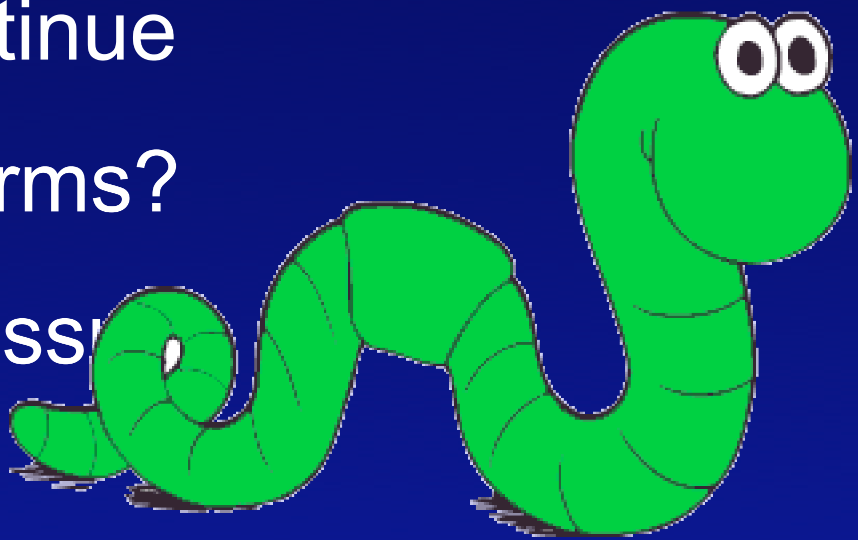
Trends in Viruses and Worms

Tom Chen
SMU
tchen@engr.smu.edu

Outline

- Viral Statistics
- What are Viruses/Worms
- Past Trends: 4 Waves
- Why Attacks Continue
- Future Super Worms?
- Some Research Iss



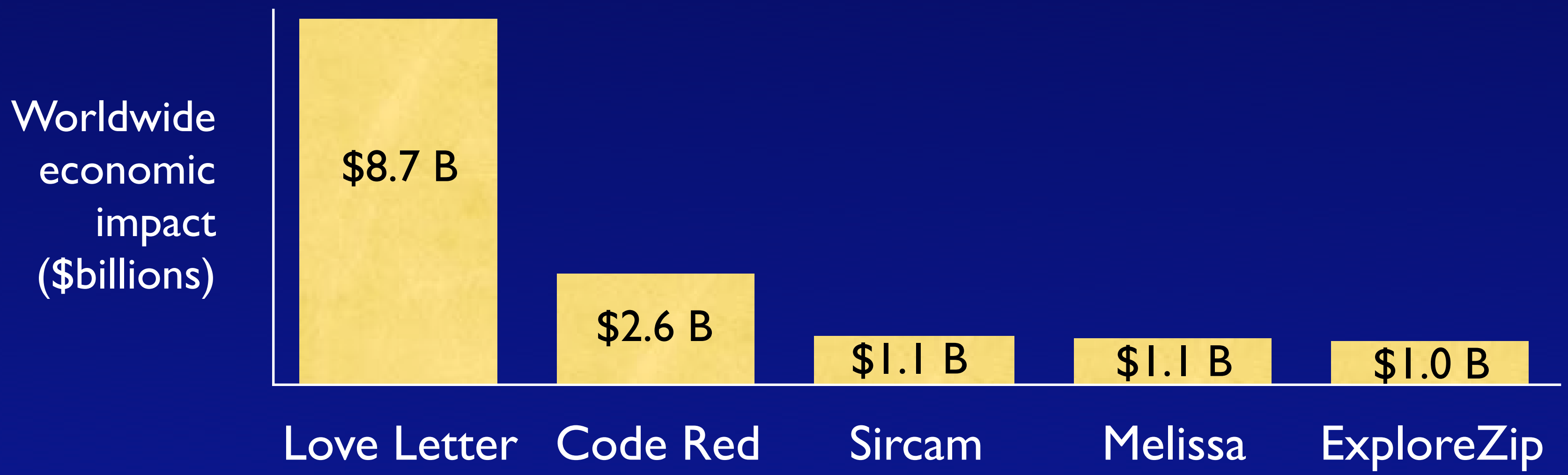
Virus/Worm Highlights

- 1979 John Shoch and Jon Hupp at Xerox
- 1983 Fred Cohen
- 1988 Robert Morris Jr
- 1992 Virus creation toolkits, Mutation Engine
- 1995 Concept macro virus
- 1999 Melissa (March), ExploreZip (June)
- 2000 Love Letter (May)
- 2001 Sircam (July), Code Red I+II (July-Aug.), Nimda (Sep.)
- 2003 Slammer (Jan.), Blaster (Aug.), Sobig.F (Aug.)

24+ years ...

