Introduction to Computer Security

Chapter 6: Malicious Software

Chi-Yu Li (2019 Spring)
Computer Science Department
National Chiao Tung University

Malware

- NIST [SOUP13] defines malware as:

> "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."

[SOUP13] Souppaya, M., and Scarfone, K. *Guide to Malware Incident Prevention and Handling for Desktops and Laptops*. NIST Special Publication SP 800-83, July 2013.

Outline

- Types of Malicious Software (Malware)
- Advanced Persistent Threat
- Propagation
  - Infected Content – Viruses
  - Vulnerability Exploit – Worms
  - Social Engineering – Spam E-Mail, Trojans
- Payload
  - System Corruption
  - Attack Agent – Zombie, Bots
  - Information Theft – Keyloggers, Phishing, Spyware
  - Stealthing – Backdoors, Rootkits
- Countermeasures

Broad Classification of Malicious Software (Malware)

- Propagation: how it spreads or propagates to reach the desired targets
  - Infection of existing content by viruses
    - Subsequently spread to other systems
  - Exploit of software vulnerabilities by worms or drive-by-downloads
    - Allow the malware to replicate
  - Social engineering attacks
    - Convince users to bypass security mechanisms to install Trojans or to respond to phishing attacks
- Payloads: how it performs once a target is reached
  - Corruption of system or data files
  - Theft of service: make the system a zombie agent
  - Theft of information
  - Stealing/hiding its presence on the system

Other Classification Approaches

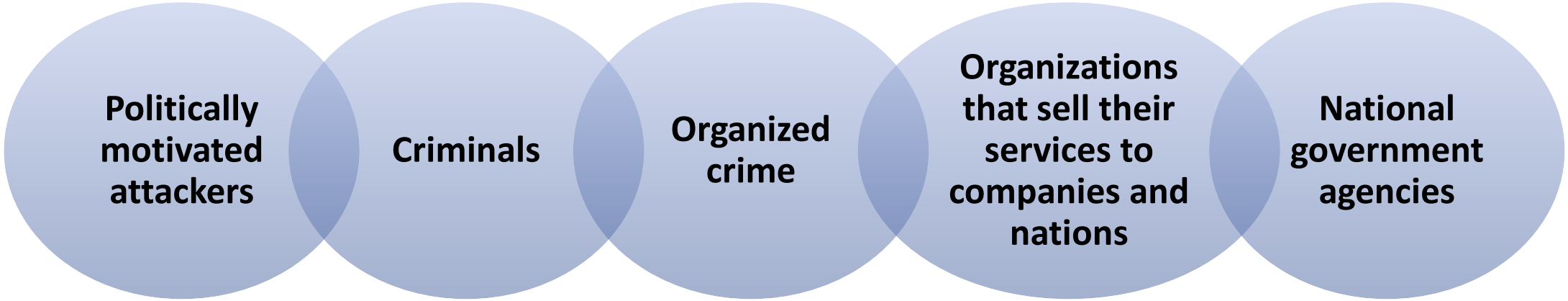
- Independent?
  - Need a host program: parasitic code such as viruses
  - Independent, self-contained programs: worms, Trojans, and bots
- Replicate?
  - Yes: viruses and worms
  - No: Trojans and spam e-mail

Attack Kits

- Malware development/deployment: considerable technical skills
  - Early 1990s: virus-creation toolkits
  - 2000s: more general attack kits
- Widely used toolkits: Zeus, Blackhole, Sakura, Phoenix, etc.
- Toolkits: known as “crimeware”
  - A variety of propagation mechanisms and payload modules
    - Even novices can combine, select, and deploy
  - A large number of new variants can be generated by attackers

Attack Sources

- Attacker change
  - Individuals → more organized/dangerous attack sources



  - changing resource available and motivation behind the rise of malware
    - A large underground economy: the sale of attack kits, access to compromised hosts, and stolen information

Advanced Persistent Threats (APTs)

- Well-resourced, persistent threats
  - Selected target, usually business or political
  - Characteristics: careful target selection, persistent, and stealthy
  - High profile attacks: Aurora, RSA, APT1, etc.
    - e.g., exposing One of China's Cyber Espionage Units (video)
- Advanced
  - Using a wide variety of intrusion technologies and (custom) malware
  - Components may not be technically advanced but are carefully seleted

APTs (Cont.)

