Lecture 5 – Malware

University of Illinois

ECE 422/CS 461

Goals

- By the end of this lecture you should:
  - Understand malware motives and means of attacks
  - Be able to classify malware by motives and means
  - know common malware defenses

Malware: Definition and Goals

- "Malware" is short for malicious software and is typically used as a catch-all term to refer to any software designed to cause damage to a computer, server, or computer network

- Examples: computer viruses, worms, trojans, ransomware, spyware, backdoors, rootkits, …

Malware Damage and Prevalence

- What can malware do? Pretty much *anything:*
  - Destroy data
  - Encrypt data
  - Steal data (record video/audio/screen/keystrokes)
  - Harass users
  - Show ads
  - Launch external activity (e.g., email spam)

- ~ 6 billion malware encounters per year
- 200 million **new** malware developed per year

Malware Classifications

Malware Categories

Propagation Behaviors
(How does it run?)

- Trojan Horses
- Computer Viruses
- Internet Worms
- Exploit Kits

Payload Behaviors
(What does it do?)

- Backdoors
- Logic Bombs
- Ransomware
- Spyware
- Droppers
- Botnets

Trojan Horses

- Software that masquerades as legitimate / useful programs but actually performs malicious functions
- *Propagation behavior*: trick users into installing them

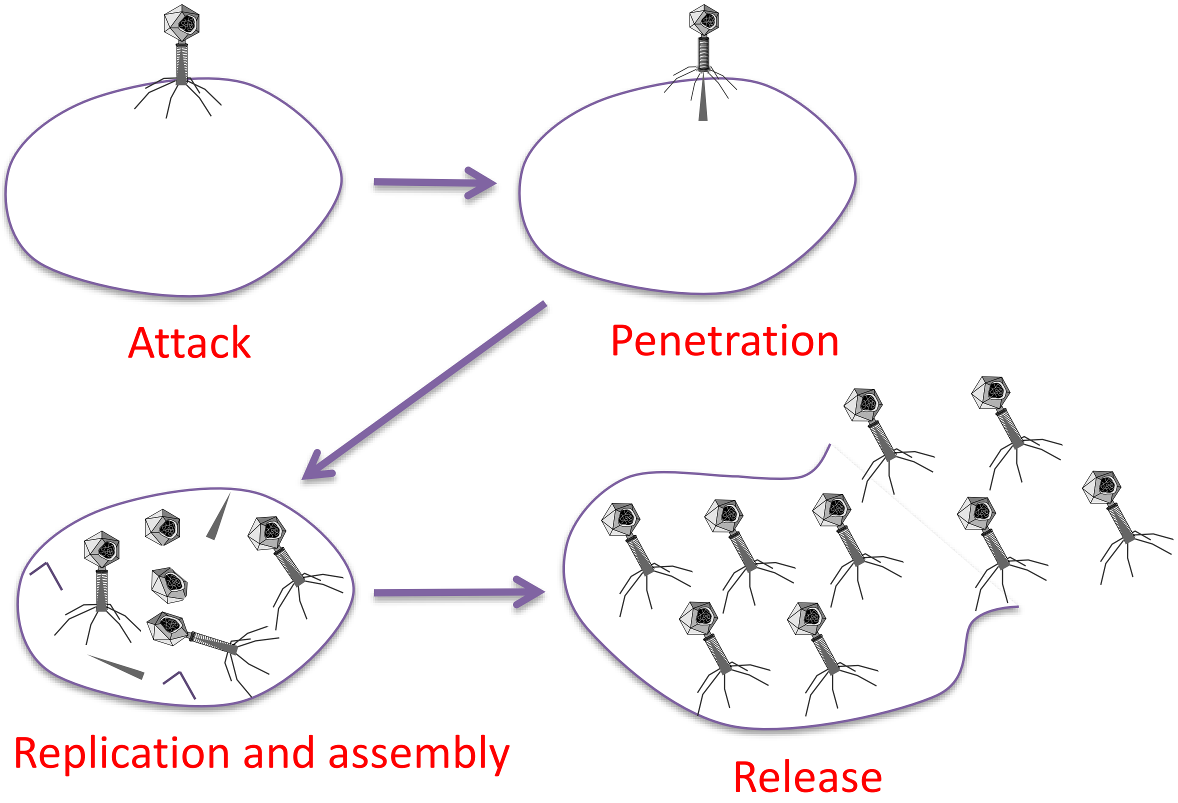
Trojan Horses

- Exacerbated by *app repackaging*
  - Android apps typically written in Java are easy to reverse engineer (there are tools to decompile an .apk installation package)
  - Add malicious payload, recompile and publish

