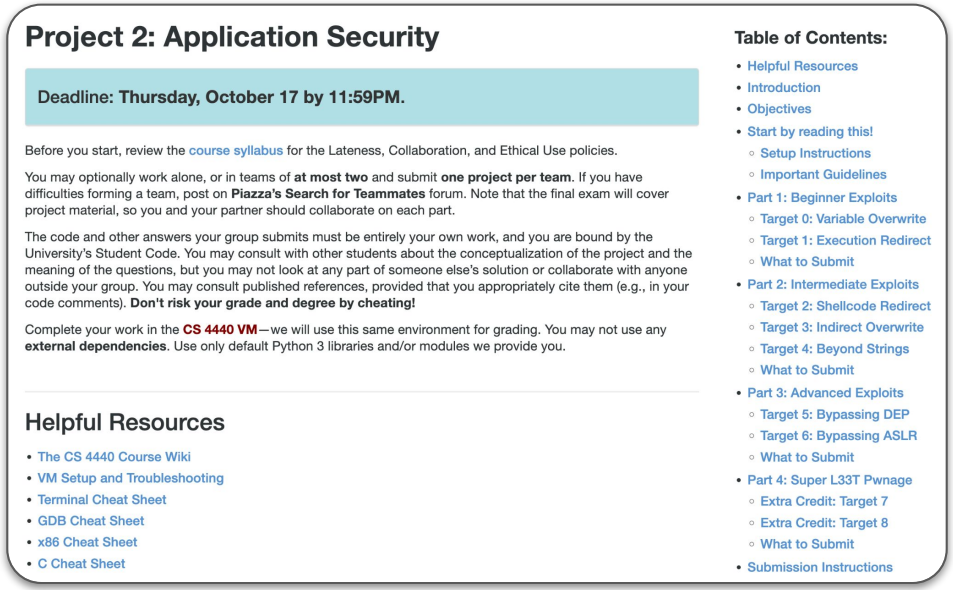
Week 7: Lecture B

Security in Practice: Malware

Thursday, October 3, 2024

Announcements

- **Project 2: AppSec** released
  - **Deadline:** Thursday, October 17th by 11:59PM



Project 2: Application Security

Deadline: Thursday, October 17 by 11:59PM.

Before you start, review the course syllabus for the Lateness, Collaboration, and Ethical Use policies.

You may optionally work alone, or in teams of **at most two** and submit **one project per team**. If you have difficulties forming a team, post on **Piazza's Search for Teammates** forum. Note that the final exam will cover project material, so you and your partner should collaborate on each part.

The code and other answers your group submits must be entirely your own work, and you are bound by the University's Student Code. You may consult with other students about the conceptualization of the project and the meaning of the questions, but you may not look at any part of someone else's solution or collaborate with anyone outside your group. You may consult published references, provided that you appropriately cite them (e.g., in your code comments). **Don't risk your grade and degree by cheating!**

Complete your work in the **CS 4440 VM**—we will use this same environment for grading. You may not use any **external dependencies**. Use only default Python 3 libraries and/or modules we provide you.

Helpful Resources

- The CS 4440 Course Wiki
- VM Setup and Troubleshooting
- Terminal Cheat Sheet
- GDB Cheat Sheet
- x86 Cheat Sheet
- C Cheat Sheet

Table of Contents:

Project 2 Progress Update

Working on Targets 0–2 0%

Working on Targets 3–4 0%

Working on Targets 5–6 0%

Finished! 0%

Haven't started :(0%

Announcements

utahsec

See Discord for meeting info!

utahsec.cs.utah.edu

Questions?

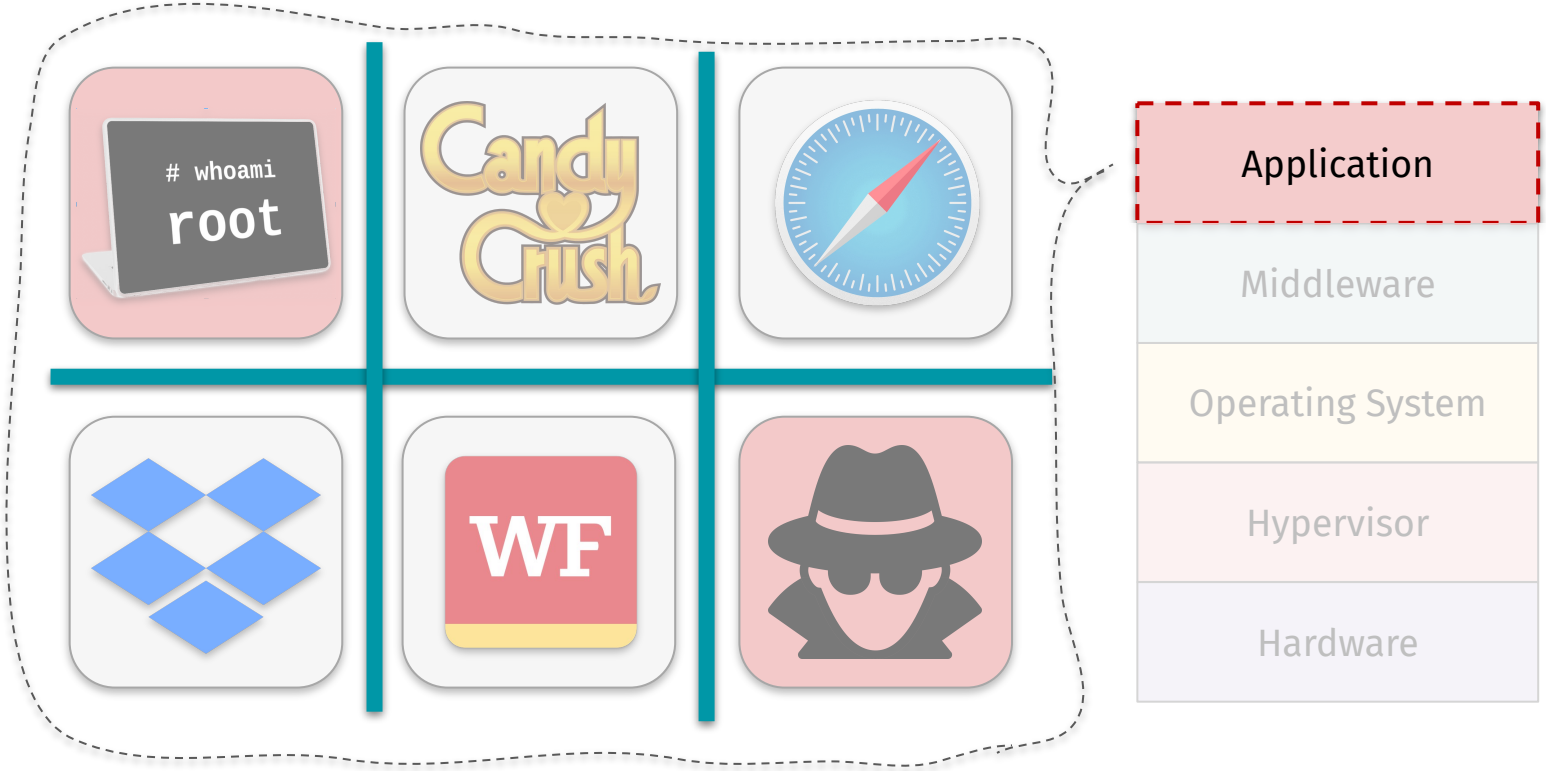
Last time on CS 4440...

Access Control
Permissions
Process Isolation

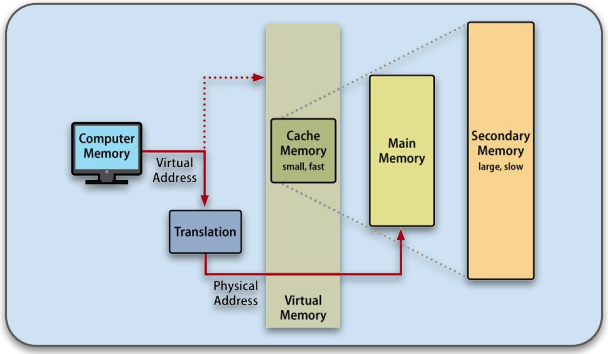
Isolating Applications

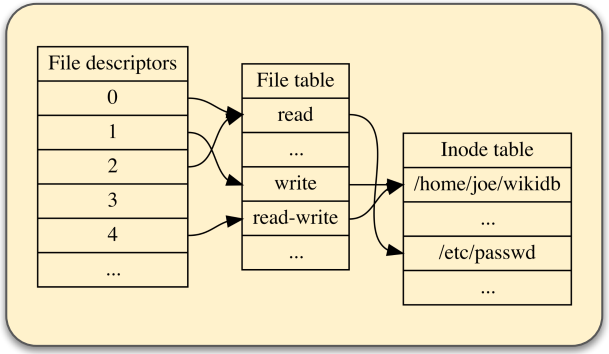
Isolating Applications

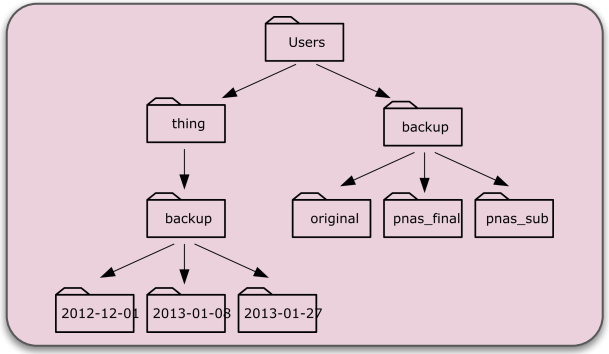
Isolating Applications

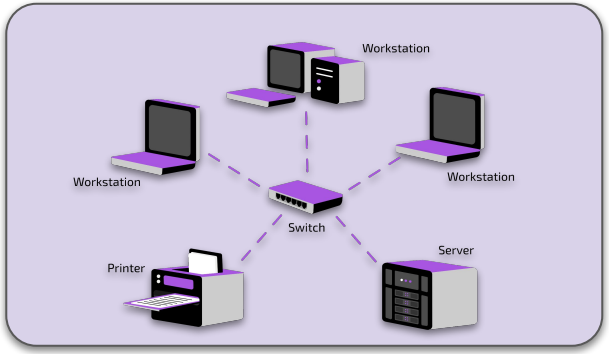


What must we protect?