- Persistent
  - Over an extended period against the target: maximize the chance of success
  - A variety of attacks may be progressively applied
    - Until the target is compromised
- Threats
  - Organized, capable, and well-funded attackers → specifically chosen targets
  - Active involvement of people in the process
  - Automated attacks tools

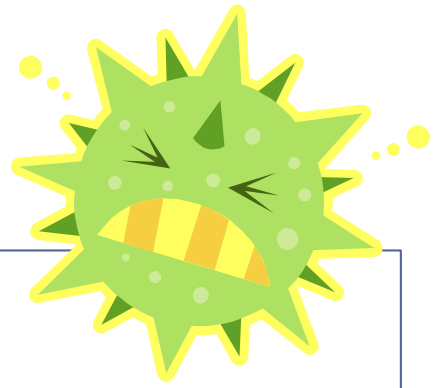
APT Attacks

- Goals
  - From: theft of intellectual property or data
  - To: physical disruption of infrastructure
- Techniques
  - Social engineering, spear-phishing email, drive-by-downloads, etc.
- Intent
  - Infecting the target with sophisticated malware
    - Multiple propagation mechanisms and payloads

Viruses: Propagation via Infected Content

- A piece of software: “infect” other programs
  - Any type of executable content
  - Modifying them to include a copy of the virus
  - Replicating and going on to infect other content
  - Easily spreading through network environments
- Can do anything that the host program is permitted to do
  - Executing secretly when the program is run
- Specific to OS and hardware
  - Taking advantage of their details and weaknesses

Virus Components



Infection mechanism

- Means by which a virus spreads or propagates
- Also referred to as the *infection vector*

Trigger

- Event or condition that determines when the payload is activated or delivered
- Sometimes known as *a logic bomb*

Payload

- What the virus does (besides spreading)
- May involve damage, or benign but noticeable activity

Virus Phases

Dormant		
idle	Activated by some event	Not all viruses have this stage

Propagation		
Placing a copy of itself into other programs or certain system areas	May not be identical to the propagating version	Each infected program will itself enter a propagation phase

Triggering	
Activated to perform its intended function	Can be caused by a variety of system events

Execution	
Performing the function	May be harmless or damaging

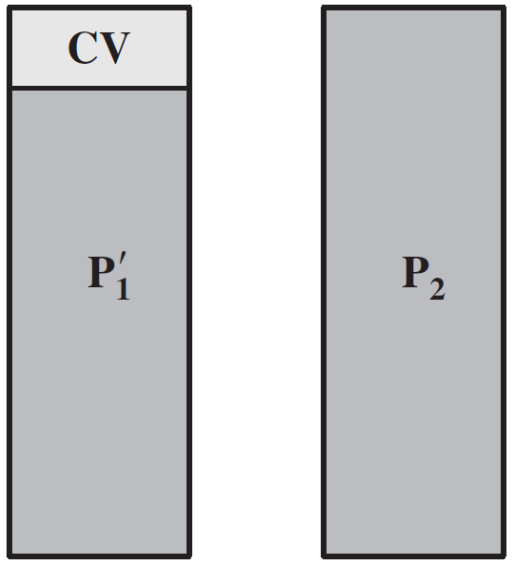
A Simple Virus

- This virus code, V, is prepended to infected programs
  - Assume that the entry point to the program is the main action block

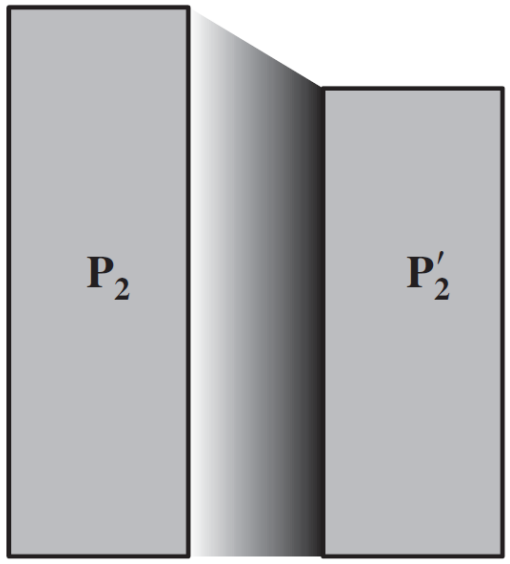
- However, it is easily detected. Why?

```
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 original program code;
end;
```