Top Viruses/Worms

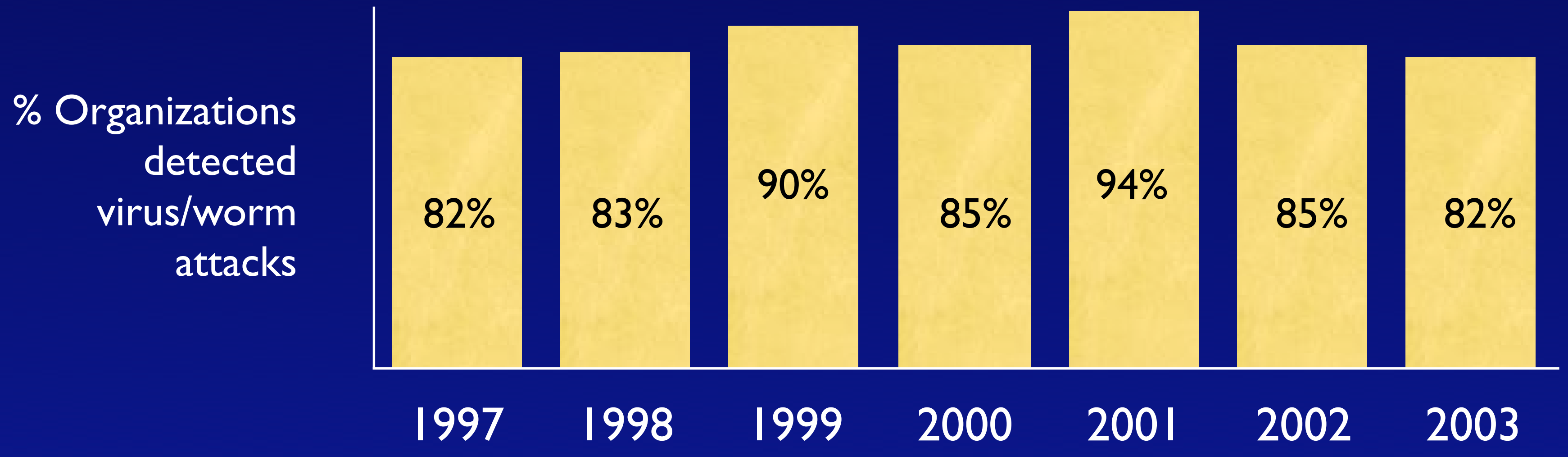
- 70,000+ viruses are known, but only hundreds “in the wild” and only a few spread well enough for major damage



*estimated by Computer Economics 2001

Prevalence

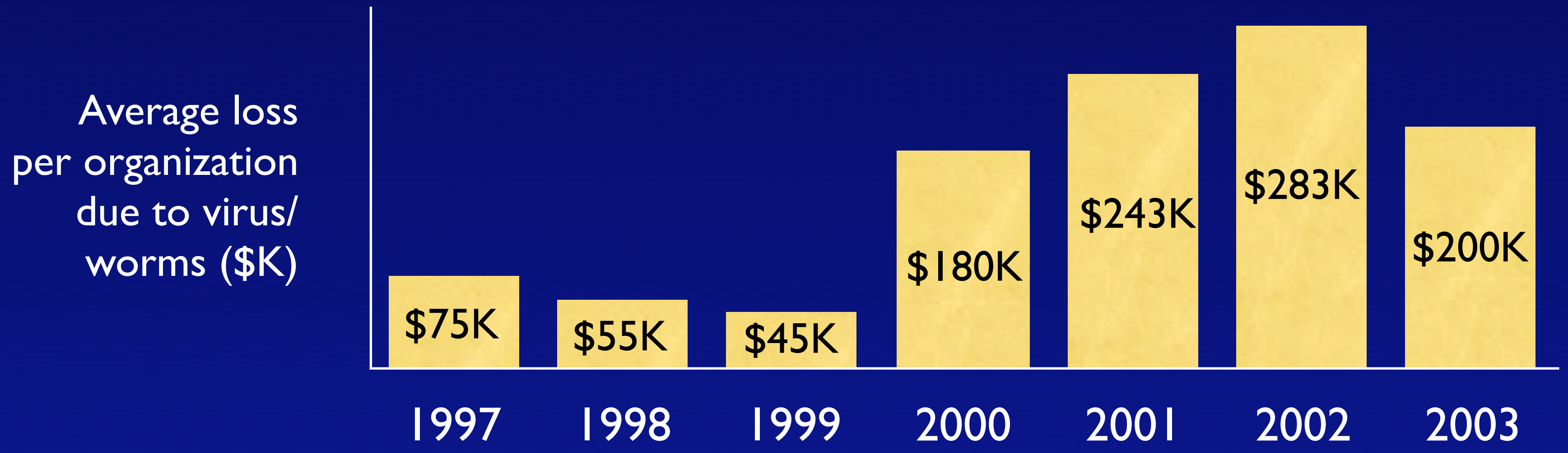
- Viruses/worms are consistently among most common attacks



*2003 CSI/FBI Computer Crime and Security Survey

Damages

- 3rd most costly security attack (after theft of proprietary info and DoS)



