Malicious Software
Computer Security
Peter Reiher
November 25, 2014

Outline

- Introduction
- Viruses
- Trojan horses
- Trap doors
- Logic bombs
- Worms
- Botnets
- Spyware
- Malware components

Introduction

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

Programs have several advantages for these purposes

- Speed
- Mutability
- Anonymity

Where Does Malicious Code Come From?

- Most commonly, it's willingly (but unwittingly) imported into the system
  - Electronic mail
  - Downloaded executables
    - Often automatically from web pages
  - Sometimes shrink-wrapped software
- Sometimes it breaks in
- Sometimes an insider intentionally introduces it

Magnitude of the 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
- 2008 CSI report shows 50% of survey respondents had virus incidents
  - Plus 20% with bot incidents
- 2009 Trend Micro study shows 50% of infected machines still infected 300 days later

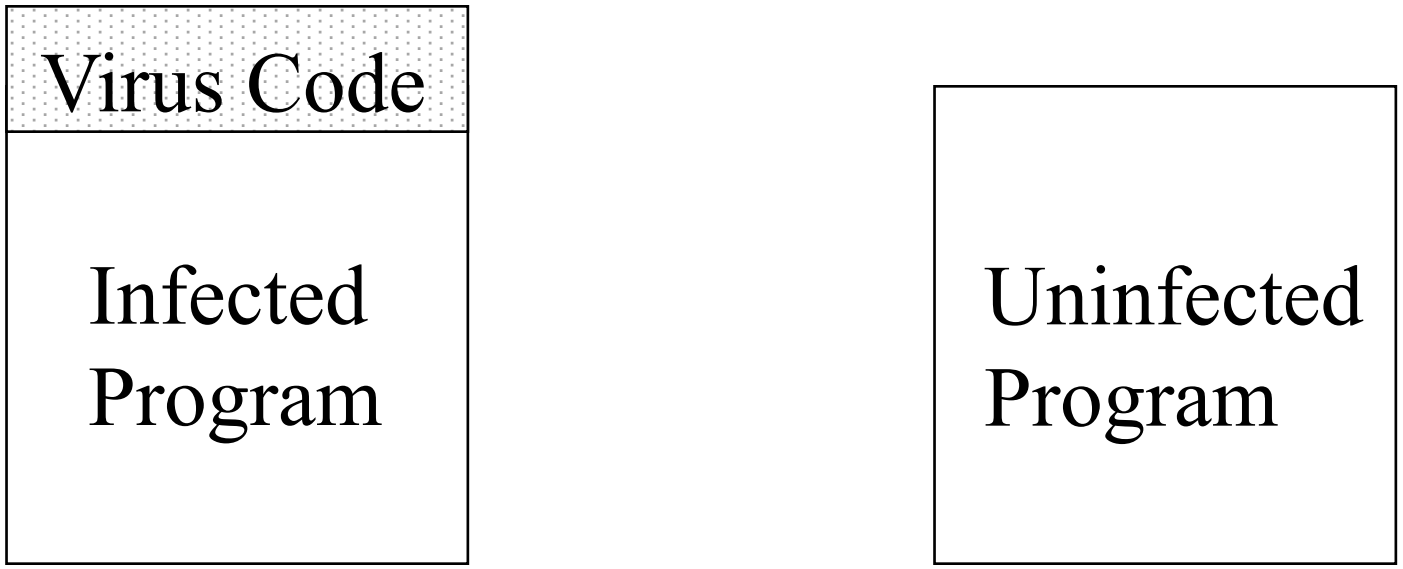
Viruses

- "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

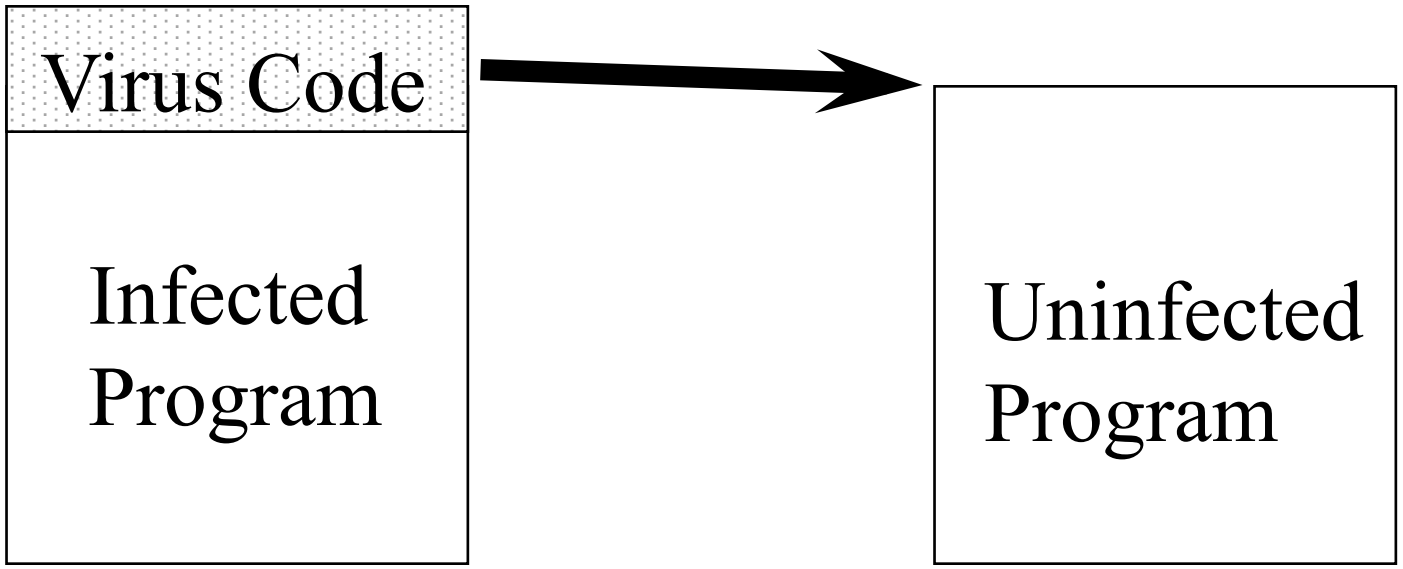
How Do Viruses Work?

- 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

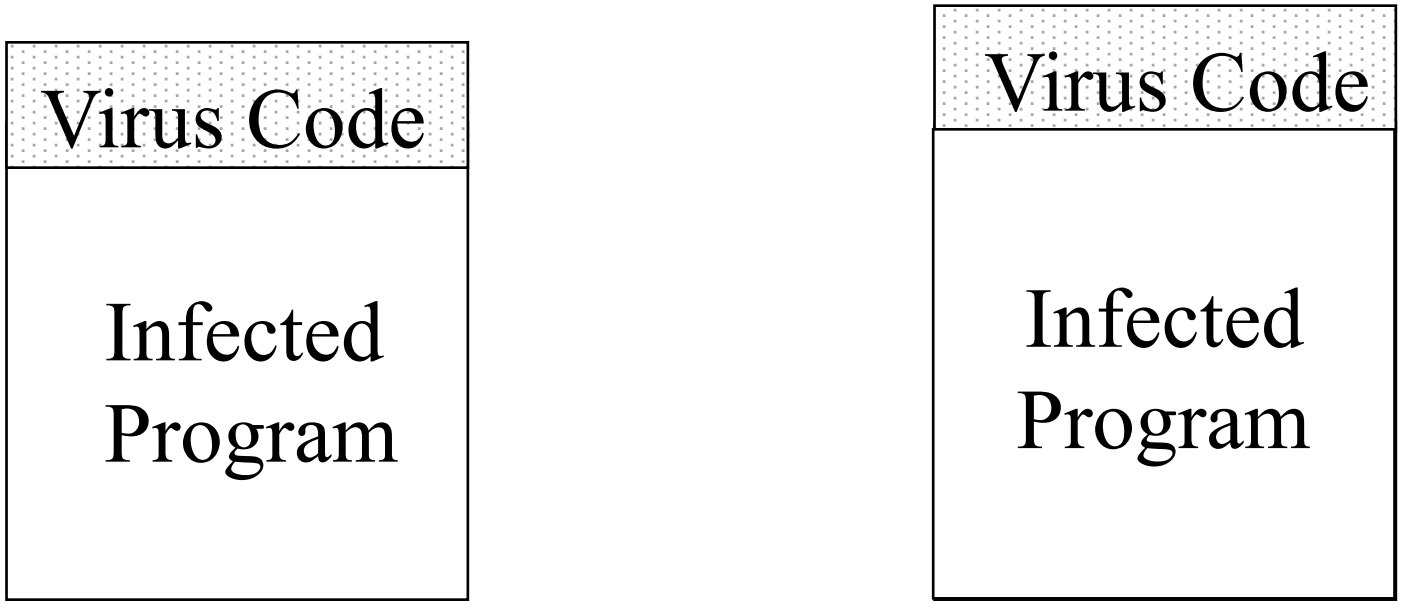
Before the Infected Program Runs



The Infected Program Runs



Infecting the Other Program



Macro and Attachment Viruses

- Modern data files often contain executables
  - Macros
  - Email attachments
- Many formats allow embedded commands to download of arbitrary executables
- Popular form of viruses
  - Requires less sophistication to get right

