

# Malicious Code

## CS 239

### Computer Security

March 15, 2004

CS 239, Winter 2004

Lecture 16  
Page 1

## Outline

- Introduction
- Viruses
- Trojan horses
- Trap doors
- Logic bombs
- Worms
- Examples

CS 239, Winter 2004

Lecture 16  
Page 2

## Introduction

Clever programmers can get software to do their dirty work for them

Programs have several advantages for these purposes

- Speed
- Mutability
- Anonymity

CS 239, Winter 2004

Lecture 16  
Page 3

## Where Does Malicious Code Come From?

- Most typically, it's willingly (but unwittingly) imported into the system
  - Electronic mail (most common today)
  - Floppy disks
  - Downloaded executables
  - Sometimes shrinkwrapped software
- Sometimes it breaks in
- Sometimes an insider intentionally introduces it

CS 239, Winter 2004

Lecture 16  
Page 4

## Is Malicious Code Really a Problem?

- Considering viruses only, by 1994 there were over 1,000,000 annual infections
  - One survey shows 10-fold increase in viruses since 1996
- In November 2003, 1 email in 93 scanned by particular survey contained a virus
- 2003 FBI report shows 82% of survey respondents had malicious code incidents
  - Around 64% had financial losses from them

CS 239, Winter 2004

Lecture 16  
Page 5

## More Alarming Statistics

- In 1992, there were around 2000 unique viruses known
- Today, Symantec's databases of viruses includes 66,000+ entries
- The numbers continue to grow

CS 239, Winter 2004

Lecture 16  
Page 6

## But Don't Get too Alarmed

- Most viruses are never found “in the wild”
- Most viruses die out quickly
- The Wild List<sup>1</sup> shows 261 active viruses worldwide
  - With another 652 or so with only a single incident reported

<sup>1</sup>[www.wildlist.org](http://www.wildlist.org)

CS 239, Winter 2004

Lecture 16  
Page 7

## How Much Do Viruses Cost?

- Group called mi2g estimated that MyDoom worm cost \$38.5 billion worldwide
  - Cleanup costs, lost productivity, etc.
- Many folks believe this (and other estimates) are bogus publicity stunts
  - Methodology lacking for real estimates

CS 239, Winter 2004

Lecture 16  
Page 8

## But Do I Really Have to Worry About Viruses?

- “After all, I run Linux/Mac OS/Solaris/BSD”
- “Aren't all viruses for Windows?”
- Mostly true in practice
  - Definitely not true in theory
- Anyone, at any time, can write and release a virus that can clobber your machine, regardless of what OS you run

CS 239, Winter 2004

Lecture 16  
Page 9

## Viruses

- “Self-replicating programs containing code that explicitly copies itself and that can ‘infect’ other programs by modifying them or their environment”
- Typically attached to some other program
  - When that program runs, the virus becomes active and infects others
- Not all malicious codes are viruses

CS 239, Winter 2004

Lecture 16  
Page 10

## How Do Viruses Work?

- When a program is run, it typically has the full privileges of its running user
- Including write privileges for some other programs
- A virus can use those privileges to replace those programs with infected versions

