CMSC414 Computer and Network Security

Malware

Yizheng Chen | University of Maryland
surrealyz.github.io

Feb 27, 2024

Agenda

- Malware
- Viruses
- Worms
- Infection cleanup and rootkis

Malware

- **Malware** (**mal**icious soft**ware**): Malicious code that is stored on and runs on victim's system
  - Sometimes called malcode (malicious code)

What can malware do?

- Deletes files
- Sends spam email
- Launches a Denial of Service (DoS) attack
- Steals private information
- Records user inputs (keylogging, screen capture, webcam capture)
- Encrypts files and demands money to decrypt them (ransomware)
- Physically damages machines
- …

How does malware get to run?

- Attacks a user- or network-facing vulnerable service
- Backdoor: Added by a malicious developer for remote access
- Social engineering: Trick the user into running/clicking/installing
- Trojan horse: Offer a good service, add in the bad
- Drive-by download: Webpage surreptitiously installs without user knowing
- Attacker with physical access downloads & runs it

Potentially from any mode of interaction (automated or not), provided sufficient vulnerability

When does malware run?

- **Some delay based on a trigger**
  - Time bomb: triggered at/after a certain time, e.g., 1st through the 19th of any month…
  - Logic bomb: triggered when a set of conditions hold, e.g., If I haven't appeared in two consecutive payrolls…
  - Can also include a backdoor to serve as ransom, e.g., "I won't let it delete your files if you pay me by Thursday…"

When does malware run?

- **Some delay based on a trigger**
  - Time bomb: triggered at/after a certain time, e.g., 1st through the 19th of any month…
  - Logic bomb: triggered when a set of conditions hold, e.g., If I haven't appeared in two consecutive payrolls…
  - Can also include a backdoor to serve as ransom, e.g., "I won't let it delete your files if you pay me by Thursday…"
- **Some attach themselves to other pieces of code**
  - Viruses: run when the user initiates something, e.g., Run a program, open an attachment, boot the machine
  - Worms: run while another program is running. No user intervention required

Self-Replicating Code

- Propagation: Spread copies of the code from machine to machine
- **Self-replicating code**: A code snippet that outputs a copy of itself
- Can be used to automatically propagate malware
  - When malware is run, the self-replicating code outputs a copy of itself and sends the code to other computers

Viruses and Worms

- Viruses and worms are both malware that automatically self-propagate
- **Virus**: Code that requires user action to propagate
  - Usually infects a computer by altering some stored code
  - When the user runs the code, the code spreads the virus to other users
- **Worm**: Code that does not require user action to propagate
  - Usually infects a computer by altering some already-running code
  - No user interaction required for the worm to spread to other users

Viruses and Worms

- Viruses and worms are both malware that automatically self-propagate
- **Virus**: Code that **requires user action** to propagate
  - Usually infects a computer by altering some stored code
  - When the user runs the code, the code spreads the virus to other users
- **Worm**: Code that **does not require user action** to propagate
  - Usually infects a computer by altering some already-running code
  - No user interaction required for the worm to spread to other users
- The difference between a virus and a worm is not always clear
  - Some malware uses both approaches together
  - Example: Trojan malware does not self-propagate, but instead requires user action

Malware: Technical Challenges

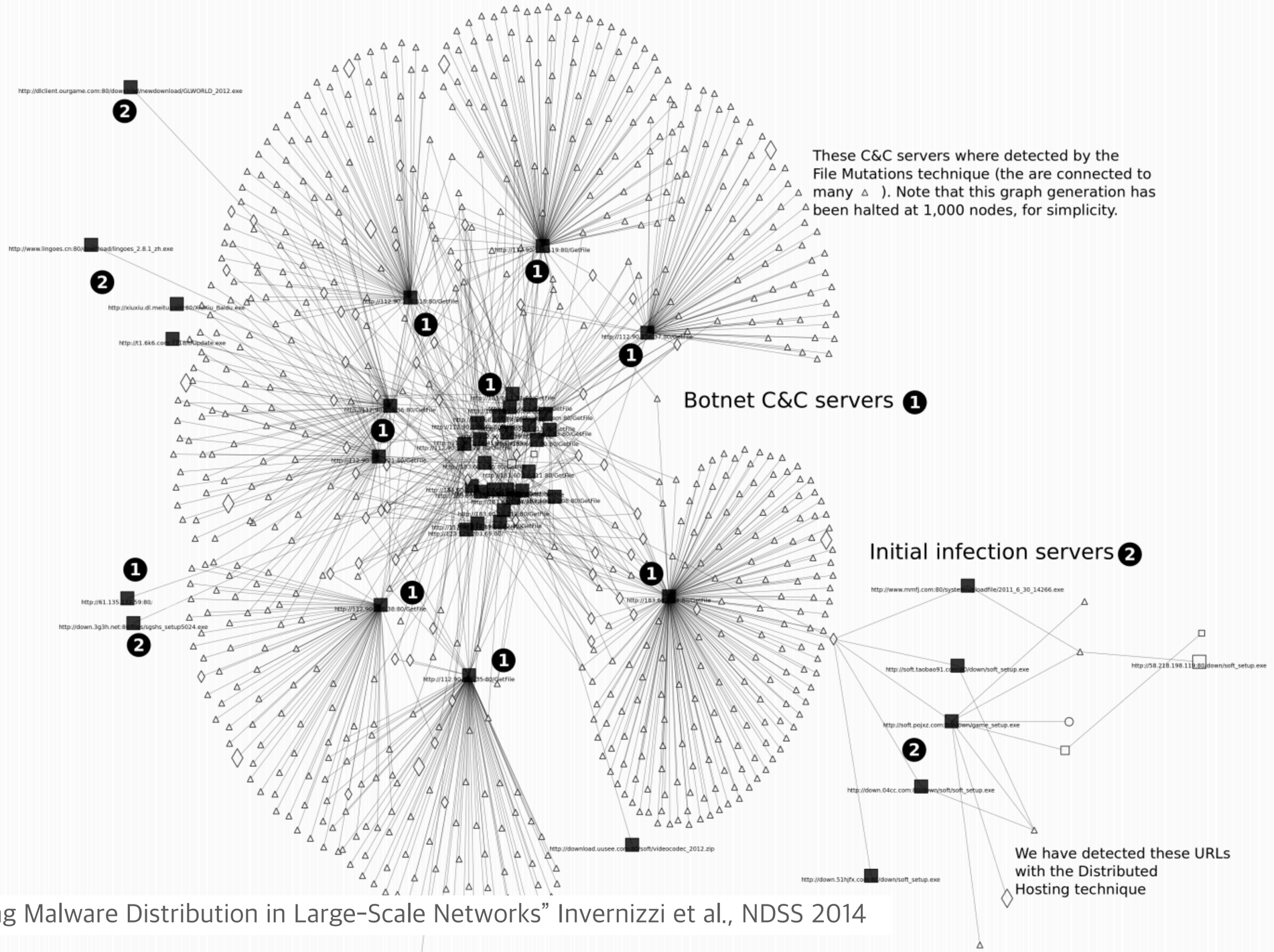
- **Viruses: Detection**
  - Antivirus software wants to detect
  - Virus writers want to avoid detection for as long as possible
  - **Evade** human response
- **Worms: Spreading**
  - The goal is to hit as many machines and as quickly as possible
  - **Outpace** human response

