Malicious Software

Dr. Demetrios Glinos
University of Central Florida

CIS3360 - Security in Computing

Readings

- "Computer Security: Principles and Practice", 3rd Edition, by William Stallings and Lawrie Brown
  - Chapter 6

Outline

- Types of Malicious Software (Malware)
- Advanced Persistent Threats
- Propagation Mechanisms
  - Infected Content
  - Vulnerability Exploit
  - Social Engineering
- Payloads
  - System Corruption
  - Attack Agent
  - Information Threat
  - Stealthing
- Countermeasures

Malware Defined

- **Malicious sofware (malware) is:**
  - a program
  - that is inserted into a system, usually covertly,
  - with the intent of
    - compromising the confidentiality, integrity, or availability of the victim's
      - data
      - applications
      - or operating system
    - or otherwise annoying or disrupting the victim.

Terminology

- There are many terms used for describing malware,
  - some overlap or are defined differently by different authors
- Some stable terms that you should know

virus –attached to executable content; infects other executables on same machine

trojan – seemingly useful program that also has a hidden and malicious function

worm – malware that can run independently and which can replicate itself on other hosts, typically on a network

spyware – malware that collects information and sends it to a remote host

adware – advertising inserted into software, producing pop-ups or redirecting

keylogger – malware that captures key strokes on compromised machines

drive-by-download – malware that attacks a client when a Web site is viewed

zombie (bot) – malware activated remotely to launch attacks on other machines

logic bomb – malware that lies dormant until a specific condition is met

backdoor (trapdoor) – mechanism that avoids normal security checks

Attack Kits and Attack Sources

- **Attack kit**
  - a set of tools for generating new malware automatically using a variety of supplied propagation and payload mechanisms
  - allows even "novice" cybercriminals to develop and deploy malware
  - for this reason, often also called crimeware

- **Attack sources**
  - originally
    - attackers were individuals, often motivated to demonstrate their technical competence to their peers
  - today
    - politically motivated attackers
    - criminals and organized crime
    - organizations that sell their services to companies and nations
    - national government agencies

Advanced Persistent Threats

- **Advanced Persistent Threat (APT)**
  - When a specific target of attack is repeatedly attacked over a period of time with a wide variety of intrusion technologies and malware, we say there is an APT to that target

- Key features of APT:
  - specific target
  - advanced – wide variety of intrusion technologies and malware
  - persistent – determined application of the attacks over an extended period
  - threat – organized, capable, and well-funded

- APTs are typically attributed to
  - state-sponsored organizations
  - some also to criminal enterprises

Propagation: Infected Content

- **Virus**
  - parasitic software fragment that attaches itself to existing executable content, such as
    - application, utility, system program, boot loader
    - also scripting code for active content in MS Word, Excel, or Adobe PDF
  - when attached
    - executes every time the host program is run
    - can do anything that the host program is allowed to do
  - viruses are specific to particular OS and hardware configurations

- viruses do 2 things
  - propagate – spread to other programs and/or systems
  - payload action - what the virus does, besides propagation
    - could be damaging, or benign but noticeable

How a Virus Works

- When the infected executable is run:
  - the virus code runs first
    - the virus attempts to spread,
    - the virus executes its payload, if its trigger conditions are met
    - and then control is passed to the original executable code

```
program V
1234567;

procedure attach-to-program;
begin
   repeat
      file := get-random-program;
   until first-program-line ≠ 1234567;
   prepend V to file;
end;

procedure execute-payload;
begin
   (* perform payload actions *)
end;

procedure trigger-condition;
begin
   (* return true if trigger condition is true *)
end;

begin (* main action block *)
   attach-to-program;
   if trigger-condition then execute-payload;
   goto main;
end;
```

source: Fig. 6.1(a)

