CS165 – Computer Security

Malware

February 22, 2024

Malware

- Adversaries aim to get code running on your computer that performs tasks of their choosing
  - This code is often called malware
- Three main challenges for adversaries
  - How do they get their malware onto your computer?
  - How do they get their malware to run?
  - How do they keep it from being detected?
- Focusing on what happens after initial exploitation

Viruses

- Is an attack that modifies programs on your host
- Approach
  - 1. Download a malware program ...
  - 2. Run the malware ...
  - 3. Searches for binaries and other code (firmware, boot sector) that it can modify ...
  - 4. Modifies these programs by adding code that the program will run
- What can an adversary do with this ability?

Viruses

- How does it work?
  - Modify executable files on your host
    - How does it do that meaningfully?

Viruses

- How does it work?
  - Modify executable files on your host
    - By knowing the executable file format
- Format for an executable file
  - Program loaders expect all binary files to comply with executable format standard (Executable and Linkable Formation, ELF) to load a program correctly
- There are several aspects, but two are important
  - Entrypoint: location to start running your program
  - Sections: parts of code and data

Viruses

- How does it work?
  - Modify executable files on your host
    - By knowing the executable file format
- What types of modifications?
  - Overwrite the program “entrypoint”
    - Add code anywhere and change “entrypoint” to start there
  - Add a new section header
    - Cause code in that section to be invoked
- All these were well known by the 1990s

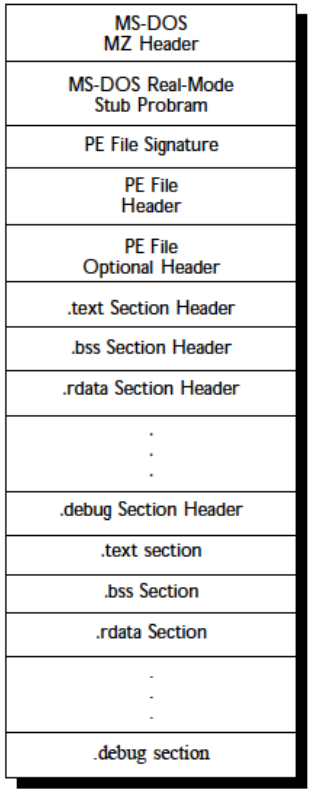


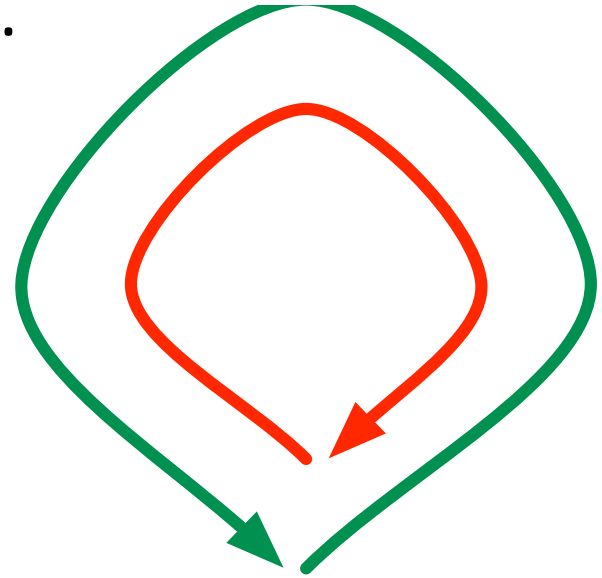
Figure 1. Overall structure of a Portable Executable file image

Virus Infection

- Keeping with the virus analogy, getting a virus to run on a computer system is called infecting the system
  - How can an adversary infect another's computer?
    - Tricking users into downloading their malware
      - E.g., Trojan horse
  - Need to also trick the user into running the malware
    - Exploiting a vulnerable program to inject code
      - E.g., memory errors
- Some systems allow an adversary to do both at once
  - E.g., phishing and email attachments

Worms

- A worm is a self-propagating program.
- As relevant to this discussion
  - 1. Exploits some vulnerability on a target host ...
  - 2. (often) embeds itself into a host ...
  - 3. Searches for other vulnerable hosts ...
  - 4. Goto (1)



- Q: Why do we care?