Botnets

- **Botnet**: A set of compromised machines (“bots”) under central control
  - Use a virus or a worm to infect many different computers
  - Every infected computer is now under the attacker’s control
  - A huge amount of resources (e.g. can be used for DoS)

Botnets

- **Botnet**: A set of compromised machines (“bots”) under central control
  - Use a virus or a worm to infect many different computers
  - Every infected computer is now under the attacker’s control
  - A huge amount of resources (e.g. can be used for DoS)
- **C&C Server**: A command-and-control (C&C) server is a computer controlled by an attacker or cybercriminal which is used to send commands to systems compromised by malware



"Nazca: Detecting Malware Distribution in Large-Scale Networks" Invernizzi et al., NDSS 2014

Viruses

- **Virus**: Code that requires user action to propagate
  - Usually infects a computer by altering some stored code
  - When the user runs the code, the code spreads the virus to other users

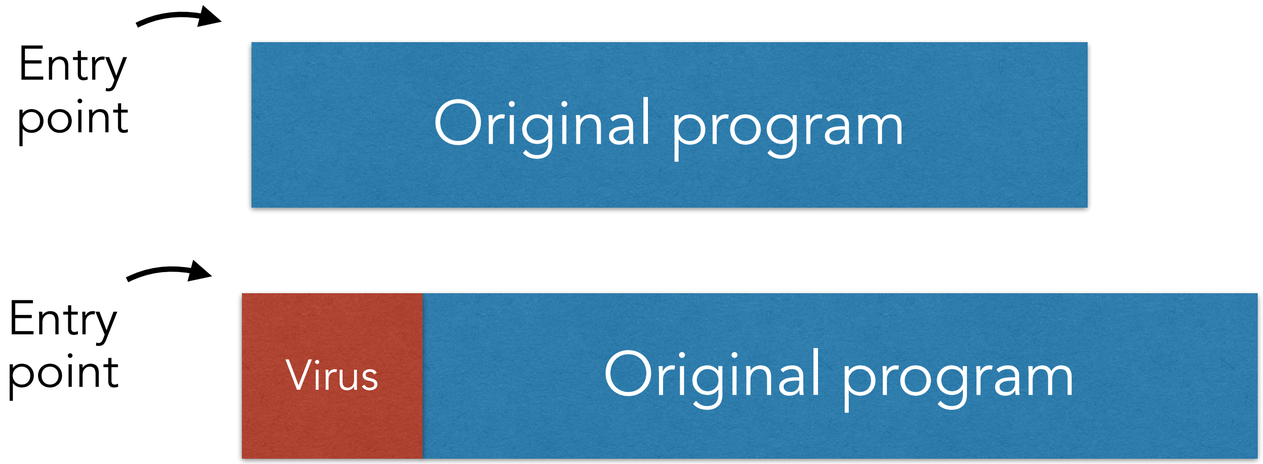
Viruses are classified by what they infect

- Infect / Modify existing code that will eventually be executed by the user
  - **Document Viruses**: Code that runs when the user opens an attachment
    - Implemented within a formatted document
    - Word documents (very rich macros)
    - PDF (Acrobat permits javascript)
    - **(Why you shouldn't open random attachments)**
  - **Boot Sector Viruses**: Code that runs when the system starts up
    - Boot sector: small disk partition at a fixed location, supposed to load the OS -> loads the virus
    - Similar: any AutoRun
    - **(Why you shouldn't plug random USB drives into your computer)**
  - **Mobile App Viruses**: Code that runs when opening an app

Propagation Strategies

- When the malcode runs, it looks for opportunities to infect more systems
  - Example: Send emails to other users with the code attached
  - Example: Copy the code to a USB flash drive (so any other users who run the files on the USB drive will be infected too)
  - Again: Don’t open random attachments! Don’t plug in random USB drives!