- A major reason to avoid 3rd party app markets
  - Less likely (though still possible) for malicious apps to appear on official app store

Computer Viruses

- Share some properties with biological viruses:

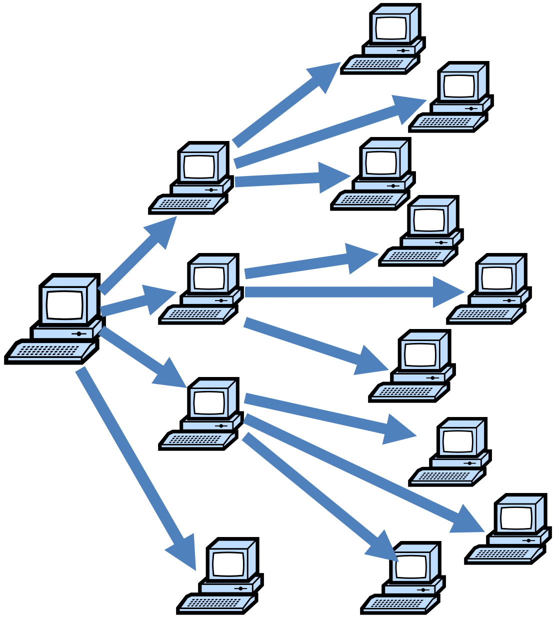


Computer Viruses

- A **computer virus** is computer code that replicates itself by **modifying other files or programs** to insert code that is capable of further replication
  - Typically need **user interaction** to replicate, e.g., opening an email attachment or using a USB drive.

Internet Worms

- A **worm** is malware that replicates and propagates across systems **automatically**
  - No user interaction required



Virus vs. Worm

- Both replicate and propagate
- Virus: have itself **eventually** executed
  - Generally infects by altering **stored** code
    - Hence, requires user interaction to activate
- Worm: have itself **immediately** executed
  - Generally infects by exploiting software bugs (e.g., buffer overflow) to alter **running** code
    - Hence, no user interaction required
  - Thus, usually spreads faster than virus

* Some people use the two terms interchangeably

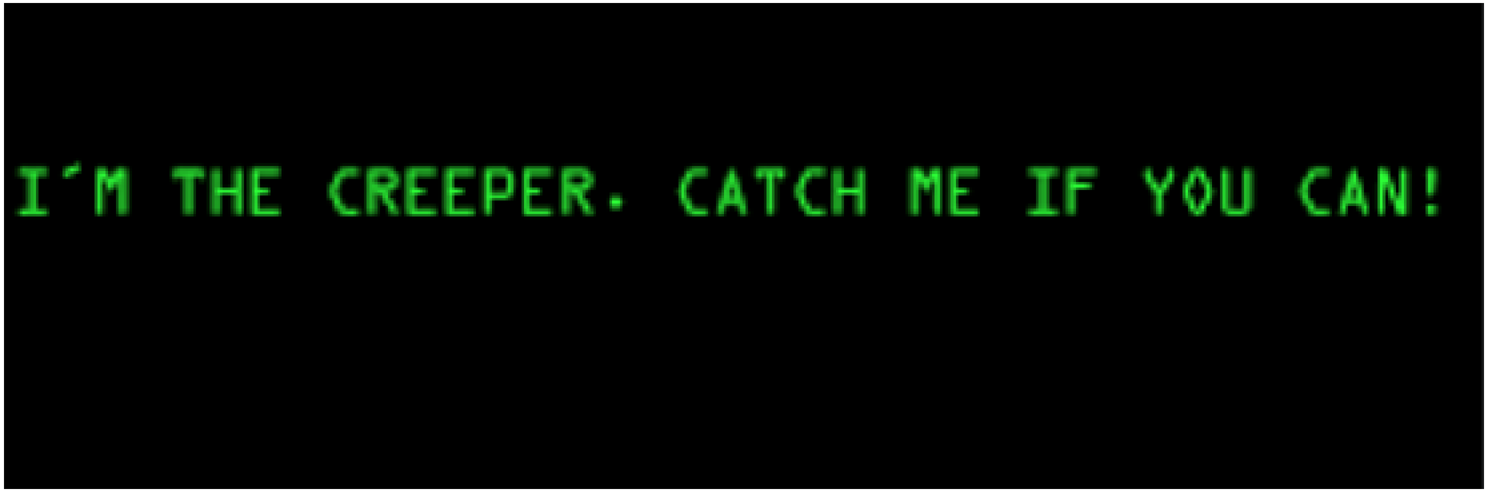
History of Virus and Worms

- 1949: John von Neumann lectures on their theoretical existence (at U of Illinois)

- 1970: appeared in a science fiction
  - Origin of the term “worm”