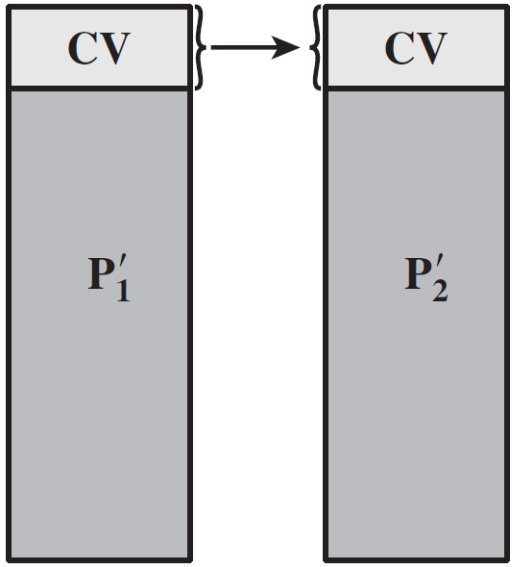
A Compression Virus



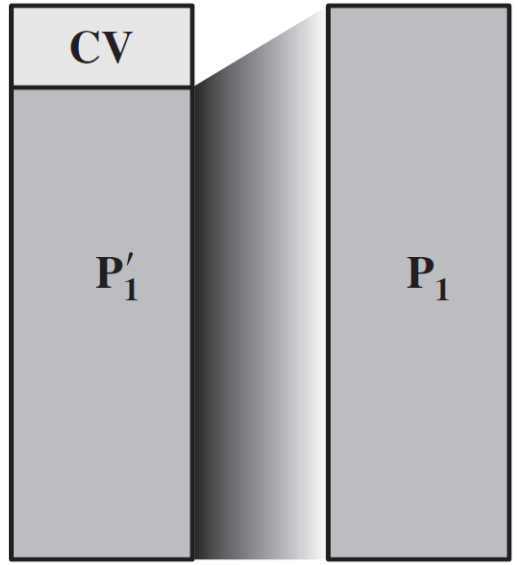
t_0: P_1' is infected version of P_1; P_2 is clean



t_1: P_2 is compressed into P_2'



t_2: CV attaches itself to P_2'



t_3: P_1' is decompressed into the original program P_1

Logic of the Compression Virus

```
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;
```

Virus Classifications (By Target)

- Boot sector infector
  - Infects a master boot record or boot record
  - Spreads when a system is booted from the disk containing the virus
- File infector
  - Infects files that OS or shell considers to be executable
- Macro virus
  - Infects files with macro or scripting code that is interpreted by an app
- Multipartite virus
  - Infects files in multiple ways or multiple types of files

Virus Classifications (By Concealment Strategy)

- Encrypted virus
  - Uses encryption to obscure its content
  - A portion creates a random encryption key and encrypts the remainder of the virus
  - When the virus replicates, a different random key is selected
- Stealth virus
  - Hides itself from detection by anti-virus software
    - Using code mutation, compression, or rootkit techniques
- Polymorphic virus
  - Creates replication copies that are functionally equivalent but have different bit patterns
    - Using inserting superfluous instructions, encryption, etc.
  - “signature” will vary with each copy

Virus Classifications (By Concealment Strategy)

- Metamorphic virus
  - Mutates with every infection
  - Difference from Polymorphic ones: rewrites itself completely at each iteration
    - Using multiple transformation techniques, etc.
  - May change both behaviors and appearance

Macro and Scripting Viruses

- Macro viruses infect scripting code used to support active content in a variety of user document types
  - Very common in mid-1990s
  - Platform independent; active content in commonly used apps
    - e.g., macros in MS Word
  - Infect documents (not executable portions of code)
    - e.g., MS Office products
    - e.g., Adobe PDF: embedded JavaScript in pdf (link)
  - Easily spread
  - Traditional file system access controls are of limited use. Why?
- Various anti-virus products: also developed tools against macro viruses
- Microsoft: increased protection against macro viruses
  - An optional macro virus protection, digital signature over macros

Macro Virus Structure

- Macro languages may have a similar syntax, but the details depend on apps
  - e.g., a MS Word macro is different from an Excel macro
- Either be saved with a document, or in a global template
- Some macros are run automatically when certain actions occur
  - e.g., macros can run when MS Word starts
  - not just perform operations on the document content, but can read/write files, and call other apps

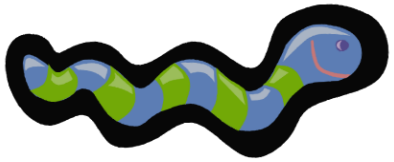
Macro Virus Structure – Example

- Melissa macro virus: a mass-mailing macro virus
  - Targets: MS Word and Outlook-based systems
    - Contained in the Document_Open macro
  - Damage: considerable network traffic
  - Hiding: disabling the Macro menu and some related security features
    - Harder for the user to stop or remove its operation
  - Infection: (1) every subsequent documents opened in the system; (2) sending infected documents to other users via email

```
macro Document_Open
    disable Macro menu and some macro security features
    if called from a user document
        copy macro code into Normal template file
    else
        copy macro code into user document being opened
    end if
    if registry key "Melissa" not present
        if Outlook is email client
            for first 50 addresses in address book
                send email to that address
                with currently infected document attached
            end for
        end if
        create registry key "Melissa"
    end if
    if minute in hour equals day of month
        insert text into document being opened
    end if
end macro
```

Worms: Propagation via Vulnerability Exploit

- Programs that actively seeks out more machines to infect
  - Each infected machine: an automated launching pad for attacks on other machines
- Some characteristics
  - Exploiting software vulnerabilities in client or server programs
  - Spreading: network connections, shared media (e.g., USB drives, CD/DVD), E-mails (in macro/script code)
  - Upon activation, they may replicate and propagate again



- First known implementation: Xerox Palo Alto Labs (early 1980s)

Worm Replication

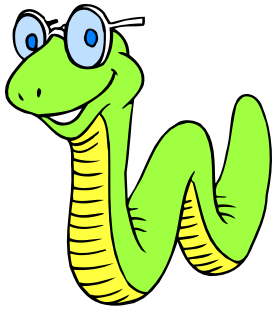
- Electronic mail or instant messenger facility
  - a copy of itself via email or an attachment via an instant message service
- File sharing
  - creating a copy of itself or infecting a file as a virus on removable media
  - using autorun mechanism by exploiting some software vulnerability
- Remote execution capability
- Remote file access or transfer capability
- Remote login capability

