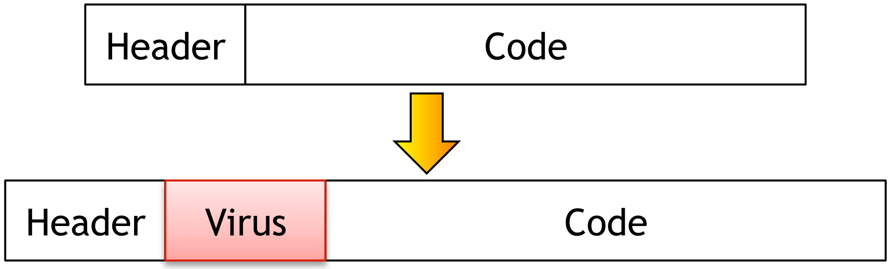
Whenever a disk read occurs

1. Interrupt 19 is triggered and transfers control to the virus
2. If the disk being read is not yet infected, the virus infects it
3. The virus then triggers interrupt 6 to allow the disk read to occur

Although Brain contained no malicious payload, it was used as a template for several more destructive viruses

Question: How do you think that users detected this virus?

There are many other types of viruses…



Executable viruses

- Attach to executable code
- Invoked when code is invoked
- ***Example:*** Jerusalem

TSR viruses

- TSR = Terminate and stay resident
- Virus stays in memory even after host process terminates (syscall interception)
- ***Examples:*** Brain and Jerusalem

Macro viruses

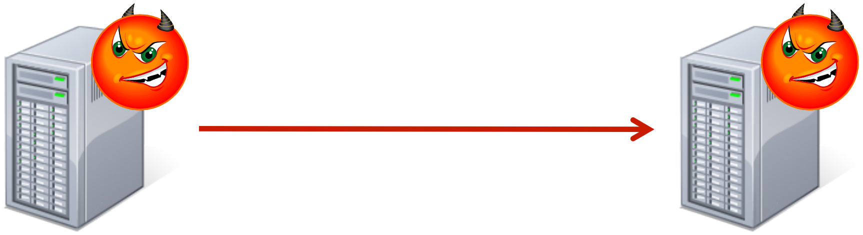
- Infect documents, not executables
- Not bound by system architecture
- ***Examples:*** Melissa

How fast can viruses spread?

In the early days, the speed with which a virus could spread was limited to the speed of human interactions

- Trading floppy disks
- Sharing spreadsheets or other documents at work
- Installing Trojan programs passed along by a friend
- …

The growing prevalence of the Internet during the late 80s and early 90s gave rise to computer worms, which are viruses capable of spreading themselves across many machines



The result: Much speedier infection rates!

Case Study: The Morris worm

The first major Internet worm was released in 1988

- Written by Robert Tappan Morris
- Launched around 5:00 PM on 2nd November 1988
- Originated at Cornell University

The worm was purportedly written to assess the size of the Internet and had no malicious payload

Unfortunately, unbounded replication of the worm brought many systems down

This worm used many techniques that we have already studied…

Eugene H. Spafford, "The Internet Worm: Crisis and Aftermath," Communications of the ACM, 32(6): 678-687, June 1989.

How did it spread?

Rather than reinvent the wheel, the Morris worm leveraged three well-known vulnerabilities to enable its spread across machines running Berkeley and Sun UNIX

Method 1: fingerd

- fingerd provides a lookup service for users' public contact information
- The version of fingerd running on many systems had a buffer overflow due to the use of the unchecked `gets()` routine

Method 2: A bug in sendmail

- sendmail is a popular e-mail routing program
- If operating in DEBUG mode, sendmail allows system commands to be transmitted over SMTP