CS 239, Winter 2004

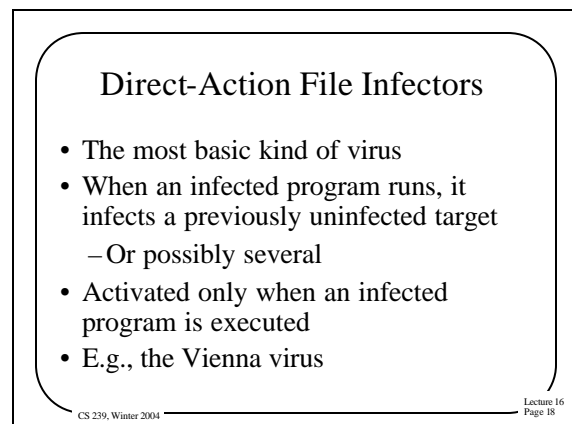
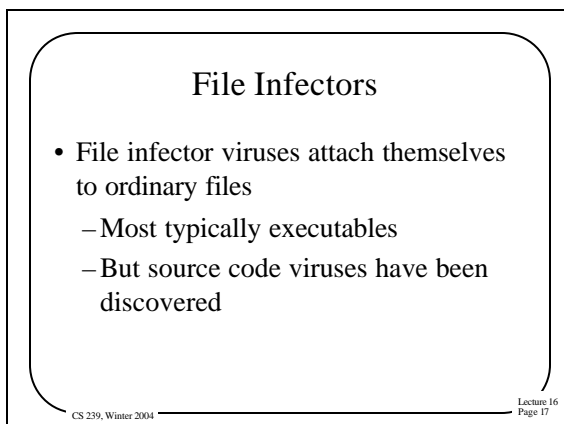
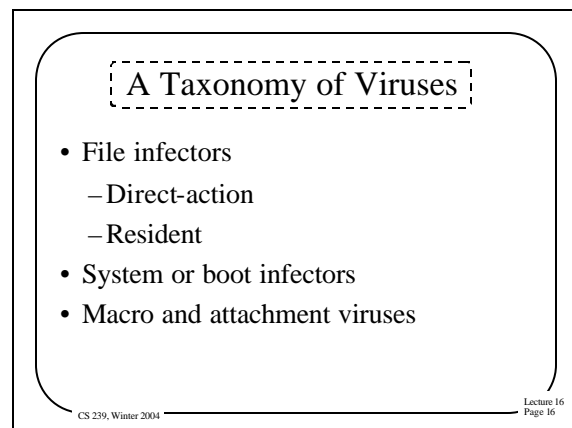
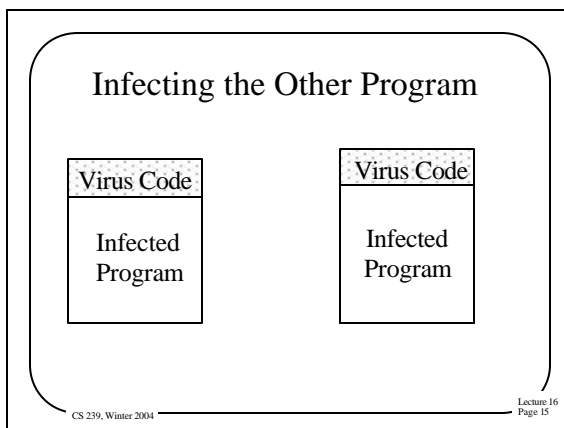
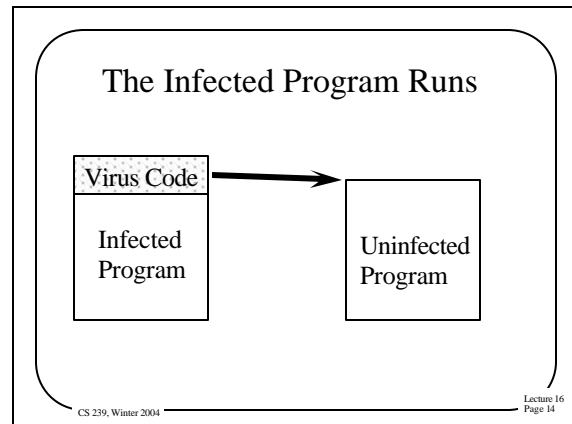
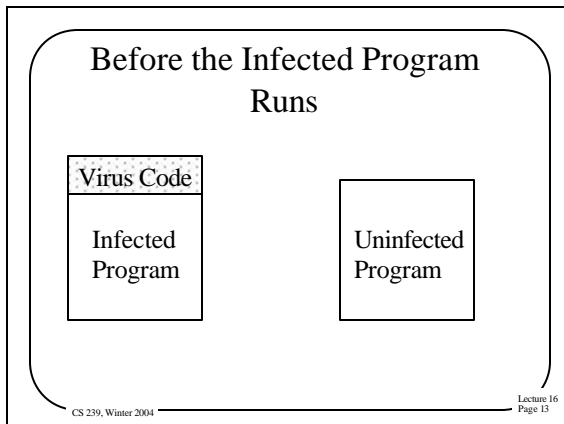
Lecture 16  
Page 11