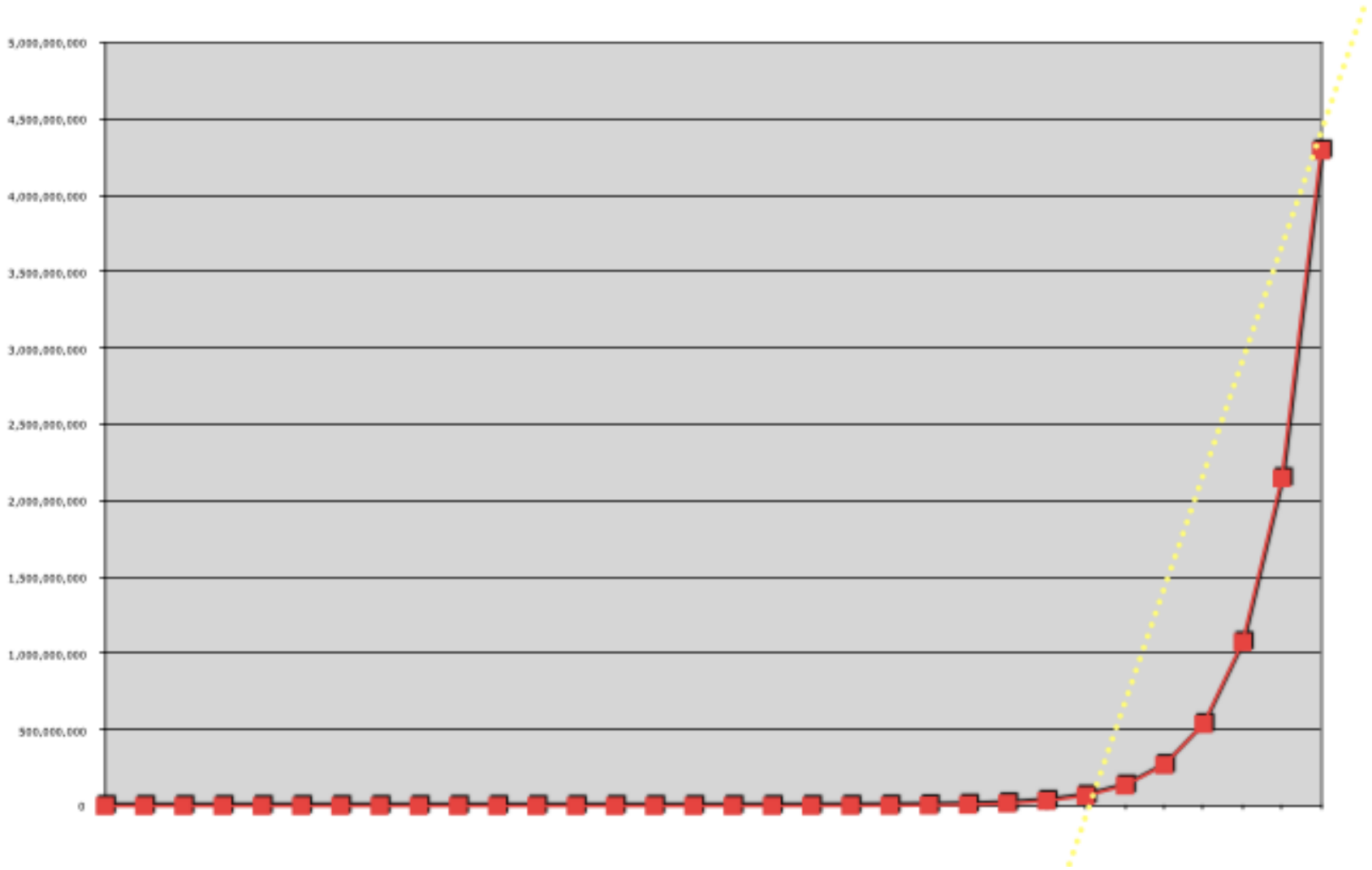
The Danger

- What makes worms so dangerous is that infection grows at an exponential rate
  - A simple model:
    - s (search) is the time it takes to find vulnerable host
    - i (infect) is the time it takes to infect a host
  - Assume that t=0 is the worm outbreak, the number of hosts infected at t=j is?