*2003 CSI/FBI Computer Crime and Security Survey

What are Viruses

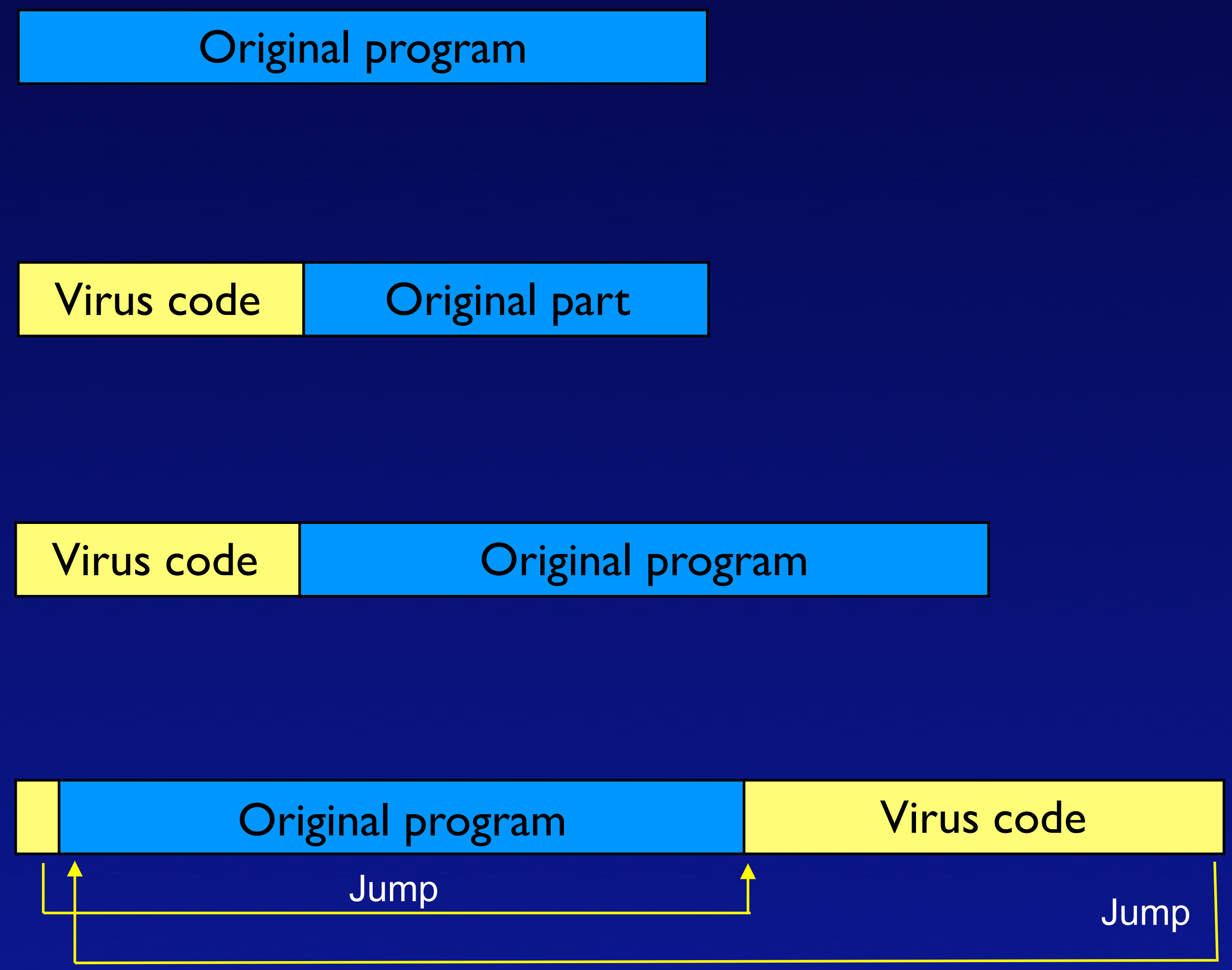
- Key characteristic: ability to self-replicate by modifying (infecting) a normal program/file with a copy of itself
  - Execution of the host program/file results in execution of the virus (and replication)
  - Usually needs human action to execute infected program