History of Virus and Worms

- 1971: first worm “Creeper” written by Bob Thomas and Ray Tomlinson
  - – Deployed on ARPANET (28 machines at that time)
  - – Benign, caused no damage, just displayed



```
I´M THE CREEPER. CATCH ME IF YOU CAN!
```

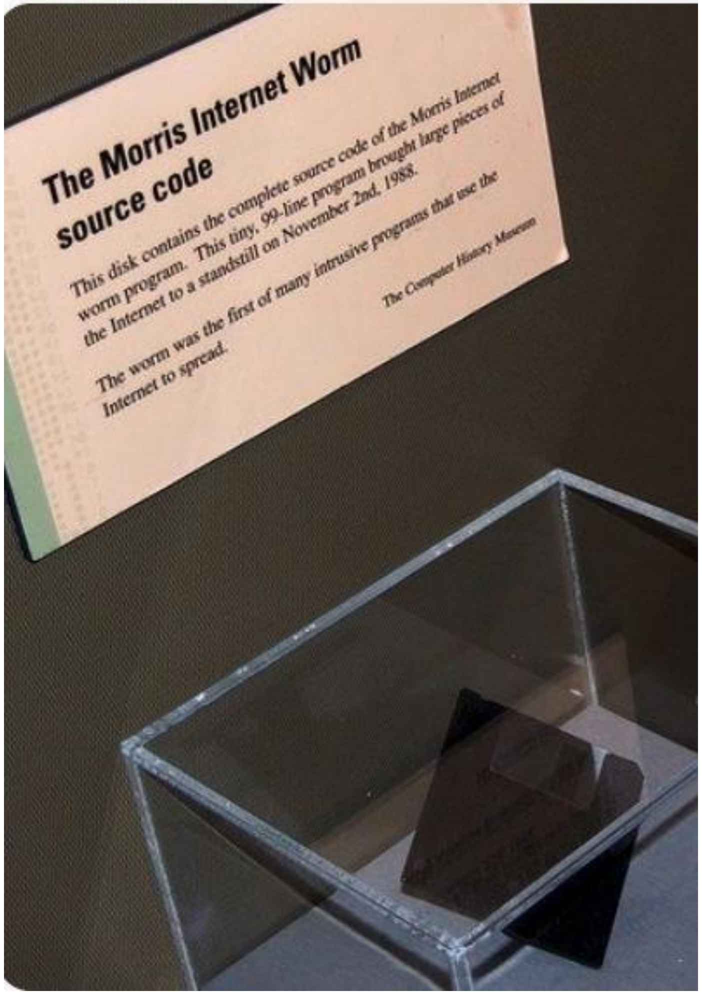
History of Virus and Worms

- 1971: first worm "Creeper" written by Bob Thomas and Ray Tomlinson
  - Deployed on ARPANET (28 machines at that time)
  - Benign, caused no damage, just displayed

- 1972: "Reaper" by Ray Tomlinson
  - A worm that propagated to delete Creeper
  - Can be considered the first anti-virus software

History of Virus and Worms

- 1988: “Morris Worm”, aka “Internet Worm”
  - Written by Robert Morris, then a graduate student in CS at Cornell
  - A landmark event in the history of Internet
    - ~60,000 machines then
  - Paralyzed thousands of machines, partitioned Internet for days



The Morris Internet Worm source code

This disk contains the complete source code of the Morris Internet worm program. This tiny, 99-line program brought large pieces of the Internet to a standstill on November 2nd, 1988.

The worm was the first of many intrusive programs that use the Internet to spread.

The Computer History Museum

Morris Worm

- Employed a series of sophisticated techniques
  - Multiple buffer overflow attacks
  - Guessed weak passwords (or no passwords)
  - Many tricks to find new targets
    - Scan local network, look through network config files, and user files that mention other hosts