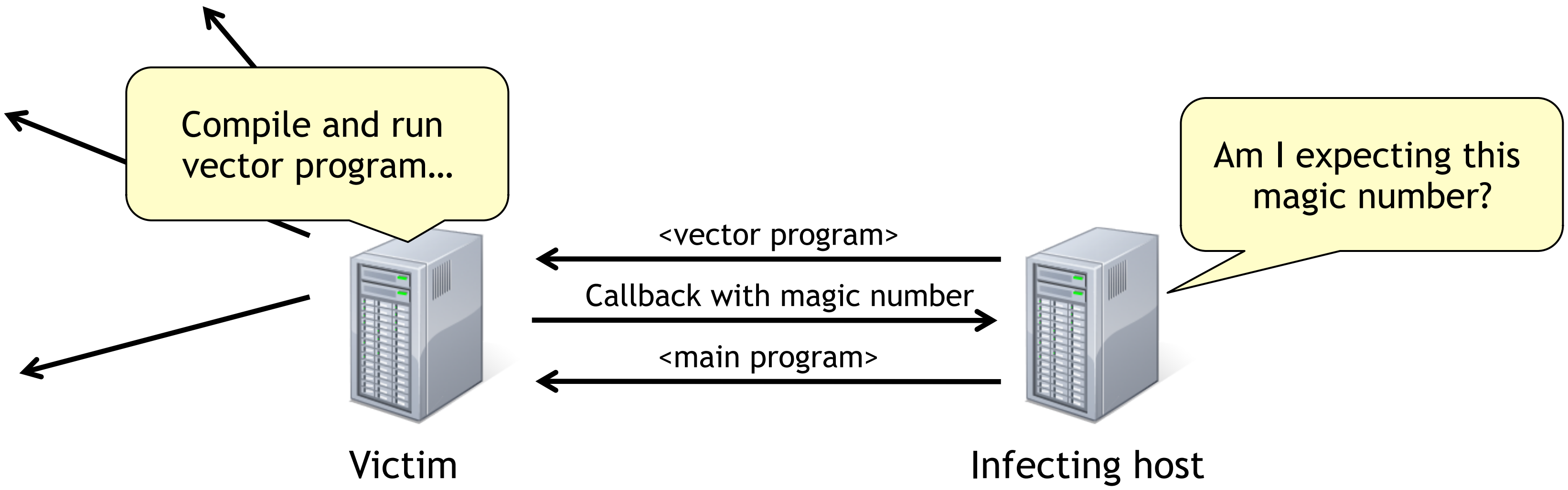
Method 3: Weak password security

- Many passwords are weak and can be broken with simple guessing
- The rsh protocol allows trusted users/hosts to bypass authentication

How did an infection proceed?

The worm consisted of two programs: a short vector (i.e., infection) program, and the main program

The vector program was 99 lines of C code, transferred using one of the previous three known exploit techniques


Compile and run vector program…
Am I expecting this magic number?
<vector program>
Callback with magic number
<main program>
Victim
Infecting host

If the (binary) attack programs would not run on the victim, the vector would delete everything to cover its tracks

What did the main program do?

The main program first gathered information about network interfaces and hosts on the local network, which was randomized to provide a set of hosts to attack

The main program then entered a simple state machine

- Read /etc/hosts.equiv and /.rhosts to look for trusted hosts to add to table
- Try to break user passwords using simple choices
- Try to break user passwords using a dictionary of 432 words
- Brute force user passwords using entire UNIX online dictionary

If a password was broken, it was used in conjunction with rsh to infect other hosts

Each state was run for short periods of time, between which the main program attempted to infect other hosts in the list to attack

The Morris worm actually took steps to prevent overloading a particular host

The worm would periodically check for copies of the worm running on the same host by connecting to a predetermined TCP socket

If another worm was found, one of the two would randomly quit

However, this didn't work terribly well…

- Race conditions in the code sometimes prevented worms from connecting
- Furthermore, 1 in 7 worms became immortal to prevent fake kills

Result: Many hosts had several copies of the worm running

It is interesting to note that the worm occasionally forked itself

- This gave the worm a new PID to prevent detection
- Also prevented the worms priority from downgrading over time

Outcomes of the Morris worm

Experts think that upwards of 6,000 hosts were infected with the Morris worm, though this number is an estimate

The Morris worm brought network security to the forefront

- More regular software patching/updating
- Broader proliferation of shadow password files
- Inspired much intrusion detection research

In response to this incident, DARPA funded CERT/CC to monitor computer vulnerabilities, and manage incident response

Robert T. Morris

- Was the first person convicted under the 1986 Computer Fraud and Abuse Act
- Is now a full professor at MIT ☺

Case Study: Code Red

Although the Morris worm was the first widespread worm, it was certainly not the last!

Software is being developed at an astounding rate

- Software is getting more complex
- Higher complexity leads to more bugs
- More bugs means more potential for exploits

The Code Red worm (v2) was launched on July 19, 2001

- Exploited a bug in Microsoft IIS server
- Infected over 359,000 hosts in under 14 hours!

Although this worm was released 13 years after the Morris worm, it is structurally very similar…

How did Code Red work?

Like the Morris worm, Code Red utilized a buffer overflow to propagate

- Microsoft IIS version 4.0 or 5.0 with the Indexing service installed

-
  ```
  /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
  NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
  NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
  NNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%
  u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531
  b%u53ff%u0078%u0000%u00=a
  ```

After a host was infected, it spawned 300-600 threads that would:

- Randomly choose an IP address and attempt to connect
- If success, attempt the above buffer overflow