How Viruses Affect Other Programs



Take over the entry point

Detection Strategies

- Signature-based detection
  - Viruses replicate by using copies of the same code
  - Capture a virus on one system and look for **bytes corresponding to the virus code** on other systems
  - Example: YARA rules, can match hex code, regex, multiple conditions, etc

Simple Example: Detect Strings that Demand Money

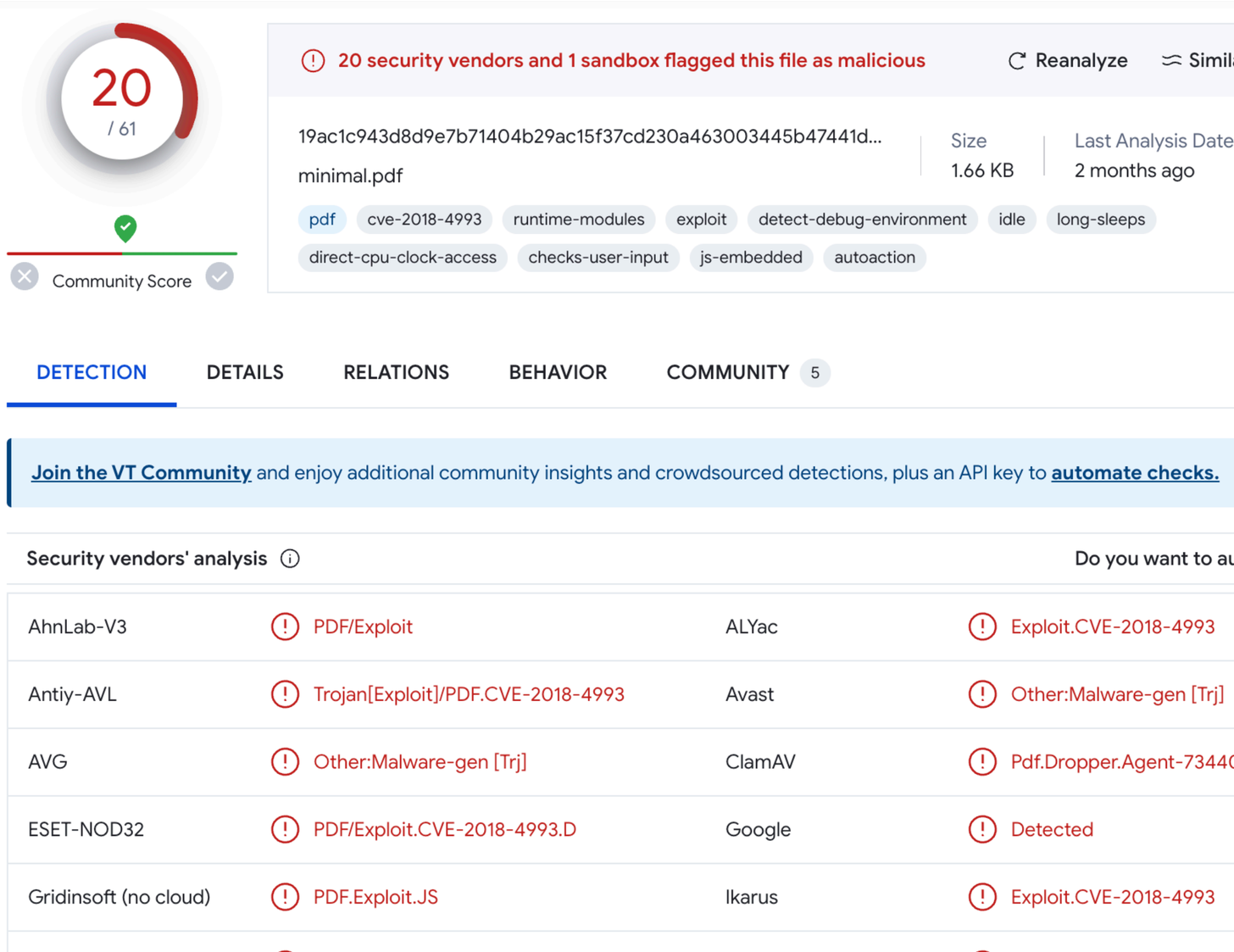
```
1 rule Example_One
2 {
3     strings:
4         $string1 = "pay"
5         $string2 = "immediately"
6
7     condition:
8         ($string1 and $string2)
9 }
```

Simple Example: Prevent specific website links or names

```
1  rule Example_Two
2  {
3     strings:
4        $MaliciousWeb1 = "www.scamwebsite.com"
5        $MaliciousWeb2 = "www.notrealwebsite.com"
6        $Maliciousweb3 = "www.freemoney.com"
7        $AttackerName1 = "hackx1203"
8        $AttackerName2 = "Hackor"
9        $AttackerName3 = "Hax"
10
11     condition:
12        any of them
13 }
```

Antivirus Software

- Antivirus software usually includes a checklist of common viruses
- Example on the right from VirusTotal:
  - 20 out of 61 AV engines detected this file as malicious



Arms Race

Mechanisms for evasive **propagation**

Mechanisms for **detection** and prevention

Want to be able to claim wide coverage for a long time

Want to be able to claim the ability to detect *many* viruses

- Attackers look for **evasion strategies**
- This arms race has influenced the **evolution** of modern malware

