Malicious Software

Malicious Software

- Also called malware.
- It is software used or created to **disrupt computer operation, gather sensitive information, or gain access to private computer systems**.
- It can appear in the form of code, scripts, active content, and other software.

Malware

- 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.”

- Programs exploiting system vulnerabilities known as malicious software or malware
  - program fragments that need a host program
    - e.g. viruses, logic bombs, and backdoors
  - independent self-contained programs
    - e.g. worms, bots
  - replicating or not
- Sophisticated threat to computer systems

Usage of Malware

- Malware is used primarily to steal sensitive personal, financial, or business information for the benefit of others.
- Also sometimes used broadly against government or corporate websites to gather guarded information, or to disrupt their operation in general.

- To partially control the user’s computer, for reasons such as:
  - To subject the user to advertising
  - To launch DDoS on another service
  - To spread spam
  - To track the user’s activity (“spyware”)
  - To commit fraud, such as identity theft and affiliate fraud

Ways of Spread of malware

- **Drive-by download**
  - unintended download of computer software from the Internet
- **Homogeneity**
- **Vulnerability**
  - A security defect in software that can be attacked by a malware.
- **Backdoor:**

Types of Malware Attacks

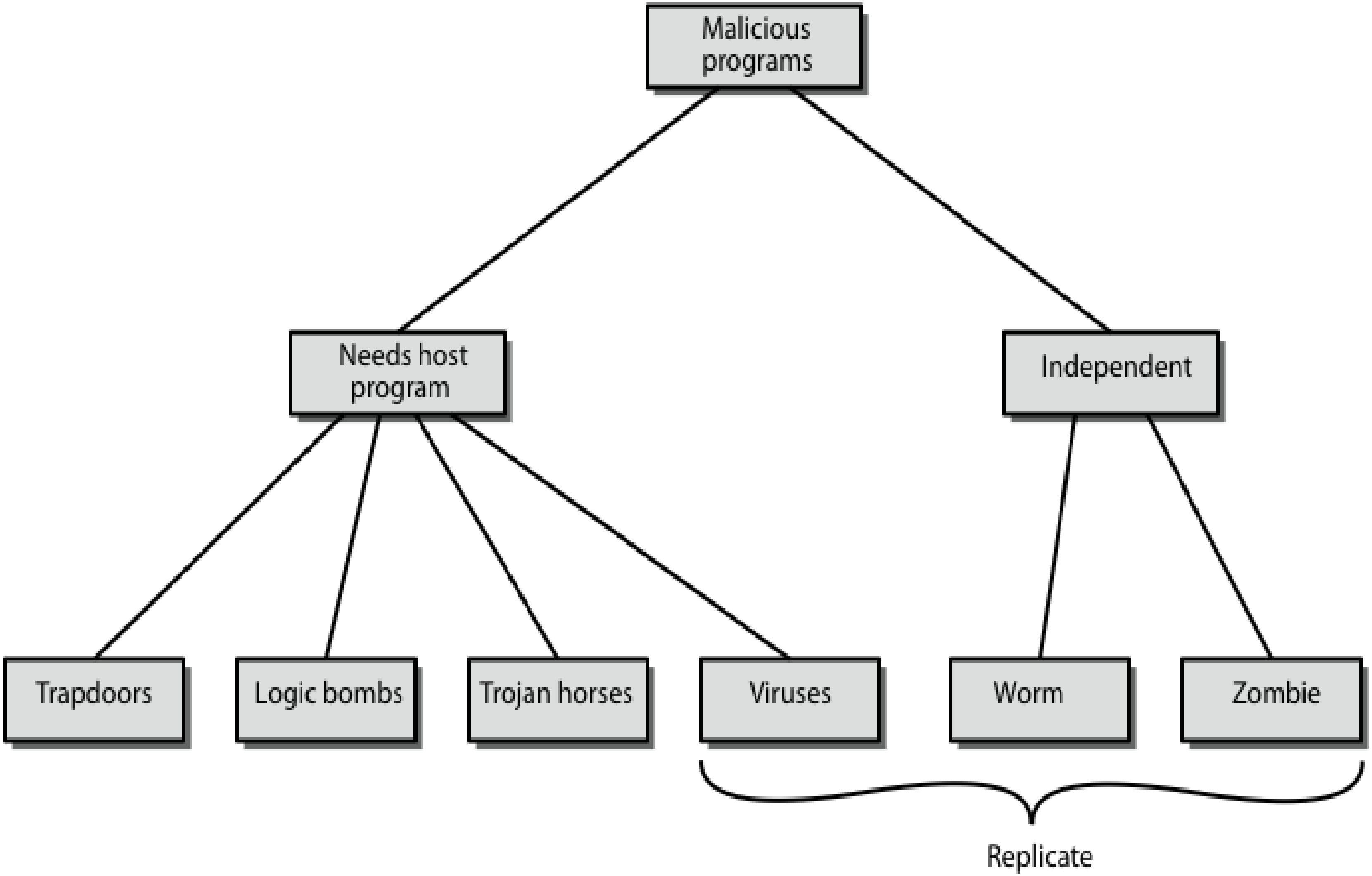
- **0-Day:** A zero-day vulnerability is an undisclosed flaw that hackers can exploit.
- **Exploit:** A threat made real via a successful attack on an existing vulnerability.
- **Privilege escalation:** Situation where the attacker gets escalated access to restricted data that is on a higher level of security.
- **Evasion:** The techniques malware maker design to avoid detection and analysis of their malware by security systems and software.