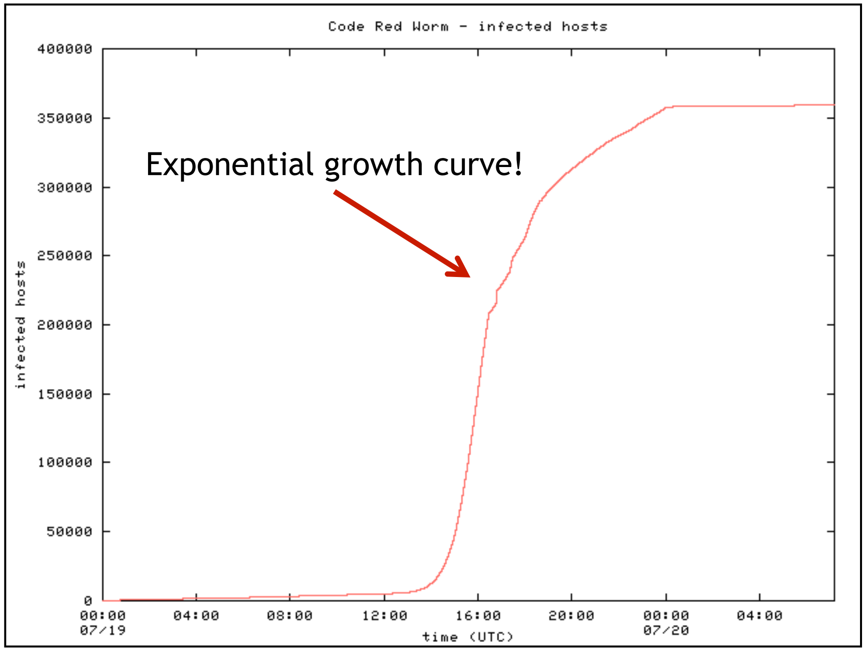
Code Red's behavior was dependent on the day of the month

- Day 1 - 19: Randomly infect other hosts
- Day 20 - 27: Carry out a packet-flooding denial of service attack on the IP address of www1.whitehouse.gov
- Day 28 - <end of month>: Sleep

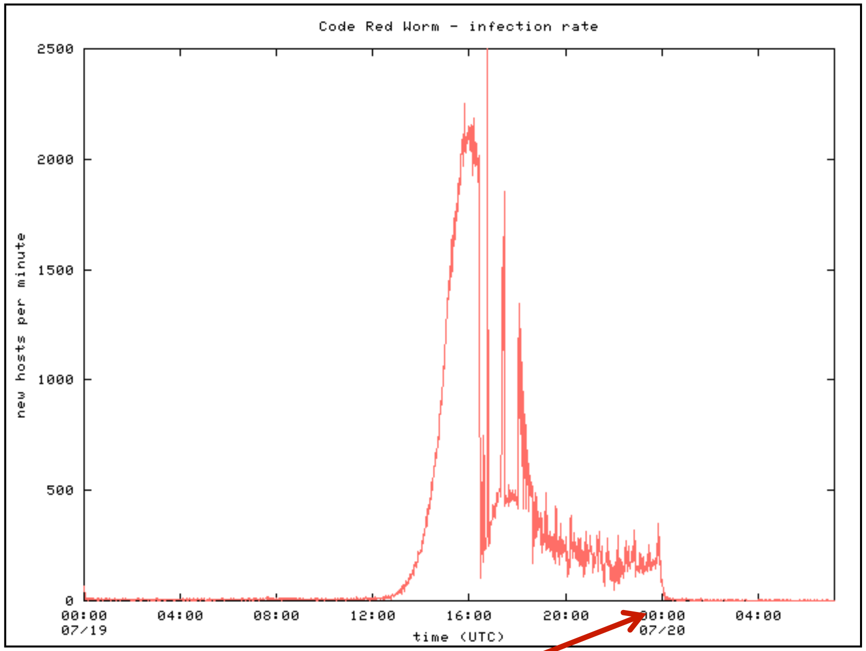
CERT® Advisory CA-2001-19 "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL
http://www.cert.org/advisories/CA-2001-19.html