## Typical Virus Actions

- 1). Find uninfected writable programs
- 2). Modify those programs
- 3). Perform normal actions of infected program
- 4). Do whatever other damage is desired

CS 239, Winter 2004

Lecture 16  
Page 12



## Resident File Infectors

- The first time a resident file infector is run, it leaves a daemon running
  - Or some other persistent and useful RAM entity
- Whenever any other program is run, the daemon infects it
- E.g., the Jerusalem virus

CS 239, Winter 2004

Lecture 16  
Page 19

## System Infectors

- Also known as boot sector infectors
- Rather than living in programs, they live in the boot sectors of disks
- Typically memory resident
- Typically infect any disk exposed to them
- E.g., the Michelangelo virus
- Multipartite viruses can live in both programs and boot sectors

CS 239, Winter 2004

Lecture 16  
Page 20

## Macro and Attachment Viruses

- Modern data files often contain executables
  - Macros
  - Email attachments
  - Ability to run arbitrary executables from many applications, embedded in data
- Easily the most popular form of new viruses
  - Requires less sophistication to get right
- Most widespread viruses today use attachments

CS 239, Winter 2004

Lecture 16  
Page 21

## Virus Toolkits

- Helpful hackers have written toolkits that make it easy to create viruses
- A typical smart high school student can easily create a virus given a toolkit
- Generally easy to detect viruses generated by toolkits
  - But we may see “smarter” toolkits

CS 239, Winter 2004

Lecture 16  
Page 22

## How To Find Viruses

- Basic precautions
- Looking for changes in file sizes
- Scan for signatures of viruses
- TSR monitoring
- Multi-level generic detection

CS 239, Winter 2004

Lecture 16  
Page 23

## Precautions to Avoid Viruses

- Don't import untrusted programs
  - But who can you trust?
- Viruses have been found in commercial shrink-wrap software
- The hackers who released Back Orifice were embarrassed to find a virus on their CD release
- Trusting someone means not just trusting their honesty, but also their caution

CS 239, Winter 2004

Lecture 16  
Page 24

## Other Precautionary Measures

- Scan incoming programs for viruses
  - Some viruses are designed to hide
- Limit the targets viruses can reach
- Monitor updates to executables carefully
  - Requires a broad definition of “executable”

CS 239, Winter 2004

Lecture 16  
Page 25

## Containment

- Run suspect programs in an encapsulated environment
  - Limiting their forms of access to prevent virus spread
- Requires versatile security model and strong protection guarantees

CS 239, Winter 2004

Lecture 16  
Page 26

## Viruses and File Sizes

- Typically, a virus tries to hide
- So it doesn't disable the infected program
- Instead, extra code is added
- But if it's added naively, the size of the file grows
- Virus detectors can look for this growth

CS 239, Winter 2004

Lecture 16  
Page 27

## Problems With Size Checking for Virus Detection

- Requires keeping carefully protected records of valid file sizes
- Won't work for files whose sizes typically change
  - E.g., Word files with possibly infected macros
- Clever viruses find ways around it
  - E.g., cavity viruses that fit themselves into “holes” in programs

CS 239, Winter 2004

Lecture 16  
Page 28

## Signature Scanning

- If a virus lives in code, it must leave some traces
- In early and unsophisticated viruses, these traces were essentially characteristic code patterns
- Find the virus by looking for the signature

CS 239, Winter 2004

Lecture 16  
Page 29

## How To Scan For Signatures

- Create a database of known virus signatures
- Read every file in the system and look for matches in its contents
- Also check every newly imported file
- Also scan boot sectors and other interesting places

CS 239, Winter 2004

Lecture 16  
Page 30

## Weaknesses of Scanning for Signatures

- What if the virus changes its signature?
- What if the virus takes active measures to prevent you from finding the signature?
- You can only scan for known virus signatures

CS 239, Winter 2004

Lecture 16  
Page 31

## Polymorphic Viruses

- A polymorphic virus produces varying but operational copies of itself
- Essentially avoiding having a signature
- Sometimes only a few possibilities
  - E.g., Whale virus has 32 forms
- But sometimes a lot