Compression Viruses

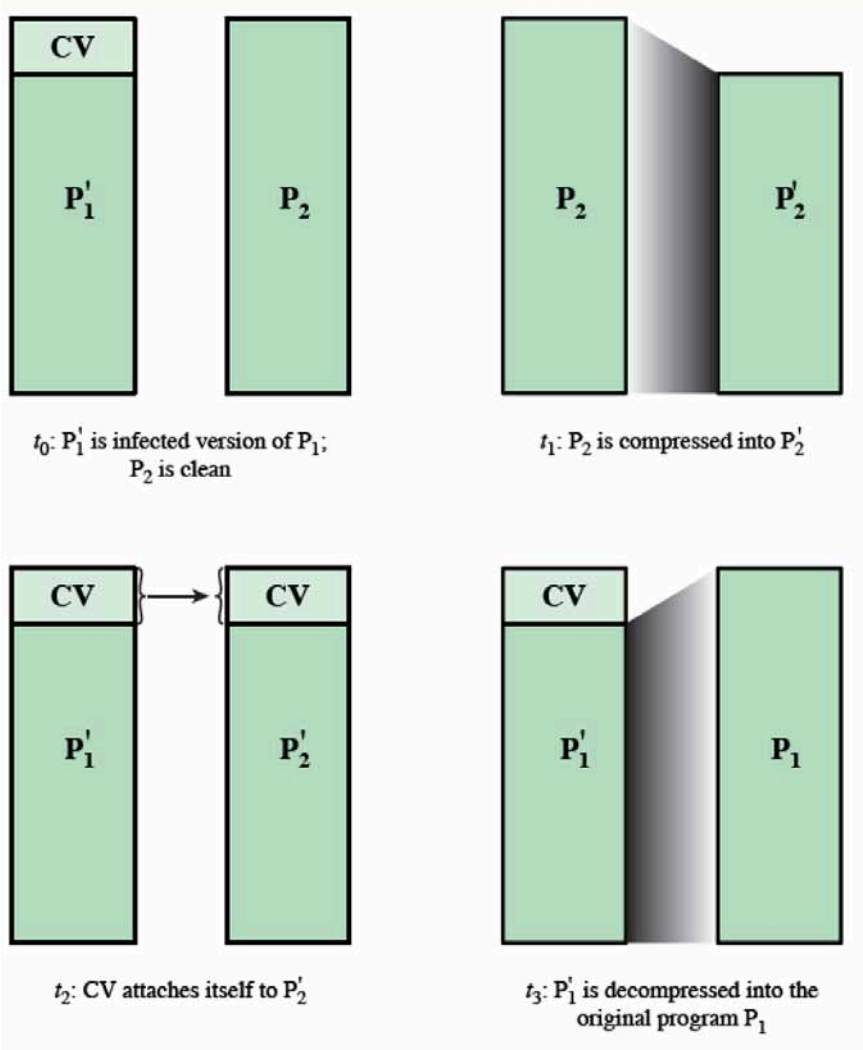
- File compression is often used by viruses to avoid detection

```
program CV
1234567;

procedure attach-to-program;
begin
   repeat
      file := get-random-program;
   until first-program-line ≠ 1234567;
   compress file;   (* t1 *)
   prepend CV to file;   (* t2 *)
end;

begin (* main action block *)
   attach-to-program;
   uncompress rest of this file into tempfile;   (* t3 *)
   execute tempfile;   (* t4 *)
end;
```

source: Fig. 6.1(b)



source: Fig. 6.2

Virus Concealment Strategies

- **Virus signatures**
  - Like all code, virus code can be detected by the distinctive bit pattern of its particular machine code instructions, even if the infected file is compressed
  - virus detection software constantly updates libraries of virus signatures
- **Concealment techniques** are often used to alter virus signatures
  - **Encrypted virus**
    - virus code includes a small encryption engine and encrypts most of its code; code is decrypted to run; each replication uses a different key
  - **Polymorpic virus**
    - during replication, uses different but functionally equivalent code, such as adding NOP instructions, using while instead of for-loops, etc.
  - **Metamorphic virus**
    - virus rewrites itself completely in place every time it is run; replications also mutated (same as for polymorphic)
  - **Stealth virus**
    - modifies OS using **"rootkit" methods** to avoid detection by anti-virus software (e.g., so executable does not appear in Task Manager)