The Danger

- What makes worms so dangerous is that infection grows at an exponential rate
  - A simple model:
    - s (search) is the time it takes to find vulnerable host
    - i (infect) is the time it takes to infect a host
  - Assume that t=0 is the worm outbreak, the number of hosts infected at t=j is
    - $2^{j/(s+i)}$
- For example, if (s+i = 1), how many infected hosts at time j=32?

The Result


5,000,000,000
4,500,000,000
4,000,000,000
3,500,000,000
3,000,000,000
2,500,000,000
2,000,000,000
1,500,000,000
1,000,000,000
500,000,000
0

Worm Impact

- In the early days, an attacker could exploit a single vulnerability to compromise many machines
  - E.g., Code Red
- Today, worm capabilities are adapted more stealthily

Malware Detection

- Industry has developed to detect malware files when installed on your system
- How to detect a malware virus?
  - Suppose you know all known malware

Malware Detection

- Industry has developed to detect malware files when installed on your system
- How to detect a malware virus?
  - Suppose you know all known malware
    - By "signature" – match all files against known malware

Malware Detection

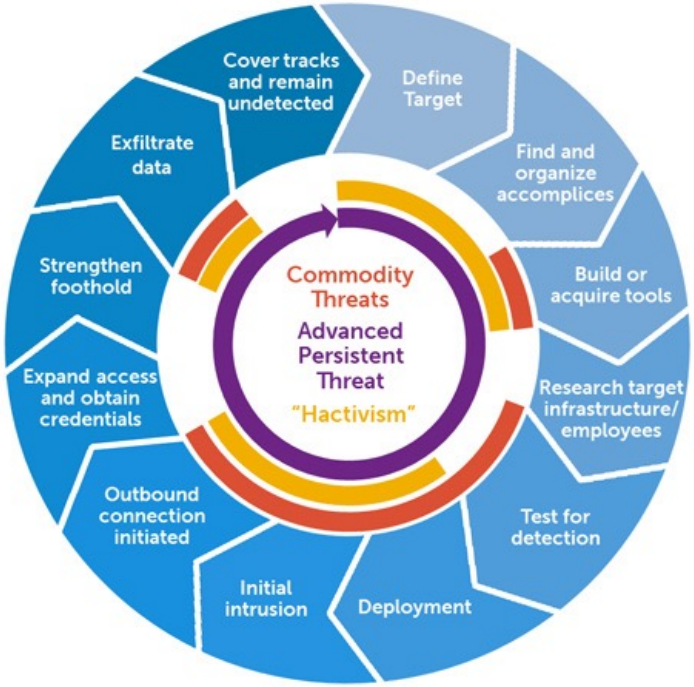
- Industry has developed to detect malware files when installed on your system
- How to detect a malware virus?
  - Suppose you know what the virus does (when run)
    - What can you monitor about a process (malware or not)?

Malware Detection

- Industry has developed to detect malware files when installed on your system
- How to detect a malware virus?
  - Suppose you know what the virus does (when run)
    - System calls (e.g., open a file, write to the file, etc.)
    - Changes to executable files