CS 239, Winter 2004

Lecture 16  
Page 32

## Stealth Viruses

- A virus that tries actively to hide all signs of its presence
- Typically a resident virus
- For example, it traps calls to read infected files
  - And disinfects them before returning the bytes
  - E.g., the Brain virus

CS 239, Winter 2004

Lecture 16  
Page 33

## Combating Stealth Viruses

- Stealth viruses can hide what's in the files
- But may be unable to hide that they're in memory
- Also, if you reboot carefully from a clean source, the stealth virus can't get a foothold

CS 239, Winter 2004

Lecture 16  
Page 34

## TSR Monitoring

- TSR - Terminate-and-Stay-Resident
  - Essentially a daemon process
- A virus detector that runs in the background
- Automatically scans (and possibly takes other actions) continuously

CS 239, Winter 2004

Lecture 16  
Page 35

## Other TSR Monitor Actions

- Signature scanning can't find new viruses
- Watching system activity for suspicious actions possibly can
- A TSR monitor can run intrusion detection systems or other code to catch new viruses

CS 239, Winter 2004

Lecture 16  
Page 36

## Multi-Level Generic Detection

- Virus detection software that is specialized to handle both known and new viruses
- Using a combination of methods
- Both continuously and on command

CS 239, Winter 2004

Lecture 16  
Page 37

## Generic Detection Tools

- Checksum comparison
- Intelligent checksum analysis
  - For files that might legitimately change
- Intrusion detection methods
  - More sophisticated than intelligent checksum analysis
  - Possibly very high overhead

CS 239, Winter 2004

Lecture 16  
Page 38

## Preventing Virus Infections

- Run a virus detection program
- Keep its signature database up to date
- Disable program features that run executables without users asking
- Make sure users are very careful about what they run

CS 239, Winter 2004

Lecture 16  
Page 39

## How To Deal With Virus Infections

- Reboot from a clean, write-protected floppy or from a clean CD ROM
  - Important to ensure that the medium really is clean
  - Necessary, but not sufficient
- If backups are available and clean, replace infected files with clean backup copies
  - Another good reason to keep backups

CS 239, Winter 2004

Lecture 16  
Page 40

## Disinfecting Programs

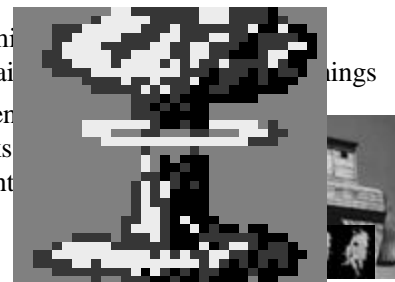
- Some virus utilities try to disinfect infected programs
  - Allowing you to avoid going to backup
- Potentially hazardous, since they may get it wrong
  - Some viruses destroy information needed to restore programs properly

CS 239, Winter 2004

Lecture 16  
Page 41

## Trojan Horses

- Seemingly harmless programs that contain malicious code
- When executed, the program performs a malicious action, such as deleting files or slaughtering livestock



CS 239, Winter 2004

Lecture 16  
Page 42

## Basic Trojan Horses

- A program you pick up somewhere that is supposed to do something useful
- And perhaps it does
  - But it also does something less benign
- Games are common locations for Trojan Horses
- Downloaded applets are increasingly popular locations
- Recently popping up in email attachments

CS 239, Winter 2004

Lecture 16  
Page 43

## Trojan Horse Login Programs

- Probably the original Trojan horse
- Spoof the login or authentication screen of a machine or service
- Capture attempts to access that service
- Then read the user IDs and the passwords

CS 239, Winter 2004

Lecture 16  
Page 44

## Trapdoors

- A secret entry point into an otherwise legitimate program
- Typically inserted by the writer of the program
- Most often found in login programs or programs that use the network
- But also found in system utilities

CS 239, Winter 2004

Lecture 16  
Page 45

## Logic Bombs

- Like trapdoors, typically in a legitimate program
- A piece of code that, under certain conditions, “explodes”
- Also like trapdoors, typically inserted by program authors
- Often used by disgruntled employees to get revenge