Propagation: Vulnerability Exploit

- **Worm**
  - An independent program that actively seeks out other machines to infect
  - Usually carries a payload

- Key features
  - runs on its own
  - spread through
    - network connections
    - shared media: USB drives, CDs, DVD data disks
    - macro or script code in email attachments or instant messenger file transfers
  - infection mechanism
    - exploitation of software vulnerabilities in client or server programs
    - e.g., servers, browsers, email systems, file transfer programs, etc.

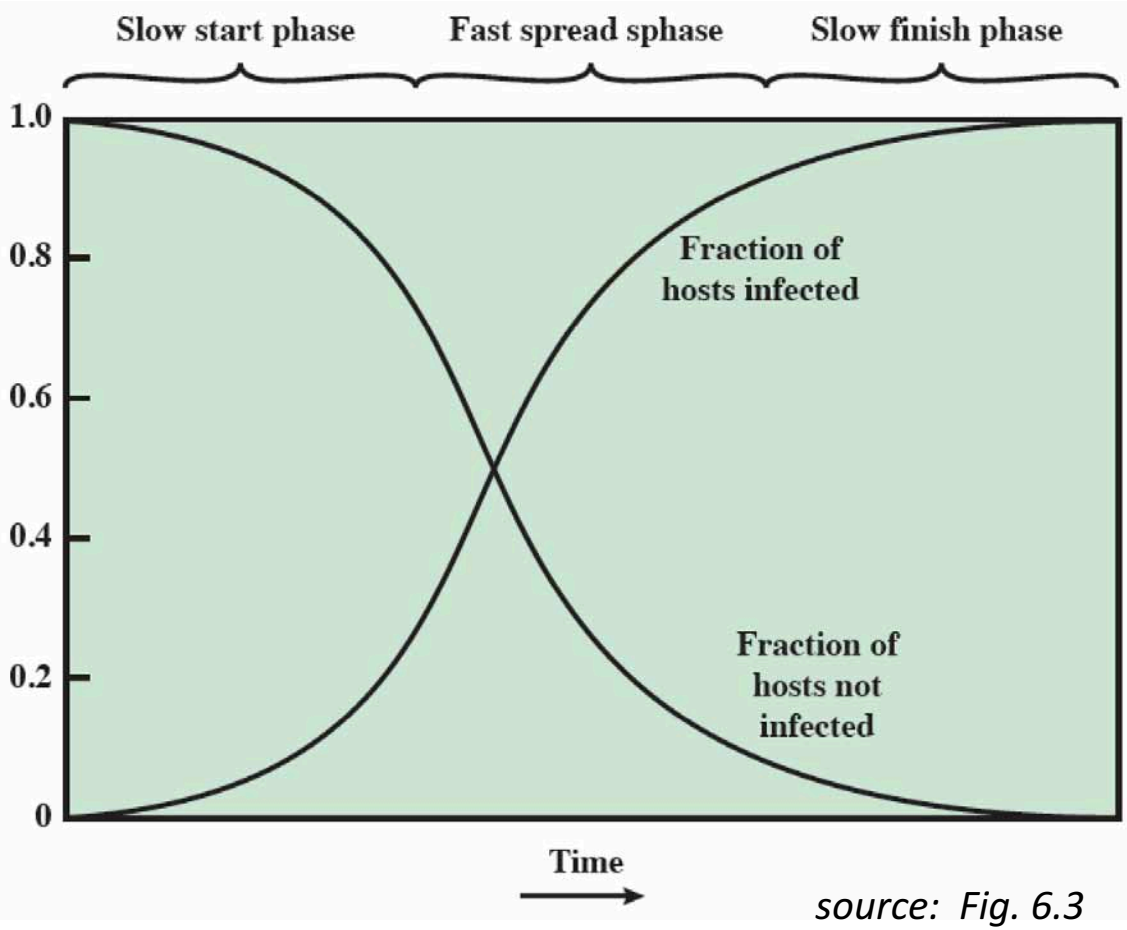
How a Network Worm Spreads

- Worm propagation
  - the worm scans the network for other machines with the particular vulnerability that the worm is designed to exploit
  - it then infects those machines
  - those newly infected machines repeat the process
    - result is a "chain reaction" of infections