Target Discovery

- Same phases as a virus: dormant, propagation, triggering, and execution

- Scanning (or fingerprinting)
  - 1st function in the propagation phase for a network worm
  - Search for other systems to infect
- Scanning strategies
  - Random
  - Hit-list
  - Topological
  - Local subnet

The Morris Worm



- Earliest significant worm infection
  - Released by Robert Morris in 1988
- Designed to spread on UNIX systems
  - Attempted to crack local password file to use login/password to logon to other systems
  - Exploited a bug in the finger protocol which reports the whereabouts of a remote user
  - Exploited a trapdoor in the debug option of the remote process that receives and sends mail
- Infection steps
  - Successful communication with the system shell (command interpreter)
  - Execute a short bootstrap problem through the shell
  - The bootstrap program calls back the parent program and downloads the remainder of the worm
  - Execute the new worm

Recent Worm Attacks

Melissa	**1998**	**e-mail worm: propagated to all of the email addresses known to the infected host; took only three days to infect over 100,000 computers**
Code Red	**July 2001**	**exploited a security hole in the Microsoft Internet Information Server (IIS) DDoS attacks against a government website by flooding Infected nearly 360,000 servers in 14 hours**
Code Red II	**August 2001**	**Installed a backdoor for a hacker to remotely execute commands**
Nimda	**September 2001**	**had worm, virus and mobile code characteristics spread using e-mail, Windows shares, Web servers, Web clients, backdoors**
SQL Slammer	**Early 2003**	**exploited a buffer overflow vulnerability in MS SQL server compact and spread rapidly: infected 90% of vulnerable hosts within 10 mins**
Sobig.F	**Late 2003**	**exploited open proxy servers to turn infected machines into spam engines produced more than one million copies of itself within the first 24 hours**
Mydoom	**2004**	**mass-mailing e-mail worm installed a backdoor in infected machines replicated up to 1,000 times per minute**

Recent Worm Attacks (Cont.)

Warezov	**2006**	**(1) creates executables in system directories; (2) sends itself as an e-mail attachment; (3) can disable security related products and updating capability**
Conficker (Downadup)	**November 2008**	**exploits a Windows buffer overflow vulnerability most widespread infection since SQL Slammer**
Stuxnet	**2010**	**restricted rate of spread to reduce chance of detection targeted industrial control systems (Iranian nuclear program) propagation: USB drives, network file shares, zero-day vulnerability exploits 1st serious use of a cyberwarfare weapon against a nation's physical infrastructure**
Duqu	**2011**	**Used code related to Stuxnet Targeted the Iranian nuclear program**
Flame family	**2012**	**Targeted Middle-Eastern countries Very successful infection strategies: infected many countries, including the systems physically isolated from Internet**
WannaCry	**2017**	**Ransomware attack: encrypted files; demanded a ransom payment to recover Very fast propagation: infected > 100,000 systems over a period of hours to days Exploited a vulnerability in the SMB file sharing service on unpatched Windows**

Microsoft Security Bulletin MS17-010 - Critical

Security Update for Microsoft Windows SMB Server (4013389)

Published: March 14, 2017

Version: 1.0

Executive Summary

This security update resolves vulnerabilities in Microsoft Windows. The most severe of the vulnerabilities could allow remote code execution if an attacker sends specially crafted messages to a Microsoft Server Message Block 1.0 (SMBv1) server.

This security update is rated Critical for all supported releases of Microsoft Windows. For more information, see the **Affected Software and Vulnerability Severity Ratings** section.

The security update addresses the vulnerabilities by correcting how SMBv1 handles specially crafted requests.

For more information about the vulnerabilities, see the **Vulnerability Information** section.

For more information about this update, see Microsoft Knowledge Base Article 4013389.

On this page

State of Worm Technology

- Multiplatform
  - A variety of platforms (Windows, UNIX); macro or scripting languages supported in popular document types
- Multi-exploit
  - Exploits against Web servers, browsers, e-mail, file sharing, other network-based apps, etc.
- Ultrafast spreading
  - Optimize the spread rate; locate as many vulnerable machines as possible in a short time period
- Polymorphic
  - Each copy has new code generated on the fly using functionally equivalent instructions and encryption techniques

State of Worm Technology (Cont.)

