Malware: Malicious Software

Viruses, Worms, Trojans, Rootkits

- Malware can be classified into several categories, depending on propagation and concealment
- Propagation
  - Virus: human-assisted propagation (e.g., open email attachment)
  - Worm: automatic propagation without human assistance
- Concealment
  - Rootkit: modifies operating system to hide its existence
  - Trojan: provides desirable functionality but hides malicious operation
- Various types of payloads, ranging from annoyance to crime

4.1 Insider Attacks

- An **insider attack** is a security breach that is caused or facilitated by someone who is a part of the very organization that controls or builds the asset that should be protected.
- In the case of malware, an insider attack refers to a security hole that is created in a software system by one of its programmers.

4.1.1 Backdoors

- A **backdoor,** which is also sometimes called a **trapdoor,** is a hidden feature or command in a program that allows a user to perform actions he or she would not normally be allowed to do.
- When used in a normal way, this program performs completely as expected and advertised.
- But if the hidden feature is activated, the program does something unexpected, often in violation of security policies, such as performing a privilege escalation.
- Benign example: **Easter Eggs** in DVDs and software

4.1.1 Backdoors inserted for debugging purpose

- A programmer is working on an elaborate biometric authentication system for a computer login program.
- She may wish to provide a special command or password that can bypass the biometric system in the event of a failure.
- It is useful during the code development and debugging, but become a risk that may allow an attacker to bypass authentication measures.

4.1.1 Deliberate Backdoors

- Deliberately insert backdoors so programmers can perform malicious actions later. For example, a program may add a backdoor through using of a sequence of keystrokes to access a digital entry system for a bank vault.
- Deliberately introduce a vulnerability to a program such as BoF. Introducing an exploitable bug into the code of an open source project, allow programmers to gain access from other machines.

4.1.2 Logic Bombs

- A **logic bomb** is a program that performs a malicious action as a result of a certain logic condition.
- The classic example of a logic bomb is a programmer coding up the software for the payroll system who puts in code that makes the program crash should it ever process two consecutive payrolls without paying him.
- Another classic example combines a logic bomb with a backdoor, where a programmer puts in a logic bomb that will crash the program on a certain date.

4.1.2 The Omega Engineering Logic Bomb

- An example of a logic bomb that was actually triggered and caused damage is one that programmer Tim Lloyd was convicted of using on his former employer, Omega Engineering Corporation. On July 31, 1996, a logic bomb was triggered on the server for Omega Engineering's manufacturing operations, which ultimately cost the company millions of dollars in damages and led to it laying off many of its employees.

4.1.2 The Omega Bomb Code

- The Logic Behind the Omega Engineering Time Bomb included the following strings:
- 7/30/96
  - Date that triggered the bomb; time bomb
- F:
  - Focused attention to volume F, which had critical files
- F:\LOGIN\LOGIN 12345
  - Login a fictitious user, 12345 (the back door). The user had supervisory and destroy permissions but no password.
- CD \PUBLIC
  - Change the current directory to the public folder
- FIX.EXE /Y F:*.*
  - Run a program, called FIX, which actually deletes everything; /Y means each file should be deleted. F:\ *.* deletes all files.
- PURGE F:\/ALL
  - Prevent recovery of the deleted files

4.1.3 Defenses against Insider Attacks

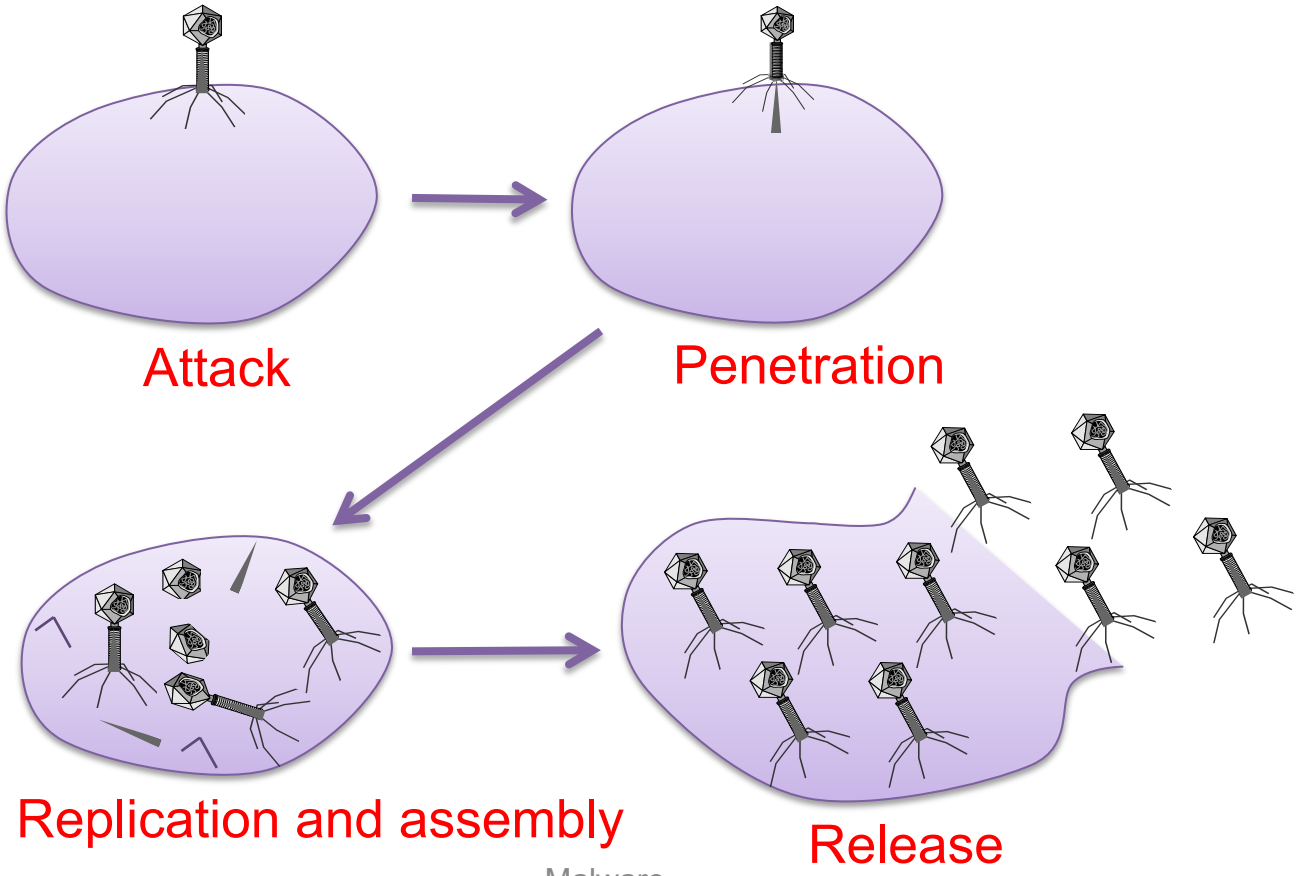
- Avoid single points of failure.
- Use code walk-throughs.
- Use archiving and reporting tools.
- Limit authority and permissions.
- Physically secure critical systems.
- Monitor employee behavior.
- Control software installations.