- Network scanning (fingerprinting) strategies
  - random -each compromised host probes random addresses using a different seed
  - hit-list - attacker first compiles a long list of vulnerable machines, and then gives a different part of the list to each infected machine
  - topological – use information on a victim machine to find others
  - local subnet – if can infect behind a firewall, then search for other machines on the local subnet

Worm Propagation Model

- Worm propagation follows the classic epidemiological model
- slow start phase
  - exponential growth
- middle phase
  - roughly linear growth
- slow finish phase
  - few remaining hosts to infect



source: Fig. 6.3

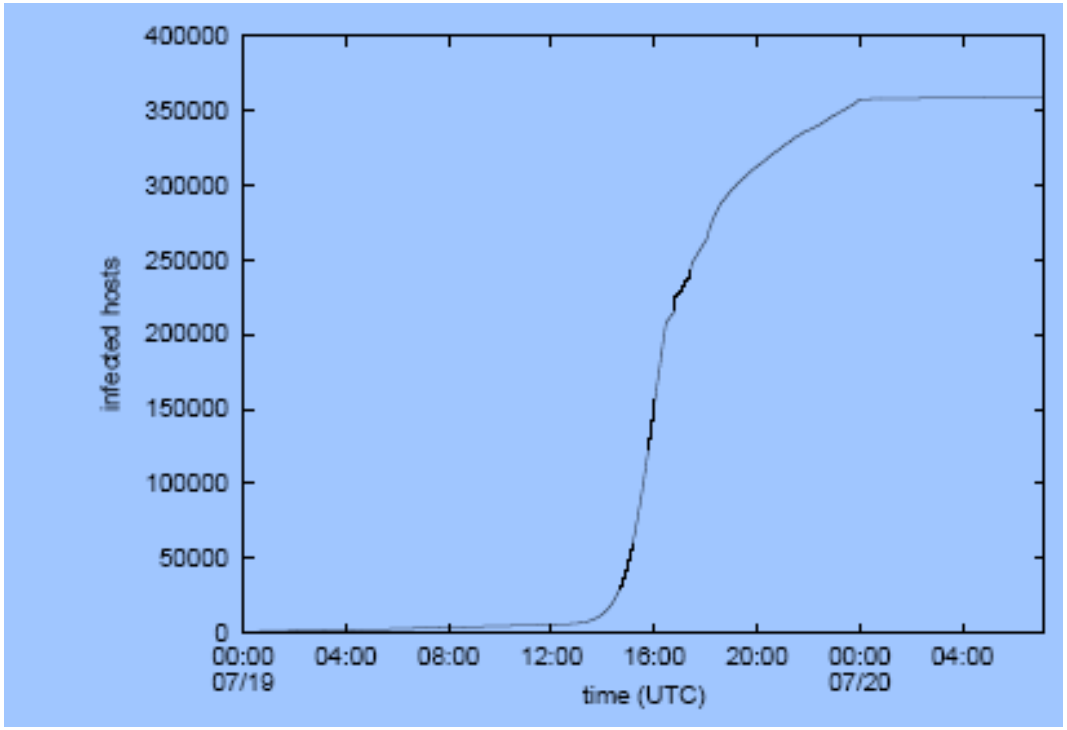
→ goal for countermeasures: catch worm in the slow start phase