- Metamorphic
  - Unleashed behavior patterns at different stages of propagation
- Transport vehicles
  - Worms can rapidly compromise a large number of systems
  - Ideal for spreading a wide variety of malicious payloads, such as DDoS bots
- Zero-day exploit
  - In 2015, 54 zero-day exploits were discovered and exploited
  - Many of these were in common computer and mobile software

Mobile Code

- Programs that can be shipped unchanged to a variety of platforms
  - Cross-platform: transmitted from a remote system to a local system and then executed on the local system
- Popular vehicles include Java applets, Active X, JavaScript and VBScript → since they are cross-platform
- Common methods of using mobile code for malicious operations
  - Interactive and dynamic websites, e-mail attachments, and downloads from untrusted sites or of untrusted software

Mobile Phone Worms

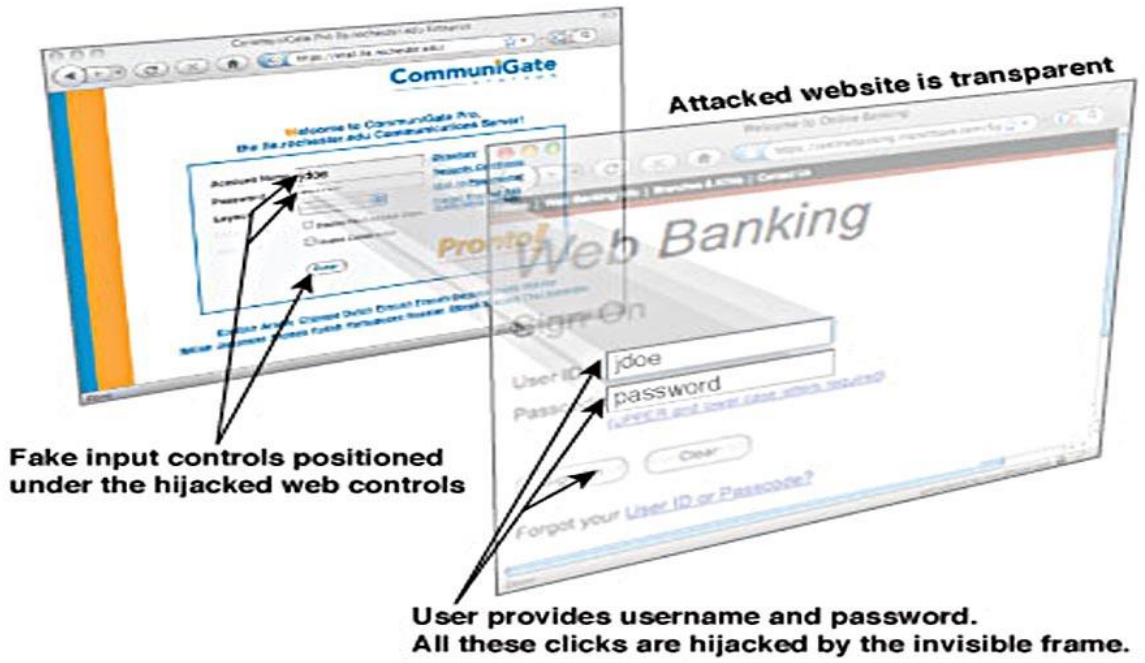
- Cabir worm: first appeared on mobile phones (2004)
  - Communicated through Bluetooth or multimedia messaging service (MMS)
  - Early mobile worms targeted Symbian, but recent ones target Android and iPhone
  - Can completely disable the phone, delete data on the phone, or force the device to send costly messages

Drive-By-Downloads

- Exploits browser vulnerabilities to download and installs malware on the system when the user views a Web page controlled by the attacker
- In most cases, they do not actively propagate
- Spread when users visit the malicious Web page
- Multiple vulnerabilities in the Adobe Flash Player and Java plugins have been exploited over many years