CS 239, Winter 2004

Lecture 16  
Page 46

## Worms

- Programs that seek to move from system to system
  - Making use of various vulnerabilities
- Other performs other malicious behavior
- The Internet worm used to be the most famous example
  - Blaster is also a worm
- Can spread very, very rapidly

CS 239, Winter 2004

Lecture 16  
Page 47

## The Internet Worm

- Created by a graduate student at Cornell in 1988
- Released (perhaps accidentally) on the Internet Nov. 2, 1988
- Spread rapidly throughout the network
  - 6000 machines infected

CS 239, Winter 2004

Lecture 16  
Page 48



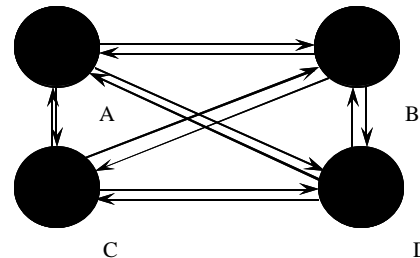
## The Effects of the Worm

- Essentially, affected systems ended up with large and increasing numbers of processes devoted to the worm
- Eventually all processes in the process table used up
- Rebooting didn't help, since other infected sites would immediately re-infect the rebooted machine

CS 239, Winter 2004

Lecture 16  
Page 49

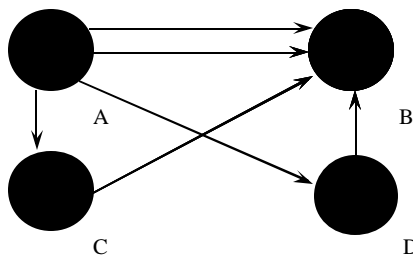
## A Visual Picture of the Infection



CS 239, Winter 2004

Lecture 16  
Page 50

## And What If Someone Reboots?



CS 239, Winter 2004

Lecture 16  
Page 51

## How Did the Internet Worm Work?

- The worm attacked network security vulnerabilities in one class of OS
  - Unix 4 BSD variants
- These vulnerabilities allowed improper execution of remote processes
- Which allowed the worm to get a foothold on a system

CS 239, Winter 2004

Lecture 16  
Page 52

## The Worm's Actions on Infecting a System

- Find an uninfected system and infect that one
- Using the same vulnerabilities
- Here's where it ran into trouble:
  - It re-infected already infected systems
  - Each infection was a new process

CS 239, Winter 2004

Lecture 16  
Page 53

## The Worm's Breaking Methods

- rsh - if the remote host is on the trusted hosts lists, simply rsh'ing could work
- fingerd - exploit a bug in the fingerd program to overwrite a buffer in a useful way
- sendmail - invoke a debugging option in sendmail and issue commands

CS 239, Winter 2004

Lecture 16  
Page 54

### What Didn't the Worm Do?

- It didn't attempt to intentionally damage a system
- It didn't attempt to divulge sensitive information (e.g., passwords)
- It didn't try hard to become root
  - And didn't exploit root access if it got superuser access

CS 239, Winter 2004

Lecture 16  
Page 55

### Stopping the Worm

- In essence, required rebooting all infected systems
  - And not bringing them back on the network until the worm was cleared out
  - Though some sites stayed connected
- Also, the flaws it exploited had to be patched

CS 239, Winter 2004

Lecture 16  
Page 56

### Effects of the Worm

- Around 6000 machines were infected and required substantial disinfecting activities
- Many, many more machines were brought down or pulled off the net
  - Due to uncertainty about scope and effects of the worm

CS 239, Winter 2004

Lecture 16  
Page 57

### How Much Did the Worm Cost?

- Hard to quantify
  - Typical for costs of computer attacks
- Estimates as high as \$98 million
  - Probably overstated, but certainly millions in down time, sysadmin and security expert time, and costs of disconnections

CS 239, Winter 2004

Lecture 16  
Page 58

### What Did the Worm Teach Us?

- The existence of some particular vulnerabilities
- The costs of interconnection
- The dangers of being trusting
- Denial of service is easy
- Security of hosts is key
- Logging is important
- We obviously didn't learn enough