Code Red spread at an alarming rate

http://www.caida.org/research/security/code-red/coderedv2_analysis.xml



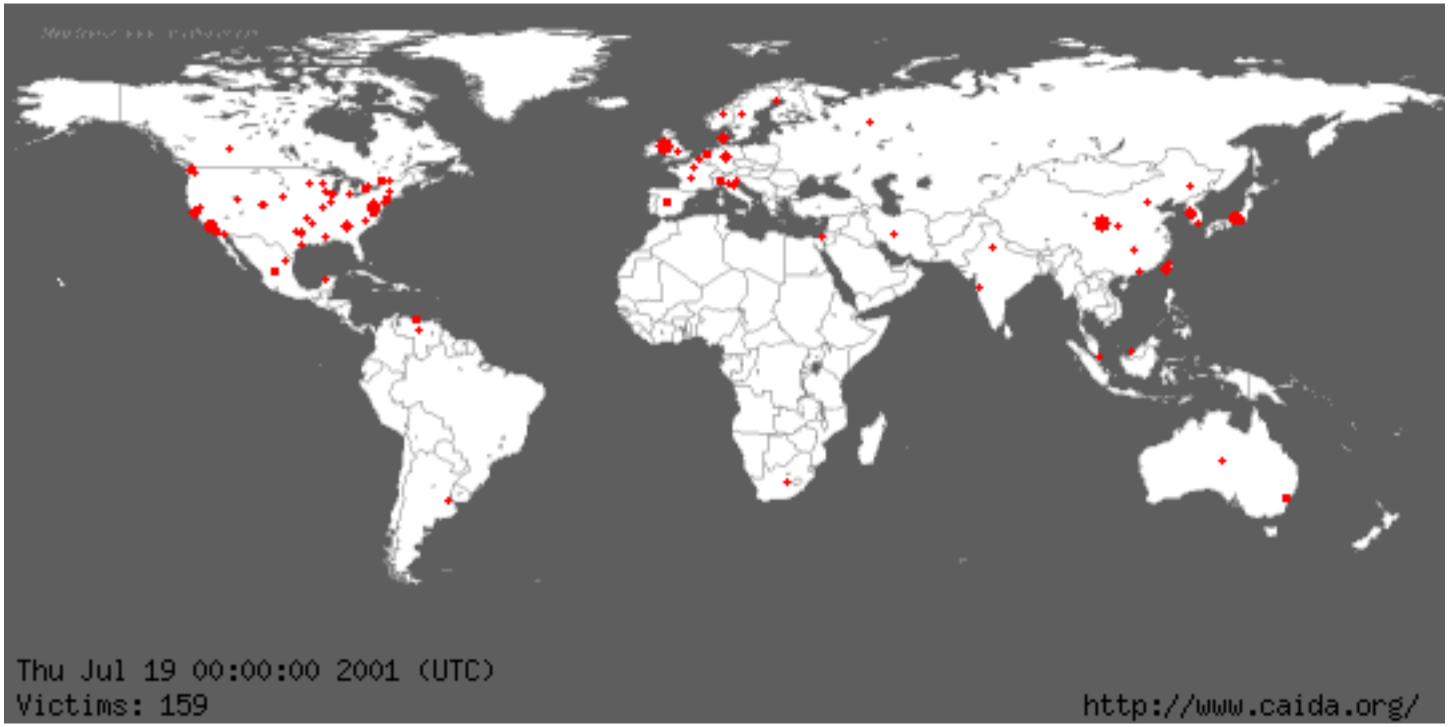
Exponential growth curve!



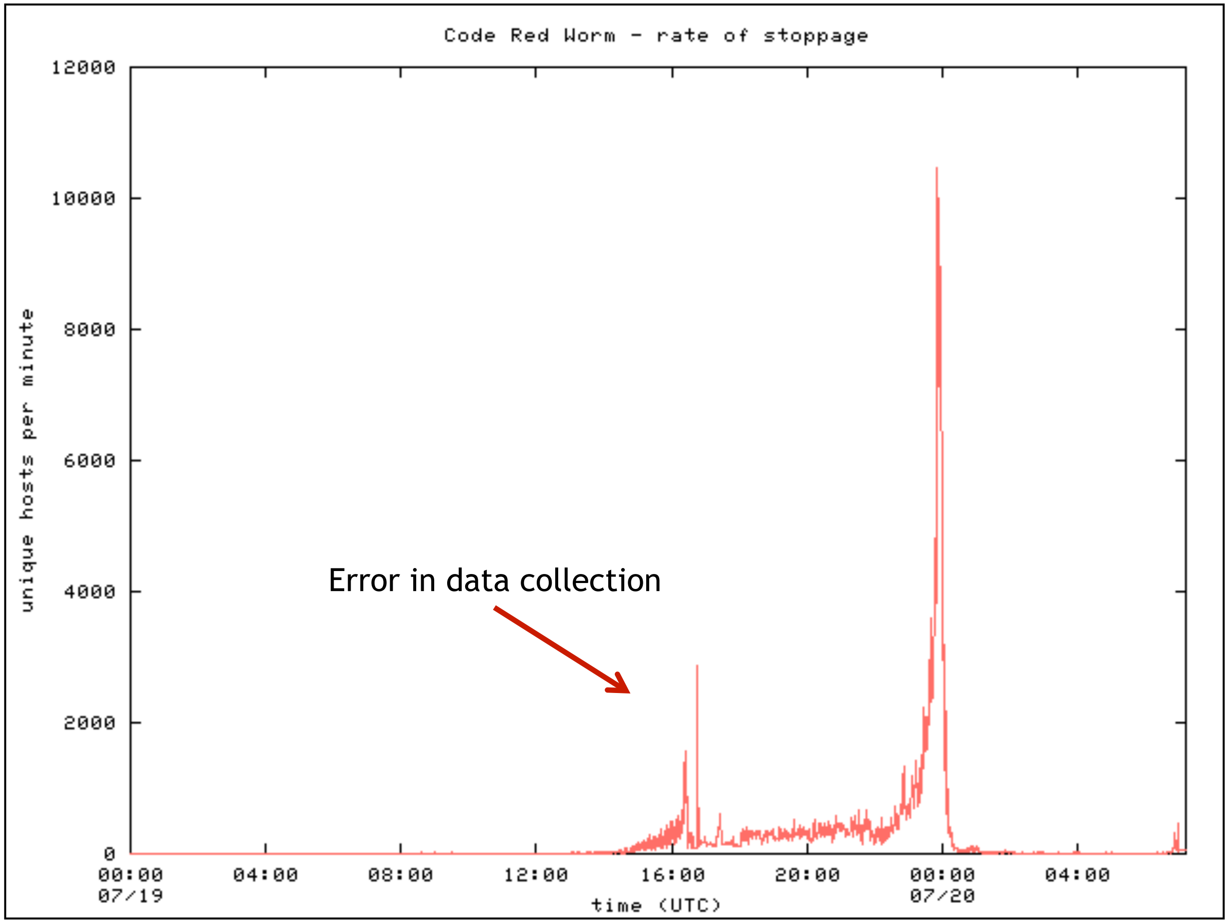
Only 13 hours were needed to infect over 359,000 hosts!