Clickjacking

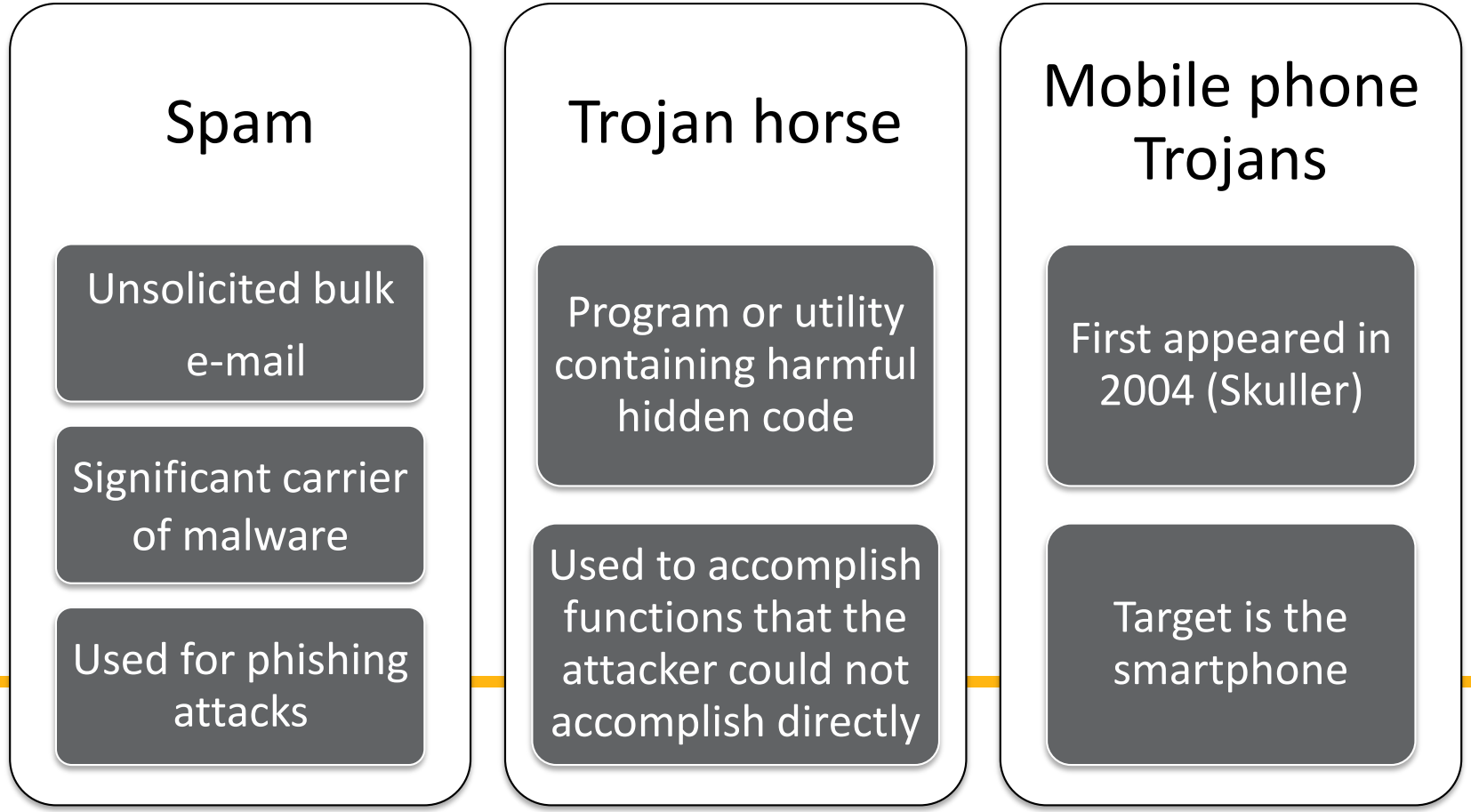
- Also known as a user-interface (UI) redress attack
  - to collect an infected user's clicks
  - Typically, using multiple transparent or opaque layers
- Keystrokes can be hijacked
  - A user can be led to believe they are typing in the password to their email or bank account
  - But, are instead typing into an invisible frame controlled by the attacker



e.g., using iFrame (HTML)

Social Engineering: Propagation via Spam E-mail, Trojans

- "Tricking" users to assist in the compromise of their own systems



Outline

- Types of Malicious Software (Malware)
- Advanced Persistent Threat
- Propagation
  - Infected Content – Viruses
  - Vulnerability Exploit – Worms
  - Social Engineering – Spam E-Mail, Trojans
- Payload
  - System Corruption
  - Attack Agent – Zombie, Bots
  - Information Theft – Keyloggers, Phishing, Spyware
  - Stealthing – Backdoors, Rootkits
- Countermeasures

Payload: System Corruption

- Data destruction and ransomware
  - Virus
    - Chernobyl: Windows-95 and 98 virus; first seen in 1998
    - Infected executable files and corrupted the entire file system (a date is reached)
  - Worm
    - Klez: mass-mailing worm infecting Windows-95 to XP systems; first seen in 2001
    - On trigger dates (13th of several months each year), caused files on the hard drive to become empty
  - Ransomware
    - PC Cyborg Trojan (1989)
    - WannaCry (2017)
    - Encrypted the user's data and demands payment for recovery

Payload: System Corruption (Cont.)

- Real-world damage
  - Causes damage to physical equipment
    - Chernobyl virus not only corrupted data, but also rewrote the BIOS code
    - Stuxnet worm targets industrial control system software → may drive the controlled equipment outside its normal operating range
    - Critical infrastructure: e.g., disrupted Ukrainian power systems in 2015 [SYMA16]
- Logic bomb
  - Code embedded in the malware that is set to "explode" when certain conditions are met
    - May alter or delete data or entire files, cause a machine to halt, etc.

Payload: Attack Agent – Zombie, Bots

- Takes over another Internet attached computer and uses that computer to launch or manage attacks
- *Botnet* – collection of bots capable of acting in a coordinated manner
- Uses:
  - Distributed denial-of-service (DDoS) attacks
  - Spamming
  - Sniffing traffic
  - Keylogging
  - Spreading new malware
  - Installing advertisement add-ons and browser helper objects (BHOs)
  - Attacking IRC chat networks
  - Manipulating online polls/games

Payload: Attack Agent – Zombie, Bots (Cont.)