- **Blended threat:** A malware package that combines the characteristics of multiple types of malware

Types of Malware

- Viruses
- Trojan horses
- Worms
- Spyware
- Zombie
- Phishing
- Spam
- Adware
- Ransomware

Taxonomy of Malicious Programs



Some terms

- Payload: The part of the malware program that actually does the damage.
- Crimeware: kits for building malware; include propagation and payload mechanisms
  - Zeus, Sakura, Blackhole, Phoenix
- APT (advanced persistent threats)
  - Advanced: sophisticated
  - Persistent: attack over an extended period of time
  - Threat: selected targets (capable, well-funded attackers)

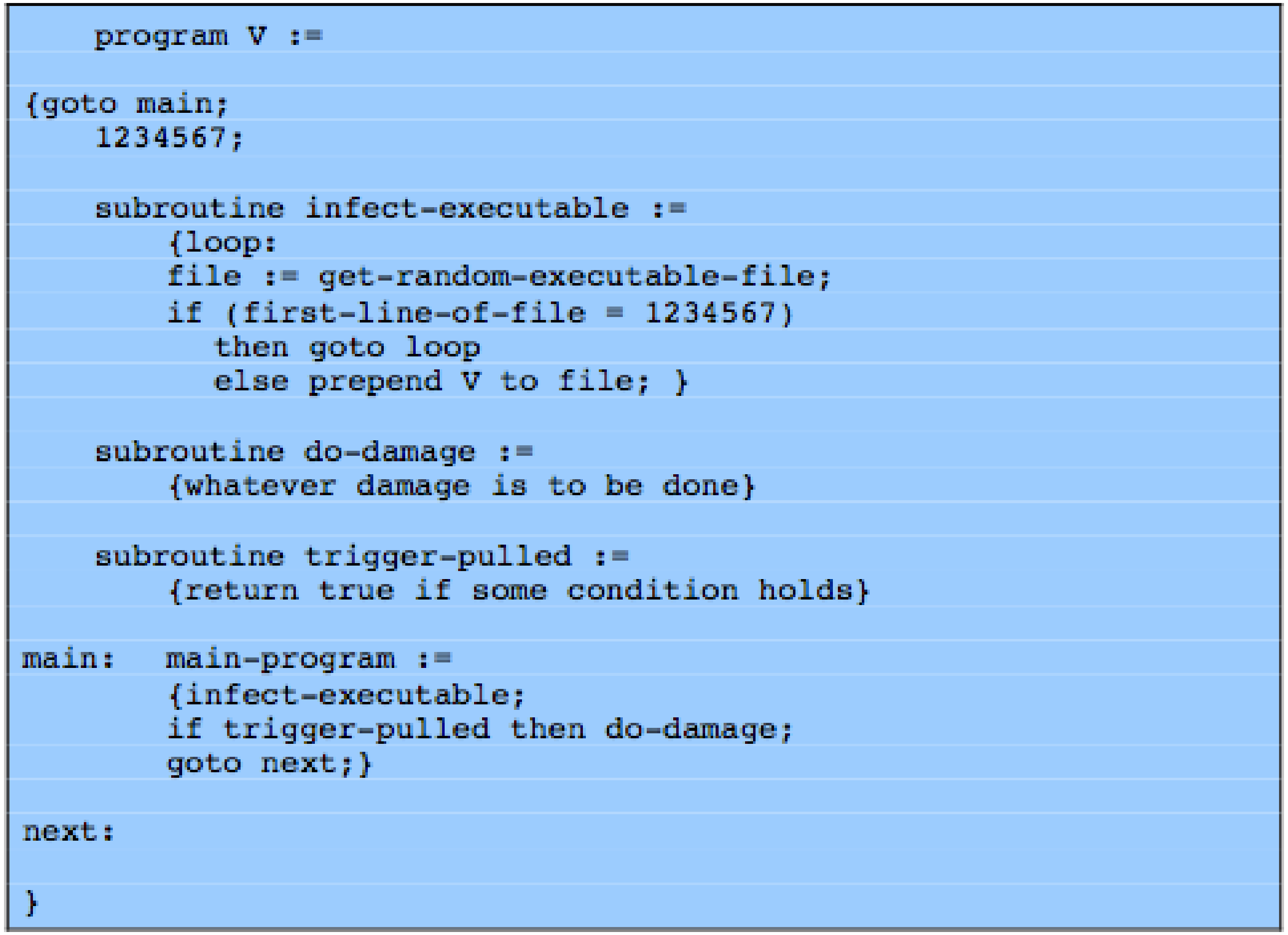
- **Signature:** Signs that are specific to either a certain type of behavior or a specific item of malware.
- **Privilege:** In computing, privilege means the access to modify a system.