Recall: Code Red stopped infecting on the 20th of each month

Question: Why was the spread rate of Code Red so much higher than that of the Morris worm?


Thu Jul 19 00:00:00 2001 (UTC)
Victims: 159
http://www.caida.org/

Throughout the day, many hosts were patched or firewalled to help prevent the spread of Code Red



Error in data collection

We got lucky here…

Characterizing the Attack

Top 10 Countries

Country	*Hosts Infected*	*Percentage*
United States	157694	43.91
Korea	37948	10.57
China	18141	5.05
Taiwan	15124	4.21
Canada	12469	3.47
United Kingdom	11918	3.32
Germany	11762	3.28
Australia	8587	2.39
Japan	8282	2.31
Netherlands	7771	2.16

Characterizing the Attack

Top 10 TLDs

TLD	*Hosts Infected*	*Percentage*
Unknown	169584	47.22
net	67486	18.79
com	51740	14.41
edu	8495	2.37
tw	7150	1.99
jp	4770	1.33
ca	4003	1.11
it	3076	0.86
fr	2677	0.75
nl	2633	0.73

No reverse lookup entry

Consistent with the overall representation of these TLDs on the Internet

Characterizing the Attack

Top 10 Domains

Domain	***Hosts Infected***	***Percentage***
Unknown	169584	47.22
home.com	10610	2.95
rr.com	5862	1.63
t-dialin.net	5514	1.54
pacbell.net	3937	1.10
uu.net	3653	1.02
aol.com	3595	1.00
hinet.net	3491	0.97
net.tw	3401	0.95
edu.tw	2942	0.82

Note: Home and small business ISPs played a huge role in the spread of Code Red!

Outcomes of Code Red

Main point: We got lucky ☺

- Code Red was a fairly benign worm
- The automatic cut-off date eased the disinfection process
- The worm relied on a flawed DDoS attack strategy

Code Red taught us a number of important lessons

- Home users play a big role in worm propagation
- Homogeneity makes the Internet susceptible to widespread attack
- Even a worm that randomly guesses IP addresses can spread at an alarming rate
- A "release and patch" mentality is detrimental

Question: If a random scanning worm can infect over 359,000 hosts in 13 hours, what could a more directed worm do?

Researchers have predicted that the spread of flash worms could happen too fast to stop

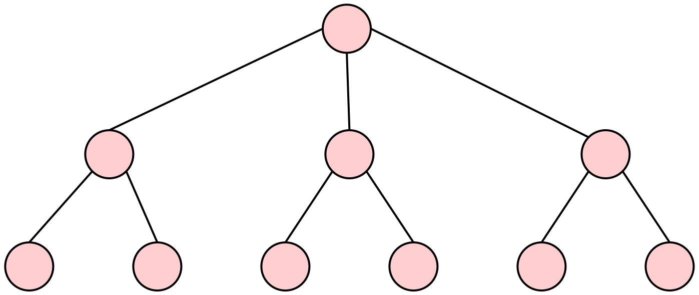
Random probing slows worms down

- Attempts to attack non-existent hosts
- Infecting hosts with minimal ability to infect others
- Hosts can be infected multiple times

Flash worms seek to spread as quickly as possible!

Phase 1: Find Vulnerable Hosts

Phase 2: Build an optimized attack tree



Phase 3: Infect!

So, how fast could a flash worm spread in the wild?