- Remote control Facility
  - Distinguishes a bot from a worm
    - Worm propagates itself and activates itself
    - Bot is initially controlled from some central facility
  - Typical means of implementation: an IRC (Internet Relay Chat) server
    - Joining a specific channel on this server and treat incoming messages as commands
    - More recent botnets use covert communication channels via protocols such as HTTP
    - Distributed control mechanisms use P2P protocols to avoid a single point of failure

Payload: Information Theft – Keyloggers, Phishing, Spyware

- Keylogger
  - Captures keystrokes to allow attackers to monitor sensitive information
  - Uses some form of filtering mechanism that only returns information close to keywords
- Spyware
  - Subverts the compromised machine to allow monitoring of activities on the system
  - Monitors history and content of browsing activity
  - Redirects certain web page requests to fake sites
  - Dynamically modifies data exchanged between the browser and certain web sites

Payload: Information Theft (Cont.)

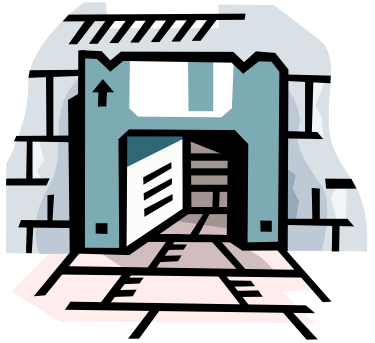
- Phishing
  - exploits social engineering to leverage the user's trust by masquerading as communication from a trusted source
  - e.g., includes a URL in a spam e-mail that links to a fake Web site
    - Mimicking the login page of a banking, gaming, or similar site
  - e.g., deceives the user that some urgent action is required

Spear-phishing

- Recipients are carefully researched by the attacker
- E-mail is crafted to specifically suit its recipient
- Often quoting a range of information to convince them of its authenticity

Payload: Stealthing – Backdoors, Rootkits

- Backdoor (aka. trapdoor)
  - A secret entry point into a program: allowing the attacker to gain access and bypass the security access procedures
  - Maintenance hook: a backdoor used by programmers to debug and test programs
    - Usually implemented as a network service listening on some non-standard port
  - Difficult to implement OS controls for backdoors in apps



Payload: Stealthing (Cont.)

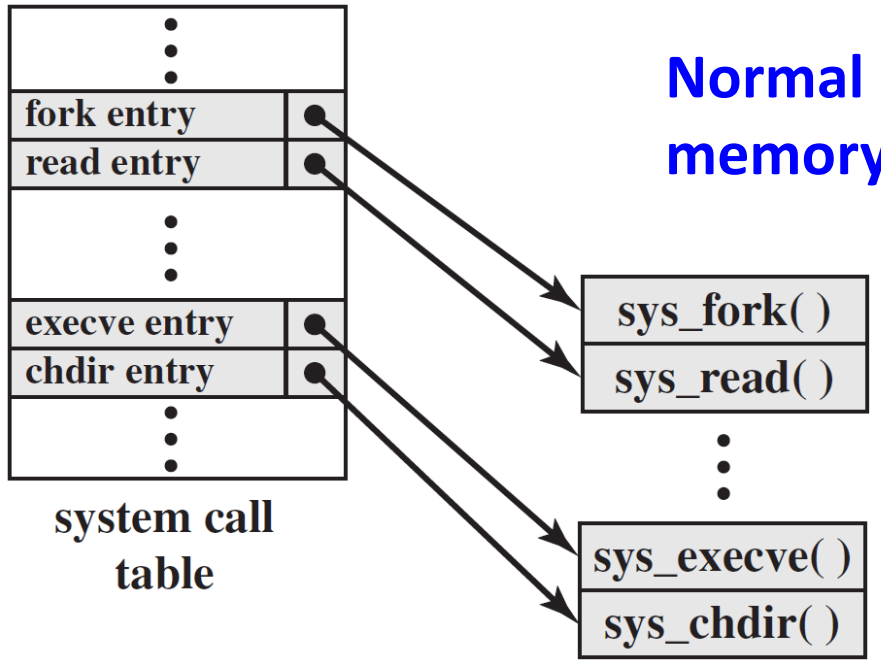
- Rootkit
  - A set of hidden programs installed on a system to maintain covert access to the system with root privileges
  - Root access: attacker has complete control of the system, and alters the system's standard functionality in a malicious and stealthy way
  - Hides by subverting the mechanisms that monitor and report on the processes, files, and registries on a computer