CS 239, Winter 2004

Lecture 16  
Page 59

### MyDoom

- Virus, worm, trapdoor, or Trojan Horse?
- Some of each, really
- Very wide spread
  - Proportionally smaller than Internet worm, but bigger total numbers
- Arrived in email posing in various guises

CS 239, Winter 2004

Lecture 16  
Page 60

### How MyDoom Works

- Usually arrives in email
- Contains an attachment with an executable (Trojan Horse)
- When attachment is opened, it alters registry entries and creates a file in a Kazaa directory (virus)
- Also tries to spread via email (worm)
- Opens a port on your machine (trapdoor)
- Also launches DDoS attack (in some variants)

CS 239, Winter 2004

Lecture 16  
Page 61

### Why Has MyDoom “Succeeded”?

- Not especially sophisticated
- Didn't introduce any new methods
- Didn't exploit any new vulnerabilities
- People still open “interesting” attachments
- Very aggressive
  - Went out to everyone
  - Can also spread via file sharing networks

CS 239, Winter 2004

Lecture 16  
Page 62

### Bagle and Netsky Worms

- In many ways similar to MyDoom
- Differences in details
- Authors of these two worms seem to be dueling
- Bagle had more clever social engineering on email messages than MyDoom
- Claimed to notify you that your email account was being revoked
  - Unless you ran the attachment . . .

CS 239, Winter 2004

Lecture 16  
Page 63

### Code Red

- A malicious worm that attacked Windows machines
- Basically used vulnerability in Microsoft IIS servers
- Became very widely spread and caused a lot of trouble

CS 239, Winter 2004

Lecture 16  
Page 64

### How Code Red Worked

- Attempted to connect to TCP port 80 (a web server port) on randomly chosen host
- If successful, sent HTTP GET request designed to cause a buffer overflow
- If successful, defaced all web pages requested from web server

CS 239, Winter 2004

Lecture 16  
Page 65

### More Code Red Actions

- Periodically, infected hosts tried to find other machines to compromise
- Triggered a DDoS attack on a fixed IP address at a particular time
- Actions repeated monthly
- Possible for Code Red to infect a machine multiple times simultaneously

CS 239, Winter 2004

Lecture 16  
Page 66

## Code Red Stupidity

- Bad method used to choose another random host
  - Same random number generator seed to create list of hosts to probe
- DDoS attack on a particular fixed IP address
  - Merely changing the target's IP address made the attack ineffective

CS 239, Winter 2004

Lecture 16  
Page 67

## Code Red II

- Used smarter random selection of targets
- Didn't try to reinfect infected machines
- Adds a Trojan Horse version of Internet Explorer to machine
  - Unless other patches in place, will reinfect machine after reboot on login
- Also, left a backdoor on some machines
- Doesn't deface web pages or launch DDoS

CS 239, Winter 2004

Lecture 16  
Page 68

## A Major Difference

- Code Red periodically turns on and tries to infect again
- Code Red II worked intensively for 24-48 hours after infection
  - Then stopped
- Eventually, Code Red II infected all infectable machines
  - Some are still infected, but they've stopped trying to spread it

CS 239, Winter 2004

Lecture 16  
Page 69

## Impact of Code Red and Code Red II

- Code Red infected over 250,000 machines
- In combination, estimated infections of over 750,000 machines
- Code Red II is essentially dead
- But Code Red is still out there

CS 239, Winter 2004

Lecture 16  
Page 70

## A Bad Secondary Effect of Code Red

- Generates lots of network traffic
- U. of Michigan study found 40 billion attempts to infect 8 fake "machines" per month
  - Each attempt was a packet
  - So that's ~1 billion packets per day just for those eight addresses
- "The new Internet locust"<sup>1</sup>

<sup>1</sup> Farnham Jahanian, talk at DARPA FTN meeting, Jan 18, 2002

CS 239, Winter 2004

Lecture 16  
Page 71

## Virus Hoaxes

- Virus hoaxes are at least as common as real viruses
- Generally arrive in email
- Usually demand instant action, on pain of something really terrible
- It's wise to check with a reliable source before taking action on such email messages
  - Or forwarding them

CS 239, Winter 2004

Lecture 16  
Page 72