4.2 Computer Viruses

- A **computer virus** is computer code that can replicate itself by modifying other files or programs to insert code that is capable of further replication.
- This self-replication property is what distinguishes computer viruses from other kinds of malware, such as logic bombs.
- Another distinguishing property of a virus is that replication requires some type of **user assistance,** such as clicking on an email attachment or sharing a USB drive.

Biological Analogy

- Computer viruses share some properties with Biological viruses



Virus Phases

- **Dormant phase.** During this phase, the virus just exists—the virus is laying low and avoiding detection.
- **Propagation phase.** During this phase, the virus is replicating itself, infecting new files on new systems.
- **Triggering phase.** In this phase, some logical condition causes the virus to move from a dormant or propagation phase to perform its intended action.
- **Action phase.** In this phase, the virus performs the malicious action that it was designed to perform, called **payload.**
  - This action could include something seemingly innocent, like displaying a silly picture on a computer's screen, or something quite malicious, such as deleting all essential files on the hard drive.

Types of Viruses

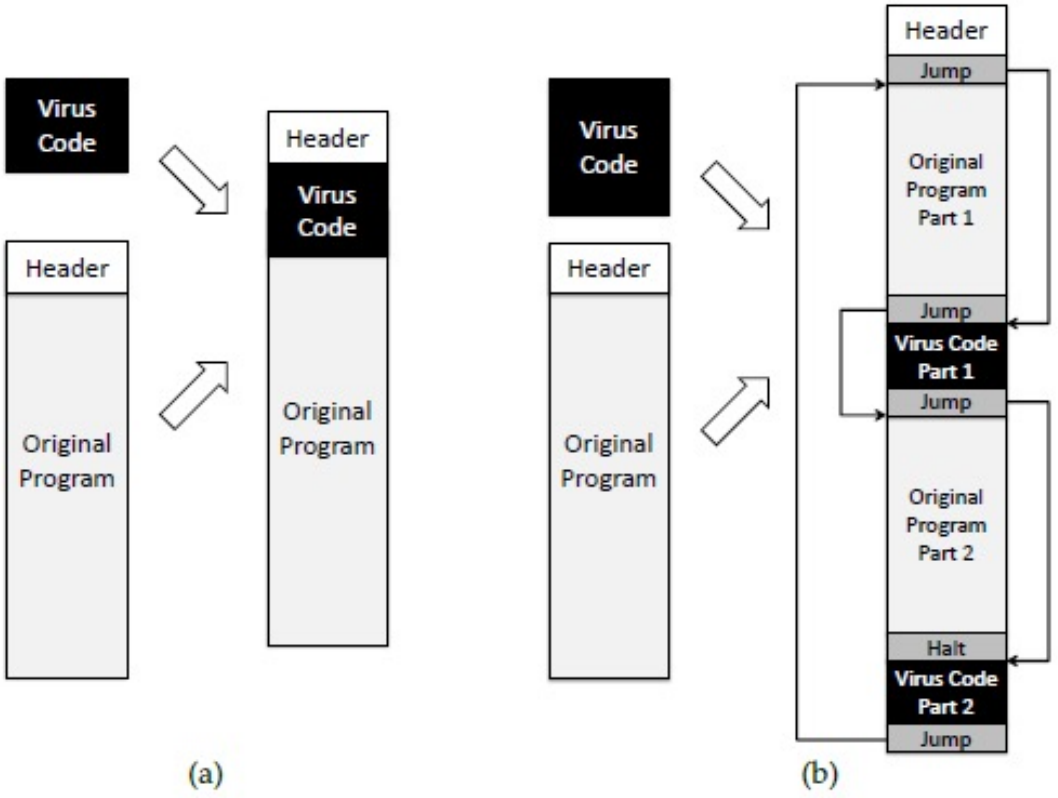
- Program Virus / file virus:
  - infects a program by modifying the file containing its object code.
  - Once the infection occurs, a program virus is sure to be run each time the infected program executes.
- Macro virus / document virus
  - When a document is opened, it searches for other documents to infect.
  - It can insert itself into the document template, which makes the newly created document infected.
  - Further propagation occurs when the infected documents are emailed to other users.

Types of Viruses

- Boot section virus:
  - Infects the code in the boot sector of a drive, which is run each time the computer is turned on or restarted.

Degrees of Complication

- Viruses have various degrees of complication in how they can insert themselves in computer code.