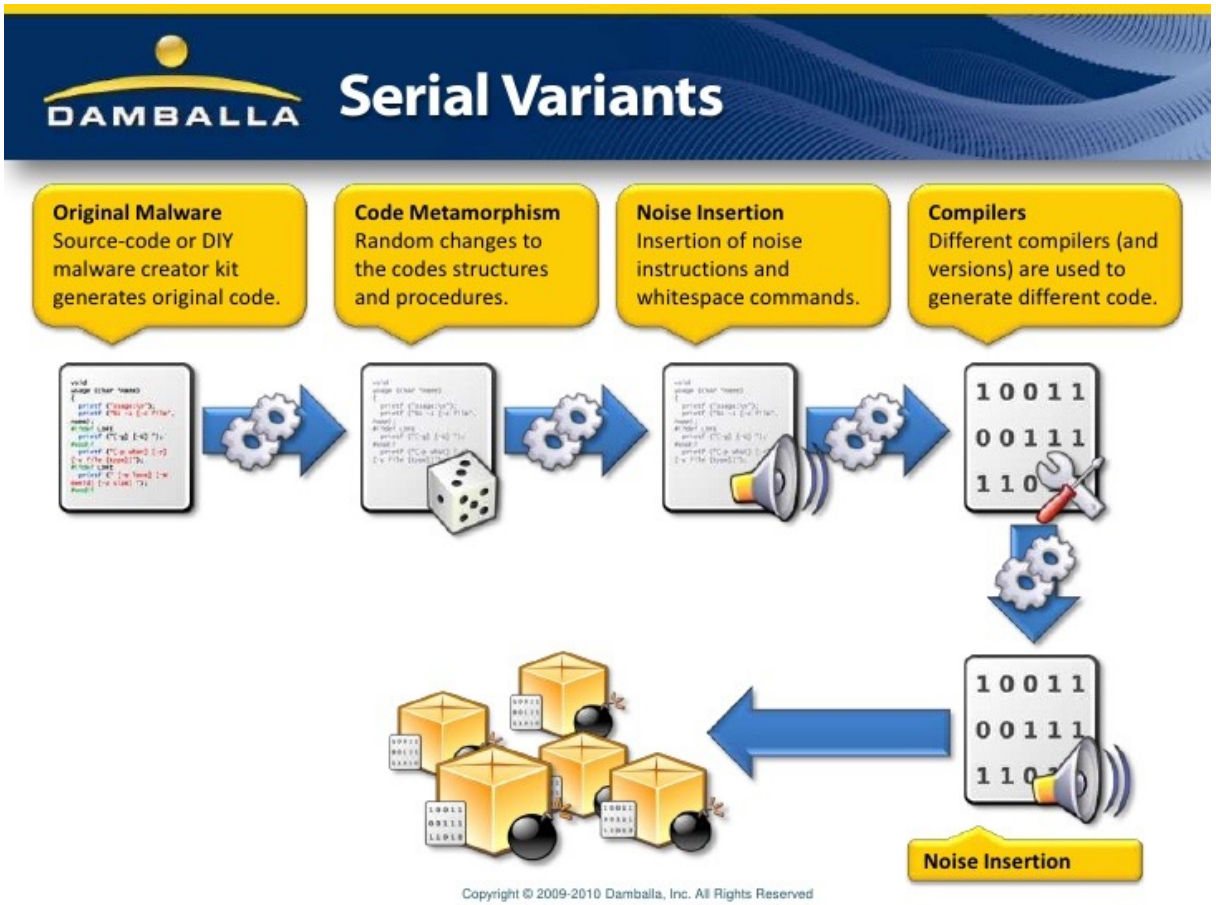
Modern Malware

- Now, malware has a much greater level of sophistication
  - Now we speak of ...
  - Advanced Persistent Malware



Malware Lifecycle

- E.g., create malware variants to bypass detection



Low-And-Slow

- Malware writers are focused on specific task
  - Criminals
  - Cyberwarfare
- Low-and-slow
  - Can exfiltrate secrets at a slow rate, especially if you don't need them right away
  - Plus, can often evade or disable defenses