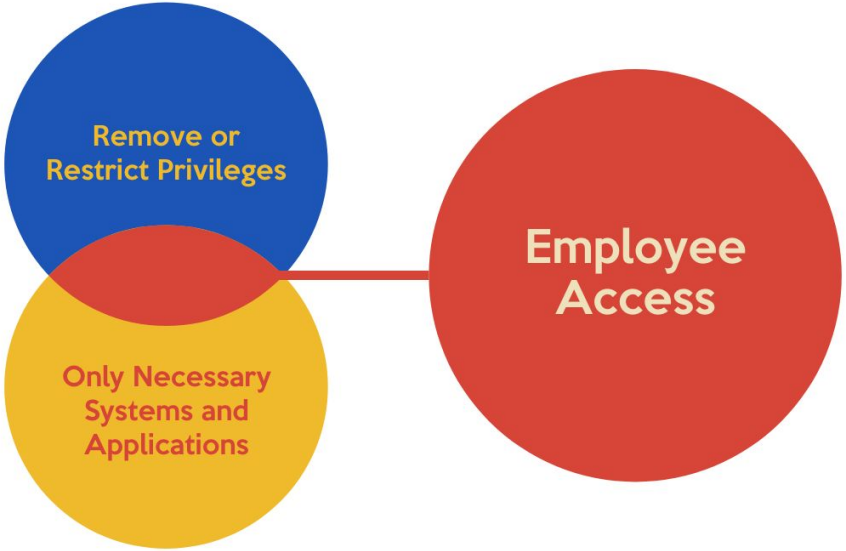
How should we protect them?

- **Principle of Least Privilege:**

Only allow **access** to **resources** that are **absolutely necessary**

How should we protect them?

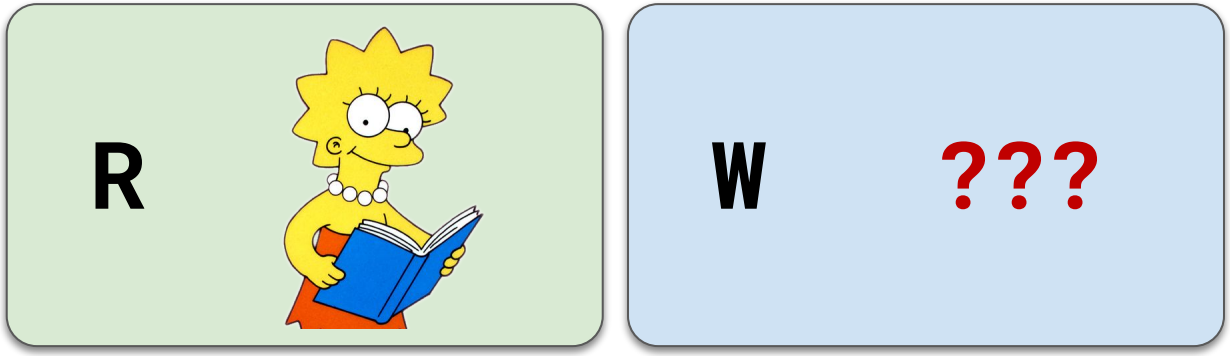
- **Principle of Least Privilege:**



Access Control

R ???

Access Control

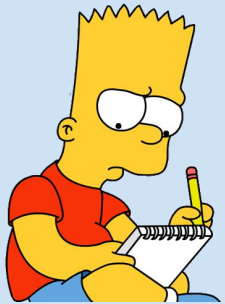


Read a file in directory D

Access Control

R

W



X ???

Read a file in directory D

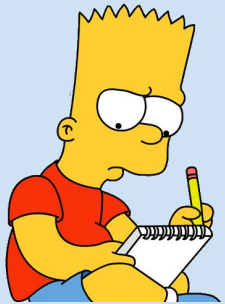
Write a file in directory D

Access Control

R

Read a file in directory D

W



Write a file in directory D

X

Execute a file in D

Implementing Access Control

```
drwxrwxr-x  2 cs4440 cs4440 bin
drwxrwxr-x  2 cs4440 cs4440 __pycache__
-rwxrwxr-x  1 cs4440 cs4440 shellcode.py
```

D = **???**

Implementing Access Control

```
drwxrwxr-x  2 cs4440 cs4440 bin
drwxrwxr-x  2 cs4440 cs4440 __pycache__
-rwxrwxr-x  1 cs4440 cs4440 shellcode.py
```

D = Directory (or a file, if “-”)

Implementing Access Control

```
drwxrwxr-x  2 cs4440 cs4440 bin
drwxrwxr-x  2 cs4440 cs4440 __pycache__
-rwxrwxr-x  1 cs4440 cs4440 shellcode.py
```

D = Directory (or a file, if “-”)

First three = **???**

Implementing Access Control

```
drwxrwxr-x  2 cs4440 cs4440 bin
drwxrwxr-x  2 cs4440 cs4440 __pycache__
-rwxrwxr-x  1 cs4440 cs4440 shellcode.py
```

D = Directory (or a file, if “-”)

First three = **owner’s** permissions

Implementing Access Control

```
drwxrwxr-x  2 cs4440 cs4440 bin
drwxrwxr-x  2 cs4440 cs4440 __pycache__
-rwxrwxr-x  1 cs4440 cs4440 shellcode.py
```

D = Directory (or a file, if “-”)

First three = **owner’s** permissions

Second three = **???**

Implementing Access Control

```
drwxrwxr-x  2 cs4440 cs4440 bin
drwxrwxr-x  2 cs4440 cs4440 __pycache__
-rwxrwxr-x  1 cs4440 cs4440 shellcode.py
```

D = Directory (or a file, if “-”)

First three = **owner’s** permissions

Second three = **group’s** permissions

Implementing Access Control

```
drwxrwxr-x  2 cs4440 cs4440 bin
drwxrwxr-x  2 cs4440 cs4440 __pycache__
-rwxrwxr-x  1 cs4440 cs4440 shellcode.py
```

D = Directory (or a file, if "-")

First three = **owner's** permissions

Second three = **group's** permissions

Last three = **???**

Implementing Access Control

```
drwxrwxr-x  2 cs4440 cs4440 bin
drwxrwxr-x  2 cs4440 cs4440 __pycache__
-rwxrwxr-x  1 cs4440 cs4440 shellcode.py
```

D = Directory (or a file, if "-")

First three = **owner's** permissions

Second three = **group's** permissions

Last three = **the world's** permissions

More Permission Puzzles!

1. Read/Write/Exec for all but group?

   ??? ??? ???

2. Read and Write only for world?

   ??? ??? ???

3. Execute only for group?

   ??? ??? ???

4. Owner can read, write, & exec; Group can only exec; and all others have no permissions.

   ??? ??? ???

More Permission Puzzles!

1. Read/Write/Exec for all but group?

```
rwx --- rwx
```

2. Read and Write only for world?

```
--- --- rw-
```

3. Execute only for group?

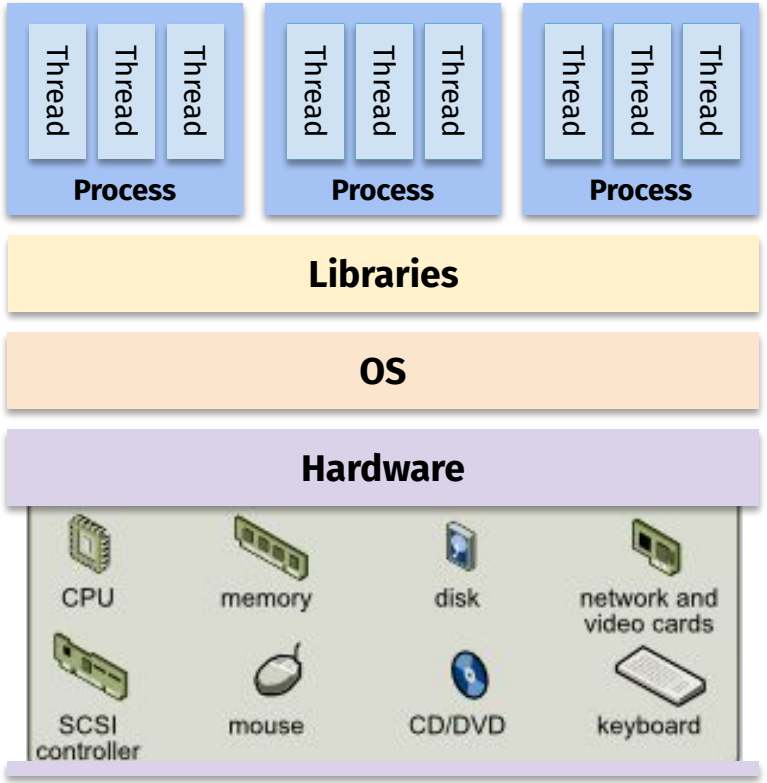
```
--- --x ---
```

4. Owner can read, write, & exec; Group can only exec; and all others have no permissions.

```
rwx --x ---
```

Process Isolation

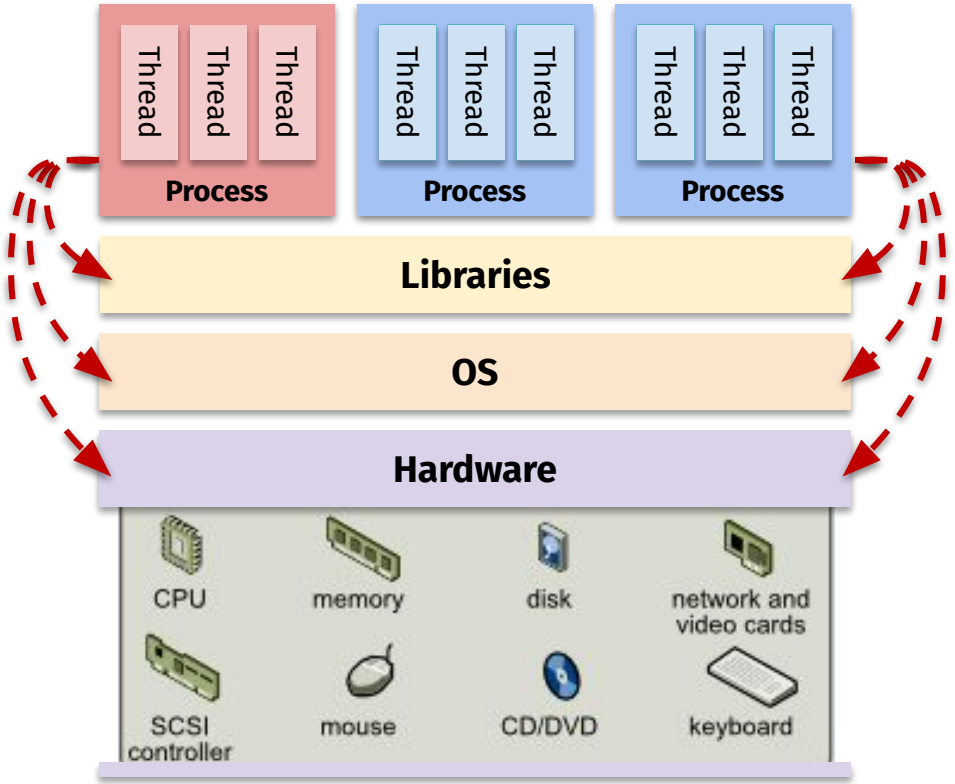
Goal: minimize damage by **isolating** every process