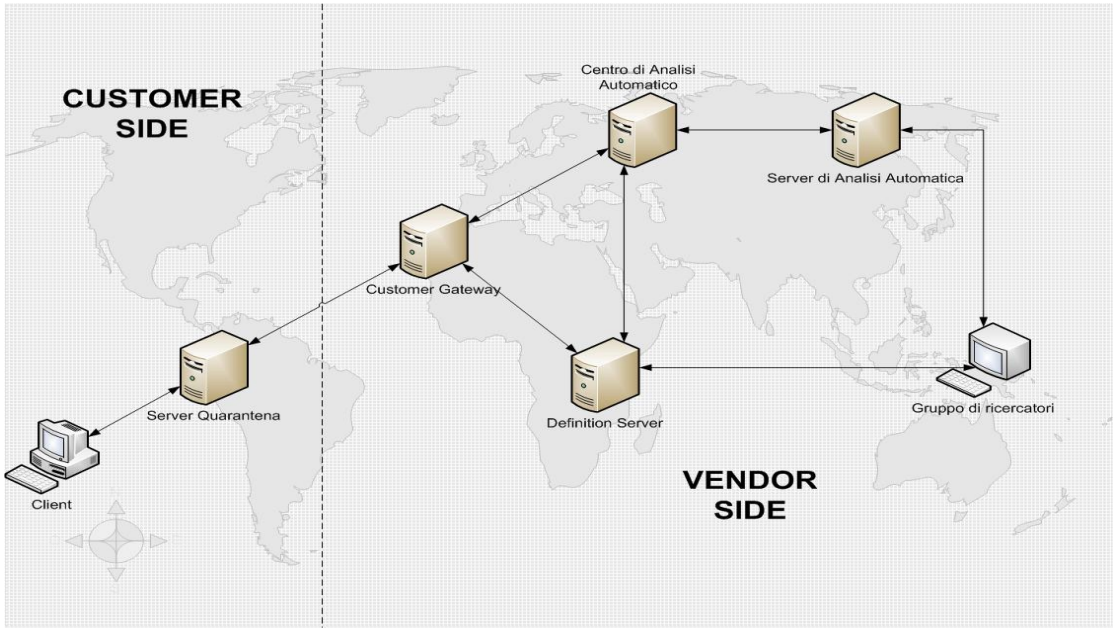


4.2.2 Defenses against viruses - Signature

- Scan compare the analyzed object with a database of signatures
- A signature is a virus fingerprint
  - E.g., a string with a sequence of instructions specific for each virus
  - Different from a digital signature
- A file is infected if there is a signature inside its code
  - Fast pattern matching techniques to search for signatures
- All the signatures together create the malware database that usually is proprietary

4.2.2 Defenses against viruses -- Signatures Database

- Common Malware Enumeration (CME)
  - aims to provide unique, common identifiers to new virus threats
  - Hosted by MITRE
  - http://cme.mitre.org/data/list.html
- Digital Immune System (DIS)
  - Create automatically new signatures



4.2.2 Defenses against viruses -- Quarantine

- A suspicious file can be isolated in a folder called quarantine:
- The suspicious file is not deleted but made harmless: the user can decide when to remove it or eventually restore for a false positive
  - Interacting with a file in quarantine it is possible only through the antivirus program
- The file in quarantine is harmless because it is encrypted
- Usually the quarantine technique is proprietary and the details are kept secret

4.2.3 Encrypted Viruses

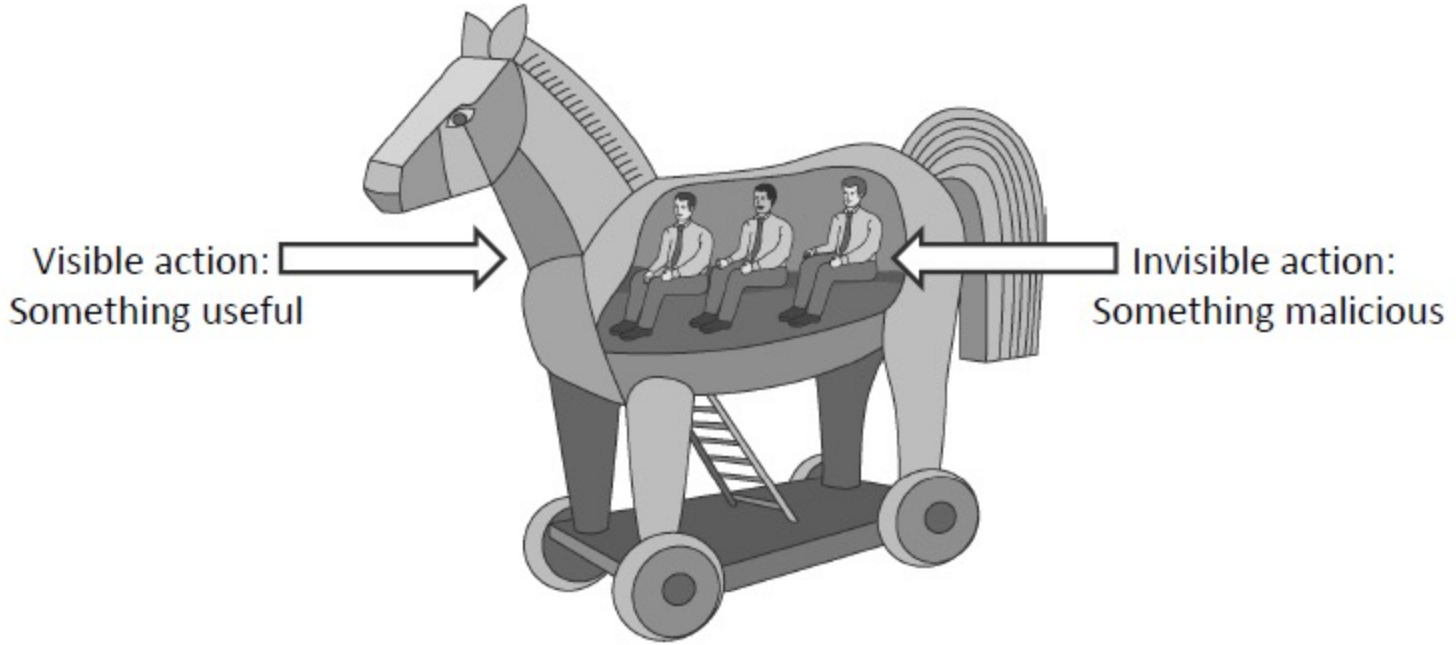
- The presence of their virus in a file is more stealthy if the main body of the program is encrypted, especially the replication code and payload
- The virus code's new structure: the decryption key, the key and the encrypted virus code.
- This structure becomes a kind of virus signature
- The arm race continues: Signature based detection → encrypted viruses → look for encryption code

4.2.4 Polymorphic and Metamorphic Viruses

- Both of them are difficult to detect because they have few fixed characteristic patterns of bits in their codes.
- Polymorphic virus
  - Using encryption.
  - Each copy of the virus is encrypted using a different key.
  - Detect by generic code for an encryption algorithm
- Metamorphic virus
  - Non-cryptographic obfuscation techniques, such as instruction reordering, inclusion of useless instructions.
  - Challenging to detect