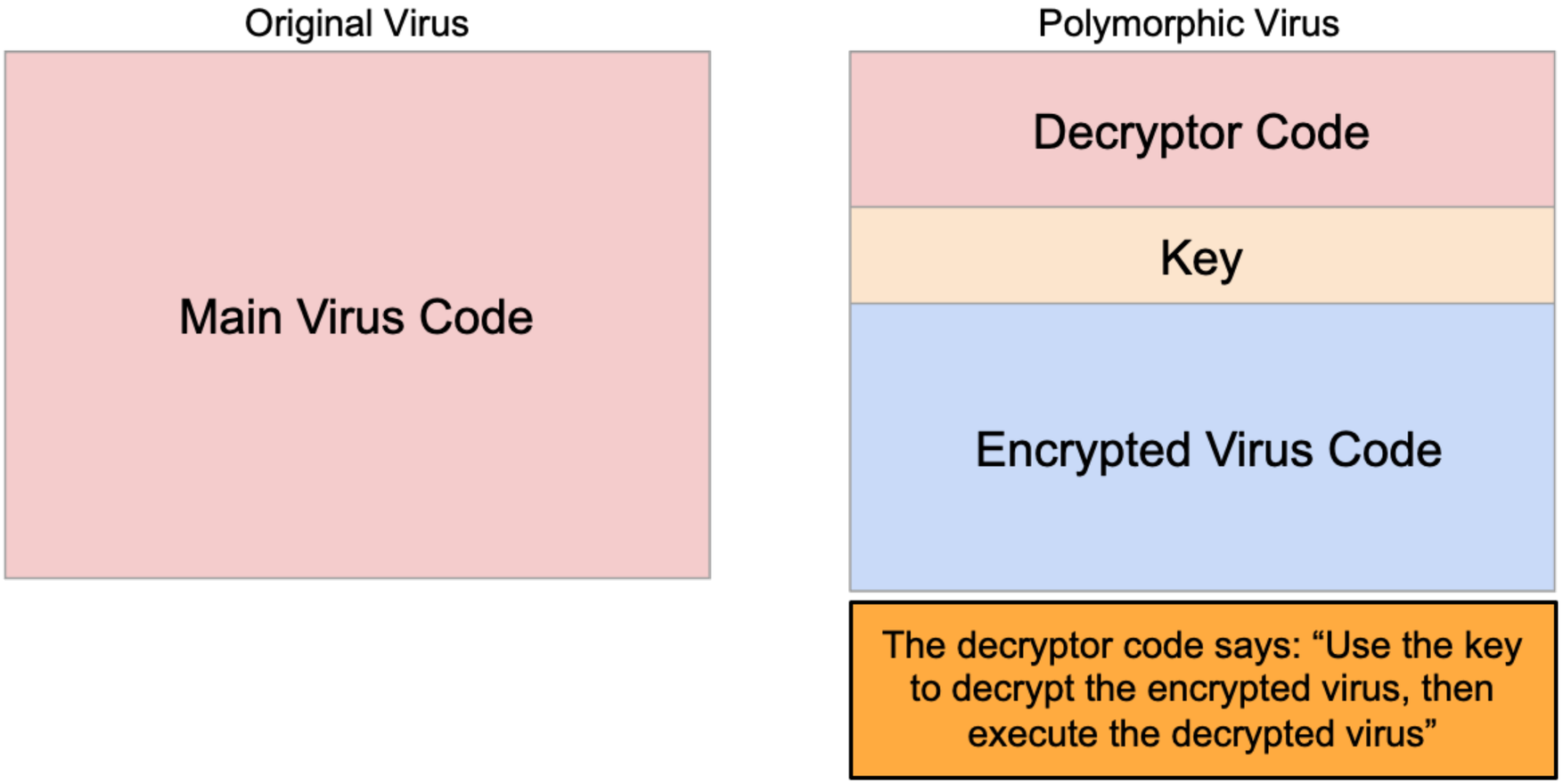
Polymorphic Code

- **Polymorphic code**: Each time the virus propagates, it inserts an encrypted copy of the code
  - The code also includes the key and decryptor
  - When the code runs, it uses the key and decryptor to obtain the original malcode
- Encryption schemes can produce different output on repeated encryptions
  - Example: Using a different key for each encryption