Process Isolation

Goal: minimize damage by **isolating** every process

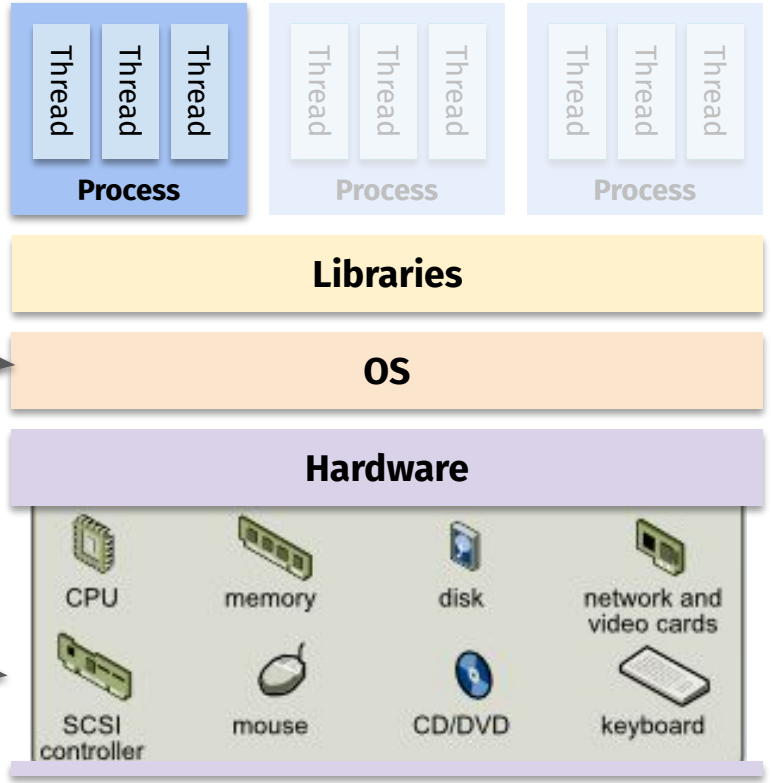
Caveat: you must **trust** all potential **isolation bridges**



Process Isolation

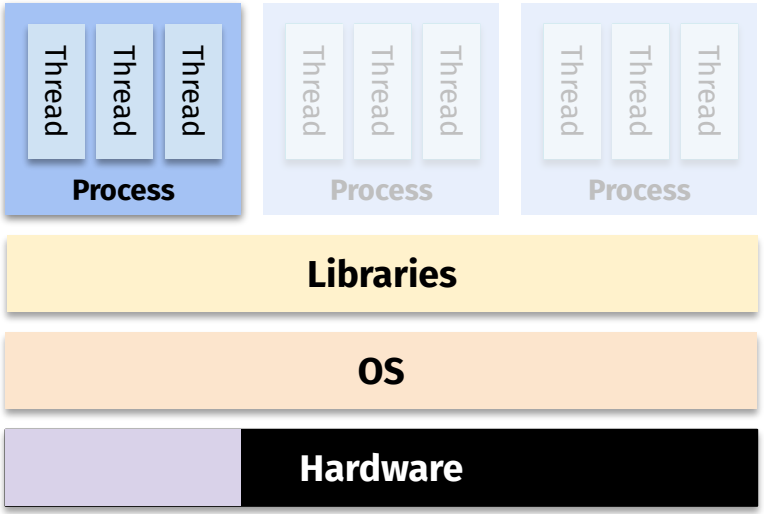
OS-level Isolation:
File system,
Network, Device

HW-level Isolation:
MMU, IO-MMU,
Exec permissions



Isolation Technique: Sandboxing

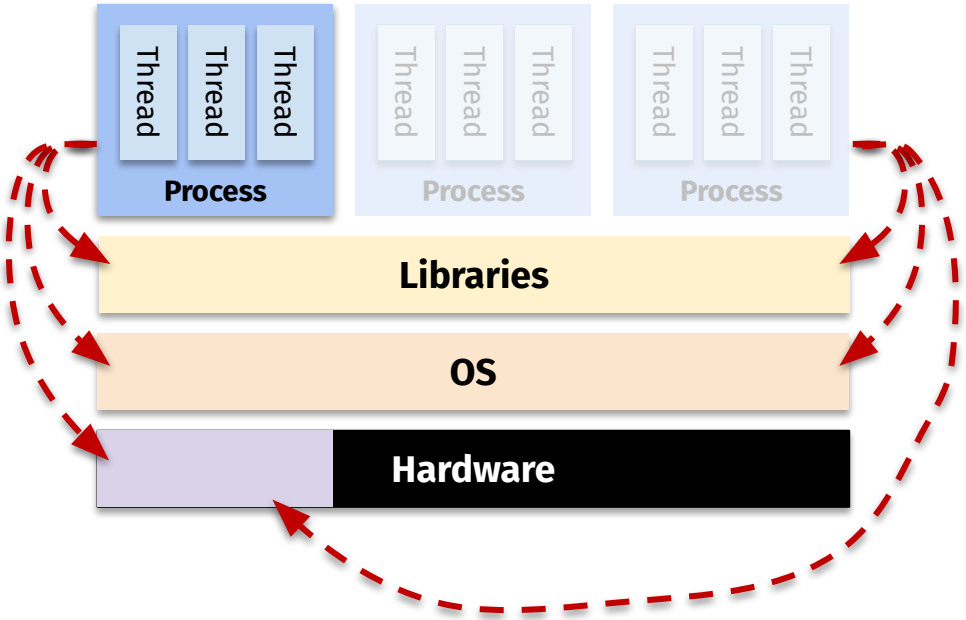
Goal: give processes the **least privileges**



Isolation Technique: Sandboxing

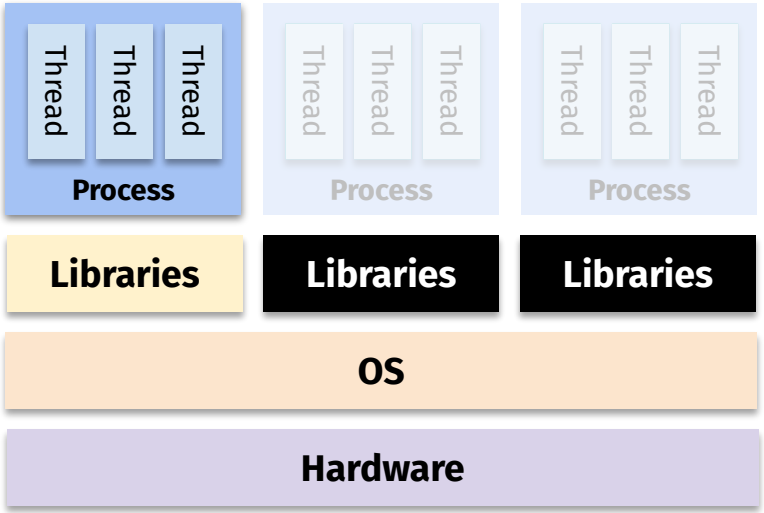
Goal: give processes the **least privileges**

Caveat: the **trusted computing base** is still very large!



Isolation Technique: Containers

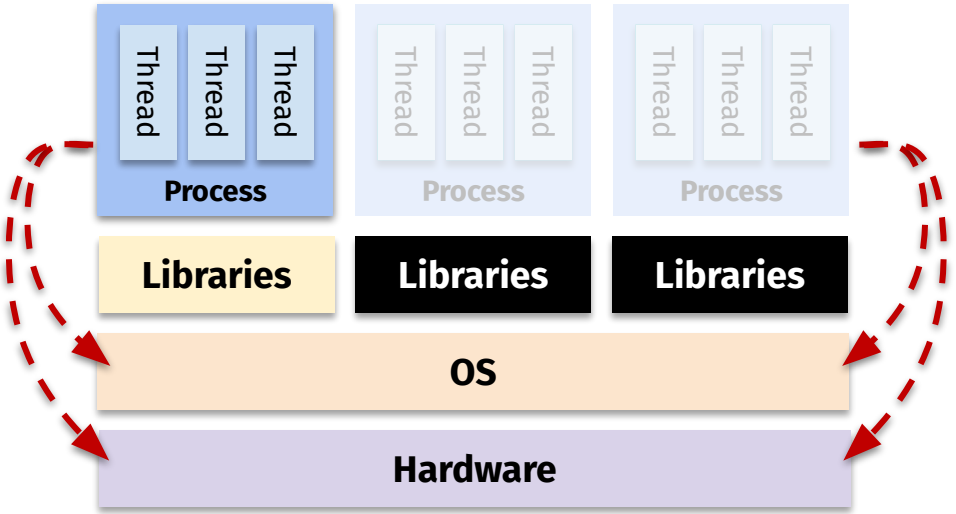
Goal: make **libraries**, **middleware** specific to each process



Isolation Technique: Containers

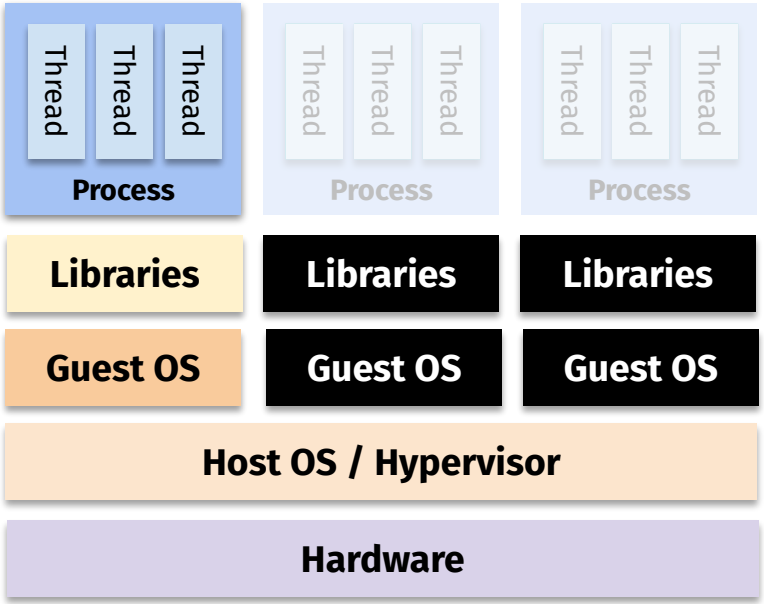
Goal: make **libraries**, **middleware** specific to each process