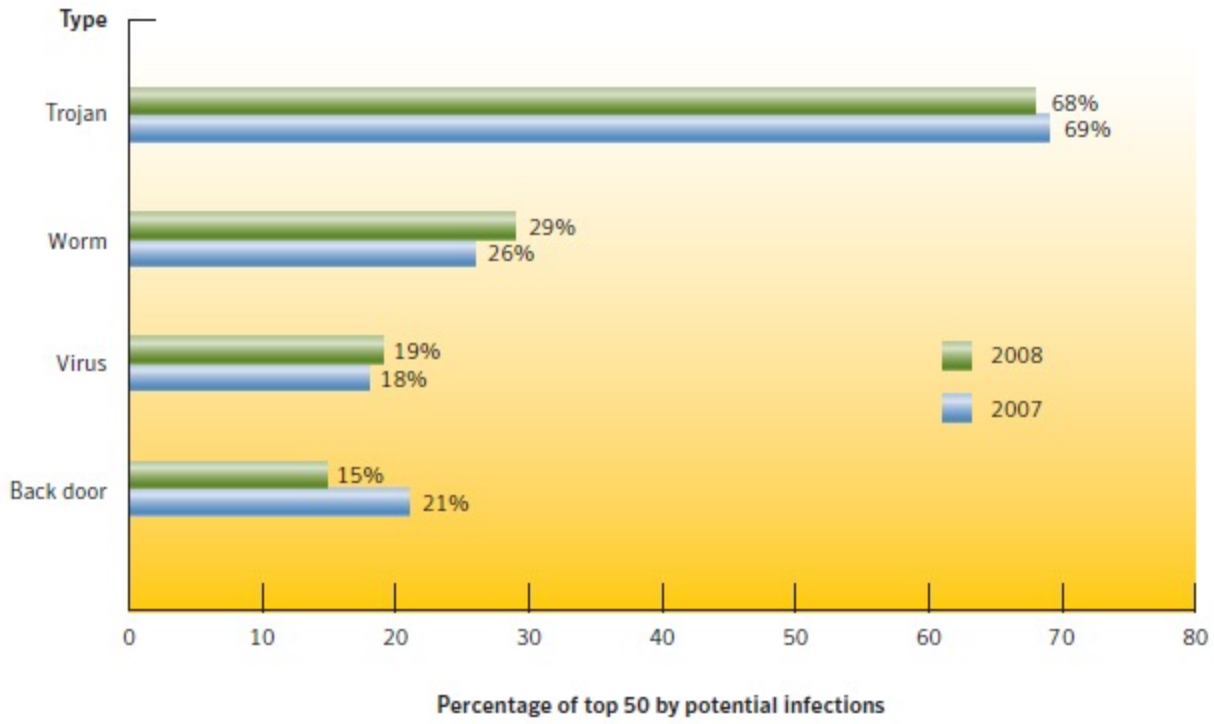
4.3 Malware Attacks -- Trojan Horses

- A **Trojan horse (or Trojan)** is a malware program that appears to perform some useful task, but which also does something with negative consequences (e.g., launches a keylogger).
- Trojan horses can be installed as part of the payload of other malware but are often installed by a user or administrator, either deliberately or accidentally.



Trojan Horse Current Trends

- Trojans currently have 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

4.3 Malware Attacks -- Computer Worms

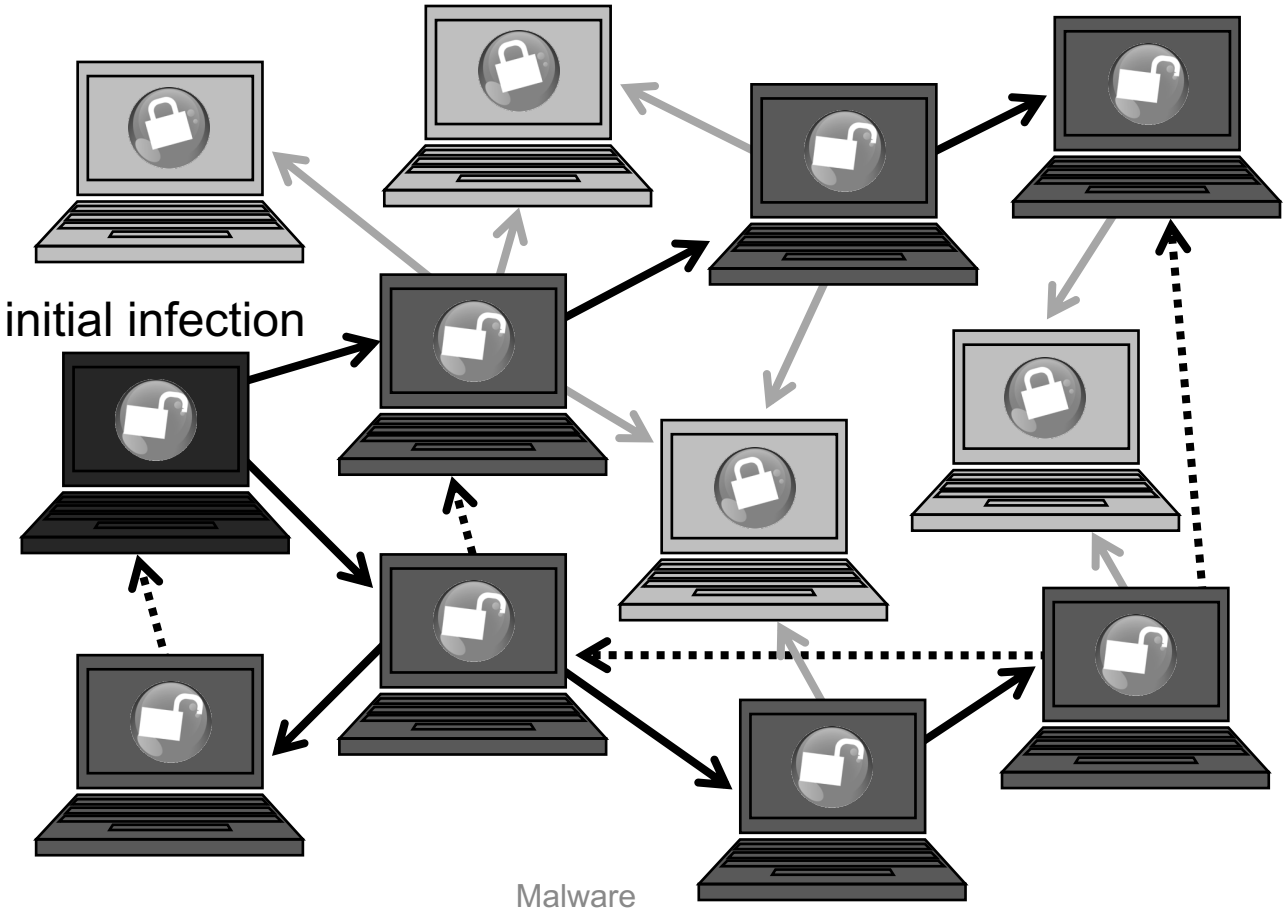
- A **computer worm** is a malware program that spreads copies of itself without the need to inject itself in other programs, and usually without human interaction.
- Thus, computer worms are technically not computer viruses (since they don't infect other programs), but some people nevertheless confuse the terms, since both spread by self-replication.
- In most cases, a computer worm will carry a malicious payload, such as deleting files or installing a backdoor.