Virus Examples

Overwriting viruses

Prepending viruses

Appending viruses



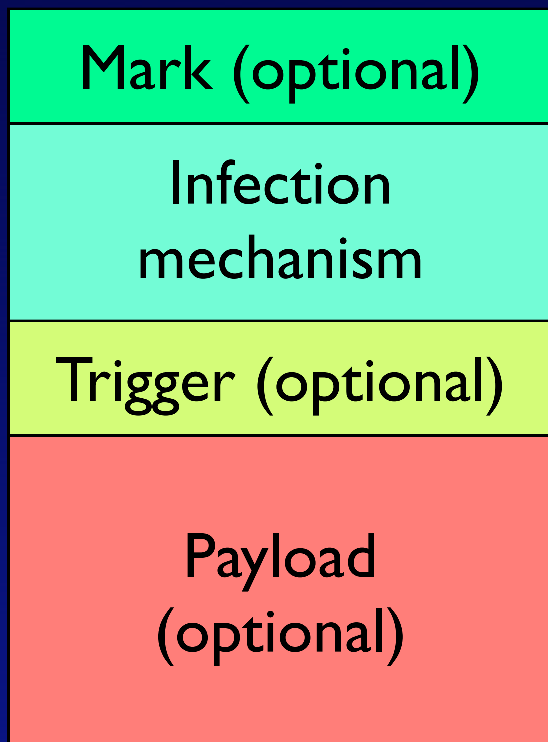
Virus Anatomy

Mark (optional)	Prevents re-infection attempts
Infection mechanism	Causes spread to other files
Trigger (optional)	Conditions for delivering payload
Payload (optional)	Possible damage to infected computer (virtually anything)

Worms

- Worm is a stand-alone program that exploits security holes to compromise other computers and spread copies of itself through the network
  - Unlike viruses, worms do not need to parasitically attach to other programs
  - Undetectable by file scanning
  - Spread by themselves without any human action

Worm Anatomy



- Structurally similar to viruses, except a stand-alone program instead of program fragment

- Infection mechanism searches for weakly protected computers through a network (ie, worms are network-based)

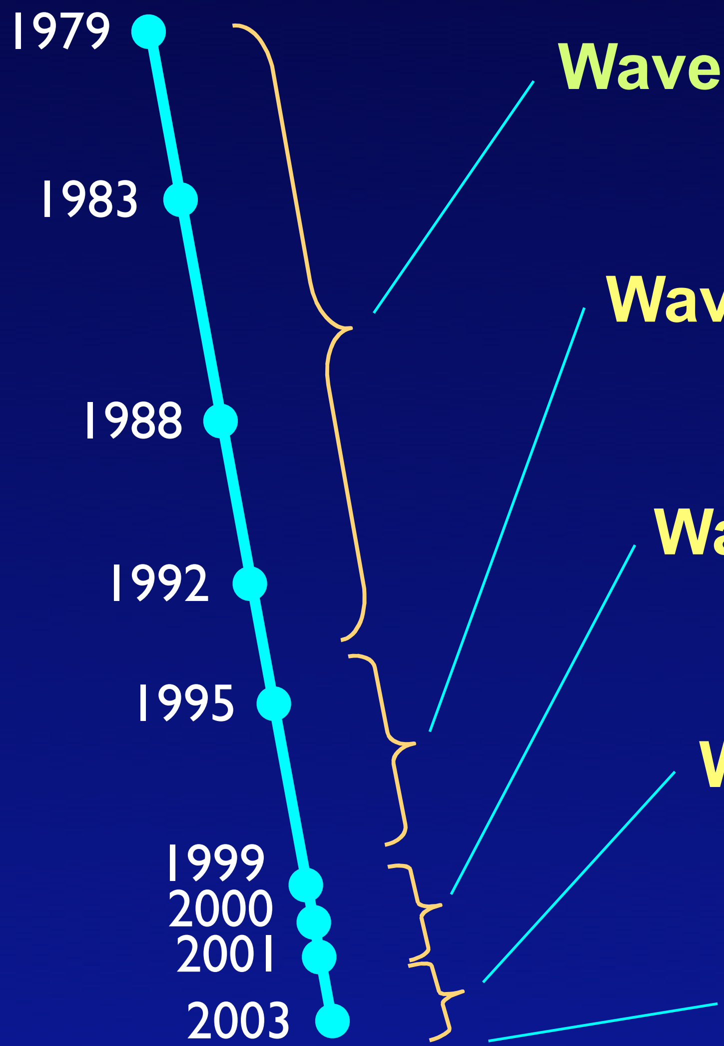
- Payload might drop a Trojan horse or parasitically infect files, so worms can have Trojan horse or virus characteristics (so-called hybrids)

My Computer... Worms?

- New vulnerabilities are continually published in Microsoft security bulletings, CERT advisories, Bugtraq, NIPC CyberNotes, MITRE CVEs,...
- SANS/FBI's Top 10 Microsoft Windows vulnerabilities (May 2003):

1	IIS server: buffer overflows, failure to handle unexpected requests
2	Remote Data Services component allows remote users to run commands with adminstrative privileges
3	SQL server: buffer overflows and various other vulnerabilities
4	Misconfiguration of network shares allows remote users full control of a host
5	Null Session connections (aka anonymous logon) allow anonymous remote users to fetch data or connect without authentication
6	LAN Manager passwords are weakly encrypted
7	User accounts with no passwords or weak passwords
8	Internet Explorer: various vulnerabilities
9	Improper permission settings allow remote access to Windows registry
10	Windows Scripting Host automatically executes .VBS scripts embedded in a file

Past Trends: 4 Waves



Wave 1 : *Experimental*

Wave 2 : *Cross platform, polymorphic*

Wave 3 : *Mass e-mailers*

Wave 4 : *Dangerous, fast, complex,...*

Super worms?