Morris Worm

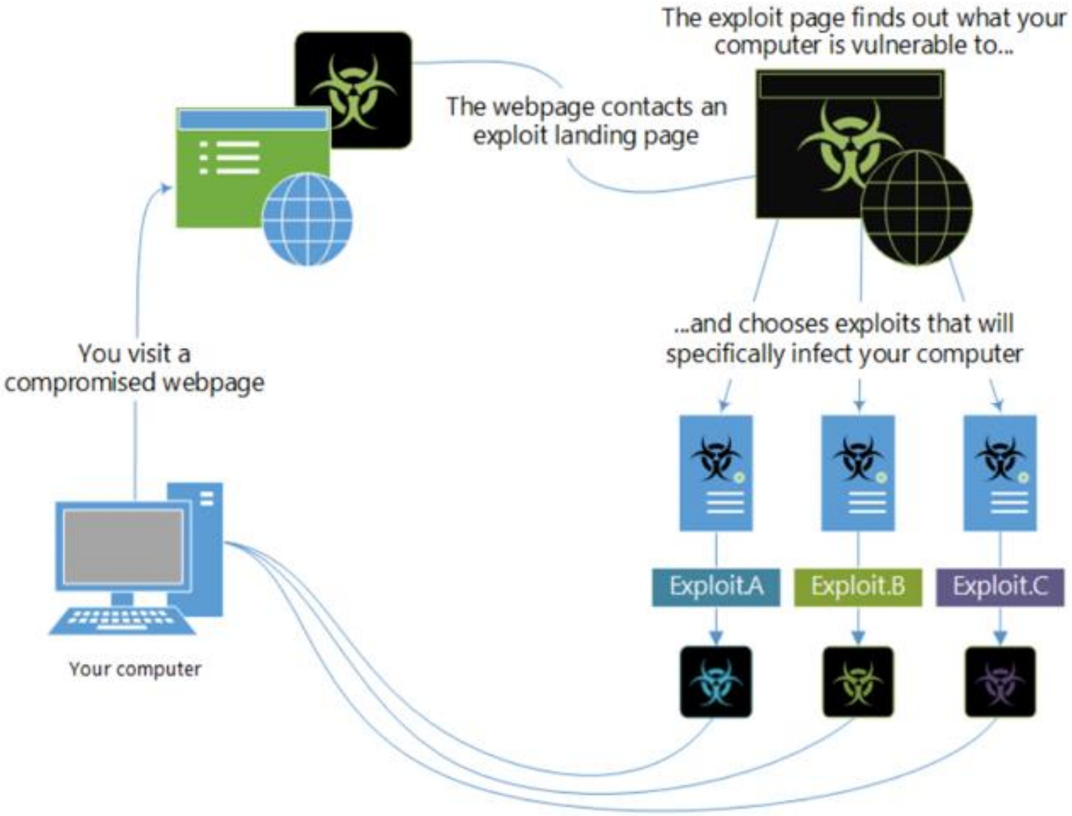
- Likely not intended to be malicious
  - Does not do anything other than infecting hosts
  - Tried to check if a host had already been infected
  - But decided to re-infect a host with a 14% probability even if it had been infected
- That last small tweak made it out of control …
  - Each host infected with many copies of the worm
  - … and each copy tried to infect other hosts
  - And this kept amplifying

Morris Worm Aftermath

- Morris became the first person convicted under the Computer Fraud and Abuse Act (CFAA, 1986)
  - 3 years of probation, 400 hours of community service, and a fine of $10,500

Exploit Kits

- Automatically choose an exploit
- Exploit-as-a-Service, often sold on black markets


The exploit page finds out what your computer is vulnerable to...
The webpage contacts an exploit landing page
You visit a compromised webpage
...and chooses exploits that will specifically infect your computer
Exploit.A
Exploit.B
Exploit.C
Your computer

Malware Categories

Propagation Behaviors
(How does it run?)

- Trojan Horses
- Computer Viruses
- Internet Worms
- Exploit Kits

Payload Behaviors
(What does it do?)