Caveat: the trusted computing base is now the **OS** and **HW**



Isolation Technique: Virtual Machine

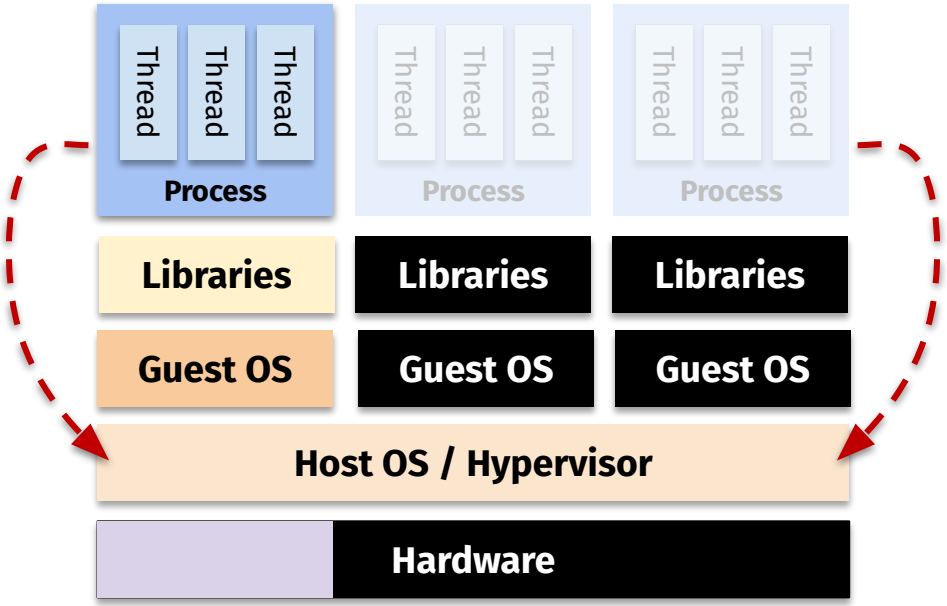
Goal: completely isolate the **OS**



Isolation Technique: Virtual Machine

Goal: completely isolate the **OS**

Caveat: the trusted computing base now the **Hypervisor**



Questions?

This time on CS 4440...

Malware
Viruses, Spyware, Worms, Rootkits
Malware Detection and Prevention

Malware: Malicious Software

- **Definition:** software (more generally, a set of instructions) that runs on a computer it **doesn't have access to** and/or does **something nefarious**

- **Goals of Malware:**
  - **???**

Malware: Malicious Software

- **Definition:** software (more generally, a set of instructions) that runs on a computer it **doesn't have access to** and/or does **something nefarious**

- **Goals of Malware:**
  - Steal private data
  - Display ads, send spam
  - Damage local machine
  - Congest a network
  - Attack other systems on the network
  - Commit online fraud
  - Gain, then grant, unauthorized access
  - Up to the attacker(s) really…

Have you (or a loved one) ever had malware?

Yes :(

0%

Not that I know of...

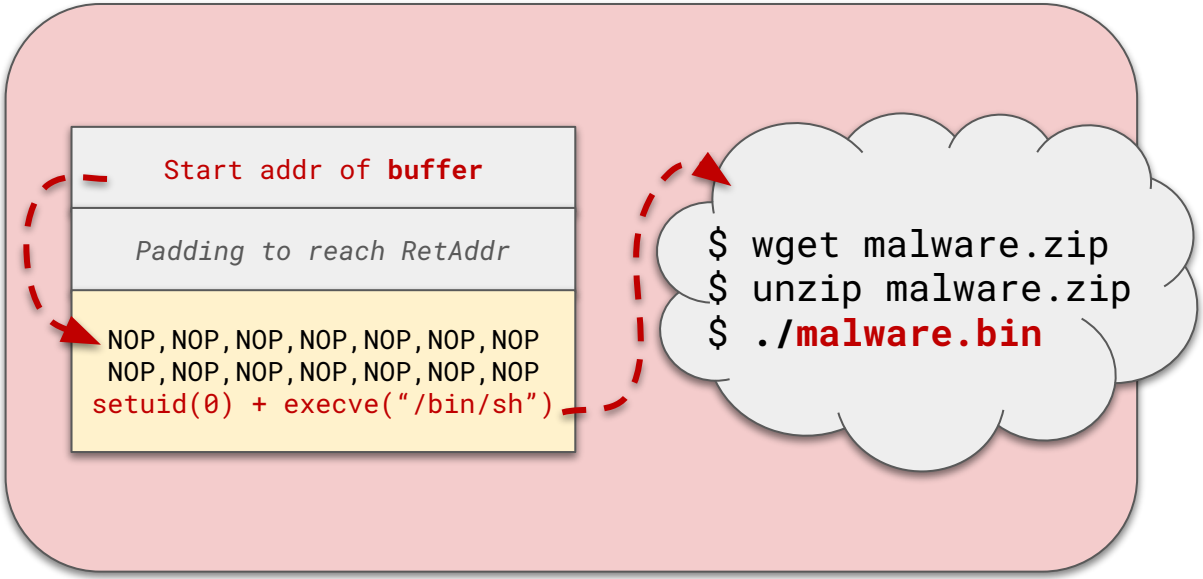
0%

Malware Infection

- Using your Project 2 skills, how could **malware** get on a victim's computer?

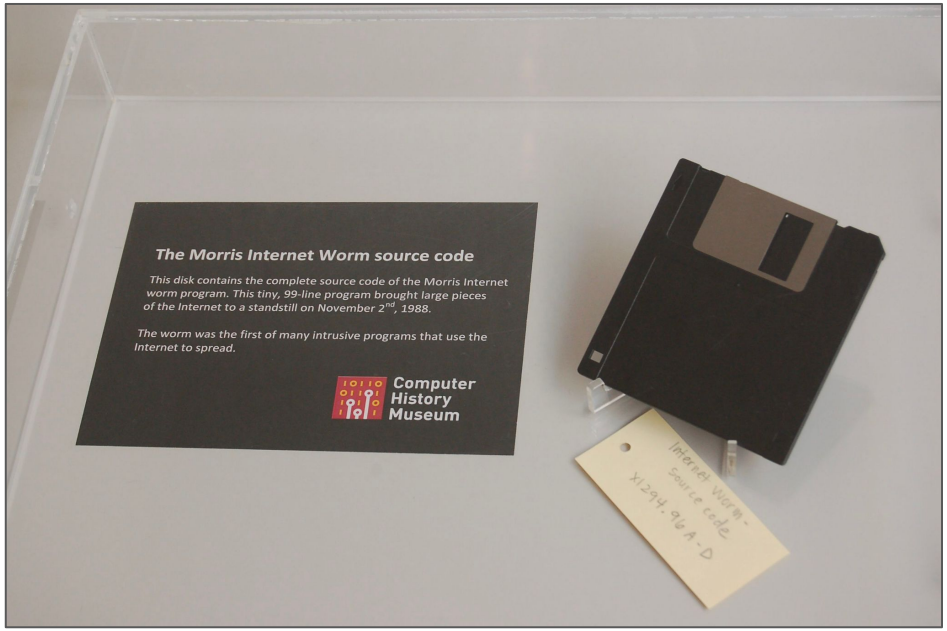
Malware Infection

- Using your Project 2 skills, how could **malware** get on a victim's computer?
  - **A local application is exploited to perform arbitrary code execution**



Case Study: the First Malware

- **1988:** The **Morris Worm**
  - First-known computer malware
- Exploited several vulnerabilities
  - UNIX's finger network service
  - UNIX sendmail
  - Weak/default network passwords
- Result: **devastated the internet**
  - Millions of dollars of damages
  - Caused a psychological shift in IT



Case Study: The Exploit Grey Market

Case Study: The Exploit Grey Market

- **Weaponizing and selling exploits**
  - A huge underground economy
    - Nation-state actors
    - Cyber-criminal gangs
- **Don't participate in this**
  - Likely to end up in bad hands regardless of who brokered it
  - E.g., authoritarian regimes
  - Likely to get people hurt **(or worse)**

Hacks Raise Fear Over N.S.A.'s Hold on Cyberweapons

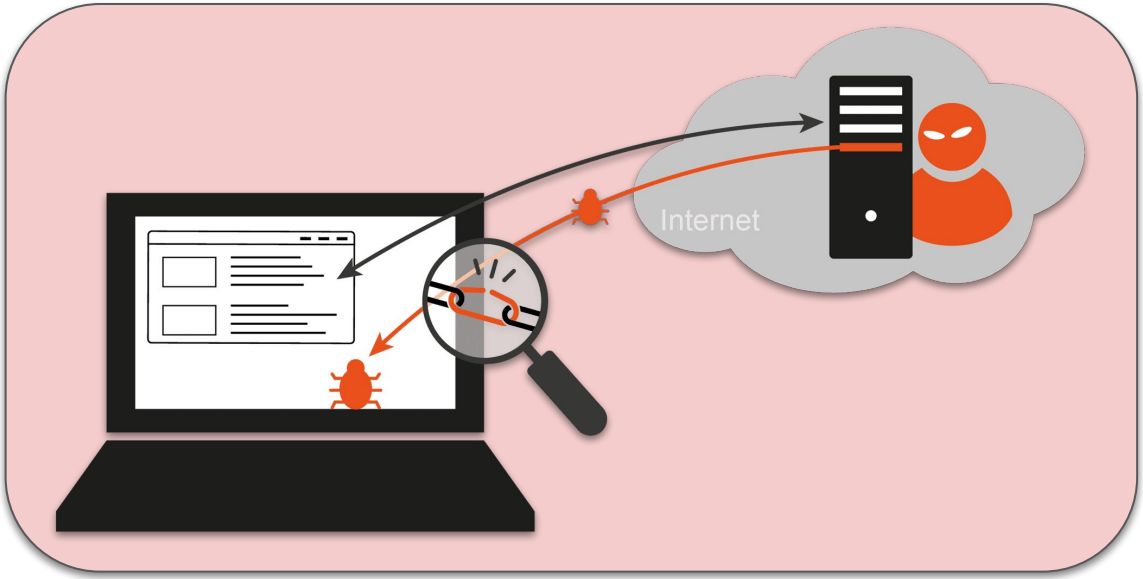
Pegasus: UAE placed spyware on Khashoggi's wife's phone months before murder

Malware Infection

- How can malware infections be facilitated **over the world wide web**?