Viruses

- Piece of software that infects programs
  - modifying them to include a copy of the virus
  - so it executes secretly when host program is run
- Specific to operating system and hardware
  - taking advantage of their details and weaknesses
- A typical virus goes through phases of:
  - dormant: *idle*
  - propagation: *copies itself to other program*
  - triggering: *activated to perform functions*
  - execution: *the function is performed*

Virus structure

- Components:
  - infection mechanism: enables replication
  - trigger: event that makes payload activate
  - payload: what it does, malicious or benign
- When infected program invoked, executes virus code then original program code
- Can block initial infection (difficult) or propagation (with access controls)



```
    program V :=

{goto main;
    1234567;

    subroutine infect-executable :=
        {loop:
        file := get-random-executable-file;
        if (first-line-of-file = 1234567)
           then goto loop
           else prepend V to file; }

    subroutine do-damage :=
        {whatever damage is to be done}

    subroutine trigger-pulled :=
        {return true if some condition holds}

main:   main-program :=
        {infect-executable;
        if trigger-pulled then do-damage;
        goto next;}

next:

}
```

Types of Viruses

- **Parasitic Virus** - attaches itself to executable files as part of their code. Runs whenever the host program runs.
- **Memory-resident Virus** - Lodges in main memory as part of the residual operating system.
- **Boot Sector Virus** - infects the boot sector of a disk, and spreads when the operating system boots up.

- **Stealth Virus** - explicitly designed to hide from Virus Scanning programs.
- **Polymorphic Virus** - mutates with every new host to prevent signature detection.
- **Metamorphic virus** - mutates with every infection, but rewrites itself completely every time.

Propagation Techniques

- Removable Storage
  - Boot sector viruses, executable viruses
    - Yamaha’s CD-R drive firmware update contained the Chernobyl virus.
- Email attachments
- Shared directories

Viruses: The Principle

- Virus attaches itself to a host that can execute instructions contained in the virus.
- When the host is invoked, the virus copies itself to other locations on the system.

Companion Infection Technique

- OS will call the virus when the user requests the companion file.

  Windows:

  - Virus is Notepad.com to hide as Notepad.exe.
  - Set the hidden attribute to prevent the virus from being seen.
  - Launch the true notebook.exe file from the virus.
  - If the user selects Start → Run and types in notebook, then windows starts the virus (notebook.com instead of notebook.exe)

Companion Infection Technique

- – Virus renames Notepad.exe to Notepad.ex_ and hides it.
- – Virus takes the place of Notepad.exe.
- – Works with shortcuts.
- – Used in the Trilisa virus / worm (2002)

Companion Infection Technique

- Virus uses alternate data stream feature of NTFS:
  - Streams look like one file in explorer and directory listings.
  - System activates the default stream, the virus.
  - Virus calls alternate stream.
  - Win2KStream Virus (2000)

Overwriting Techniques

- Virus replaces part of an executable.
- Usually the executable looses functionality.
- Users will now that there is something wrong.

Prepending Techniques

- Virus placed in front of executable.
- After virus executes, host program is called.
- Very easy for .com files.
- Easy to clean files.
  - Bliss virus had a disinfect mode built into it.
- Used by the NIMDA worm.

Appending Infection Technique

- Insert itself at the end of host file.
- Add a jump at the beginning of host file.

Stealth Techniques for Prepending and Appending

- Compress host.
- When virus calls hosts, host is uncompressed into RAM.
- Fill up total package (virus, compressed host) to same size as original host.
- Change filler so that checksum is not changed.

Macro viruses

- Became very common in mid-1990s since
  - platform independent
  - infect documents
  - easily spread
- Microsoft Office applications allow “macros” to be part of the document. The macro could run whenever the document is opened, or when a certain command is selected (Save File).
- More recent releases include protection
- Recognized by many anti-virus programs

E-Mail Viruses

- More recent development
- Melissa
  - exploits MS Word macro in attached doc
  - if attachment opened, macro activates
  - sends email to all on users address list and does local damage

Examples of computer viruses are:

– Macro virus

– Boot virus

– Logic Bomb virus

– Directory virus

– Resident virus

Virus countermeasures

- Prevention: ideal solution but difficult
- Realistically need:
  - detection: determine what occurred
  - identification: identify the specific virus
  - removal: remove all traces
- If detected but can’t identify or remove, must discard and replace infected program

Anti-virus evolution

- Virus & antivirus tech have both evolved
- Early viruses simple code, easily removed
- As viruses become more complex, so did the countermeasures
- Generations
  - first - signature scanners (bit patterns all the same)
  - second – heuristics (integrity checks; checksums)
  - third - identify actions (find by actions they do)
  - fourth - combination packages

Anti-Virus Defense

- Antivirus software on gateways:
  - User workstations
  - File servers
  - Mail servers
  - Application servers
  - Border firewalls

Anti-Virus Defense

- Virus signatures
  - Looks for small patterns indicative of a known virus.
    - Polymorphic viruses
- Heuristics
  - Looks for programs with bad behavior:
    - Attempts to access the boot sector
    - Attempts to locate all files in a directory
    - Attempts to write to an exe file
    - Attempts to delete hard drive contents

Anti-Virus Defense

- Integrity Verification
  - Generate database of hashes of important files.
  - Recalculate these hashes and compare them to known values.
- Configuration Hardening
  - Least privilege
  - Minimize active components.
  - Set warnings (e.g. against macros)
  - User education

Self-modification

- *Self-modification*, which defeats *signature scanners*, because each infected file contains a different variant of the virus.
  - *Simple self-modifications*: exchanging subroutines in the code.
  - *Encryption*: virus consists of a small decrypting module and an encrypted copy of the virus code. Key randomly generated each time.