- Backdoors
- Logic Bombs
- Ransomware
- Spyware
- Droppers
- Botnets

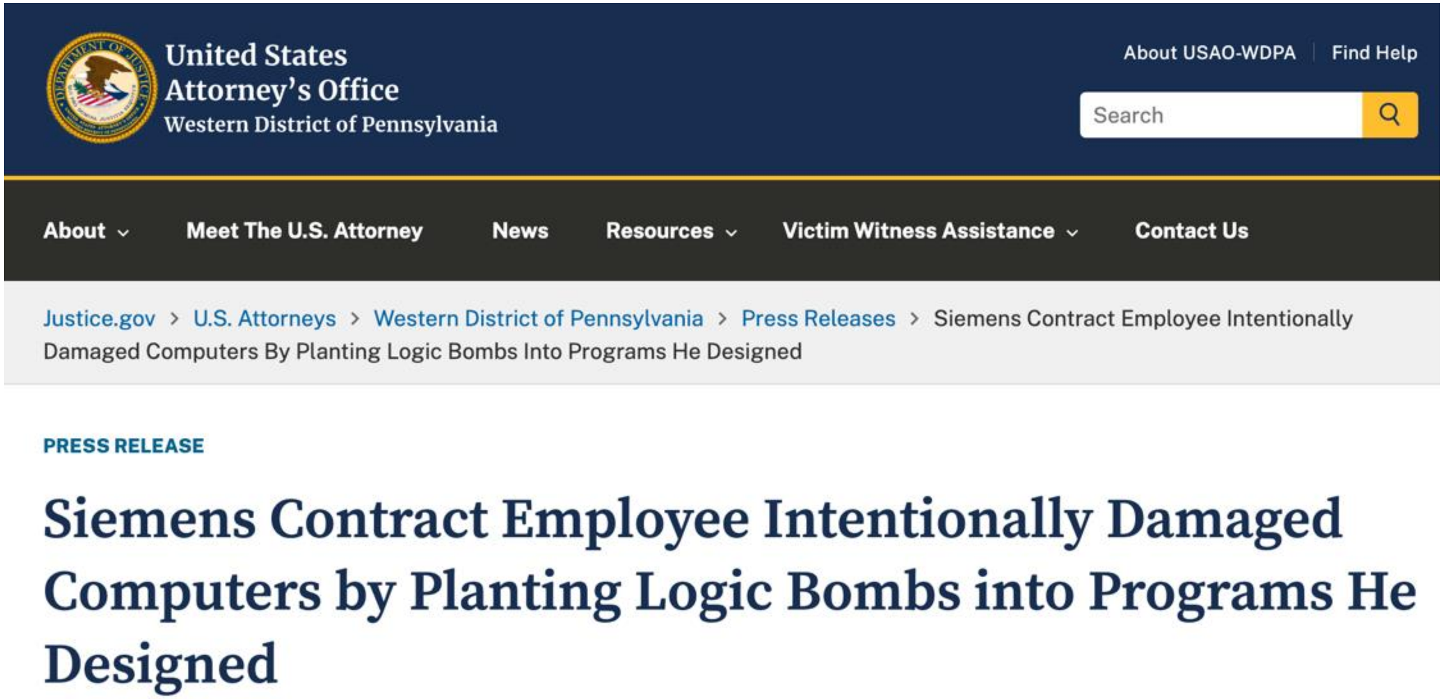
Backdoor

- An undocumented way of gaining access to a system (bypass normal authentication process)

- Often inserted into code or *supply chain* (by developers or attacks) prior to distribution

Logic Bomb

- A piece of code that set off a malicious function when specified conditions are met
  - – Often laying dormant for a long period



Justice.gov > U.S. Attorneys > Western District of Pennsylvania > Press Releases > Siemens Contract Employee Intentionally Damaged Computers By Planting Logic Bombs Into Programs He Designed

PRESS RELEASE

Siemens Contract Employee Intentionally Damaged Computers by Planting Logic Bombs into Programs He Designed

Ransomware

- Encrypts victim’s data (delete plaintext version), promise to decrypt only if a ransom is paid

- On the rise, account for 20%~70% of attacks
- Average ransom is now 2.73 million dollars

WannaCry (May 2017)

- Worm + ransomware
- Exploited vulnerabilities in Windows that were patched by Microsoft in March
  - Some Windows versions were no longer supported (e.g., Windows XP and Windows 2003)
- Affected > 300,000 computers in > 150 countries
- Total damage might be billions of dollars

Spyware

- Software that gathers information without victim’s knowledge
    - Keyloggers
    - Take screenshots
    - Record and transmit GPS locations
    - Record and transmit videos & pictures
    - ……

Droppers

- Malware whose main purpose is to install another malware

- Why would a dropper be useful?
  - May want to decide the concrete attack later
  - Sell the victim to the highest bidder

- Persistent vs. non-persistent: whether it is erased after installation of another malware

Botnets

- A network of compromised machines (bots) under (unified) control of attacker (botmaster)

- A new bot “phones home” to rendezvous with botnet *command-and-control* (C&C)
- Botmaster uses C&C to push out commands
  - Common example: distributed denial of service, email spam, cryptocurrency mining

Rootkits

- A rootkit subverts / modifies the operating system to hide its existence
  - Hard to detect using software that relies on OS itself (e.g., anti-virus software that monitors system calls)
  - Hard to disable / remove / uninstall

Sony XCP Rootkit

- In 2005, Sony distributed CDs with auto-installed rootkit-based DRM software
  - Not disclosed, even in licensing agreement
  - Also spyware: sent user listening habits
  - Sony paid $5.75M to settle multiple class-action lawsuits

Malware Defense

Malware Defense Overview

- Proactive security
  - Better design and better coding to avoid bugs
  - Find and fix bugs (e.g., software testing)
  - Anticipate bugs, isolate untrusted software

- Reactive security
  - Identify malware (e.g., intrusion detection) Today
  - Investigate incidents and respond quickly

Intrusion Detection System (IDS)