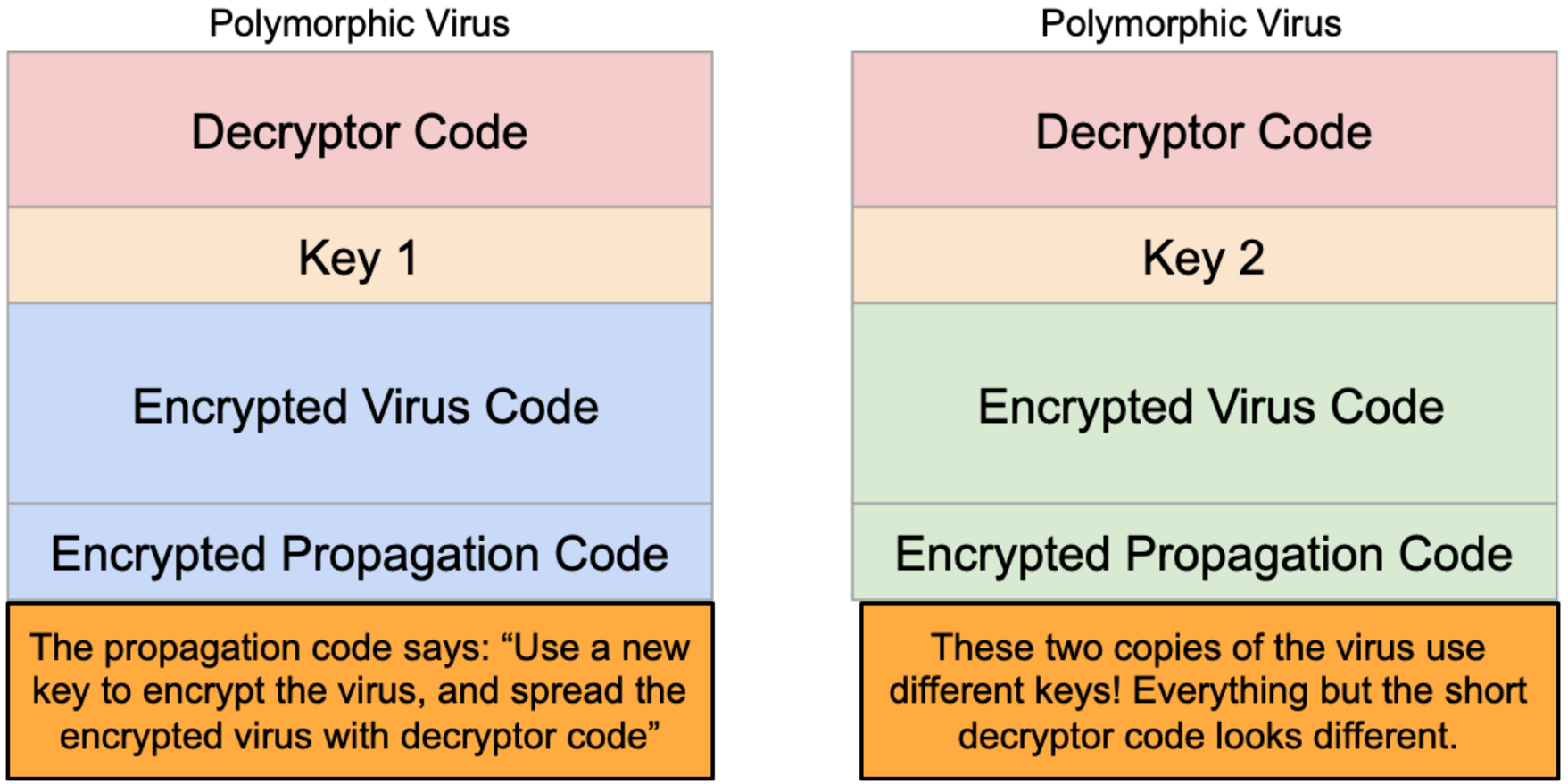
Polymorphic Code

- **Polymorphic code**: Each time the virus propagates, it inserts an encrypted copy of the code
  - The code also includes the key and decryptor
  - When the code runs, it uses the key and decryptor to obtain the original malcode
- Encryption schemes can produce different output on repeated encryptions
  - Example: Using a different key for each encryption
- Encryption is being used for **obfuscation**, not confidentiality
  - The goal is to evade detection by making the virus look different
  - The goal is not to prevent anyone from reading the virus contents
  - Weaker encryption algorithms can be used, and the key can be stored in plaintext

Polymorphic Code



Polymorphic Code



Polymorphic Code: Defenses

- Strategy #1: Add a signature for detecting the decryptor code
  - Issue: Less code to match against → More false positives
  - Issue: The decryptor code could be scattered across different parts of memory

Polymorphic Code: Defenses

- Strategy #1: Add a signature for detecting the decryptor code
  - Issue: Less code to match against → More false positives
  - Issue: The decryptor code could be scattered across different parts of memory
- Strategy #2: Safely check if the code performs decryption
  - Execute the code in a sandbox
  - Analyze the code structure without executing the code
  - Issue: Legitimate programs might perform similar operations too (e.g. decompressing ZIP files)
  - Issue: How long do you let the code execute? The decryptor might only execute after a long delay.

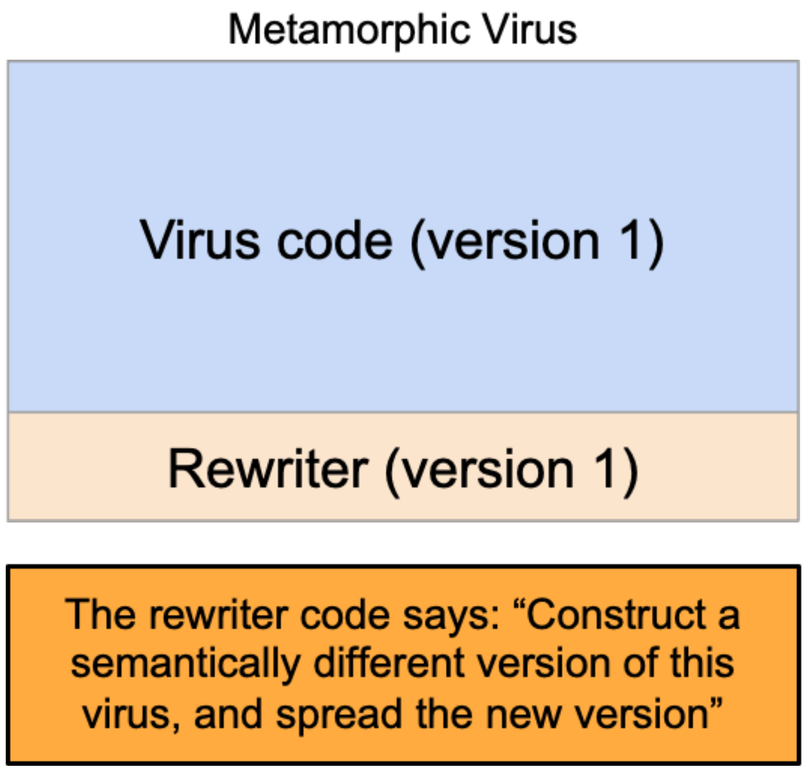
Arms Race: How to Evade?