- *Polymorphic code*: the decryption module is also modified on each infection. No parts stay the same. Detection (by using an emulator, or by statistical pattern analysis of virus body) much more difficult.
- *Metamorphic code*: virus rewrites itself completely each time. A *metamorphic engine* is needed. Virus very large and complex. W32/Simile >14000 loc; 90% of it is the metamorphic engine.

Worms

Worms

Worms:

- Propagates across a network
- Typically, does not require user action for propagation.

Virus:

- Infects files.
- Typically requires user interaction.

Taxonomy of Computer Worms

- Taxonomy based on:
  - target discovery
  - carrier
  - activation
  - payloads
  - attackers

Target Discovery

- the mechanism by which a worm discovers new targets to infect
- How do worms propagate
  - Scanning worms
  - Meta-server worm
    - Ask server for hosts to infect
  - Topological worm
    - Use information from infected hosts
  - Contagion worm
    - Propagate parasitically along with normally initiated communication

Scanning worms

- Scanning : entails probing a set of addresses to identify vulnerable hosts. ⍰
  - sequential : working through a IP address block using an ordered set of addresses ⍰
  - random : trying address out of a block in a pseudo-random fashion ⍰

Code-Red, Nimda, Slammer Worm

Hit-list Scanning

- Provide the worm with a list of potentially vulnerable machines. ⍰
- The worm, when released onto an initial machine on this hit-list, begins scanning down the list. ⍰
- When it infects a machine, it divides the hitlist in half, communicating half to the recipient worm, keeping the other half.

- Random scanning is inefficient : ⍰
  - many addresses are probed multiple times
  - ⍰no means for a randomly scanning worm to effectively determine when all vulnerable machines are infected
- Permutation scanning:
  - a worm can detect that a particular target is already infected ⍰
  - all worms share a common pseudo random permutation of the IP address space

Spread of Scanning Worms

- The speed of scanning worms is limited by:
  - ⍰Density of vulnerable machines ⍰
  - Design of the scanner ⍰
  - The ability of edge routers to handle a potentially significant increase in new, diverse communication
- Scanning is highly anomalous behavior. ⍰
- Effective detection: defenses designed to stop an entire family of worms

Topological Worms : Internal Target Lists

- Many applications contain information about other hosts providing vulnerable services.
- Topological worm searches for local information to find new victims by trying to discover the local communication topology
  - The original Morris worm used topological techniques including Network Yellow pages, /etc/hosts, and other sources to find new victims.

Topological Worms

- The spread is slower as compared to scanning worms. ⍰
- Can bypass defenses by communicating information known by one instance to other instances. ⍰
- May present a global anomaly, the local traffic may appear normal. ⍰
- Highly distributed sensors may be needed to detect topological worms

Target Discovery : Passive Worms

- A passive worm does not seek out victim machines.
- Instead, it either waits for potential victims to contact the worm or rely on user behavior to discover new targets
- Example: Gnuman, CRclean

Worm Components

- Warhead
- Propagation Engine
- Target Selection Algorithm
- Scanning Engine
- Payload

Worm Warhead

- A piece of code that exploits a vulnerability on the target system
  - Exploits such as Buffer Overflow Exploits
  - File Sharing Attacks
  - E-mail
  - Common Misconfigurations

Worm Propagation Engine

- After gaining access, the worm must transfer itself to the target machine.
- Some worms are completely contained in the warhead.
- File Transfer Mechanisms
  - FTP
  - TFTP
  - HTTP
  - SMB (MS Server Message Block)
    - Windows file sharing
    - Unix servers running SAMBA

Worm Target Selection Algorithm

- Once the worm has gained control of a target, it starts looking for new targets.
  - E-mail addresses
  - Host lists
  - Trusted Systems
  - Network Neighborhood
  - DNS queries
  - Randomly selected IP address.

Worm Scanning Engine

- Once targets are identified, the worm scans for the original vulnerability.

Worm Payload

- Some specific action done on behalf of the attacker.
- Opening up a backdoor.
- Planting a distributed denial of service attack.
- Performing complex calculations:
  - password cracking
  - math research (actually happened)

Worm Spread

- Worm spread is limited
  - Diversity of machines
    - Tiny worm
      - targeted only machines running security software from a medium company
      - was successful in infecting most machines.
    - Worms can contain support for multiple entry methods.
  - Too many victims crash
  - Fast worms can cause network congestion

Worm Trends

- Multiplatform worms
- Multiexploit worms
- Zero-day exploit worms
  - No chance to patch
- Fast-spreading worms: Warhol / Flash
  - pre-scan targets
- Polymorphic worms
  - Change appearance
- Metamorphic worms
  - Change functionality

Worm Defenses

- Antivirus tools
- Fast patching services
- Firewalling
  - Block arbitrarily outbound connections
  - Prevents spreading
- Establishment of Incident Response Capabilities