Notable Computer Worms

- Morris worm (1988)
  - Infected early UNIX systems
  - Harmless, but ***reinfection rate*** had same effect as denial of service
- Melissa (1998)
  - email worm; first to combine virus-trojan-worm features in one package
- ILOVEYOU (2000)
  - Email worm disguised as a ***love letter*** named **LOVE-LETTER-FOR-YOU.TXT.vbs**
  - Sent itself to everyone in the user's address book
  - Also destructive: replaced documents and photos with copies of itself
- Blaster (2003) and Sasser (2004)
  - Exploited ***buffer overflow*** in several Windows versions
  - Sasser caused Delta Airlines to cancel some flights
- Conficker (2008)
  - Allowed attacker to launch DoS, install spyware, or send spam
  - ***Disabled safe mode & auto-update, and killed antimalware***

Empirical Data for Code Red Worm

- Cumulative total of unique IP addresses infected by the first outbreak of **Code-Red v2** on July 19-20, 2001
  - Exploited ***buffer overflow in Microsoft IIS web server***
  - Malicious action: Launch denial-of-service attacks on selected sites
  - ***Infected over 350,000 hosts on the first day***



Source:
David Moore, Colleen Shannon, and Jeffery Brown. Code-Red: a case study on the spread and victims of an Internet worm, CAIDA, 2002

Propagation: Using Client-Side Vulnerabilities

- An infected or malicious server can propagate by exploiting vulnerabilities in a client-side application when the client connects to the server
  - the malware does not actively propagate like a worm
  - instead, the malware waits for an ususpecting user to connect to the server
- **Drive-by download**
  - here, it is a browser vulnerability that is exploited
  - when victim views a Web page controlled by the attacker, the server exploits the broswer bug to download and install malware on the victim's machine
- **Clickjacking**
  - also known as *user-interface (UI) redress attack*
  - involves altering the apparent UI so that when the user intends to select (click) one component, the click event is processed by another component, which may redirect the user to a malicious page or take other action
  - typically uses multiple transparent or opaque layers, placing a button over or under a legitimate button, etc.

Propagation: Social Engineering

- **Social engineering**
  - involves tricking users into assisting in the compromise of their systems or personal information
  - examples
    - user clicks link in some SPAM email
    - user innocently permits installation and execution of some Trojan horse program or scripting code
- a classic social engineering attack (from the Grimms' fairy tale):
  - Little Red Riding Hood's grandmother letting the Big Bad Wolf into the house