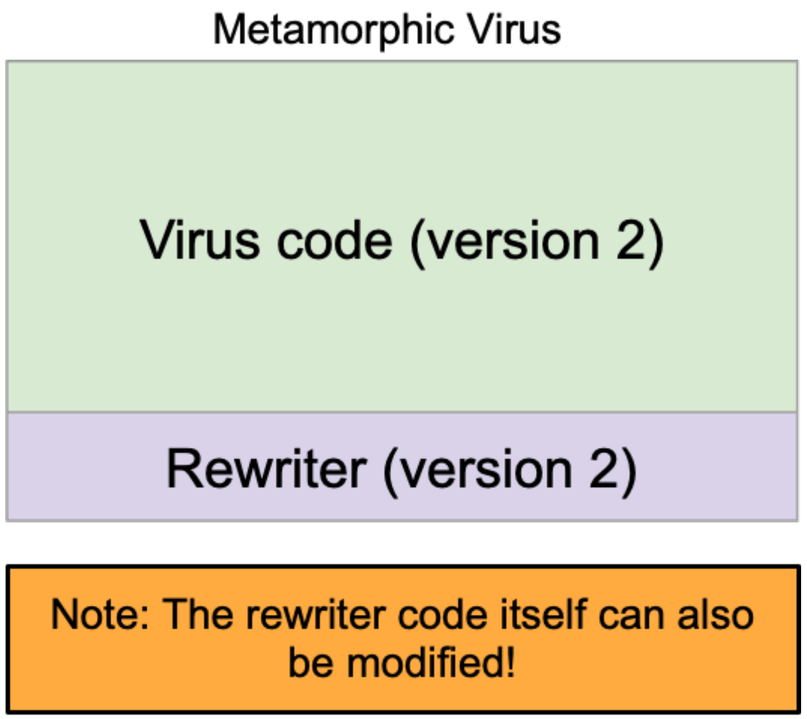
- Idea #1: Change the decrypter
  - True polymorphic viruses: use endless number of decrypters
- Idea #2: Change the decrypted code itself

Metamorphic Code

- **Metamorphic code**: Each time the virus propagates, it generates a **semantically different version** of the code
  - The code performs the **same high-level action**, but with **minor differences in execution**
  - Difference in low-level semantics
- Include a code rewriter with the virus to change the code randomly each time
  - Renumber registers
  - Change order of conditional (if/else) statements
  - Reorder independent operations
  - Replace a low-level algorithm with another (e.g. mergesort and quicksort)
  - Add some code that does nothing useful (or is never executed)

Metamorphic Code





```
5A              pop  edx
BF04000000      mov  edi,0004h
8BF5            mov  esi,ebp
B80C000000      mov  eax,000Ch
81C288000000    add  edx,0088h
8B1A            mov  ebx,[edx]
899C8618110000  mov  [esi+eax*4+00001118],ebx

58              pop  eax
BB04000000      mov  ebx,0004h
8BD5            mov  edx,ebp
BF0C000000      mov  edi,000Ch
81C088000000    add  eax,0088h
8B30            mov  esi,[eax]
89B4BA18110000  mov  [edx+edi*4+00001118],esi
```

Figure 4: Win95/Regswap using different registers in new generations

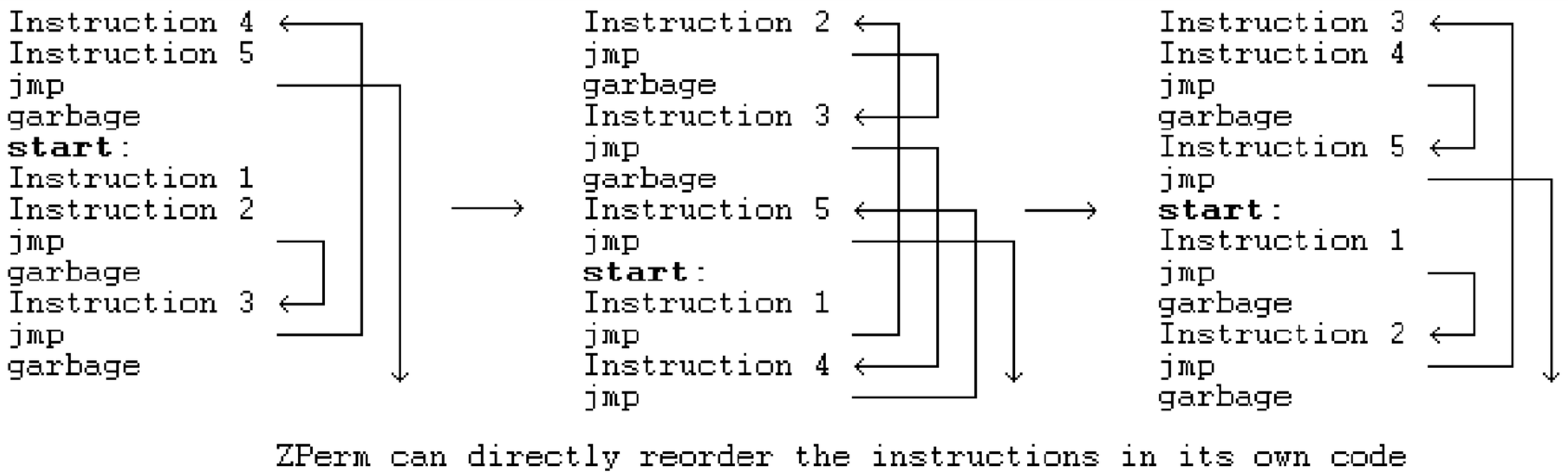


Figure 7. Zperm.A inserts JMP instruction into its code

```
a. An early generation:

C7060F000055   mov       dword ptr [esi],5500000Fh
C746048BEC5151 mov       dword ptr [esi+0004],5151EC8Bh

b. And one of its later generations:

BF0F000055     mov       edi,5500000Fh
893E           mov       [esi],edi
5F             pop       edi
52             push      edx
B640           mov       dh,40
BA8BEC5151     mov       edx,5151EC8Bh
53             push      ebx
8BDA           mov       ebx,edx
895E04         mov       [esi+0004],ebx

c. And yet another generation with recalculated ("encrypted") "con-
stant" data.

BB0F000055     mov       ebx,5500000Fh
891E           mov       [esi],ebx
5B             pop       ebx
51             push      ecx
B9CB00C05F     mov       ecx,5FC000CBh
81C1C0EB91F1   add       ecx,F191EBC0h ; ecx=5151EC8Bh
894E04         mov       [esi+0004],ecx
```

Figure 6: Example of code metamorphosis of Win32/Evol

Metamorphic Code: Defense

- Behavioral detection
  - Need to analyze **behavior** instead of **syntax**
  - Look at the effect of the instructions, not the appearance of the instructions
  - Antivirus company analyzes a new virus to find a **behavioral signature**