Malware Infection

- How can malware infections be facilitated **over the world wide web**?
  - **Vulnerable client connects to a malicious server/host; drive-by-download**



Case Study: Malvertising

- **Idea:** booby-trap malware in seemingly-benign ads
- **Common target:** browser **content rendering** engines
  - Adobe Flash
  - JavaScript
  - ActiveX
  - Java applets
- Somewhat rare nowadays

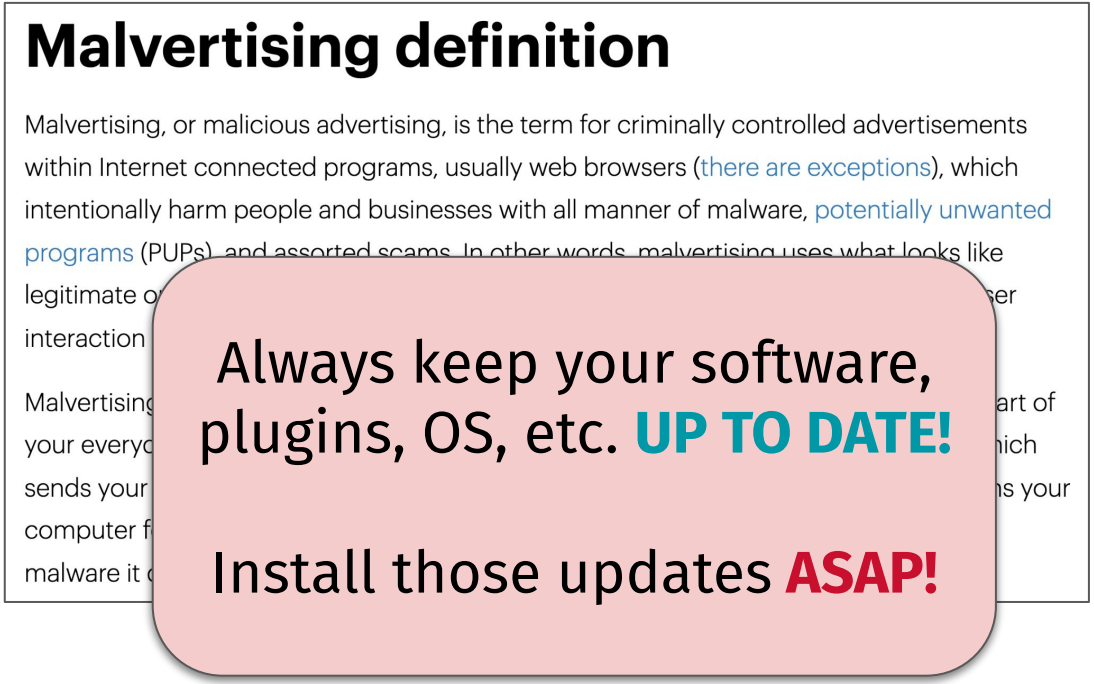
Malvertising definition

Malvertising, or malicious advertising, is the term for criminally controlled advertisements within Internet connected programs, usually web browsers (there are exceptions), which intentionally harm people and businesses with all manner of malware, potentially unwanted programs (PUPs), and assorted scams. In other words, malvertising uses what looks like legitimate online advertising to distribute malware and other threats with little to no user interaction required.

Malvertising can appear on any advertisement on any site, even the ones you visit as part of your everyday Internet browsing. Typically, malvertising installs a tiny piece of code, which sends your computer to criminal command and control (C&C) servers. The server scans your computer for its location and what software is installed on it, and then chooses which malware it determines is most effective to send you.

Case Study: Malvertising

- **Idea:** booby-trap malware in seemingly-benign ads
- **Common target:** browser **content rendering** engines
  - Adobe Flash
  - JavaScript
  - ActiveX
  - Java applets
- Somewhat rare nowadays



Malware Infection

- What if we **can't** install malware by **remotely exploiting** an application?

Malware Infection

- What if we **can't** install malware by **remotely exploiting** an application?
  - **Social engineering attacks:** tricking users into **installing malware themselves**!

Case Study: Scareware

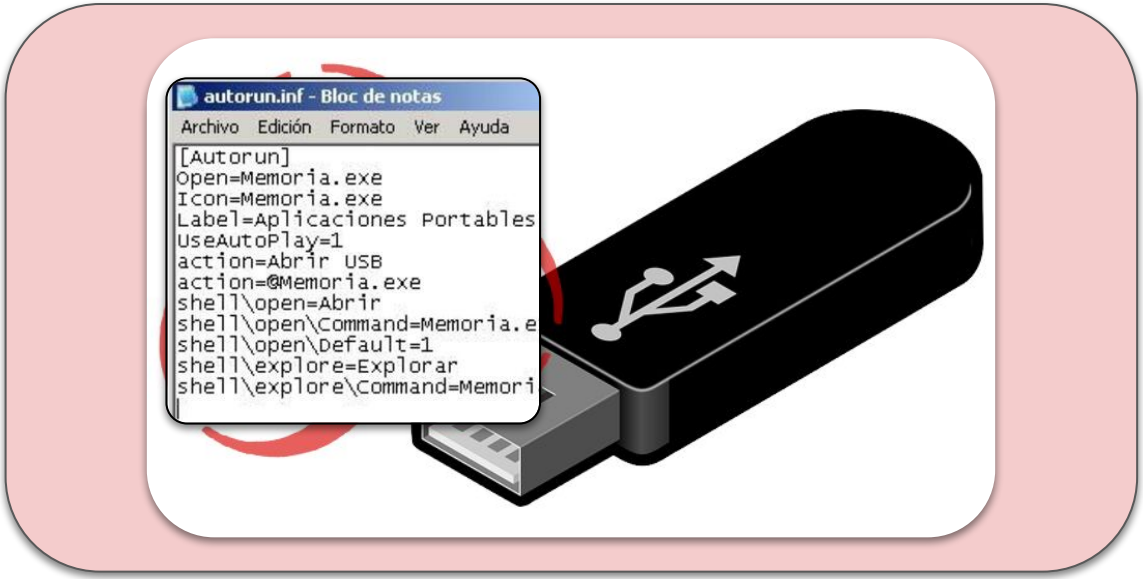
- **Idea:** trick victim into downloading “anti-virus” software… that itself is really **just a piece of malware**
- Was really common in mid-2000s
- **Common target:** children, elderly, inexperienced computer users, etc.
- Nowadays: **ransomware**

Malware Infection

- **How else** can malicious software get on victim computers?

Malware Infection

- **How else** can malicious software get on victim computers?
  - **Malicious hardware plugged-in; automatically executes code**



Case Study: People are Naive

Users Really Do Plug in USB Drives They Find

Matthew Tischer[†] Zakir Durumeric[‡†] Sam Foster[†] Sunny Duan[†]
Alec Mori[†] Elie Bursztein[◇] Michael Bailey[†]

[†] University of Illinois, Urbana Champaign [‡] University of Michigan [◇] Google, Inc.
{tischer1, sfoster3, syduan2, ajmori2, mdbailey}@illinois.edu
zakir@umich.edu elieb@google.com

***Abstract*—We investigate the anecdotal belief that end users will pick up and plug in USB flash drives they find by completing a controlled experiment in which we drop 297 flash drives on a large university campus. We find that the attack is effective with an estimated success rate of 45–98% and expeditious with the first drive connected in less than six minutes. We analyze the types of drives users connected and survey those users to understand their motivation and security profile. We find that a drive's appearance does not increase attack success. Instead, users connect the drive with the altruistic intention of finding the owner. These individuals are not technically incompetent, but are rather typical community members who appear to take more** median time to connection of 6.9 hours and the first connection occurring within six minutes from when the drive was dropped. Contrary to popular belief, the appearance of a drive does not increase the likelihood that someone will connect it to their computer. Instead, users connect all types of drives unless there are other means of locating the owner—suggesting that participants are altruistically motivated. However, while users initially connect the drive with altruistic intentions, nearly half are overcome with curiosity and open intriguing files—such as vacation photos—before trying to find the drive's owner.

Case Study: People are Naive

Users Really Do Plug in USB Drives They Find

Matthew Tischer† Zakir Durumeric‡† Sam Foster† Sunny Duan†

Success rate of people to plugging-in random USB thumb drives: **45–98%**

Abstract—We investiga
will pick up and plug in U
a controlled experiment i
a large university campus
with an estimated success
the first drive connected in less than six minutes. We analyze
the types of drives users connected and survey those users to
understand their motivation and security profile. We find that
a drive's appearance does not increase attack success. Instead,
users connect the drive with the altruistic intention of finding the
owner. These individuals are not technically incompetent, but are
rather typical community members who appear to take more

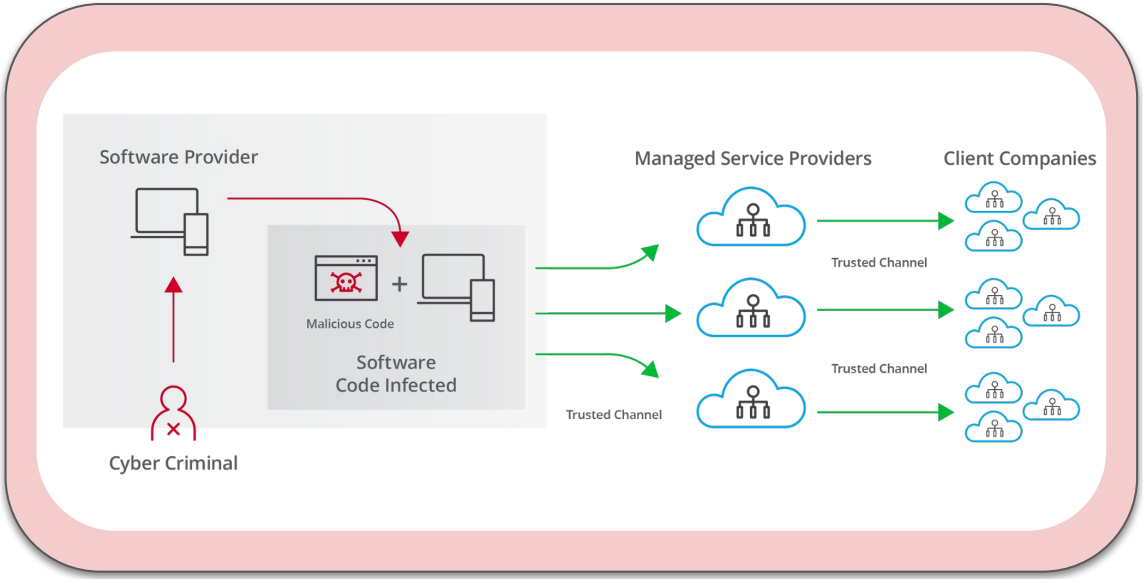
rs and the first connection
en the drive was dropped.
rance of a drive does not
e will connect it to their
t all types of drives unless
there are other means of locating the owner—suggesting that
participants are altruistically motivated. However, while users
initially connect the drive with altruistic intentions, nearly half
are overcome with curiosity and open intriguing files—such
as vacation photos—before trying to find the drive's owner.