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 toolkits are getting smarter

How To Find Viruses

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

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

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"

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
  - No use to run in tightly confined mode if user allows it to get out

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 look for this growth
- Won't work for files whose sizes typically change
- Clever viruses find ways around it
  - E.g., cavity viruses that fit themselves into "holes" in programs

Signature Scanning

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

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
- Can use same approach for other kinds of malware

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

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
  - Storm worm had more than 54,000 forms

Polymorphism By Hand

- Malware writers have become professional and security-aware
- They know when their malware has been identified
  - And they know the signature used
  - Smart ones subscribe to all major anti-virus programs
- They change the malware to remove that signature and re-release it

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

Combating Stealth Viruses

- Stealth viruses can hide what's in the files
- But may be unable to hide that they're in memory
- Careful reboot from clean source won't allow stealth virus to get a foothold
- Concerns that malware can hide in other places, like peripheral memory

Other Detection Methods

- Checksum comparison
- Intelligent checksum analysis
  - For files that might legitimately change
- Intrusion detection methods
  - E.g., look for attack invariants instead of signatures
- Identify and handle "clusters" of similar malware

Preventing Virus Infections

- Run a virus detection program
  - Almost all serious organizations do this
  - And many still get clobbered
- Keep its signature database up to date
  - Modern virus scanners do this by default
- Disable program features that run executables without users asking
  - Quicktime had this problem a few years ago
- Make sure users are careful about what they run
- Also make sure users are careful about what they attach to computers