Spam

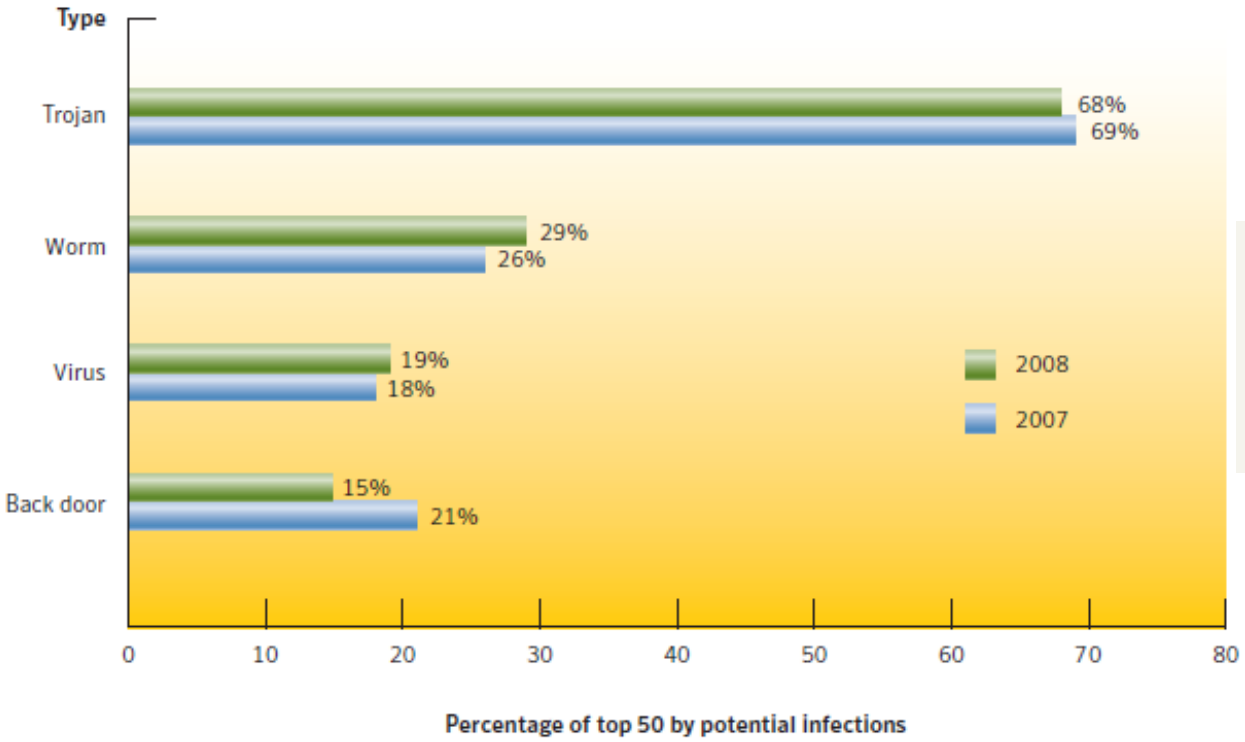
- **Spam**
  - unsolicited bulk e-mail
  - estimated to be 90% of all email traffic
  - most spam is just advertising
  - some contains attachments that contain Trojan horse programs or scripting code that, if run, installs malware on the user's machine
  - spam is often used for a **phishing** attack
    - that directs the user to a fake Web site that mimics a legitimate site
      - e.g., spoofing the user's online banking site
    - or tricks the user into entering personal information into a bogus Web page form so the attacker can steal the user's identity
  - recent decline in e-mail spam
    - along with rapid rise in malware attacks through social media networks

Trojan Horses

- **Trojan Horse** (or simply **Trojan**)
  - a useful, or apparently useful, program or utility that contains hidden code that, when invoked, performs some unwanted or harmful action
    - used to perform a malicious action that the attacker could not do directly
  - typically, some user action is required
    - e.g, downloading seemingly useful app from a mobile phone app store
    - but some Trojans bypass this requirement by exploiting some software vulnerability to enable automatic installation and execution
    - however, unlike a worm, such a Trojan does not replicate
  - Mocmex Trojan (2008)
    - Distributed via Chinese-made ***digital photo frames***
    - Malware installed when photo frame connected to a Windows computer to load photos; collected and transmitted passwords to attacker
  - smartphones have been targets of Trojans since 2004
    - usually distributed on official and unofficial app stores

Popularity of Trojans

- Trojans currently have the largest infection potential
  - Often exploit browser vulnerabilities
  - Typically ***used to download other malware*** in multi-stage attacks.



Source:
Symantec Internet Security Threat Report, April 2009

Payload: System Corruption

- **System corruption** payload
  - an attack that targets the integrity of a systems hardware, software, or user data
- Chernobyl virus (1998)
  - when trigger date reached, deleted data on Win 95/98 by overwriting the first megabyte of the hard drive with zeroes
- Klez mass-email worm (2001)
  - mailed itself to address book contacts on Win 95/98/XP systems
  - used **logic bomb**: on trigger dates (13th of the month) deleted contents of files
- WannaCry cryptoworm (May 2017)
  - worldwide attack targeting Win XP/7/8/10 and Windows Server 2008,12,16
  - contained **ransomware**:
    - encrypted user files and demanded a ransom payment in Bitcoin to obtain decryption key