Worm Development

- Typically spread by exploiting vulnerabilities (e.g. BoF) in application run by Internet-connected computer systems that have a security hole.
- Write code for
  - Generation of target list
    - Random hosts on the internet
    - Hosts on LAN
    - Divide-and-conquer
  - Installation and execution of payload
  - Querying/reporting if a host is infected
- Distributed graph search algorithm
  - Forward edges: infection
  - Back edges: already infected or not vulnerable

Worm Propagation

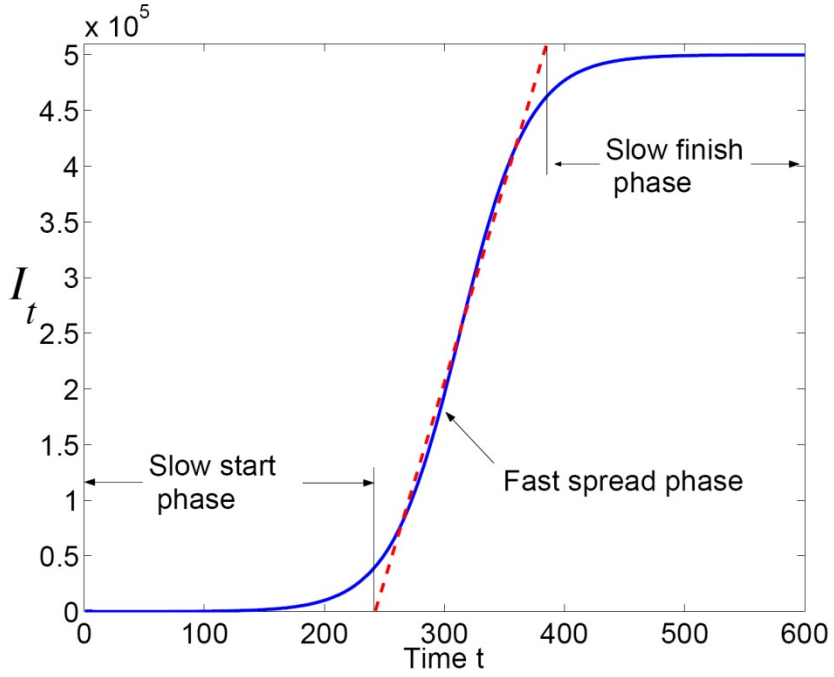
- Worms propagate by finding and infecting vulnerable hosts.
  - They need a way to tell if a host is vulnerable
  - They need a way to tell if a host is already infected.



Propagation: Theory

- Classic epidemic model
  - N: total number of vulnerable hosts
  - $I(t)$: number of infected hosts at time t
  - $S(t)$: number of susceptible hosts at time t
  - $I(t) + S(t) = N$
  - β: infection rate
- $I(0) = 1$
- $S(0) = N - 1$
- $I(t+1) = I(t) + \beta I(t) S(t)$
- $S(t+1) = N - I(t+1)$

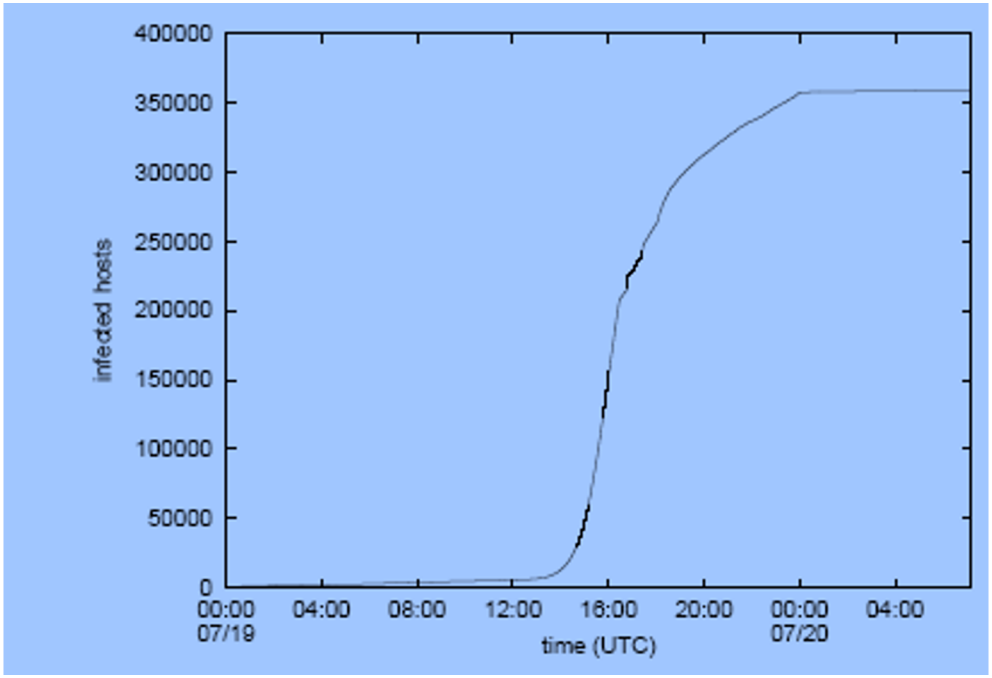
Source:
Cliff C. Zou, Weibo Gong, Don Towsley, and Lixin Gao. The Monitoring and Early Detection of Internet Worms, IEEE/ACM Transactions on Networking, 2005.