- Worm defense approaches include:
  - signature-based worm scan filtering: define signatures
  - filter-based worm containment (focus on contents)
  - payload-classification-based worm containment (examine packets for anomalies)
  - threshold random walk scan detection (limit the rate of scan-like traffic)
  - rate limiting and rate halting (limit outgoing traffic when a threshold is met)

Email Worm

- Email worm goes into a user’s contact/address book and chooses every user in that contact list.
- It then copies itself and puts itself into an attachment; then the user will open the attachment and the process will start over again

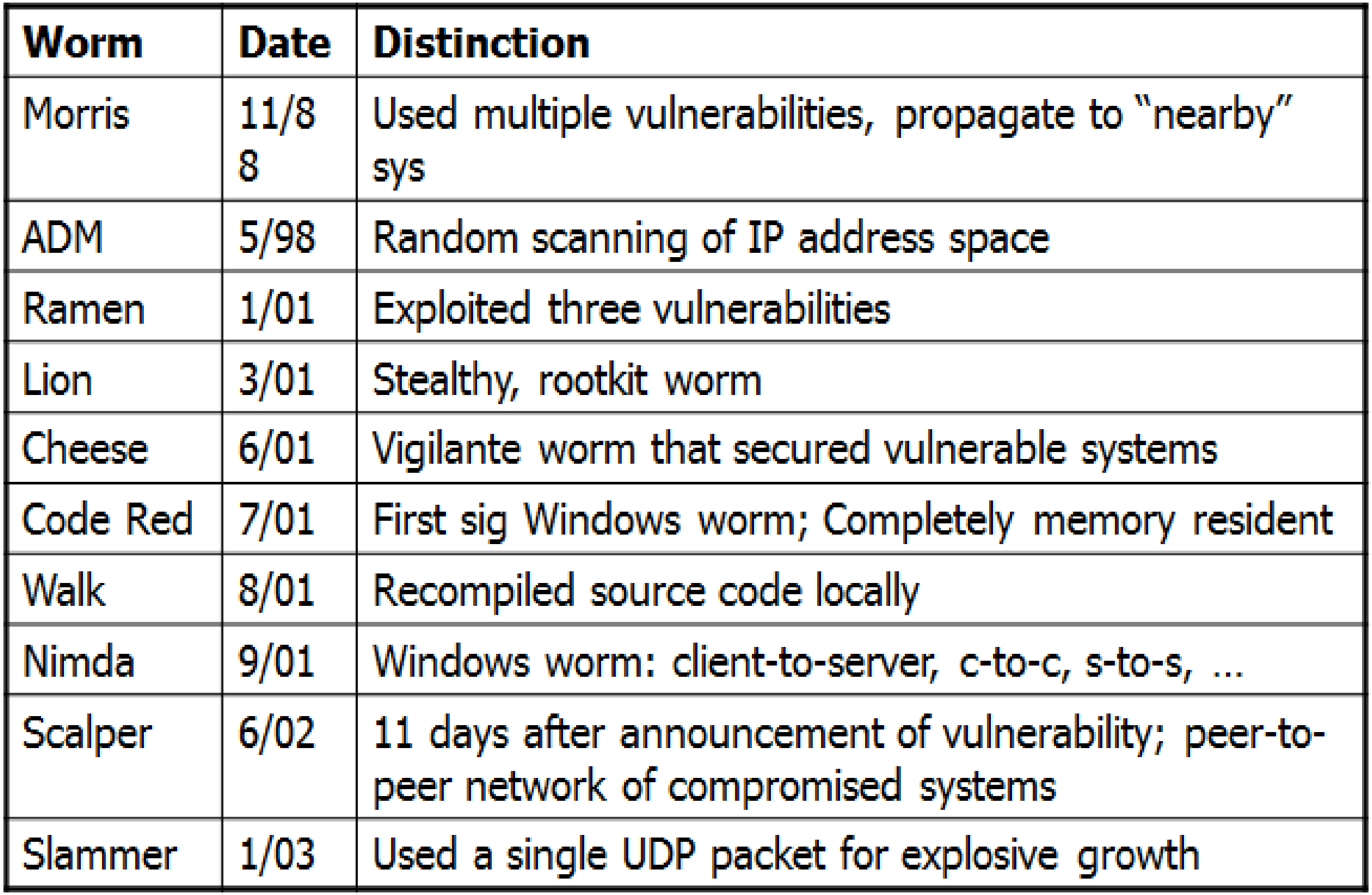
Internet Worms

- A internet worm is designed to be conspicuous to the user.
- The worms scans the computer for open internet ports that the worm can download itself into the computer.
- Once inside the computer the worms scans the internet to infect more computers.