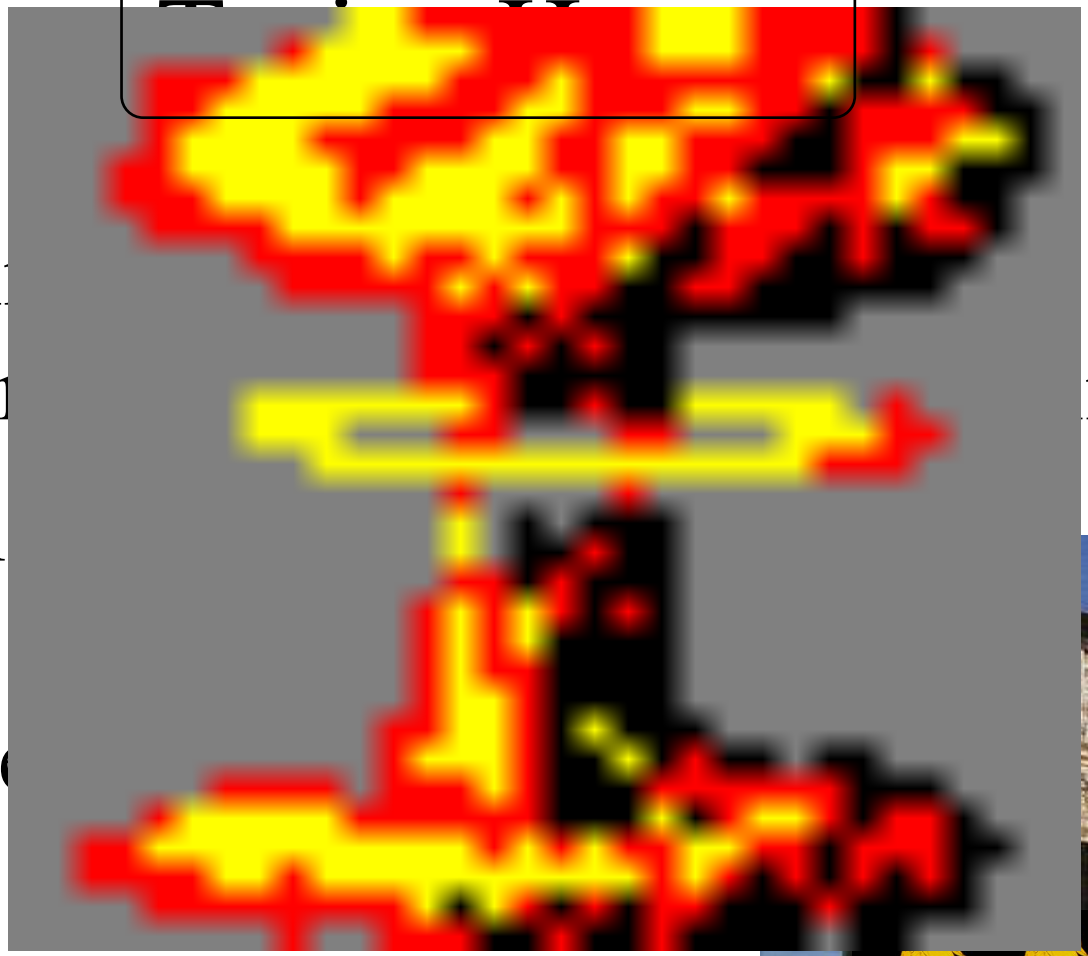
How To Deal With Virus Infections

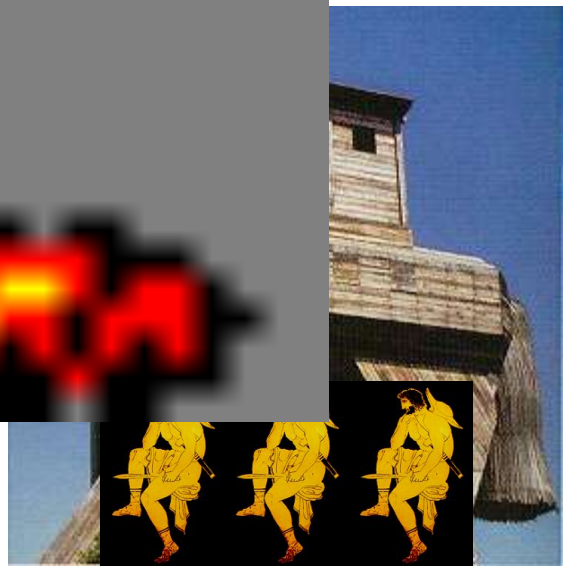
- Reboot from a clean, write-protected medium
  - Vital that the medium really is clean
  - Necessary, but not sufficient
- If backups are available and clean, replace infected files with clean backup copies
  - Another good reason to keep backups
- Proof-of-concept code showed infection of firmware in peripherals . . .

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

- Seemi
  contai ings
  - When
  Greeks
  slaught





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 a common location host program
- Downloaded applets are also popular
- Frequently found in email attachments
- Bogus security products also popular
- Flash drives are a hardware vector

Recent Trends in Trojan Horses

- Qakbot Trojan steals online banking credentials
- Android/IoS Trojan targeting Hong Kong protestors
- Trojan targeting customers of Islamic banks
  - Using man-in-the-middle techniques to overcome 2 factor authentication
  - Other similar Trojans floating around, including a toolkit for them
- Citadel Trojan stole sensitive info from petrochemical companies

Trapdoors

- Also known as back doors
- 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

Trapdoors and Other Malware

- Malware that has taken over a machine often inserts a trapdoor
- To allow the attacker to get back in
  - If the normal entry point is closed
- Infected machine should be handled carefully to remove such trapdoors
  - Otherwise, attacker comes right back

Logic Bombs

- Like trapdoors, typically in a legitimate program
- Code that "explodes" under certain conditions
- Often inserted by program authors
- Previously used by primarily by disgruntled employees to get revenge
  - Former TSA employee got two years in prison for planting one in 2009
- Beginning to be used by nation state cyber attacks
  - South Korean banks and media companies hit with major logic bomb in March 2013

Extortionware

- Attacker breaks in and does something to system
  - Demands money to undo it
  - "Break-in" often via social engineering
    - E.g., claiming it will cure another infection
- Encrypting vital data is common
  - Some incidents also encrypted backups
- Unlike logic bombs, not timed or triggered

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, Slammer, Witty are other worms
- Can spread very, very rapidly

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

How Did the Internet Worm Work?

- The worm attacked vulnerabilities in Unix 4 BSD variants
- These vulnerabilities allowed improper execution of remote processes
- Which allowed the worm to get a foothold on a system
  - And then to spread

The Worm's Actions

- Find an uninfected system and infect that one
- Here's where it ran into trouble:
  - It re-infected already infected systems
  - Each infection was a new process
  - Caused systems to wedge
- Did not take intentional malicious actions against infected nodes

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
- Why didn't firewalls stop it?
  - They weren't invented yet

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

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

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

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

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

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

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
- Didn't turn on periodically

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
  - Except for periodic reintroductions of it
- But Code Red is still out there

Stuxnet

- Scary worm that popped up in 2010
- Targeted at SCADA systems
  - Particularly, Iranian nuclear enrichment facilities
- Altered industrial processes
- Very specifically targeted

Where Did Stuxnet Come From?