Propagation: Practice

- Cumulative total of unique IP addresses infected by the first outbreak of Code-RedI v2 on July 19-20, 2001

The propagation of the worm has three phases: slow start, fast spread, and slow finish.



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

4.3.3 Rootkits

- A rootkit alter system utilities or the OS itself to prevent detection
  - Infect Windows process monitor utility, which list current running processes
  - Infect utilities that allow the user to browse files, such as Windows Explore to hide files on disk
  - Rootkits are often used to hide the malicious actions of other types of malware such as Trojan Horse
- User-mode rootkits
  - Alter system utilities or libraries on disk
  - Insert code to another user-mode process's address space to alter its behavior, such as DLL injection
- Kernel mode rootkits
  - Loaded as device drivers which allows user to easily install drivers for keyboard, audio or video devices.
  - Function hooking to achieve stealth. Modify kernel memory to replace OS functions with customized versions.

4.3.3 Rootkits

- Detecting Rootkit
  - Two scans of file system
  - High-level scan using the Windows API
  - Raw scan using disk access methods
  - Discrepancy reveals presence of rootkit
  - Could be defeated by rootkit that intercepts and modifies results of raw scan operations

4.3.4 Zero-day attacks

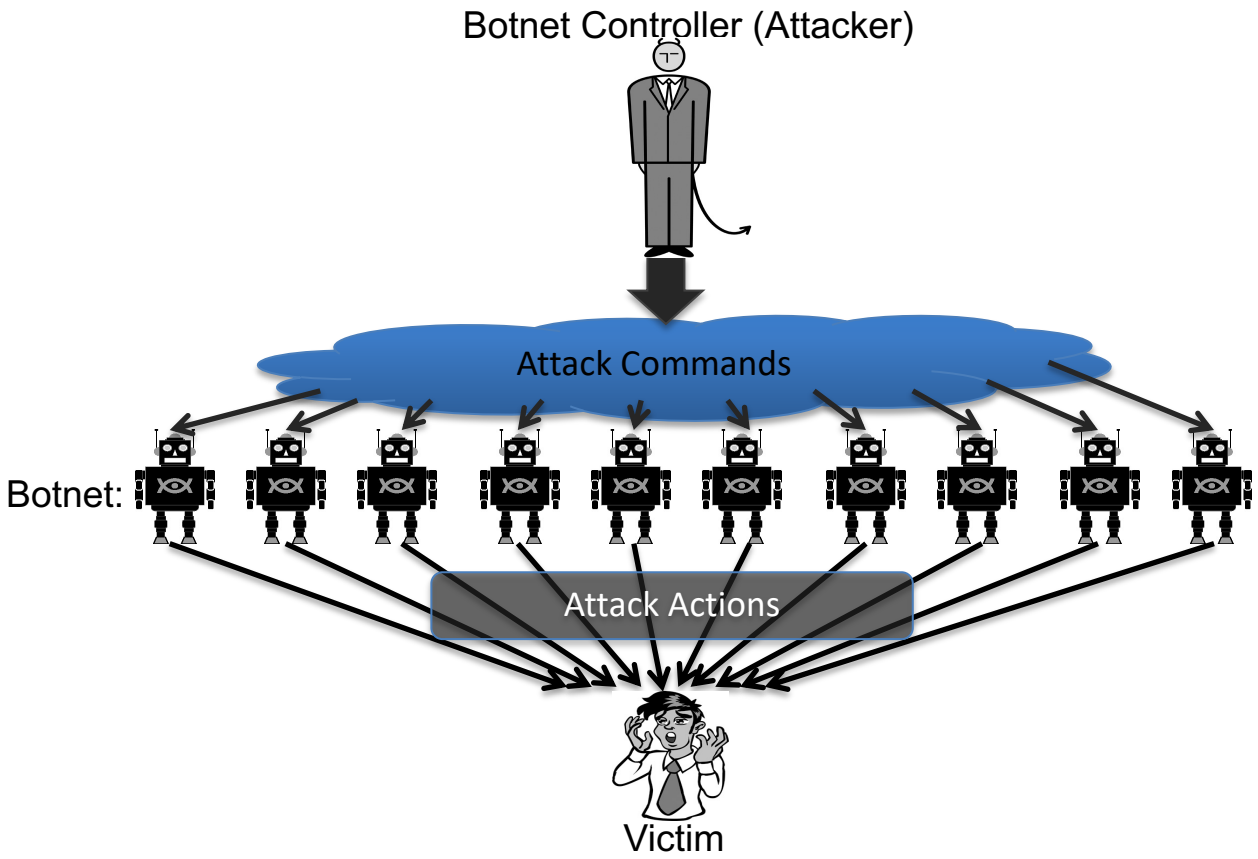
- A zero-day attack is an attack that exploits a vulnerability that was previously unknown, even to the software designers who create the system containing this vulnerability.
- Which means a malware attack exploits a vulnerability that the developers did not know about.