The Morris Worm

- One of best know worms
- Released by Robert Morris in 1988
  - Affected 6,000 computers; cost $10-$100 M
- Various attacks on UNIX systems
  - cracking password file to use login/password to logon to other systems
  - exploiting a bug in the finger protocol
  - exploiting a bug in sendmail
- If succeed have remote shell access
  - sent bootstrap program to copy worm over

Some historical worms of note



Worm	Date	Distinction
Morris	11/88	Used multiple vulnerabilities, propagate to "nearby" sys
ADM	5/98	Random scanning of IP address space
Ramen	1/01	Exploited three vulnerabilities
Lion	3/01	Stealthy, rootkit worm
Cheese	6/01	Vigilante worm that secured vulnerable systems
Code Red	7/01	First sig Windows worm; Completely memory resident
Walk	8/01	Recompiled source code locally
Nimda	9/01	Windows worm: client-to-server, c-to-c, s-to-s, …
Scalper	6/02	11 days after announcement of vulnerability; peer-to-peer network of compromised systems
Slammer	1/03	Used a single UDP packet for explosive growth

Trojan Horse

Trojan Horse

- **Trojan horse** is a malicious program that is disguised as legitimate software.
- Trojan horse programs cannot replicate themselves, in contrast to some other types of malware, like viruses or worms.
- A Trojan horse can be deliberately attached to useful software by a cracker, or it can be spread by tricking users into believing that it is a useful program.

Types of attack

- Credit Card Information (often used for domain registration, shopping with your credit card)
- Any accounting data (E-mail passwords, Dial-Up passwords, Web Services passwords)
- Email Addresses (Might be used for spamming)
- Work Projects (Steal your presentations and work related papers)

Types of Trojans

- Erasing or overwriting data on a computer, corrupting files in a subtle way
- Spreading other malware, such as viruses (dropper)
- Setting up networks of zombie computers in order to launch DDoS attacks or send Spam.
- logging keystrokes to steal information such as passwords and credit card numbers (key logger)
- Phish for bank or other account details, which can be used for criminal activities.
- Installing a backdoor on a computer system.

Propagation Methods

- By visiting a rogue websites
- Lack of security in instant message applications
- Attachments on e-mail messages may contain Trojans

Well Know Trojans

- AceBot Trojan
- Secup Trojan
- Dmsys

Trojan Horses Defenses

- Tripwire could find substitutes for executables.
- Filter email attachments that are executable.

Rootkits

Rootkit

- A **rootkit** is a tool that is designed to hide itself and other processes, data, and/or activity on a system

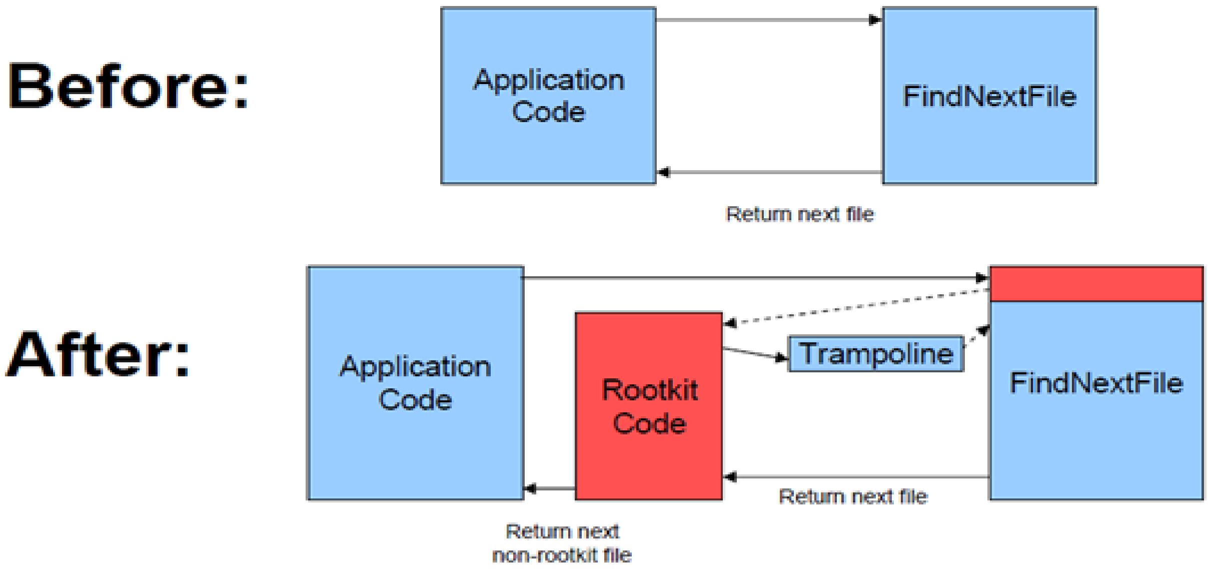
➢ An attacker wants to:

  - Survive system restart
  - Hide processes
  - Hide services
  - Hide listening TCP/UDP ports
  - Hide kernel modules
  - Hide drivers

Rootkits

- Rootkits are Trojan horse backdoor tools that modify existing operating system software so that an attacker can keep access to and hide on a machine.
- Installing a Rootkit on a Target System
  - Hacker have root level access on target system
  - Gain root level access by compromising system via buffer overflow, password attack, social engineering
  - Rootkit allows hacker to get back onto system with root level privilege

How Rootkit Works



- Overwrite first few bytes of target function with a jump to rootkit code
- Create "trampoline" function that first executes overwritten bytes from original functio then jumps back to original function
- When function is called, rootkit code executes
- Rootkit code calls trampoline, which executes original function

Types of Rootkits

- User-level Rootkits
- Kernel-level Rootkits

User-level rootkits

- Replace utilities such as ps, ls, ifconfig, etc
- Replace key libraries
- Detectable by utilities like tripwire