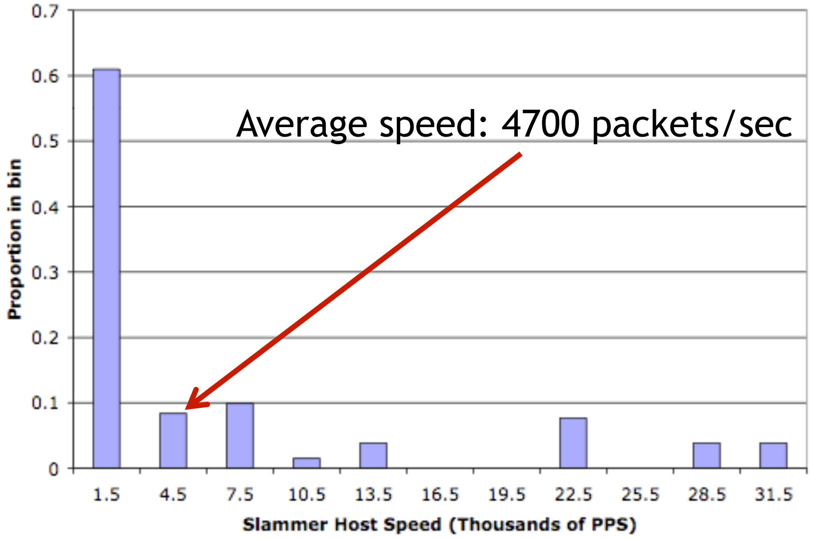
Stuart Staniford, David Moore, Vern Paxson, and Nicholas Weaver, "The Top Speed of Flash Worms," Proceedings of the ACM Workshop on Rapid Malcode (WORM), Oct. 2004.

Predicting a worm means that we first need to characterize it

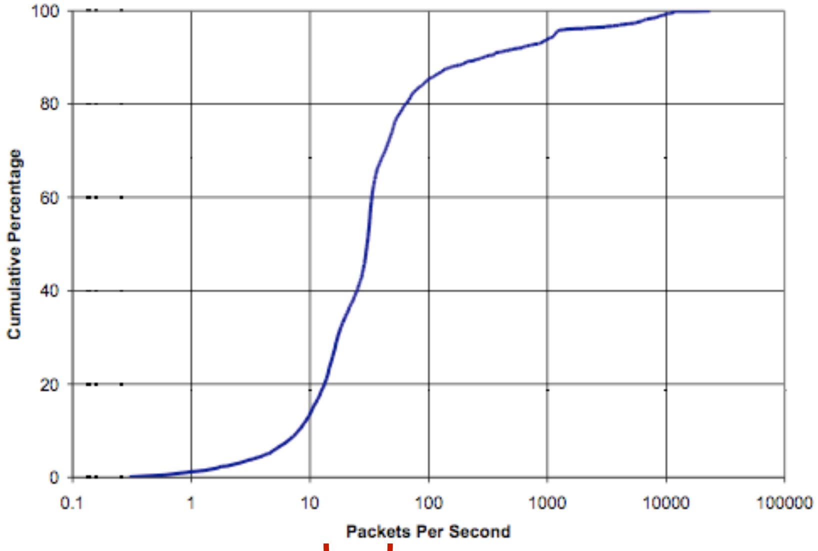
How small could a flash worm be?

- The Slammer worm used a single 404-byte UDP packet!

How fast could a flash worm propagate?



Average speed: 4700 packets/sec



60% of nodes infected by Witty sent between 11 and 60 1090-byte packets/sec. This would be between 29.67 and 161.88 Slammer-sized packets/sec.

Flash worms use an optimized distribution tree

Average Internet latency distribution is 103ms.

- Sending Slammer sized packets at 2700pps, 227 packets can be sent before the first infection
- This motivates a wide and shallow infection tree

Time to infection can be modeled as follows:

Number of hosts to infect

Size of worm packet

Number of addresses sent to each node

Latency

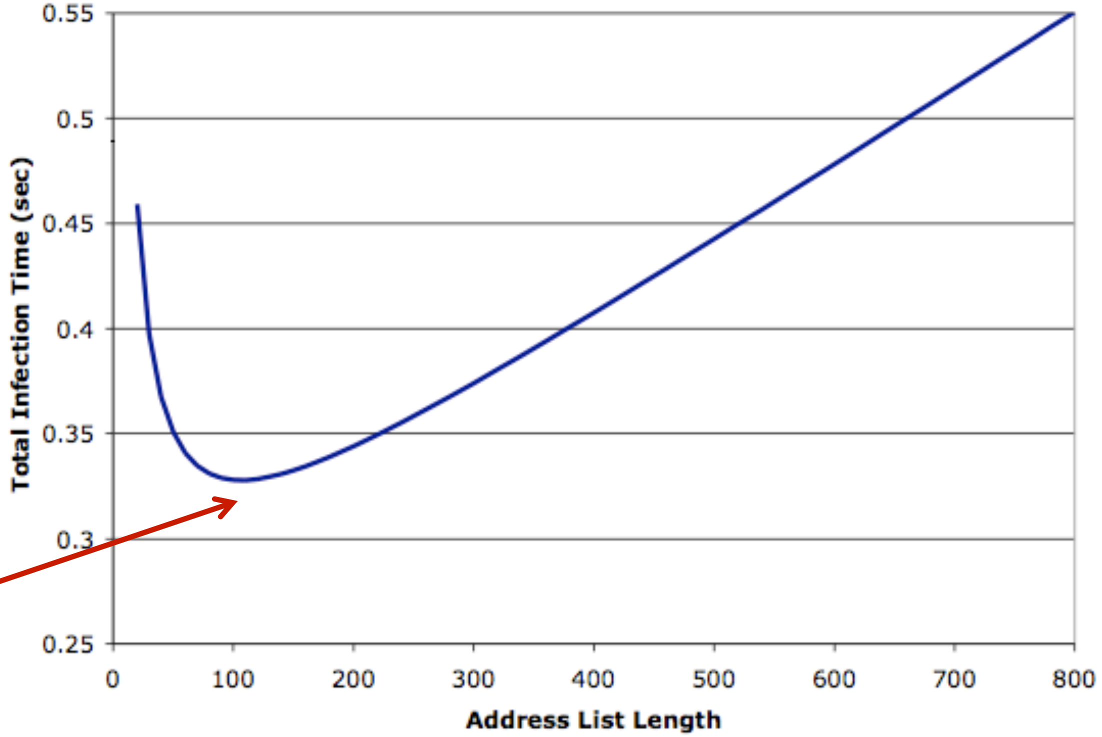
$$t_I = \underbrace{\frac{N(W+4A)}{(A+1)B}}_{\text{Time to infect first level in tree}} + \underbrace{\frac{AW}{b}}_{\text{Time to infect second level in tree}} + \underbrace{2L}_{\text{Parallelized latency}}$$