4.3.4 Zero-day attacks -- Heuristic Analysis

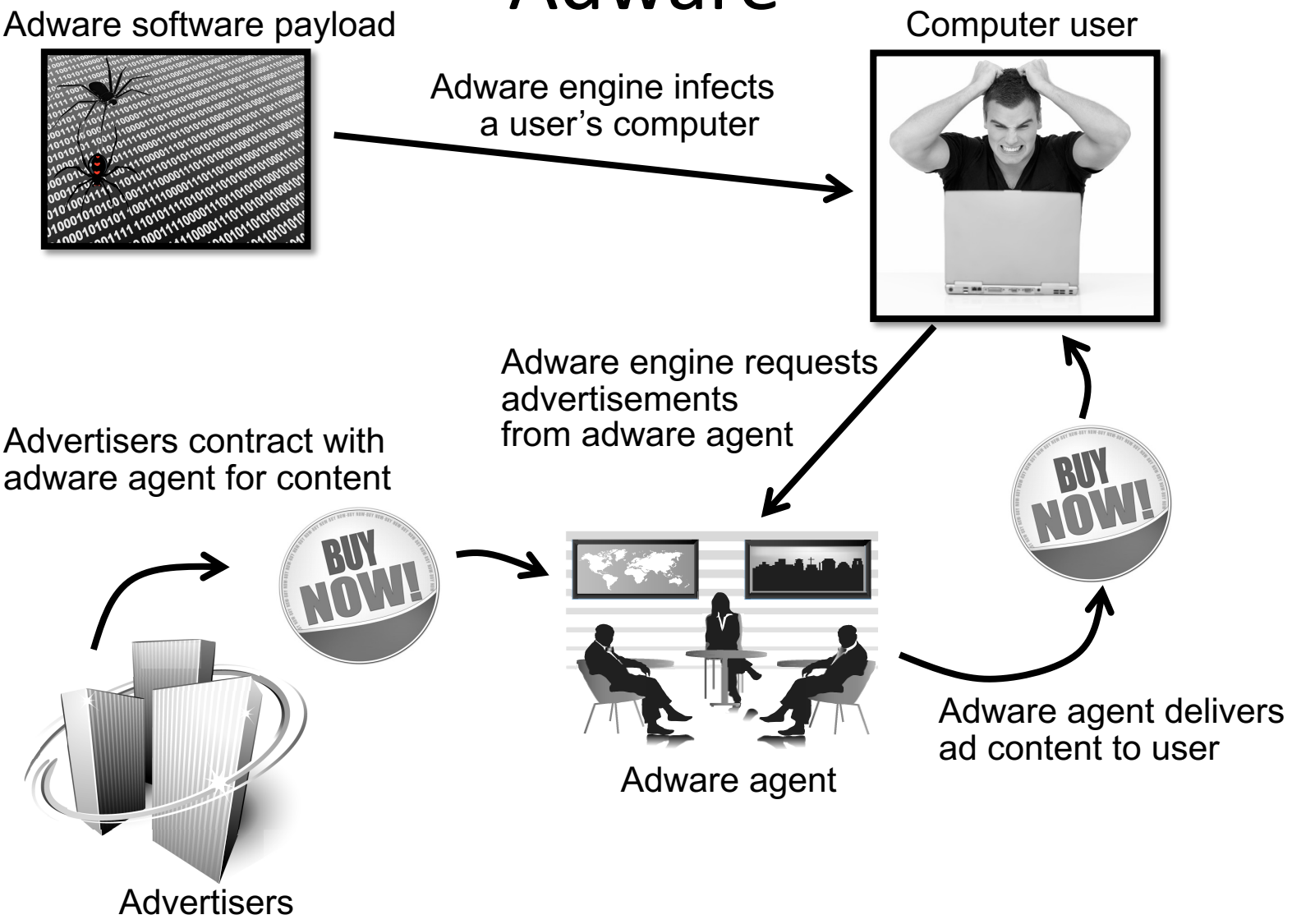
- Useful to identify new and “zero day” malware
- Code analysis
  - Based on the instructions, the antivirus can determine whether or not the program is malicious, i.e., program contains instruction to delete system files,
- Execution emulation
  - Run code in isolated emulation environment
  - Monitor actions that target file takes
  - If the actions are harmful, mark as virus
- Heuristic methods can trigger false alarms

4.3.5 Bontnets

- Malware can turn a computer in to a **zombie**, which is a machine that is controlled externally to perform malicious attacks, usually as a part of a **botnet**.

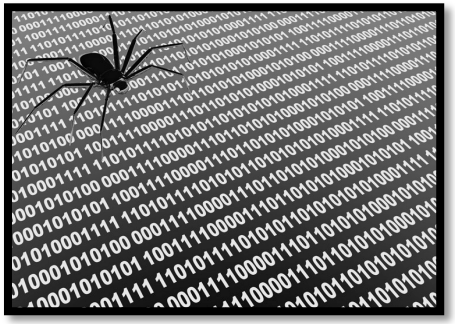


4.4 Privacy-Invasive Software -- Adware



4.4 Privacy-Invasive Software -- Spyware

Spyware software payload



1. Spyware engine infects a user's computer.

Computer user

2. Spyware process collects keystrokes, passwords, and screen captures.

3. Spyware process periodically sends collected data to spyware data collection agent.

Spyware data collection agent

Countermeasures

Best Practices

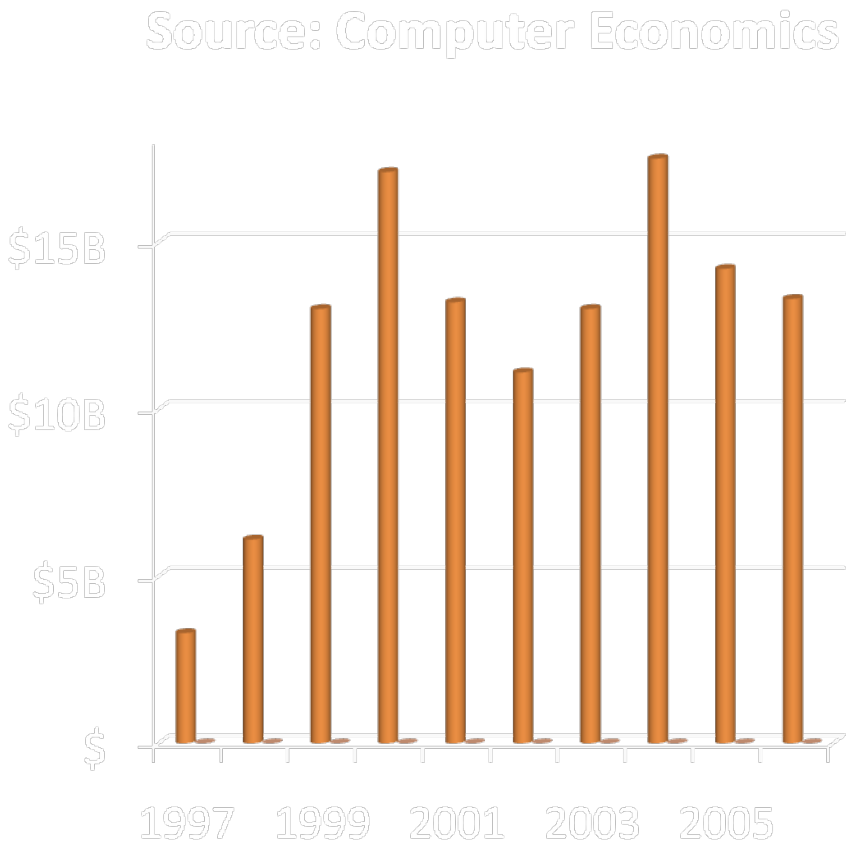
- Employ system diversity as much as possible. This will limit damage from software-specific vulnerabilities.
- Try to limit software installations to systems.
- Turn off auto-execution.
- Employ a principle of least privilege for sensitive systems and data paths.