Wave 1

- 1979 John Shoch and Jon Hupp - Xerox worms
- 1983 Fred Cohen
- 1986 Brain virus
- 1987 Christma Exec virus
- 1988 Robert Morris worm
- 1989 Wank worm
- 1990 Stoned virus

Wave 1 - Early Ideas

- 1949 John von Neumann postulated "self-reproducing automata" - computing machines that could build copies and pass on their programming
- 1975 John Brunner's novel "The Shockwave Rider" described a global computer network and a self-replicating, network-crawling "tapeworm" program

Wave 1

- 1971 Bob Thomas (BBN) wrote “creeper” program that moved around ARPAnet and displayed message on computer screens challenging people to catch it
  - An annoyance more than serious program
  - In response, others wrote “reaper” programs to chase and delete “creeper” programs (first antivirus)

Wave 1 - First Worms

- 1979 John Shoch and Jon Hupp at Xerox PARC coined “worm” after network-based “tapeworm” monster in John Brunner’s “The Shockwave Rider”
  - Experimented with worms for overnight diagnostics on internal Ethernet LAN
  - Worms programmed with limited lifetimes and suicide response to special “kill” packet

Wave 1 - First Viruses

- One worm mysteriously got out of control and crashed several computers
- 1983 Fred Cohen (PhD student at USC) conceived, wrote and demonstrated first documented virus
- 1986 Brain virus written by 2 Pakistani programmers used stealth to try to fool DOS utilities looking for its presence

Wave 1 - Christmas Tree

- 1987 Christma Exec virus spread by email, promising to display a Christmas tree graphic, but secretly emailed copies of itself to user’s list of outgoing mail addresses, using user’s name (to entice recipients to open the attachment)
  - Early example of social engineering attack

Wave 1 - Internet Worm

- Nov. 2, 1988 Robert Morris Jr (Cornell student) released worm that disabled 6,000 computers - 10% of Internet at the time
  - Programming bug caused worm to re-infect already infected computers, until they crashed
  - Brought worms/viruses to public awareness

Wave 1 - Internet Worm

- First to use combination of attacks to spread
  - Buffer overflow of Unix “finger” daemon: caused victim computers to run a shell code
  - Debug mode of “sendmail” program: caused victims to run set of commands to copy the worm
  - Cracked password files: guessed common words from a dictionary

Wave 1 (cont)

- 1989 WANK (worms against nuclear killers) worm spread through DECnet by guessing default accounts and passwords (often not changed), spreading anti-war propaganda
- Stoned, Jerusalem, other viruses - mostly targeted to DOS

Wave 1 Trends

- Most viruses are limited to DOS and spread slowly by diskettes
- Experiments with worms (Xerox, Morris) hard to control
- Beginnings of stealth viruses and social engineering attacks

Wave 2

- 1992 Polymorphic generators (MtE, SMEG, NED), virus construction toolkits (VCL, PS-MPC)
- 1994 Pathogen, Queeg polymorphic viruses
- 1995 Concept macro virus
- 1996 Boza, Tentacle, Punch viruses for Windows
- 1997 Bliss virus for Linux
- 1998 CIH virus, HLLP.DeTroie virus

Wave 2

- Encryption attempts to hide a recognizable signature (code pattern) from file-scanning antivirus software by scrambling virus body
  - But decryption routine (prepended and unencrypted) is constant (detectable)
- Polymorphism continually permutes appearance - no more than few bytes common between generations

Wave 2 - Polymorphism

- 1992 Dark Avenger's user-friendly Mutation Engine (MtE) let anyone add polymorphism to any virus
  - Followed by others: TPE, NED, DAME
  - Created high risk of false alarms for antivirus
- 1994 Pathogen and Queeg: complicated viruses created by Black Baron's SMEG

Wave 2 - Toolkits

- 1992 Virus Creation Lab: user-friendly virus construction toolkit allowed “script kiddies” to generate hundreds of viruses with little programming skill
  - Followed by PS-MPC and other toolkits
  - Antivirus companies flooded with thousands of (lame) viruses
  - Best known example: 2001 Anna Kournikova VBScript email virus

Wave 2 - Win32 Viruses

- 1995 Concept macro virus for Word for Windows95
  - Macro viruses: easy to write and cross-platform (mostly targeted to MS Office)
- 1996 Boza, Tentacle, Punch, other viruses target Windows95
- 1997 Bliss: first virus for Linux

Wave 2 (cont)

- 1998 CIH (Chernobyl) destructively overwrites PC hard disks with random data and overwrites flash ROM BIOS firmware - PCs cannot boot up
- 1998 HLLP.DeTroie virus: first to transmit private data from infected PCs to virus creator

Wave 2 Trends

- Most viruses target Windows
- Macro viruses go cross-platform
- Large-scale autogeneration of viruses and easy polymorphism