- Example: Ransomware encrypts files *and* changes the victim's desktop background
- Really hard to craft behavioral signatures

Metamorphic Code: Defense

- Behavioral detection
  - Need to analyze **behavior** instead of **syntax**
  - Look at the effect of the instructions, not the appearance of the instructions
  - Antivirus company analyzes a new virus to find a **behavioral signature**
- Subverting behavioral detection
  - Delay analysis by waiting a long time before executing malcode
  - Detect that the code is being analyzed (e.g. running in a debugger or a virtual machine) and choose different behavior
  - Antivirus can look for these subversion strategies and skip over them

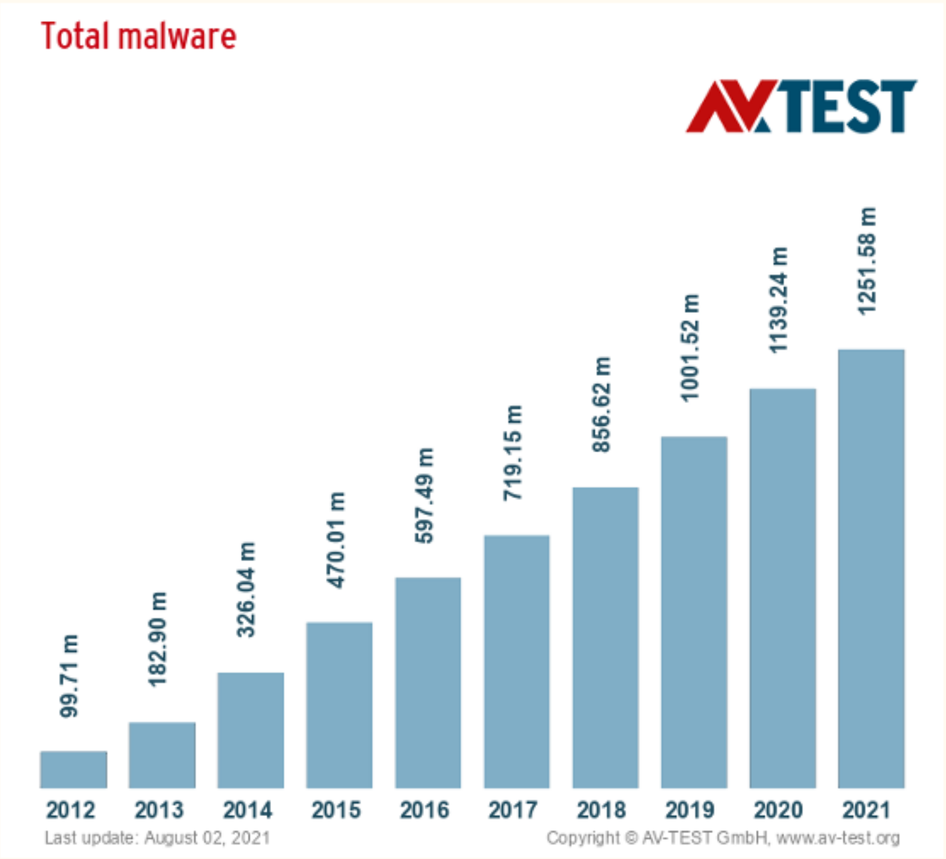
Defense: Flag Unfamiliar Code

- It is impossible to write a perfect algorithm to separate malicious code from safe code
- Antivirus software can look for new, unfamiliar code
  - Keep a central repository of previously-seen code
  - If some code has never been seen before, treat it as more suspicious
  - The central repository can store secure cryptographic hashes of previously-seen code snippets for efficiency (the software hashes code and see if the hash matches a hash in the repository)
- Issue: false positives

Virus Counts May Be Exaggerated

AV-TEST
The Independent IT-Security Institute
Magdeburg Germany

Every day, the AV-TEST Institute registers over 350,000 new malicious programs (malware) and potentially unwanted applications (PUA). These are examined and classified according to their characteristics and saved. Visualization programs then transform the results into diagrams that can be updated and produce current malware statistics.



Takeaway: Antivirus companies might overcount different versions of one virus

Agenda

- Worms
- Infection cleanup and rootkis

Worms

- **Worm**: Malware code that does not require user action to propagate
  - Usually infects a computer by altering some already-running code
  - Unlike malware, no user interaction is required for the worm to spread to other users

Propagation Strategies

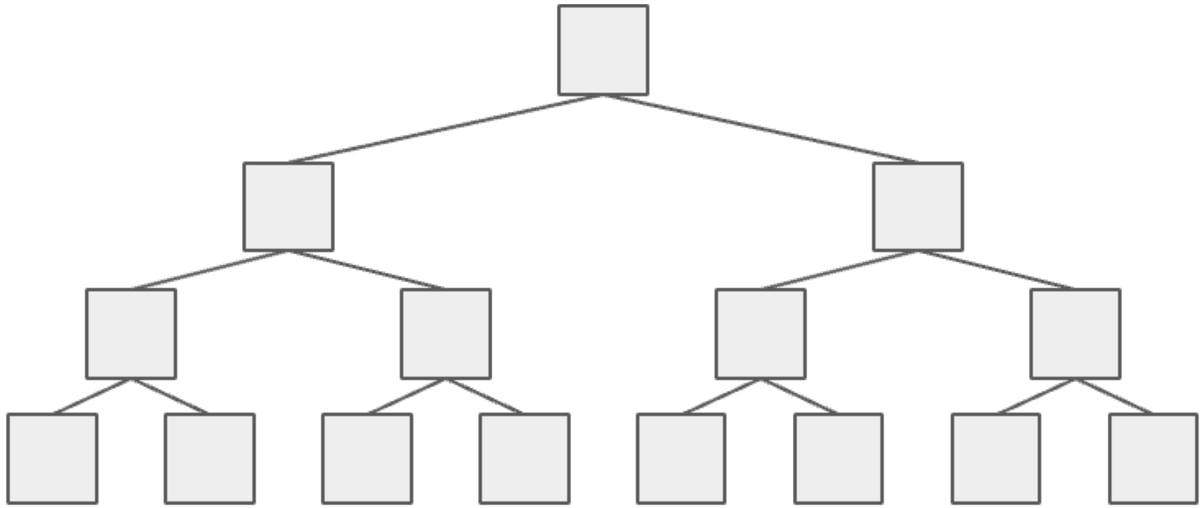
- How does the worm find new users to infect?
  - **Randomly choose machines**: generate a random 32-bit IP address and try connecting to it
  - **Search worms**: Use Google searches to find victims
  - **Scanning**: Look for targets (can be limited by bandwidth)
  - **Target lists**
    - Pre-generated lists (hit lists)
    - Lists of users stored on infected hosts
    - Query a third-party server that lists other servers
  - **Passive**: Wait for another user to contact you, and reply with the infection