Undecidable Detection Problems

- Detection of a virus
  - by its appearance
  - by its behavior
- Detection of an evolution of a known virus
- Detection of a triggering mechanism
  - by its appearance
  - by its behavior
- Detection of a virus detector
  - by its appearance
  - by its behavior
- Detection of an evolution of
  - a known virus
  - a known triggering mechanism
  - a virus detector

4.5.4 Economics of Malware -- Financial Impact

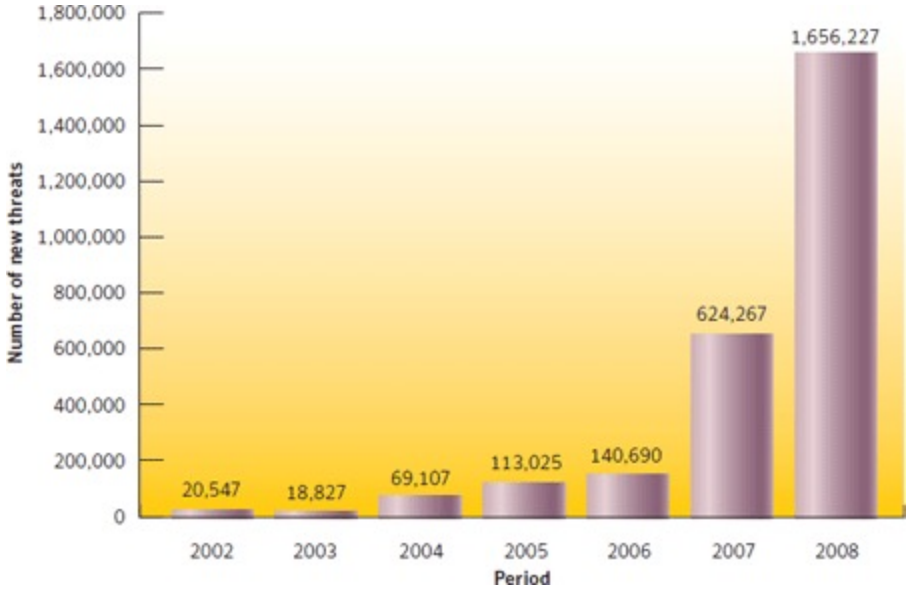
- Malware often affects a large user population
- Significant financial impact, though estimates vary widely, up to $100B per year (mi2g)
- Examples
  - LoveBug (2000) caused $8.75B in damages and shut down the British parliament
  - In 2004, 8% of emails infected by W32/MyDoom.A at its peak
  - In February 2006, the Russian Stock Exchange was taken down by a virus.



Economics of Malware

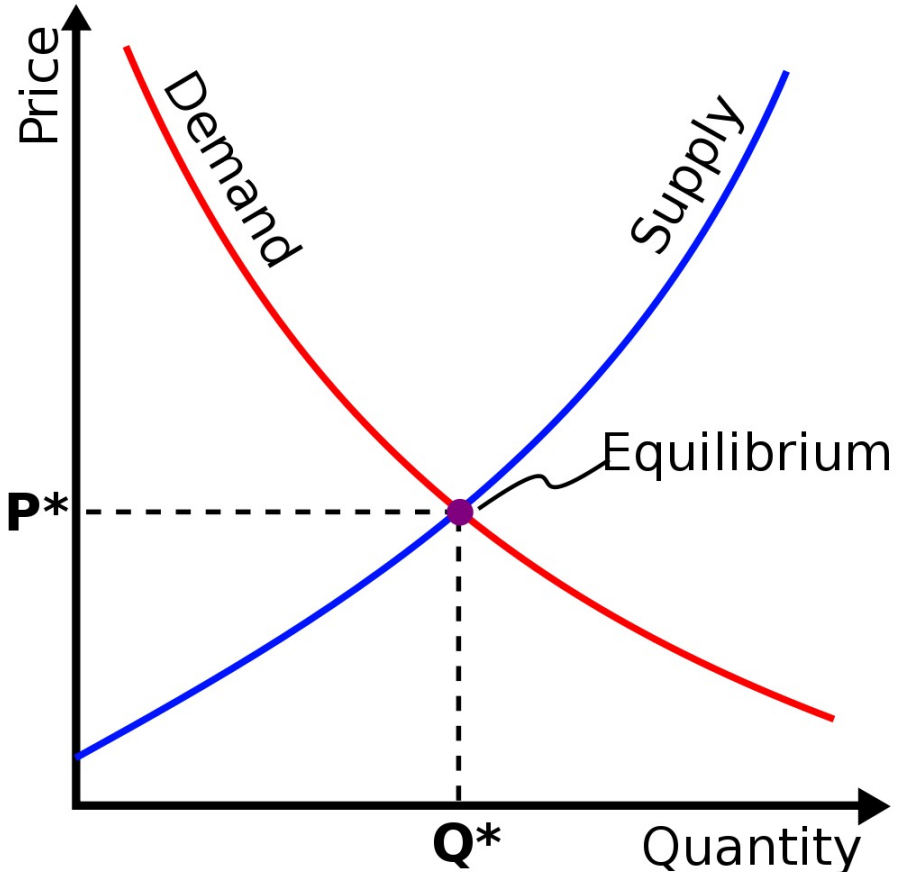
- New malware threats have grown from 20K to 1.7M in the period 2002-2008
- Most of the growth has been from 2006 to 2008
- Number of new threats per year appears to be growing an exponential rate.

Source: Symantec Internet Security Threat Report, April 2009



Professional Malware

- Growth in professional cybercrime and online fraud has led to demand for professionally developed malware
- New malware is often a custom-designed variations of known exploits, so the malware designer can sell different "products" to his/her customers.
- Like every product, professional malware is subject to the laws of supply and demand.
  - Recent studies put the price of a software keystroke logger at $23 and a botnet use at $225.



Resources

- Computer Emergency Response Team
  - Research center funded by the US federal government
  - Vulnerabilities database
- Symantec
  - Reports on malware trends
  - Database of malware
- Art of Computer Virus Research and Defense by Peter Szor