Malware Infection

- How could an attacker **maximize their infection spreading** potential?

Malware Infection

- How could an attacker **maximize their infection spreading** potential?
  - **Supply chain attacks:** hack into **key software provider** and inject virus into it



Case Study: SolarWinds Breach

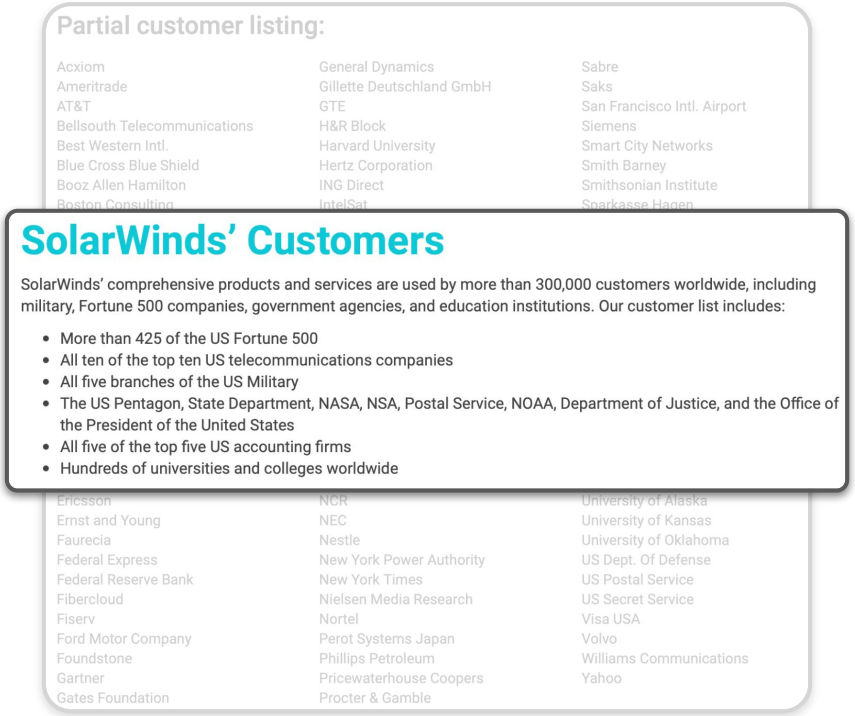
- **Idea:** infect software provider that serves major targets

Partial customer listing:

Acxiom	General Dynamics	Sabre
Ameritrade	Gillette Deutschland GmbH	Saks
AT&T	GTE	San Francisco Intl. Airport
Bellsouth Telecommunications	H&R Block	Siemens
Best Western Intl.	Harvard University	Smart City Networks
Blue Cross Blue Shield	Hertz Corporation	Smith Barney
Booz Allen Hamilton	ING Direct	Smithsonian Institute
Boston Consulting	IntelSat	Sparkasse Hagen
Cable & Wireless	J.D. Byrider	Sprint
Cablecom Media AG	Johns Hopkins University	St. John's University
Cablevision	Kennedy Space Center	Staples
CBS	Kodak	Subaru
Charter Communications	Korea Telecom	Supervalu
Cisco	Leggett and Platt	Swisscom AG
CitiFinancial	Level 3 Communications	Symantec
City of Nashville	Liz Claiborne	Telecom Italia
City of Tampa	Lockheed Martin	Telenor
Clemson University	Lucent	Texaco
Comcast Cable	MasterCard	The CDC
Credit Suisse	McDonald's Restaurants	The Economist
Dow Chemical	Microsoft	Time Warner Cable
EMC Corporation	National Park Service	U.S. Air Force
Ericsson	NCR	University of Alaska
Ernst and Young	NEC	University of Kansas
Faurecia	Nestle	University of Oklahoma
Federal Express	New York Power Authority	US Dept. Of Defense
Federal Reserve Bank	New York Times	US Postal Service
Fibercloud	Nielsen Media Research	US Secret Service
Fiserv	Nortel	Visa USA
Ford Motor Company	Perot Systems Japan	Volvo
Foundstone	Phillips Petroleum	Williams Communications
Gartner	Pricewaterhouse Coopers	Yahoo
Gates Foundation	Procter & Gamble	

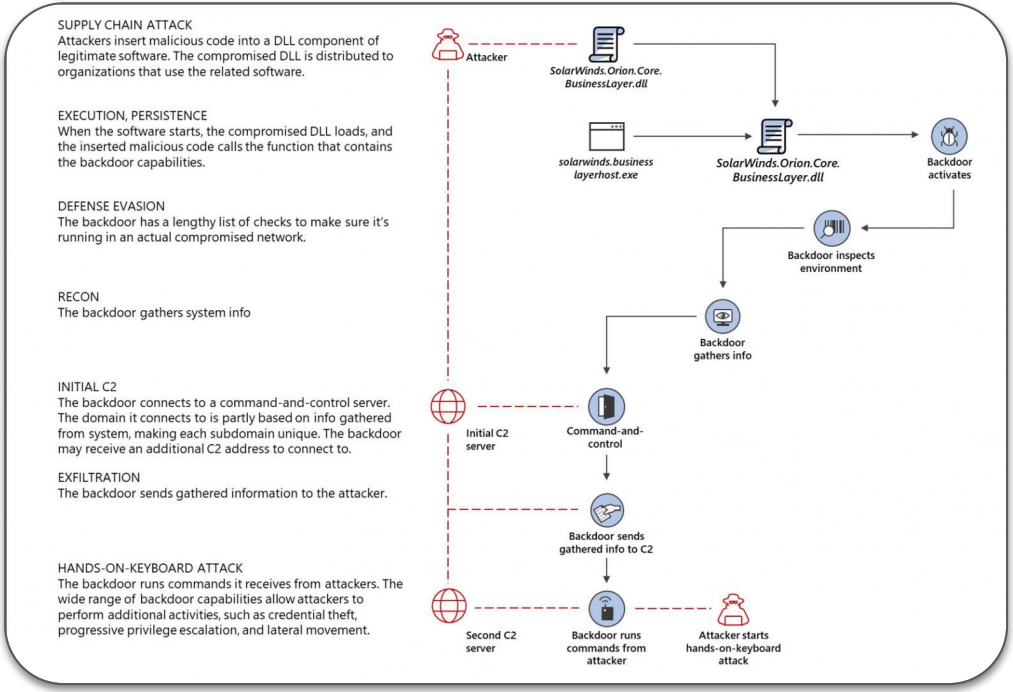
Case Study: SolarWinds Breach

- **Idea:** infect software provider that serves major targets



Case Study: SolarWinds Breach

- **Idea:** infect software provider that serves major targets
- Inject malware within their development process
- When deployed, attacker gets access **to all supplied targets**



OS-level Security

- Your OS is also software too!
  - Arguably the most vital software
  - OS exploits are the most sought after exploits in today's market
- **Is one Operating System more vulnerable than its peers?**
  - Microsoft Windows?
  - Apple MacOS?

Perceptions of Windows security?

Nobody has responded yet.

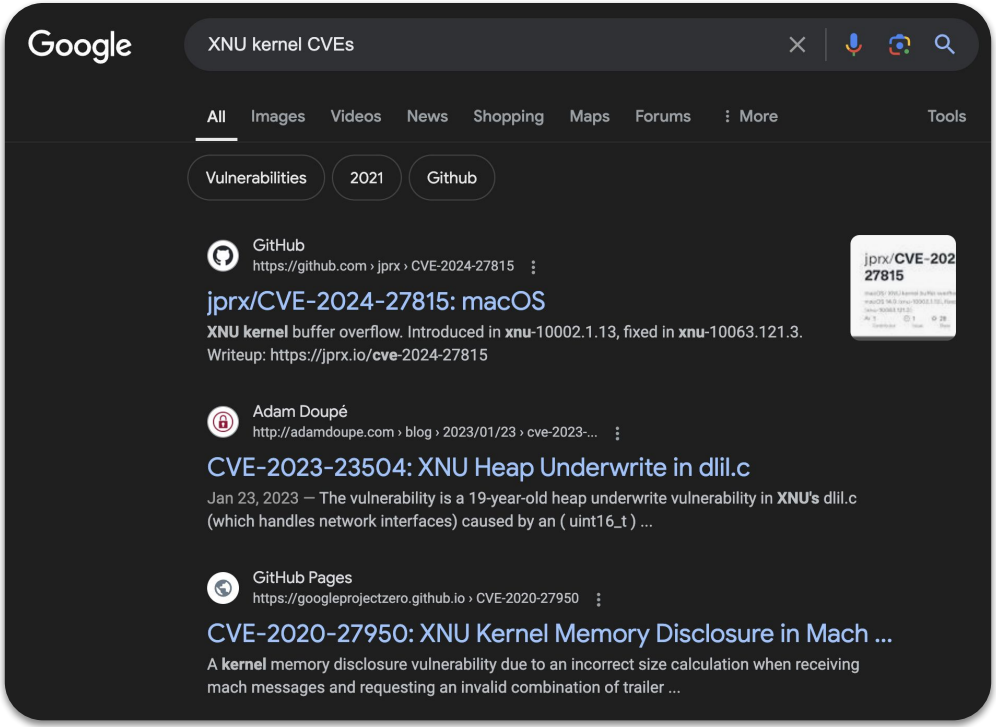
Hang tight! Responses are coming in.

Perceptions about MacOS / iOS security?

Nobody has responded yet.

Hang tight! Responses are coming in.

But... MacOS is safe, right?