Kernel-level rootkits

- Replace key kernel functions
- Prevents user processes from accessing critical kernel data structures.
- Kernel Mode Rootkit Capabilities
  - File & Directory Hiding
  - Process Hiding
  - Network Port Hiding
  - Promiscuous Mode Hiding
  - Execution Redirection
  - Device Interception and Control

Rootkit Defenses

- Preventing Root Kits
  - Harden systems and apply patches.
- Detect Root Kits
  - File Integrity Checking (Signatures)
  - Root Kit Identification
    - By Tripwire, Honeypots

Zombies

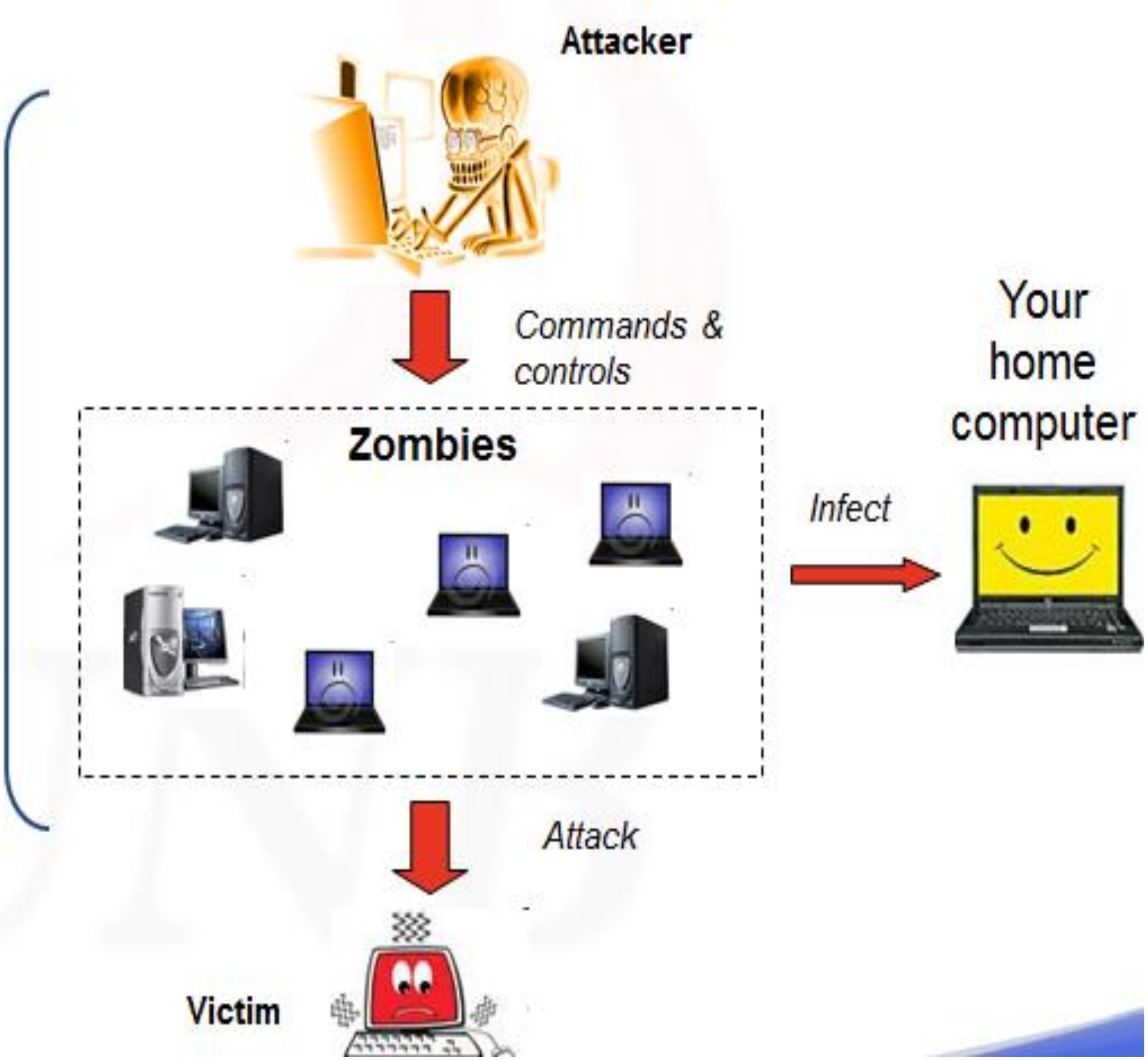
What is Botnets?

- A Botnet is a network of compromised computers under the control of a remote attacker.
- Bot program runs silently in the background, awaiting instructions from the attacker

Botnets consist of:

- Bot herder
  - The attacker controlling the malicious network (also called a Bot master)
- Bot
  - A compromised computers under the Bot herders control (also called zombies, or drones).
- Bot Client
  - The malicious Trojan installed on a compromised machine that connects it to the Botnet.
- Command and Control Channel (C&C)
  - The communication channel the Bot herder uses to remotely control the bots.

Botnet topology mainly refers to the organization of C&C channels between zombies and an attacker.