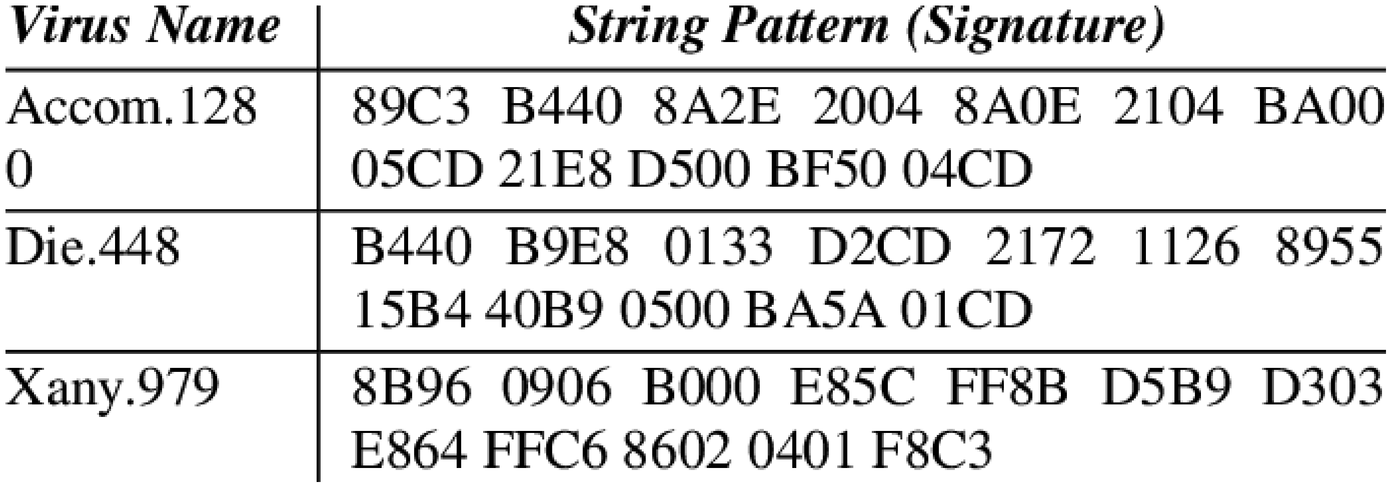
- Device or software that monitors networks or systems for malicious or suspicious activities
  - “Intrusion” & “activities” are broader than malware

- Three main types:
  - Signature-based: What does malware *look like*?
  - Heuristic-based: How does malware behave?
  - Anomaly-based: How does normal software behave?

Signature-based IDS

(Not to confuse with digital signatures in cryptography)

- Scan the analyzed object, compare with a dataset of malware features (signatures)
  - E.g., a sequence of instructions



Virus Name	*String Pattern (Signature)*
Accom.1280	89C3 B440 8A2E 2004 8A0E 2104 BA00 05CD 21E8 D500 BF50 04CD
Die.448	B440 B9E8 0133 D2CD 2172 1126 8955 15B4 40B9 0500 BA5A 01CD
Xany.979	8B96 0906 B000 E85C FF8B D5B9 D303 E864 FFC6 8602 0401 F8C3

Rad et al.. Evolution of Computer Virus Concealment and Anti-Virus Techniques: A Short Survey. 2011

Signature-based IDS Challenges

- Polymorphic malware
  - Decryption engine + (randomized) encrypted body
  - Look for decryption engine in detection?

- Metamorphic malware
  - Contains a code rewriter to generate **functionally equivalent** but **semantically different** code

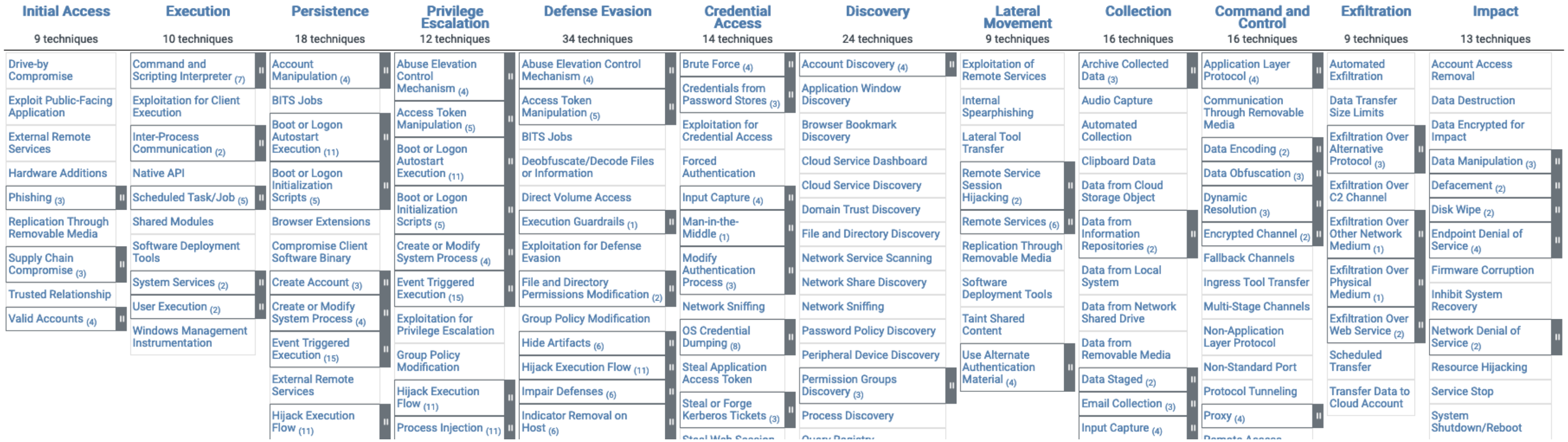
- Over 90% of malware have these capabilities