Wave 3

1999 Happy99 worm

Melissa macro virus

PrettyPark, ExploreZip worms

2000 BubbleBoy virus, KAK worm

Love Letter worm

Hybris worm

2001 Anna Kournikova worm

Wave 3 - Mass E-mailers

- Jan 1999 Happy99 worm spread as e-mail attachment “happy99.exe”, displayed fireworks on screen for New Years Day 1999
  - Secretly modifies WSOCK32.DLL to e-mail second message (with worm) after every message sent

Wave 3 - Melissa

- March 1999 Melissa macro virus set new record, infecting 100,000 computers in 3 days
  - Launched MS Outlook and mailed itself to 50 addresses in address book
  - Infected Word normal.dot template

Wave 3 - PrettyPark

- Mid 1999 PrettyPark worm spread as e-mail with an attachment “PrettyPark.exe” showing icon of South Park character
  - Installs itself into system folder and modifies registry to ensure it runs
  - Emails itself to addresses in Windows address book
  - Sends password data to certain IRC servers

Wave 3 - ExploreZip

- June 1999 ExploreZip worm appeared to be WinZip file attached to e-mail
  - If executed, it displayed an error message but secretly installs itself into System directory
  - E-mails itself via Outlook or Exchange to recipients in unread inbox messages, and replies to all incoming messages with a copy of itself

Wave 3 - KAK Worm

- Jan 2000 KAK worm was an embedded VBScript in HTML e-mail message with no visible text
  - Previewing or opening message in Outlook executes the script
  - Worm copies itself into Windows start-up folder, and attaches itself as a signature in all outgoing e-mail

Wave 3 - Love Letter

- May 2000 Love Letter worm demonstrated social engineering attack, pretending to be e-mail love letter
  - Attachment appears to be text but is VBScript that infects Windows and System directories and various file types
  - E-mails itself via Outlook to everyone in address book, infects shared directories, tries to spread by IRC channels

Wave 3 - Dynamic Plug-ins

- Oct 2000 Hybris worm spread by e-mail, modifying WSOCK32.DLL file to send itself as a second message to same recipient after every normal message sent
- Connected to a newsgroup to download encrypted plug-ins (code updates)
  - Potentially very dangerous - worm can get new instructions or payload at any time

Wave 3 Trends

- Mass e-mailing becomes most popular infection vector
  - Attacks increase in speed and scope
- Social engineering becomes common
- Worms begin to become dangerous (data theft, dynamic plug-ins)

Wave 4

2001 Ramen, Davinia worms

Lion, Gnutelman worms

Sadmind worm

Sircam, Code Red I, Code Red II worms

Nimda worm

Badtrans, Klez, Bugbear worms

2002

Gibe worm

Slapper worm

Winevar worm

2003 Lirva, Sapphire/Slammer worms

Fizzer worm

Blaster, Welchia/Nachi, Sobig.F worms

Wave 4 - Linux Worms

- Linux is targeted by Ramen worm (Jan 2001) and Lion worm (March 2001)
- Lion is dangerous
  - Steals password data, installs rootkit “t0rn” (hides presence of worm from “syslogd” and other system utilities), installs distributed DoS agent “TFN2K”, installs backdoor root shells, listens on certain ports for remote instructions

Wave 4 - More Vectors

- Feb 2001 Gnutelman/Mandragore worm infects users of Gnutella peer-to-peer networks
  - Disguises itself as a searched file
- Blended attacks:
  - May 2001 Sadmind worm targets Sun machines and Microsoft IIS web servers
  - July 2001 Sircam spreads by e-mail and network shares

Wave 4 - A Modern Worm

- July 12, 2001 Code Red I version 1 worm targeted buffer overflow vulnerability in Microsoft IIS servers
  - Tried to install DoS agent targeted to "www.whitehouse.gov"
  - Programming bug caused worm to probe same set of IP addresses instead of generate random addresses, so spread was slow

Wave 4 - Code Red

- Week later, Code Red I version 2 fixed programming bug and spread much faster
  - Infected 359,000 computers in 14 hours (peak rate of 2,000 computers per minute)
- Aug 4, Code Red II used same exploit, spawning 300 parallel threads on each machine to probe for new victims
  - Probing caused DoS-like congestion

Wave 4 - New Sophistication

- Sept 2001 Nimda worm used blended attack via 5 vectors:
  - E-mailed itself using its own SMTP engine
  - Infected MS IIS web servers via buffer overflow exploit
  - Infected network shares
  - Added Javascript to web pages, infected any web browser
  - Backdoors left by Code Red and Sadmind

Wave 4 (cont)

- Nimda infected 450,000 computers in 12 hours
  - Spreading rate caused DoS-like congestion
  - Extensively modified registry and System directory to conceal its presence and make hard to remove
  - Created backdoor administrative account for remote control