- Bot master
  - sends viruses, worms, etc. to unprotected PCs
  - Direct attacks on home PC without patches or firewall
  - Indirect attacks via malicious HTML files that exploit vulnerabilities (especially in MS Internet Explorer)
  - Malware attacks on peer-to-peer networks

Bot

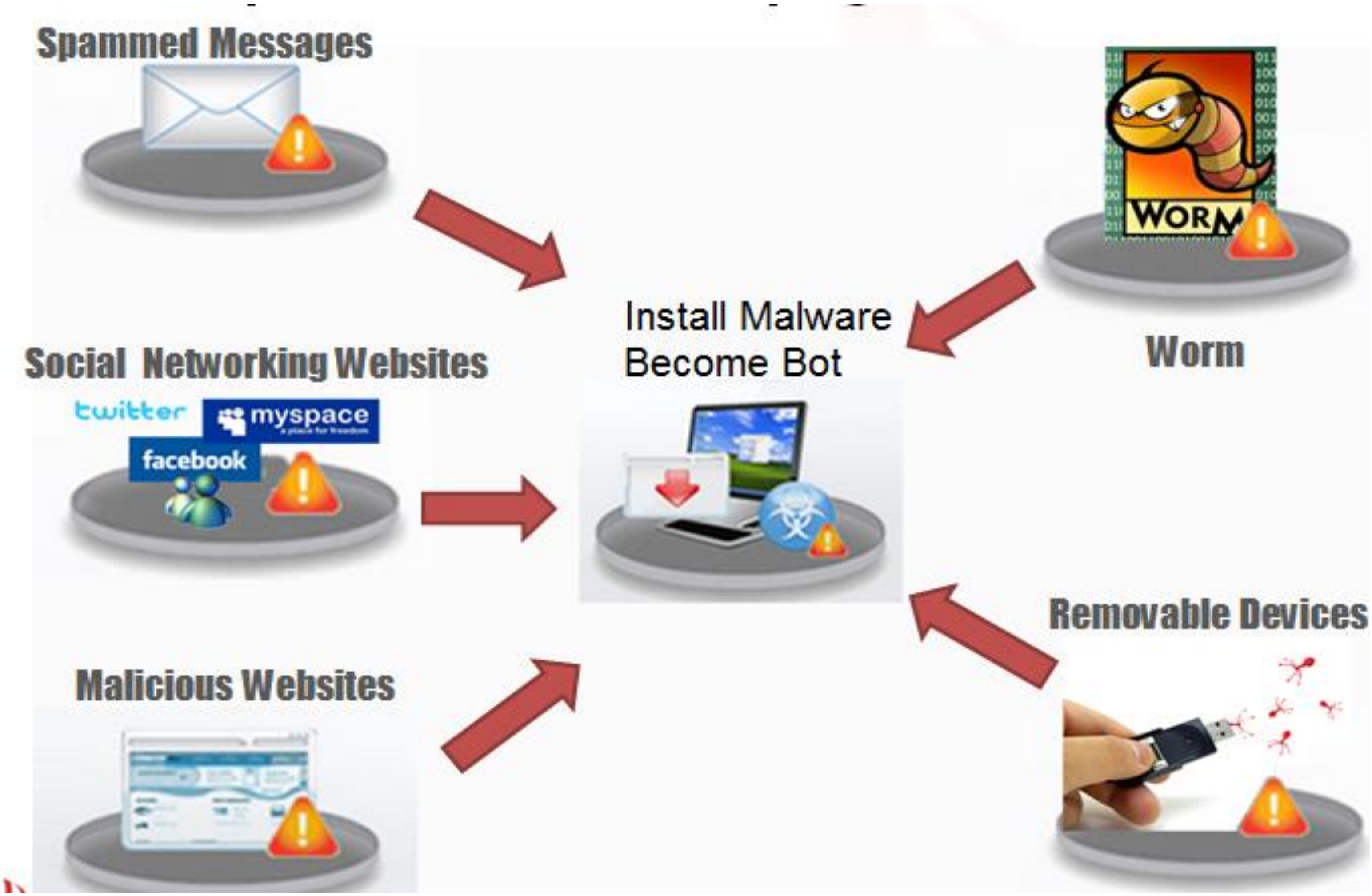
- *Bot - a small program to remotely control a computer*
- Characterized by
  - Remote control & communication (C&C) channels to command a victim
    - *For ex., perform denial-of service attack, send spam*
  - The implemented remote commands
    - *For ex., update bot binary to a new version*
  - The spreading mechanisms to propagate it further
    - *For ex., port scanning, email*

C&C channel

- Means of receiving and sending commands and information between the botmaster and the zombies.
- Protocols imply (to an extend) a botnet's communication topology.
  - The topology provides trades-off in terms of bandwidth, affectivity, stealth

- Today, bot herders primarily rely on these three protocols for their C&C:
  - Internet Relay Chat (IRC) Protocol
  - Hyper-Text Transfer Protocol (HTTP)
  - Peer-to-Peer (P2P) networking protocols.

Popular Botnets Propagation Methods



Detection

- Complicated organization of botnets & variety of cover-up techniques make detection of botnets challenging
- Botnet malware use encryption techniques to avoid being detected by **signature-based** Intrusion detection system