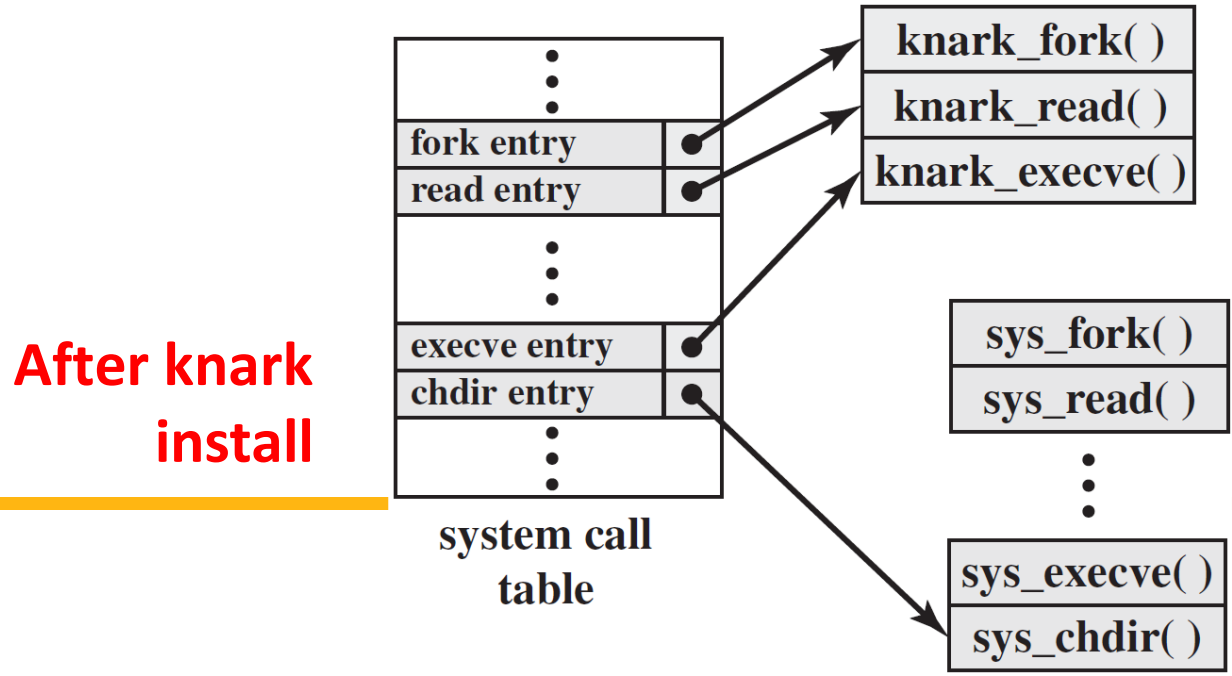
Payload: Stealthing (Cont.)

- Rootkit Classification Characteristics
  - **Persistent**: activates each time the system boots
  - **Memory based**: cannot survive a reboot, but can be harder to detect
  - **User mode**: intercepts calls to APIs and modifies returned results
  - **Kernel mode**: can intercept calls to native APIs in kernel mode
  - **Virtual machine based**: installs a lightweight virtual machine monitor, and then runs the OS in a virtual machine above it
  - **External mode**: located outside the normal operation mode (e.g., in BIOS or system management mode)

Payload: Stealthing (Cont.)

- Kernel mode rootkits: next generation
  - Making changes inside the kernel and co-existing with the OS code
  - ➔ Make their detection much harder
  - A primary target: system calls
- Example: system call table modification by the knark Rootkit





Countermeasures

- What is the ideal solution?
  - Prevention: do not allow malware to get into the system
  - However, it is nearly impossible
- NIST suggests four main elements of prevention
  - Policy: e.g., having all patches installed
  - Awareness: e.g., having a training course
  - Vulnerability mitigation: e.g., disabling unused services
  - Threat mitigation: e.g., having anti-virus/IDS software installed
- If prevention fails
  - Detection, identification, and removal

Requirements for Countermeasures

- Generality: handle a wide variety of attacks
- Timeliness: respond quickly → limit the number of infected programs
- Resiliency: resistant to evasion techniques
- Minimal DoS costs: minimal reduction in capacity or service
- Transparency: no modification to existing OSs, app software, hardware
- Global and local coverage: against attack sources both from outside and inside the network

Host-based Scanners: Anti-virus Software

First generation: simple scanners

- Require a malware signature to identify the malware
- Limited to the detection of known malware

Second generation: heuristic scanners

- Execute the questionable program in VM or decompile them and analyze source codes
- Another approach is integrity checking

Third generation: activity traps

- Memory-resident programs that identify malware by its actions rather than its structure in an infected program

Fourth generation: full-featured protection

- Packages consisting of a variety of anti-virus techniques used in conjunction
- Include scanning and activity trap components and access control capability - e.g., using firewall to limit the malware propagation

Other Countermeasures

- Generic decryption (GD)
  - detect the most complex polymorphic viruses and malware
  - based on the behavior: when a file containing a polymorphic virus is executed, the virus must ***decrypt*** itself to activate
  - executable files are run through a GD scanner
    - Including CPU emulator, virus signature scanner, emulation control module, etc.
  - the emulator begins interpreting instructions in the target code, one at a time
    - Searching for decryption routines
  - Drawback?
    - To determine how long to run each interpretation

Other Countermeasures (Cont.)

- Host-based Behavior-Blocking software
  - Integrate with the OS and monitors program behavior in real time for malicious actions
  - Block potentially malicious actions, before they have a chance to affect the system
  - Block suspicious software in real time, so it has an advantage over anti-virus detection techniques such as fingerprinting or heuristics
  - Drawback? Malicious codes must run on the target before all its behaviors can be identified
    - It can cause harm before it has been detected

Other Countermeasures (Cont.)

- Spyware detection and removal
- Rootkit countermeasures
- Perimeter scanning approaches
- Distributed Intelligence gathering approaches

Questions?