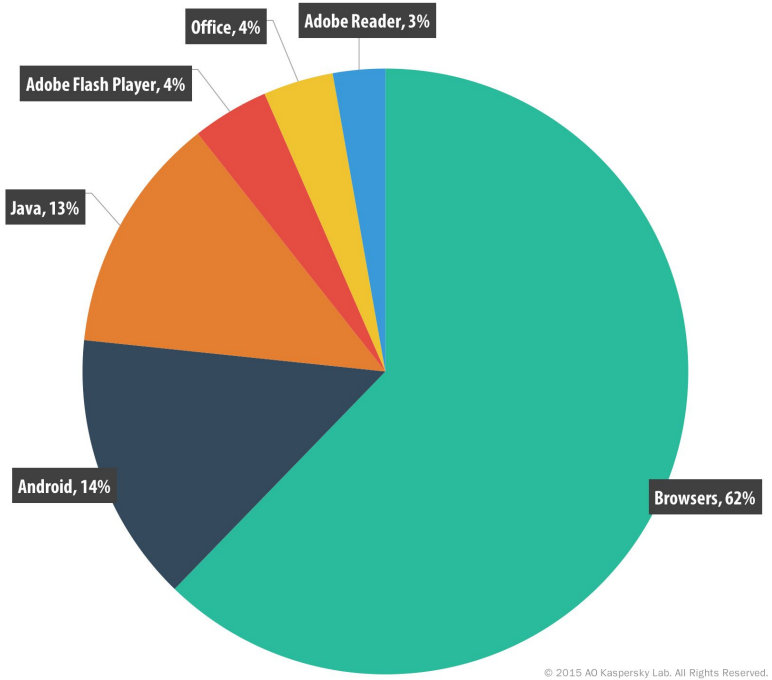


But... MacOS is safe, right?

CVE-2024-44165	A logic issue was addressed with improved checks. This issue is fixed in macOS Ventura 13.7, iOS 17.7 and iPadOS 17.7, visionOS 2, iOS 18 and iPadOS 18, macOS Sonoma 14.7, macOS Sequoia 15. Network traffic may leak outside a VPN tunnel. **Published:** September 16, 2024; 8:15:51 PM -0400	*V4.0:*(not available) *V3.1:* 7.5 HIGH *V2.0:*(not available)
CVE-2024-44164	This issue was addressed with improved checks. This issue is fixed in iOS 17.7 and iPadOS 17.7, macOS Ventura 13.7, macOS Sonoma 14.7, macOS Sequoia 15. An app may be able to bypass Privacy preferences. **Published:** September 16, 2024; 8:15:51 PM -0400	*V4.0:*(not available) *V3.1:* 7.1 HIGH *V2.0:*(not available)
CVE-2024-44163	The issue was addressed with improved checks. This issue is fixed in macOS Ventura 13.7, macOS Sonoma 14.7, macOS Sequoia 15. A malicious application may be able to access private information. **Published:** September 16, 2024; 8:15:51 PM -0400	*V4.0:*(not available) *V3.1:* 5.5 MEDIUM *V2.0:*(not available)
CVE-2024-44161	An out-of-bounds read was addressed with improved bounds checking. This issue is fixed in macOS Ventura 13.7, macOS Sonoma 14.7, macOS Sequoia 15. Processing a maliciously crafted texture may lead to unexpected app termination. **Published:** September 16, 2024; 8:15:51 PM -0400	*V4.0:*(not available) *V3.1:* 5.5 MEDIUM *V2.0:*(not available)
CVE-2024-44160	A buffer overflow issue was addressed with improved memory handling. This issue is fixed in macOS Ventura 13.7, macOS Sonoma 14.7, macOS Sequoia 15. Processing a maliciously crafted texture may lead to unexpected app termination. **Published:** September 16, 2024; 8:15:50 PM -0400	*V4.0:*(not available) *V3.x:*(not available) *V2.0:*(not available)
CVE-2024-44158	This issue was addressed with improved redaction of sensitive information. This issue is fixed in iOS 17.7 and iPadOS 17.7, macOS Ventura 13.7, macOS Sonoma 14.7, macOS Sequoia 15. A shortcut may output sensitive user data without consent. **Published:** September 16, 2024; 8:15:50 PM -0400	*V4.0:*(not available) *V3.1:* 5.5 MEDIUM *V2.0:*(not available)

Our Vulnerable World

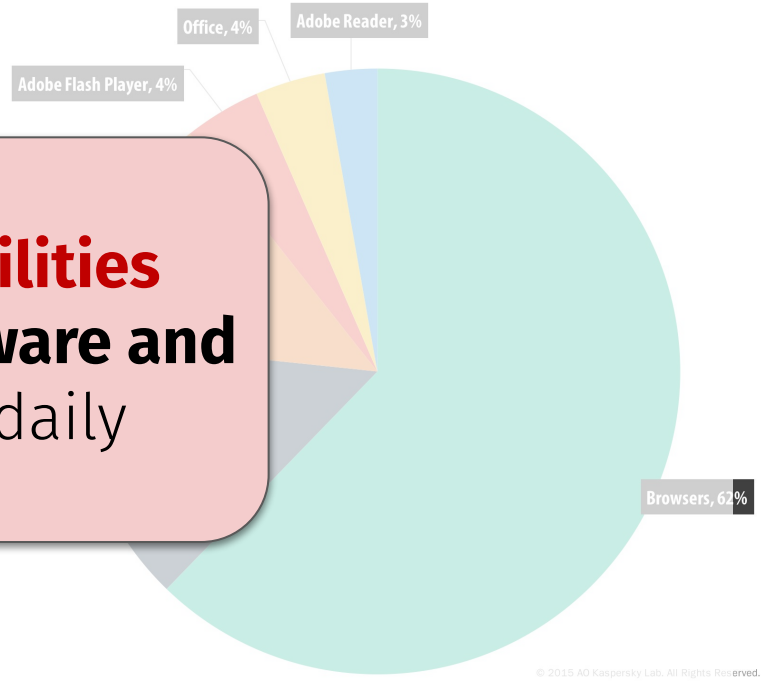
- Kaspersky Lab's 2015 report
- Modern exploits are multi-stage
- Attackers "mastered" non-Windows OSs
  - Linux, MacOS, iOS aren't as safe as you think!



Our Vulnerable World

- Kaspersky Lab's 2015 report
- Modern exploits
- Attackers "maste
  - Linux, MacOS, i

Critical vulnerabilities exist in **every software and system** we use daily

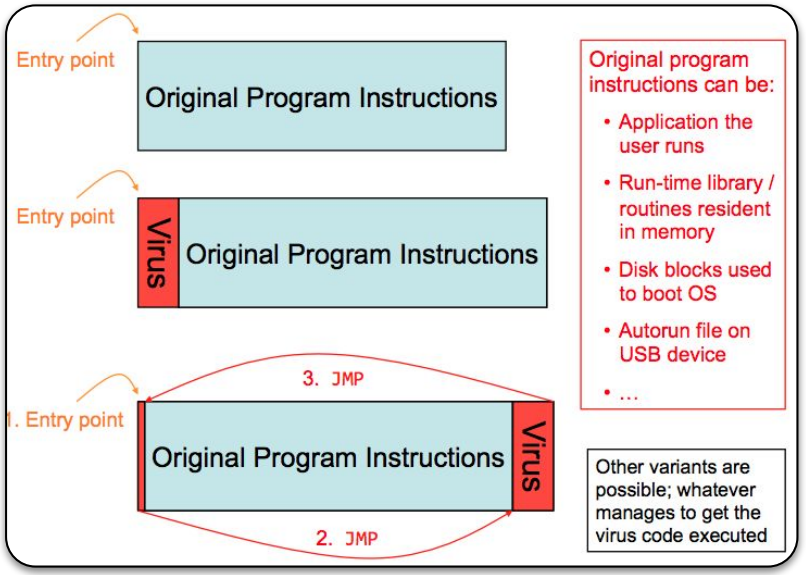


Questions?

Today's Malware "Zoo"

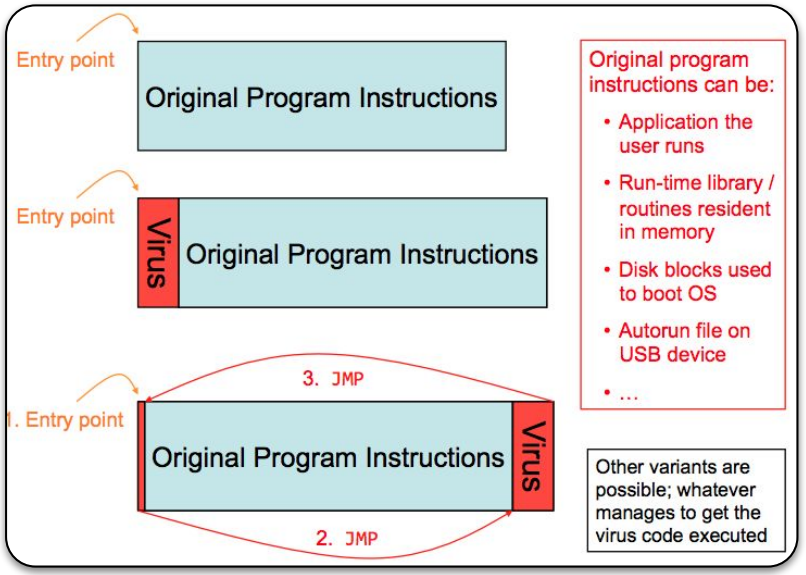
Viruses

- Analogous to viruses in biology
- **Self-replicating software** that infects other programs by modifying them to inject a version of itself



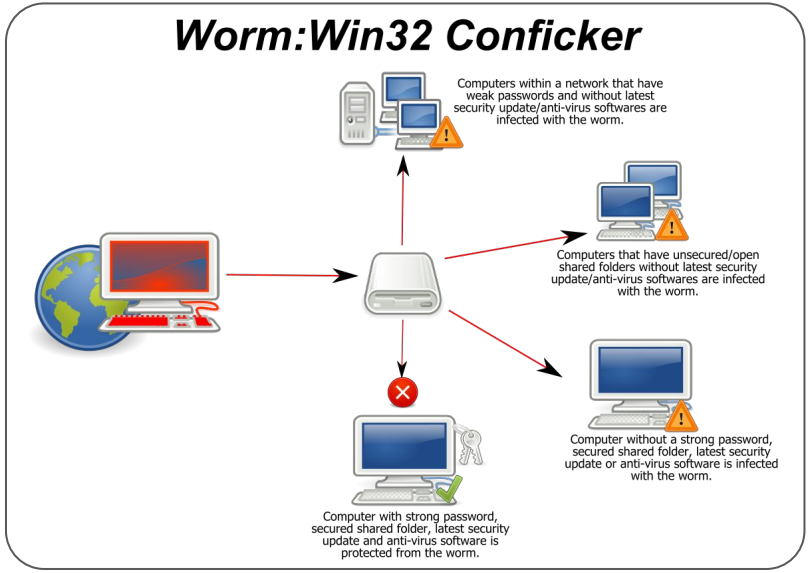
Viruses

- Analogous to viruses in biology
- **Self-replicating software** that infects other programs by modifying them to inject a version of itself
- Can **mutate to avoid detection** by changing parts of their code
  - E.g., “polymorphic”, “metamorphic” viruses