Bandwidth

What is the optimal number of addresses to send to each 2nd level host?

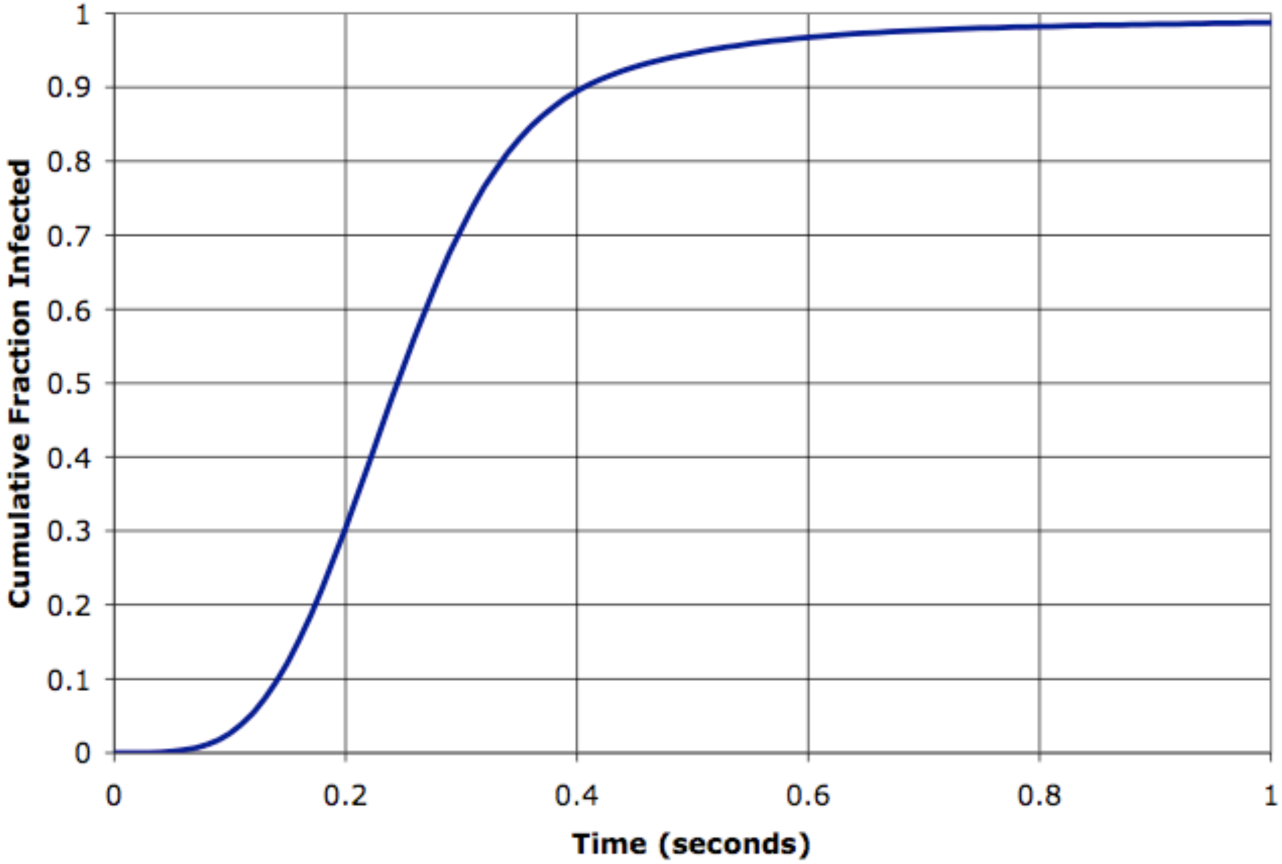
Assumptions:

- N = 1,000,000 hosts to infect
- W = 404 bytes
- Initial link can send 240,000 Slammer-sized packets/sec
  - 75% of a 1 Gbps link
  - B = .75 Gbps
- L = 103 ms
- b = 1 Mbps

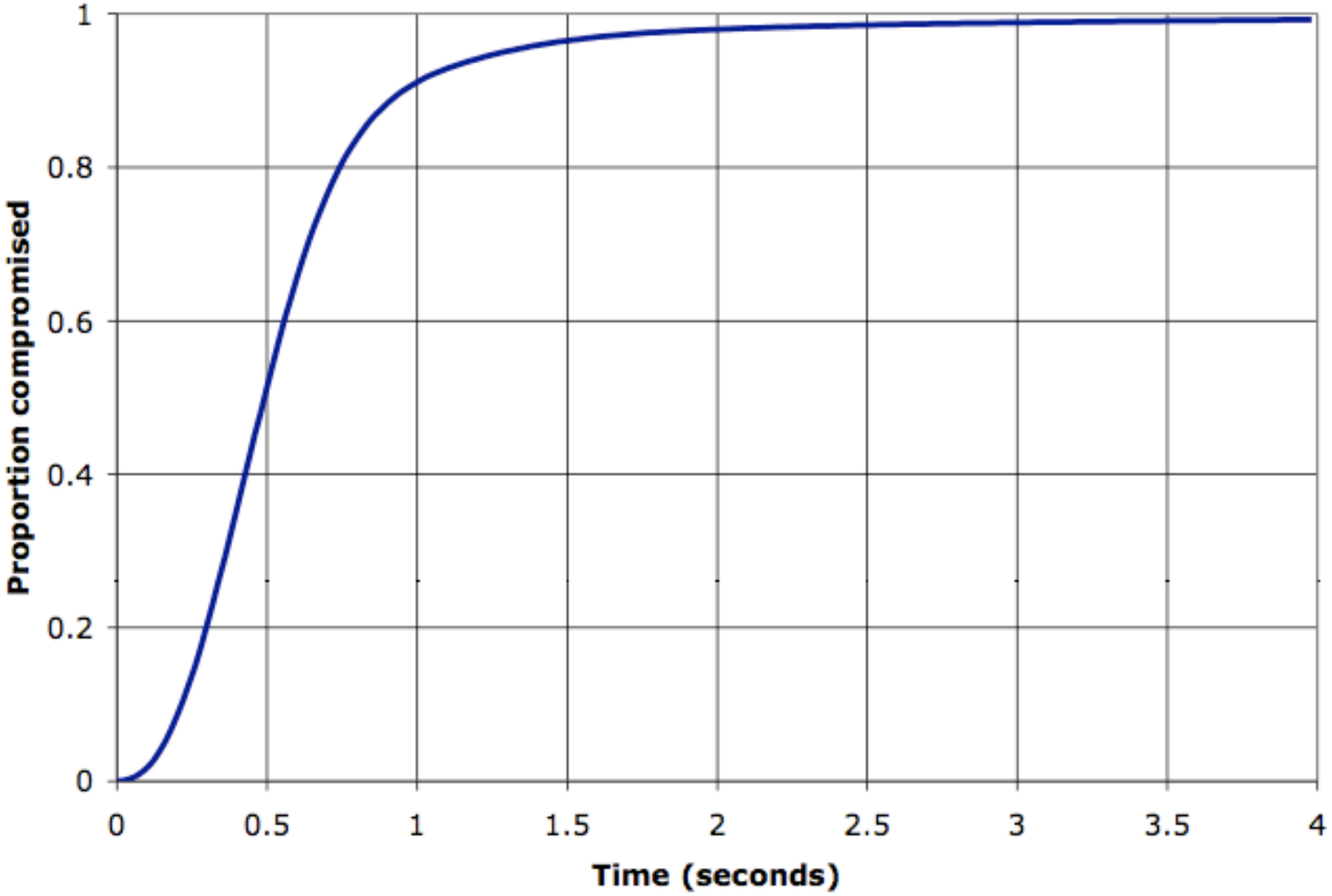


Optimum: 107 addresses to each 2nd level host

A UDP flash worm could infect 95% of 1 million susceptible hosts in 510ms!



A TCP flash worm could infect 95% of 1 million susceptible hosts in 3.3 seconds!



Conclusions

Malicious code has been around for a very long time

In the early days, computer viruses and Trojan horses moved at the speed of human-to-human interactions

Worms spread much faster by leveraging constant node connectivity

- Over the years, the propagation techniques used by worms have not changed
- More aggressive propagation mechanisms allow newer worms to spread faster

Flash worms are quite scary, but sensitive to minor problems in the network

- Excellent "worse case" for analyzing worm defenses