- Stuxnet was very sophisticated
  - Speculated to be from unfriendly nation state(s)
  - New York Times claims White House officials confirmed it (no official confirmation, though)
- Research suggests SCADA attacks do not need much sophistication, though
  - Non-expert NSS Labs researcher easily broke into Siemans systems
- Duqu worm might be Stuxnet descendent
  - Appears to be stealing certificates

Worm, Virus, or Trojan Horse?

- Terms often used interchangeably
- Trojan horse formally refers to a seemingly good program that contains evil code
  - Only run when user executes it
  - Effect isn't necessarily infection
- Viruses seek to infect other programs
- Worms seek to move from machine to machine
- Don't obsess about classifications

Botnets

- A collection of compromised machines
- Under control of a single person
- Organized using distributed system techniques
- Used to perform various forms of attacks
  - Usually those requiring lots of power

What Are Botnets Used For?

- Spam (90% of all email is spam)
- Distributed denial of service attacks
- Hosting of pirated content
- Hosting of phishing sites
- Harvesting of valuable data
  - From the infected machines
- Much of their time spent on spreading

Botnet Software

- Each bot runs some special software
  - Often built from a toolkit
- Used to control that machine
- Generally allows downloading of new attack code
  - And upgrades of control software
- Incorporates some communication method
  - To deliver commands to the bots

Botnet Communications

- Originally very unsophisticated
  - All bots connected to an IRC channel
  - Commands issued into the channel
- Most sophisticated ones use peer technologies
  - Similar to some file sharing systems
  - Peers, superpeers, resiliency mechanisms
  - Conficker's botnet uses peer techniques
- Stronger botnet security becoming common
  - Passwords and encryption of traffic

Botnet Spreading

- Originally via worms and direct break-in attempts
- Then through phishing and Trojan Horses
  - Increasing trend to rely on user mistakes
- Conficker uses multiple vectors
  - Buffer overflow, through peer networks, password guessing
- Regardless of details, almost always automated

Characterizing Botnets

- Most commonly based on size
  - Estimates for Conficker over 5 million
  - Zeus-based botnets got 3.6 million machines in US alone
  - Trend Micro estimates 100 million machines are members of botnets
- Controlling software also important
- Other characteristics less examined

Why Are Botnets Hard to Handle?

- Scale
- Anonymity
- Legal and international issues
- Fundamentally, if a node is known to be a bot, what then?
  - How are we to handle huge numbers of infected nodes?

Approaches to Handling Botnets

- Clean up the nodes
  - Can't force people to do it
- Interfere with botnet operations
  - Difficult and possibly illegal
  - But some recent successes
- Shun bot nodes
  - But much of their activity is legitimate
  - And no good techniques for doing so

Spyware

- Software installed on a computer that is meant to gather information
- On activities of computer's owner
- Reported back to owner of spyware
- Probably violating privacy of the machine's owner
- Stealthy behavior critical for spyware
- Usually designed to be hard to remove

What Is Done With Spyware?

- Gathering of sensitive data
  - Passwords, credit card numbers, etc.
- Observations of normal user activities
  - Allowing targeted advertising
  - And possibly more nefarious activities

Where Does Spyware Come From?

- Usually installed by computer owner
  - Generally unintentionally
  - Certainly without knowledge of the full impact
  - Via vulnerability or deception
- Can be part of payload of worms
  - Or installed on botnet nodes

Malware Components

- Malware is becoming sufficiently sophisticated that it has generic components
- Two examples:
  - Droppers
  - Rootkits

Droppers

- Very simple piece of code
- Runs on new victim's machine
- Fetches more complex piece of malware from somewhere else
- Can fetch many different payloads
- Small, simple, hard to detect

Rootkits

- Software designed to maintain illicit access to a computer
- Installed after attacker has gained very privileged access on the system
- Goal is to ensure continued privileged access
  - By hiding presence of malware
  - By defending against removal

Use of Rootkits

- Often installed by worms or viruses
  - E.g., the Pandex botnet
  - But Sony installed rootkits on people's machines via music CDs
- Generally replaces system components with compromised versions
  - OS components
  - Libraries
  - Drivers

Ongoing Rootkit Behavior

- Generally offer trapdoors to their owners
- Usually try hard to conceal themselves
  - And their other nefarious activities
  - Conceal files, registry entries, network connections, etc.
- Also try to make it hard to remove them
- Sometimes removes others' rootkits
  - Another trick of the Pandex botnet