Wave 4 - Armored Worms

- “Armored” worms attack and disable antivirus programs
  - Klez (Oct 2001), Bugbear (Oct 2001), Winevar (Nov 2002), Avril (Jan 2003) look for common antivirus processes and stop them, scan hard drive for key antivirus files and delete them
  - Winevar also masquerades as a Trojan version of an antivirus program

Wave 4 - Dangerous

- Worms become more dangerous
  - Gibe worm (March 2002) pretends to be e-mailed Microsoft security bulletin and patch, but secretly installs backdoor
  - Badtrans (Nov 2001), Bugbear, Lirva, Fizzer (May 2003) worms install keystroke logging Trojan horses
  - Lirva e-mails password data to virus writer, and downloads Back Orifice to infected PCs (gives complete remote control)

Wave 4 - Proof-of-Concepts

- Jan 2003 Sapphire/Slammer worm demonstrated that simple worm (in only one 404-byte UDP packet) could spread very fast
  - Targeted Microsoft SQL servers, hit 90 percent of vulnerable hosts within 10 minutes (120,000 machines)
  - At peak rate, infection doubled every 8.5 seconds - reached peak rate of 55,000,000 scans/sec after only 3 minutes

Wave 4 - Proof-of-Concepts

- Aug 12, 2003 Blaster targeted DCOM RPC vulnerability on Win2000 and WinXP - demonstrated majority of PCs are vulnerable
  - Infected 400,000 computers but not nearly the maximum potential spreading rate due to novice programming
  - Carried DoS agent targeted at "www.windowsupdate.com"

Wave 4 - Proof-of-Concepts

- Aug 19, 2003 Sobig.F was 6th variant of Sobig, spread by e-mail among Windows PCs
  - At peak rate, Sobig.F was 1 out of every 17 e-mail messages
  - Produced 1 million copies within 24 hours
  - Preprogrammed stopping date and empty payload suggests intention as proof-of-concept

Wave 4 Trends

- New infection vectors (Linux, P2P, IRC, IM,...)
- Blended attacks (combined vectors)
- Dynamic code updates (via IRC, web,...)
- Dangerous payloads
- Active attacks on antivirus software
- Fast and furious spreading

Wave 4 Trends

- Shorter time between discovery of vulnerability and a worm exploiting it
- Series of variants of a worm appear quickly
  - Most likely different authors - coordinated efforts?

Why Attacks Continue

- Worm outbreaks continue regularly despite antivirus software, firewalls, intrusion detection systems, e-mail filters
- Sometimes portrayed as escalating conflict between virus writers (innovating) and antivirus developers (catching up), but problem is larger involving entire computer industry

Why Attacks Continue

- Attacks will continue as long as computers have vulnerabilities that can be exploited
  - Software is written in unsecure manner, eg, vulnerable to buffer overflows
  - When vulnerabilities are announced, many people do not apply patches (too inconvenient, too frequent, sometimes unstable)

Why Attacks Continue

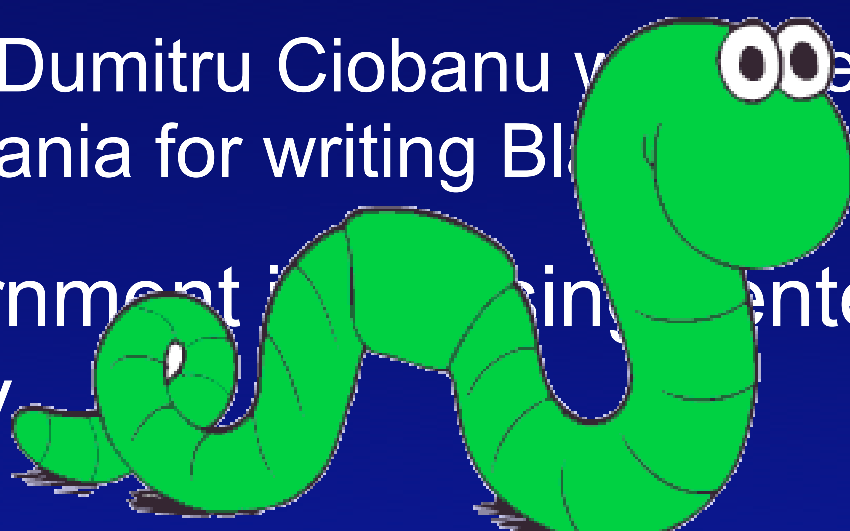
- Who is held accountable?
  - Software vendors have acknowledged their responsibility to produce secure software but have avoided accountability (financial liability)
    - New lawsuits charge MS monolithic IT culture creates weakness
  - Virus writers are the criminals, but hard to identify and prosecute

Why Attacks Continue

- Viruses/worms are hard to trace to creator from analysis of code, unless there are accidental clues left
  - Most skilled virus writers are too good to get caught
  - Prosecuted get light sentences: Robert Morris - 3 years probation, $10,000 fine; Onel de Guzman for LoveLetter - released due to lack of laws in Philippines; Jan De Wit for Anna Kournikova - 150 hours community service

Why Attacks Continue

- Government cracking down on virus writers to set an example?
  - Teenager Jeffrey Lee Parson was just arrested for writing Blaster.B variant
  - Dan Dumitru Ciobanu w… ested in Romania for writing Bl…
- Governm… i… sin… entence limits in Nov



Future Super Worms?