Logic Bombs

- **Logic bomb**
  - a malicious action that is triggered by the evaluation of a ***logical expression***
  - Example: a payroll system that crashes if two periods go by without a paycheck for the developer
- Often a key component of data-corupting malware
- Can be combined with a **backdoor** (a hidden feature that allows access or action)
  - Example: backdoor that allows attacker to disable bomb when ransom paid
- **2008 Fannie Mae logic bomb:**
  - Programmed to erase all data 3 months after programmer discharged
  - Discovered before target date, so bomb did not explode
- **Y2K Problem distinguished:**
  - Original intent: save memory at a time when memory was not cheap or plentiful
  - No malicious intent, so not a logic bomb

The Omega Engineering Logic Bomb

- **Bomb caused much damage**
  - Triggered on July 31, 1996
  - Files on the company's manufacturing server were destroyed
  - Caused $10 million in damages
  - Caused layoffs of 80 employees

- **Programmed by system administrator Tim Lloyd**
  - The only backup tapes were in Lloyd's possession and had been erased
  - Convicted of computer espionage
    - sentenced to 41 months in prison
    - ordered to pay $2 million in restitution

Payload: Attack Agent

- **Zombie, bot (robot), or drone**
  - malware that secretly takes over another Internet-attached computer and uses it to launch or manage attacks that are difficult to trace to the bot's creator
  - controlled by a remote control facility
- Bots are typically
  - planted on hundreds of thousands of unsuspecting victim machines
    - the collection is known as a **botnet**
  - capable of coordinated action under the control of the remote control facility, often referred to as the **bot herder**
- Botnets are used for
  - Distributed denial of service (DDoS) attacks
  - spamming
  - spreading malware
  - installing adware
  - etc.

Payload: Information Theft

- Credential theft: **Keyloggers, and Spyware**
  - at-risk credentials: usernames/passwords for banking, gaming, and related sites
  - **keyloggers** – record keystrokes, from which analysis can reveal credentials
  - **spyware** – monitor additional activity, such as history and content of browser activity
- Identity theft: **Phishing**
  - exploits social engineering to leverage user's trust by masquerading as communications from a trusted source
  - typically, spam email that directs user to a fake Web site controlled by attacker
  - fake Web site mimics a legitimate Web site
  - unsuspecting user enters login credentials on fake Web site
  - **spear-fishing**
    - variant in which a targeted user is researched and phishing email designed to contain some personal information, to convince user to trust it
- **Reconnaissance:** the above are just special cases; all kinds of info can be harvested

Payload: Stealthing

- **Stealthing**
  - refers to methods used by malware to hide its presence on the infected system and to provide covert access to that system
- **Backdoor (trapdoor):**
  - a secret entry point into a program that bypasses normal security checks
  - distinguish from **maintenance hook**
    - legitimate backdoor inserted by a developer to support debug and testing
  - ***Trigger*** could be:
    - Special command sequence
    - Special username and/or password
    - Deliberate insertion of vulnerability (e.g., failure to check for buffer overflow)
  - ***Easter Eggs*** distinguished
    - Harmless undocumented features unlocked by secret password or keystrokes

Rootkits

- A **rootkit** is malware that *modifies OS components to hide its existence*
  - Example: modifying the Windows Process Monitor utility not to show the rootkit in the process list

- **User-mode rootkits** (relatively easier to detect)
  - These alter system utilities or libraries on disk
  - May insert code into another user-mode process's address space
  - can be detected by the OS kernel

- **Kernel-mode rootkits** (generally more difficult)
  - These affect kernel functions
  - May use *function hooking* (i.e., modifying kernel *functions*)
    - e.g., to conceal its presence
  - May *modify data structures* used by kernel components
    - e.g., permissions files, system registry

Copy Protection Rootkit

- This rootkit was distributed with ***SONY BMG music CDs in 2005***

- Installed when music CD was placed in computer drive
  - Only affected Windows systems
  - Took advantage of Windows "AutoPlay" feature
  - Modified system files to hide its presence
  - Prevented unauthorized copying of music files

- Malicious attackers soon learned to name their files so that they would also be hidden by this rootkit

- SONY relented after public outcry.