Heuristic/Rule-based IDS

- Detect malware based on what it **does**
  - – Modern jargon: *Endpoint Detection & Response (EDR)*

- Resilient to minor changes in malware
- Rely on (proprietary) datasets, like signature-based IDS

ATT&CK Enterprise Matrix



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
9 techniques	10 techniques	18 techniques	12 techniques	34 techniques	14 techniques	24 techniques	9 techniques	16 techniques	16 techniques	9 techniques	13 techniques
Drive-by Compromise	Command and Scripting Interpreter (7)	Account Manipulation (4)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Brute Force (4)	Account Discovery (4)	Exploitation of Remote Services	Archive Collected Data (3)	Application Layer Protocol (4)	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	Exploitation for Client Execution	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Credentials from Password Stores (3)	Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
External Remote Services	Inter-Process Communication (2)	Boot or Logon Autostart Execution (11)	Boot or Logon Autostart Execution (11)	BITS Jobs	Exploitation for Credential Access	Browser Bookmark Discovery	Lateral Tool Transfer	Automated Collection	Data Encoding (2)	Exfiltration Over Alternative Protocol (3)	Data Encrypted for Impact
Hardware Additions	Native API	Boot or Logon Initialization Scripts (5)	Boot or Logon Initialization Scripts (5)	Deobfuscate/Decode Files or Information	Forced Authentication	Cloud Service Dashboard	Remote Service Session Hijacking (2)	Clipboard Data	Data Obfuscation (3)	Exfiltration Over C2 Channel	Data Manipulation (3)
Phishing (3)	Scheduled Task/Job (5)	Browser Extensions	Create or Modify System Process (4)	Direct Volume Access	Input Capture (4)	Cloud Service Discovery	Remote Services (6)	Data from Cloud Storage Object	Dynamic Resolution (3)	Exfiltration Over Other Network Medium (1)	Defacement (2)
Replication Through Removable Media	Shared Modules	Compromise Client Software Binary	Event Triggered Execution (15)	Execution Guardrails (1)	Man-in-the-Middle (1)	Domain Trust Discovery	Replication Through Removable Media	Data from Information Repositories (2)	Encrypted Channel (2)	Exfiltration Over Physical Medium (1)	Disk Wipe (2)
Supply Chain Compromise (3)	Software Deployment Tools	Create Account (3)	Exploitation for Privilege Escalation	Exploitation for Defense Evasion	Modify Authentication Process (3)	File and Directory Discovery	Software Deployment Tools	Data from Local System	Fallback Channels	Exfiltration Over Web Service (2)	Endpoint Denial of Service (4)
Trusted Relationship	System Services (2)	Create or Modify System Process (4)	Group Policy Modification	File and Directory Permissions Modification (2)	Network Sniffing	Network Service Scanning	Taint Shared Content	Data from Network Shared Drive	Ingress Tool Transfer	Scheduled Transfer	Firmware Corruption
Valid Accounts (4)	User Execution (2)	Event Triggered Execution (15)	Hijack Execution Flow (11)	Group Policy Modification	OS Credential Dumping (8)	Network Share Discovery	Use Alternate Authentication Material (4)	Data from Removable Media	Multi-Stage Channels	Transfer Data to Cloud Account	Inhibit System Recovery
	Windows Management Instrumentation	External Remote Services	Process Injection (11)	Hide Artifacts (6)	Steal Application Access Token	Network Sniffing		Data Staged (2)	Non-Application Layer Protocol		Network Denial of Service (2)
		Hijack Execution Flow (11)		Hijack Execution Flow (11)	Steal or Forge Kerberos Tickets (3)	Password Policy Discovery		Email Collection (3)	Non-Standard Port		Resource Hijacking
				Impair Defenses (6)	Steal Web Session	Peripheral Device Discovery		Input Capture (4)	Protocol Tunneling		Service Stop
				Indicator Removal on Host (6)		Permission Groups Discovery (3)			Proxy (4)		System Shutdown/Reboot
						Process Discovery			Remote Access		
						Query Registry					