- General epidemic model predicts the rate of spreading as

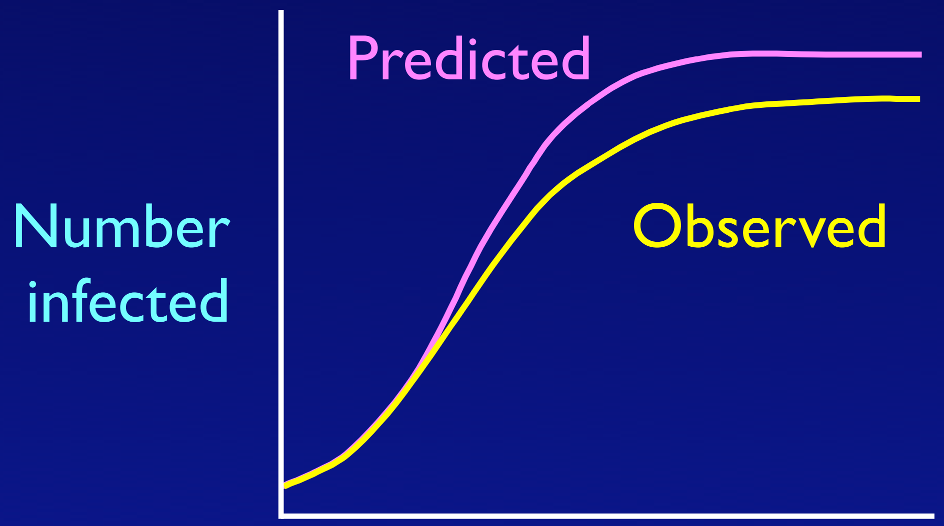
$$\frac{d}{dt} S = - bSI$$

$$\frac{d}{dt} I = bSI$$

b = infection rate parameter
S = number susceptibles
I = number infected

Super Worms (cont)

- But observed worm outbreaks tend to slow down more quickly than predicted



Super Worms (cont)

- Epidemics naturally slow down when many become infected (then infectives tend to contact already infected)
- Worm outbreaks slow down for same reason
- Second factor is network congestion caused by heavy random probing
  - Worms effectively work against themselves

Super Worms (cont)

- Super worms (aka Warhol worms, flash worms, pulse worms) seek to saturate vulnerable population within few seconds or few minutes, not hours
- Possible if probing to new victims is efficient and coordinated, then spreading rate may be sustained
  - Network does not become congested with high volume of inefficient probes

Super Worms (cont)

- Coordinated probing is theoretically possible in several ways (not seen yet)
  - A hitlist of vulnerable hosts is pre-scanned and programmed into worm initially
  - Address range is continually divided whenever worm copies itself (each worm covers a separate address subrange)
  - Worms are coordinated centrally (eg, via IRC channel) - can also download updates

Some Research Issues

- Fast worms must be contained ("quarantined") automatically, cannot depend on manual methods (eg, patching)
- Network infrastructure must be equipped to
  - Automatically detect worm outbreaks
  - React to quarantine new worms

Research (cont)

- Worm detection
  - New worms will have unknown signature, but worms typically exploit known vulnerabilities
  - Vulnerability exploited by Code Red I was known for a month; Sapphire/Slammer targeted vulnerability known for 6 months
  - Although known, people did not patch PCs so worms were successful

Research (cont)

- Worm writers use known vulnerabilities because easier than discovering new security holes, and they want to ensure worms will spread
  - Hence may be possible to recognize new worm by detecting attempts to exploit known vulnerabilities (behavior recognition approach)

Research (cont)

- If new worm is detected, how to quarantine?
  - Routers may be equipped with advanced packet filtering to selectively block worm traffic
  - How many routers and where? We have been looking at epidemiology and metastasis models for answers

Research (cont)

- In epidemic theory, “herd immunity” is concept that entire population can be protected by immunization of sufficient fraction (but not all) of population
  - Applied in medicine to eliminate smallpox
  - Concept implies certain number of advanced routers at key locations may be sufficient to protect entire Internet from new epidemics
  - Epidemic models may point to key router locations

Conclusions

- Worm outbreaks continue to be commonplace, innovations continue
  - Past worms have tended to be proof-of-concepts, but future worms may be more dangerous as well as fast
  - In past, dangerous worms were slow enough to stop, future worms may be too fast

Conclusions

- Virus research has been little compared to scope/importance of problem
  - Outbreaks are so commonplace, they have become viewed as routine costs
  - But more research is [illegible]d
- Also, research has [illegible]d exclusively on “mic[illegible]ev[illegible]virus code) - no “macr[illegible]” (n[illegible]w[illegible]level) research