Propagation Strategies

- How does the worm find new users to infect?
  - **Randomly choose machines**: generate a random 32-bit IP address and try connecting to it
  - **Search worms**: Use Google searches to find victims
  - **Scanning**: Look for targets (can be limited by bandwidth)
  - **Target lists**
    - Pre-generated lists (hit lists)
    - Lists of users stored on infected hosts
    - Query a third-party server that lists other servers
  - **Passive**: Wait for another user to contact you, and reply with the infection
- How does the worm force code to run?
  - Buffer overflows for code injection
  - A web worm might propagate with an XSS vulnerability

Modeling Worm Propagation

- Worms can potentially spread extremely quickly because they parallelize the process of propagating/replicating
- More computers infected = more computers to spread the worm further
- Viruses have the same property, but usually spread more slowly, since user action is needed to activate the virus



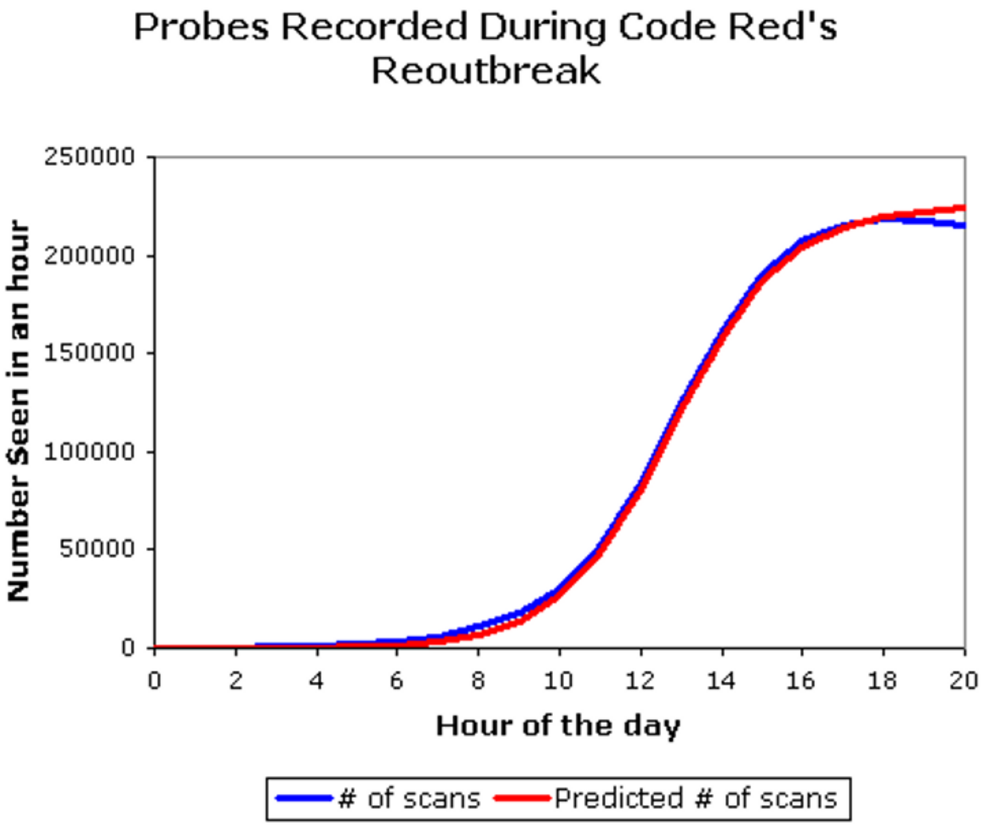
If each infected computer can infect two more computers, we get exponential growth!

Modeling Worm Propagation

- The number of infected hosts grows **logistically**
  - **Initial growth is exponential**:

    More infected hosts = more opportunities to infect
  - **Later growth slows down**: Harder to find new non-infected hosts to infect

- Logistic growth is a good model for worm propagation
  - e.g., Code Red Worm, DoS attack against the White House



Infection Cleanup

- If we find malware on a system, how do we get rid of it?
- May require restoring and repairing many files
  - Antivirus companies sell software that helps with disinfection
- What if the malware executed with administrator privileges?
  - The entire computer is potentially compromised
  - The operating system might be compromised too
  - Best solution: Rebuild the system from data backups and a fresh copy of the operating system
- What if malware infected the tools used to rebuild the operating system?
  - There is no good way of cleaning up malware using only tools in the system!

Rootkit

- Rootkit: Malcode in the operating system that hides its presence
  - Note that the operating system controls disk storage, running processes, etc.
- Rootkits are can be very hard to detect and eliminate
  - Often the best recovery solution is to delete everything and start over