Anomaly-based IDS

- Enumerating malware behavior is hard
  - New malware comes out every week
  - Arms race between attacker and defender

- Instead, describe a program's normal behavior and alert if deviating from normal behaviors

Monitoring System Calls

- Used by both rule- and anomaly-based IDS
- System calls are the bridge between users processes and the OS
  - e.g., creating a new process, accessing a file, etc.
- Malware cannot cause significant damage without using system calls

Forrest IDS: N-Gram Monitoring

- “A Sense of Self for Unix Processes”
  - [Forrest et al. 1996]

- Describes normal behaviors for a process using short sequences of system calls (N-Grams)

Forrest IDS: Training

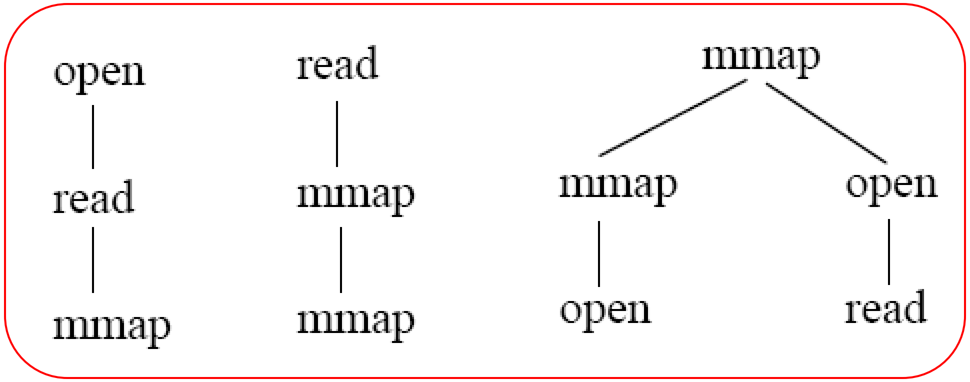
- Slide a window over a given system call trace and extract unique sequences of length N

Example (N=3): open, read, mmap, mmap, open, read, mmap System Call trace

Unique Sequences

open, read, mmap
read, mmap, mmap
mmap, mmap, open
mmap, open, read

Database



Forrest IDS: Runtime Monitoring

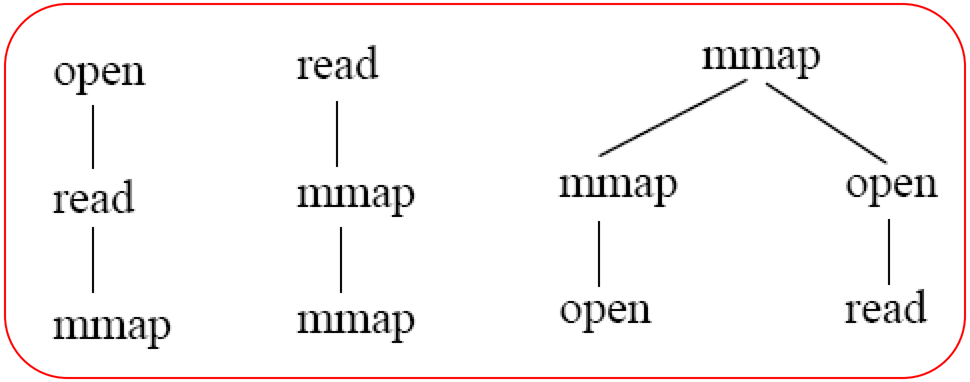
- Monitor the system call trace as the program issues them; Raise alert if detecting a sequence not seen during training

Example (N=3): mmap, open, read, mmap, read, …

Unique Sequences

open, read, mmap
read, mmap, mmap
mmap, mmap, open
mmap, open, read

Database



open
read
mmap

read
mmap
mmap

mmap
mmap
open
open
read

Intrusion Detection Systems

- Typically combine all three styles, and recently, more advanced machine learning techniques

- Overall accuracy still a problem in practice
  - False negatives: lose security
  - False positives: alert fatigue

Summary

- Common malware types and classification
  - By propagation behaviors: trojan, virus, worm, …
  - By payload behaviors: backdoor, ransomware, spyware, botnet, rootkit, …

- Common (reactive) defense: IDS
  - Signature-, heuristic-/rule-, anomaly-based