Countermeasures

- **Anti-virus software**
  - originally developed to counter virus infections
  - now address all types of malware, but name remains for historical reasons

- Countermeasure mechanisms seek to mitigate malware by
  - prevention
  - detection
  - identification
  - removal

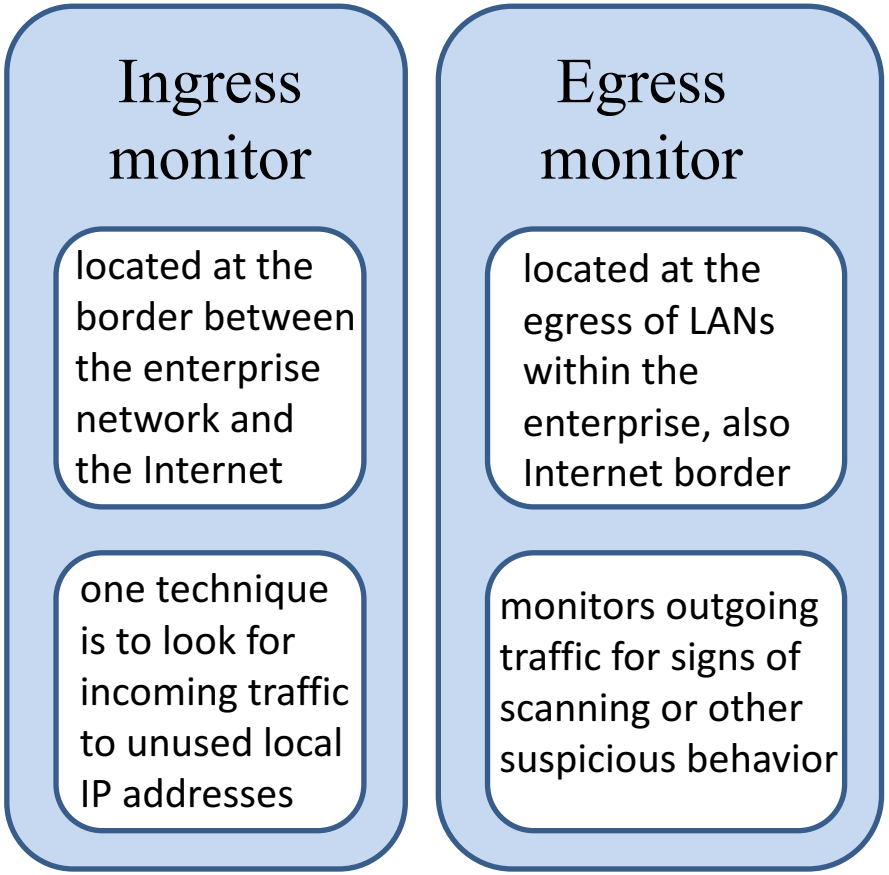
- Countermeasures can be implemented as
  - host-based scanners – e.g., the PCs that can be infected
  - perimiter scanners - organization firewalls, IDS,
    - e.g., email and Web service proxies

Host-Based Scanners

- First generation: **simple scanners**
  - requires a malware signature to identify
  - limited to detection of known malware
  - can also check for changes in file sizes
- Second generation: **heuristic scanners**
  - uses heuristic rules to search for probable malware instances
    - e.g., rule to detect beginning of encryption loop of a polymorphic virus
  - also uses integrity checking
- Third generation: **activity traps**
  - memory-resident programs that identify malware by its activity, not structure
    - e.g., actions that attempt to modify system files
- Fourth generation: **combined approaches**
  - use a variety of techniques, including
    - **generic decryption**: execute in a sandbox to detect encrypted malware
    - **behavior blocker**: monitor program execution at the instruction level

Perimiter Scanning Approaches

- An organization's perimeter includes
  - firewall server
  - IDS server
- Anti-virus software is typically included in services running on perimeter servers
  - e-mail services
  - Web proxy services
- May also be included in the traffic analysis component of the IDS
- May include intrusion prevention measures, blocking the flow of any suspicious traffic
- Approach is limited to scanning for malware



Types of monitoring software