Example: Sirefef

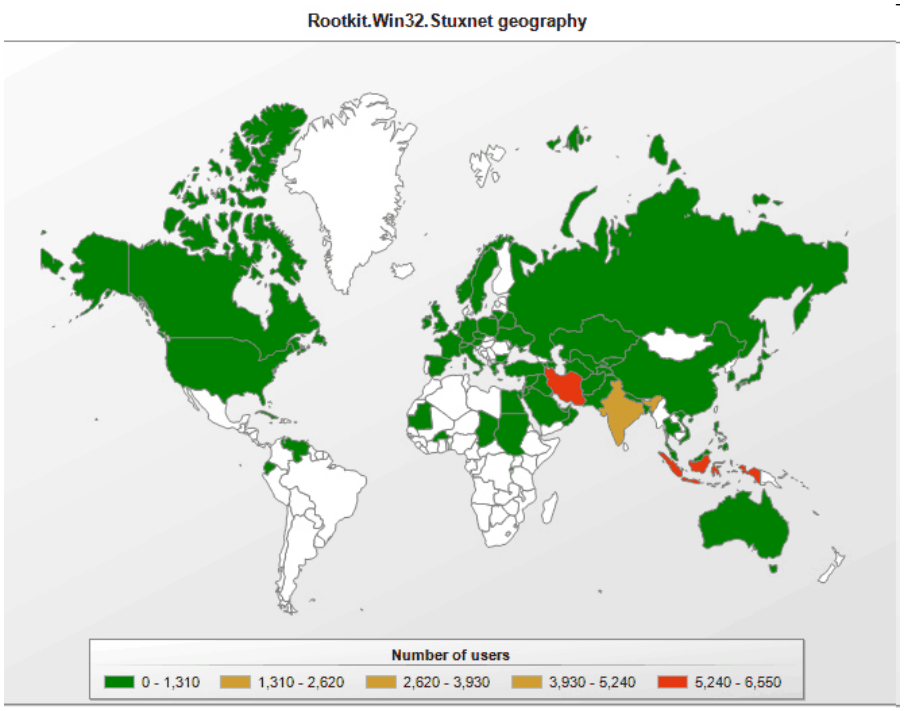
- Windows malware – from fake software update
- Technical summary
  - https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Virus:Win32/Sirefef.R
  - Attack:"Sirefef gives attackers full access to your system"
  - Runs as a Trojan software update (GoogleUpdate)
  - Runs on each boot by setting a Windows registry entry
- Does a variety of malicious things
  - Downloads code to run C&C communication
  - Some versions replace device drivers
  - Steal software keys and crack password for software piracy
  - Downloads other files to propagate the attack to other computers

Example: Sirefef

- Stealthy: “while using stealth techniques in order to hide its presence”
  - “altering the internal processes of an operating system so that your antivirus and anti-spyware can't detect it.”
  - Disables defenses, such as: Windows firewall, Windows defender
  - Changes: Browser settings
  - Changes: Windows registry
    - Resets registry change if manually “fixed”
- Microsoft: “This list is incomplete”

Example: Stuxnet

- Slides from Symantec



Example: Stuxnet

Stuxnet: Overview

- June 2010: A worm targeting Siemens WinCC industrial control system.
- Targets high speed variable-frequency programmable logic motor controllers from just two vendors: Vacon (Finland) and Fararo Paya (Iran)
- Only when the controllers are running at 807Hz to 1210Hz. Makes the frequency of those controllers vary from 1410Hz to 2Hz to 1064Hz.
- http://en.wikipedia.org/wiki/Stuxnet

Example: Stuxnet

- Very carefully designed malware for a specific industrial control environment
  - Fake update using stolen keys from a Windows driver vendor
  - Compromise/disable a variety of antivirus software to evade detection
  - Self-spreading through USB drives installed on infected computers to propagate in an air-gapped system
  - Infect application used to program the programmable logic controllers of centrifuges to inject malicious code
  - Erase malicious code from application's code viewer

Example: Stuxnet

- Stuxnet includes several modern malware facets
  - Reconnaissance: Learn the victim configuration
  - Infection (virus): Trojan device driver and PLC programming application
  - Stealth: Knock out antivirus detection and remove malicious code from GUI
  - Propagation (worm): Through USB drives – no network
- A well-funded adversary can be very difficult to stop

Conclusions

- Adversaries ultimately aim to run their code (malware) on victim systems
- In the early days, infection (viruses) and propagation (worms) were relatively straightforward
- Modern malware has to work around various detection methods (often AI-based these days)
- And aims to remain undetected (stealthy) and stay resident on the victim system (persistent)
  - Advanced persistent threats

Questions