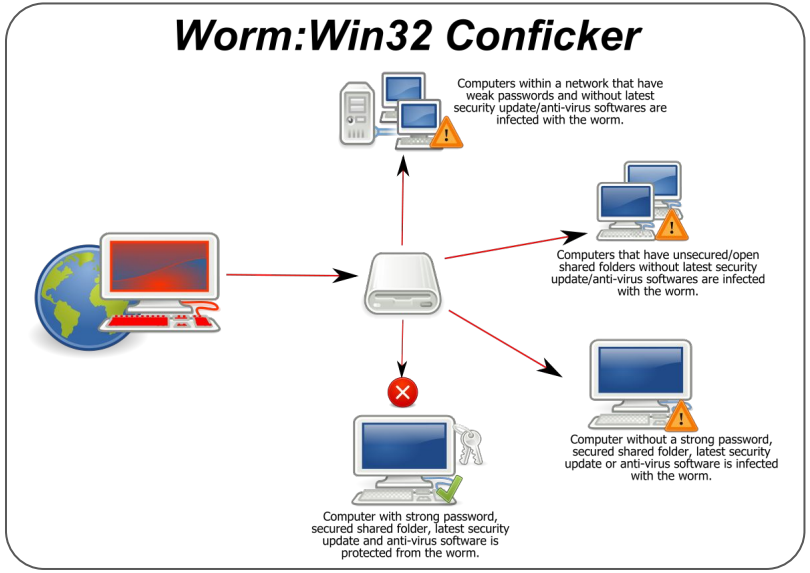
Worms

- Self-replicating software that infects **other systems** by automatically spreading over a connected network
- Fast-spreading worms are a big threat (fueled by **software homogeneity**)



Worms

- Self-replicating software that infects **other systems** by automatically spreading over a connected network
- Fast-spreading worms are a big threat (fueled by **software homogeneity**)
- Famous worms (and exploited software):
  - **2003:** Slammer Worm (**Microsoft's SQL Server**)
  - **2008:** Conficker Worm (**Windows NetBIOS**)

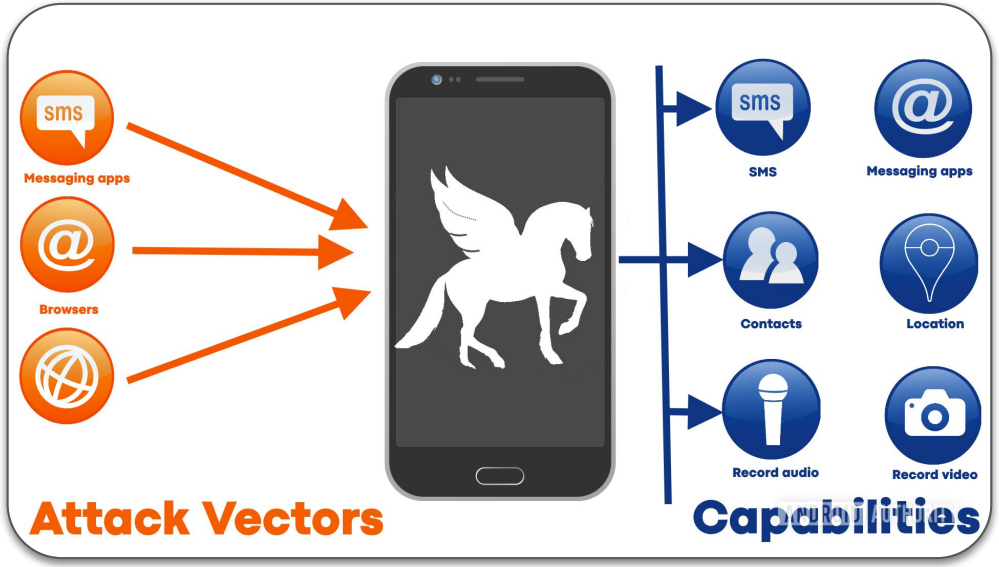


Adware

- Software that incessantly displays **advertisements**
  - Pop-up ads
  - Opening web pages
  - False search engine results
  - Redirecting URL clicks
- Often needs some form of **user interaction** to install

Spyware

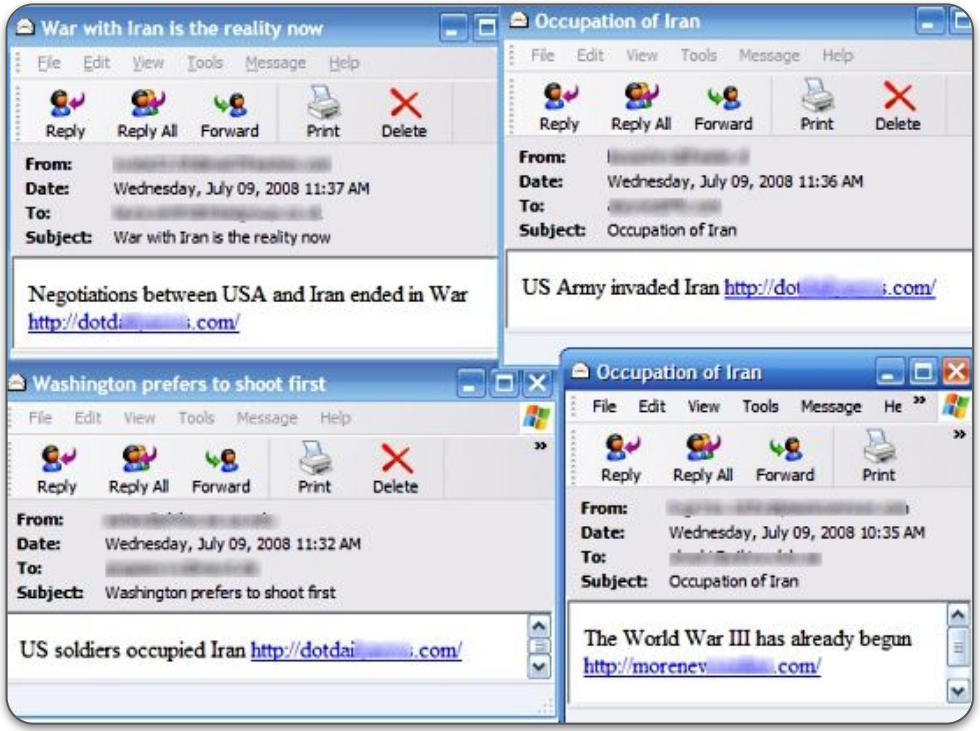
- Software that tracks and **sensitive user information**
  - Keystrokes
  - Passwords
  - Web searches
  - GPS Location
  - Installed/accessed apps
- Collects, sends to a third party
  - Parental Control applications
  - Nation-state spyware (Pegasus)



Trojan Horses

- Software that **tricks user into installing** by masquerading as a benign, safe application

- **Common examples:**
  - Adware
  - Malicious attachments
    - E-Cards (Storm Worm)
    - Intriguing links
  - Fake anti-virus applications
  - Ransomware



Trojan Horses

- Software that **tricks user into installing** by masquerading as a benign, safe application

- **Common examples:**
  - Adware
  - Malicious attachments
    - E-Cards (Storm Worm)
    - Intriguing links
  - Fake anti-virus applications
  - Ransomware

PRIVACY AND SECURITY

The FBI Thinks Ransomware Victims Should 'Just Pay Up'

Chris Mills
10/26/15 10:26pm • Filed to: CRIME

58.1K 109 8

Cryptolocker 2.0

Your personal files are encrypted

Info

Your **important files were encrypted** on this computer: photos, videos, documents , etc. You can verify this by click on see files and try to open them.

Encryption was produced using **unique** public key RSA-4096 generated for this computer. To decrypt files, you need to obtain **private** key.

The single copy of the private key, which will allow you to decrypt the files, is located on a secret server on the Internet; **the server will destroy the key within 72 hours after encryption completed**. After that, nobody and never will be able to restore files.

To retrieve the private key, you need to pay 0.5 bitcoins.

Your files will be lost without payment on:

As documented in numerous Nicolas Cage movies, the FBI has a fairly strict 'don't negotiate with the terrorists' policy. Unless you're a company that's had your files encrypted, in which case you should probably just pay the ransom. Welp.

According to *Security Ledger*, the advice comes from Joseph Bonavolonta, the Assistant Special Agent in Charge of the FBI's CYBER and Counterintelligence Program in the Boston office. He said that "the ransomware is that good," and

Rootkits

- Software designed to **maintain attacker's control** over a system
  - I.e., **root-level access**
- Typically a payload of other malware (e.g., viruses, worms)
- Maintain **stealth**, **undetectability**

Rootkits

- Software designed to **maintain attacker's control** over a system
  - I.e., **root-level access**
- **Stealth Measures:**
  - Intercept system calls responsible listing files, processes, etc.
  - Filter out the malware's files and processes to avoid being seen

Rootkits

- Software designed to **maintain attacker's control** over a system
  - I.e., **root-level access**
- **Stealth Measures:**
  - Intercept system calls responsible listing files, processes, etc.
  - Filter out the malware's files and processes to avoid being seen

Sony BMG copy protection rootkit scandal

3 languages

Article Talk | Read Edit View history

From Wikipedia, the free encyclopedia

Screenshot of the Sony CD audio player, playing Switchfoot's fifth studio album *Nothing Is Sound*.

A scandal erupted in 2005 regarding Sony BMG's implementation of copy protection measures on about 22 million CDs. When inserted into a computer, the CDs installed one of two pieces of software that provided a form of digital rights management (DRM) by modifying the operating system to interfere with CD copying. Neither program could easily be uninstalled, and they created vulnerabilities that were exploited by unrelated malware. One of the programs would install and "phone home" with reports on the user's private listening habits, even if the user refused its end-user license agreement (EULA), while the other was not mentioned in the EULA at all. Both programs contained code from several pieces of copylefted free software in an apparent infringement of copyright, and configured the operating system to hide the software's existence, leading to both programs being classified as rootkits.

Sony BMG initially denied that the rootkits were harmful. It then released an uninstaller for one of the programs that merely made the program's files visible while also installing additional software that could not be easily removed, collected an email address from the user and introduced further security vulnerabilities.

Following public outcry, government investigations and class-action lawsuits in 2005 and 2006, Sony BMG partially addressed the scandal with consumer settlements, a recall of about 10% of the affected CDs and the suspension of CD copy-protection efforts in early 2007.

Rootkits

- Software designed to **maintain attacker's control** over a system
  - I.e., **root-level access**
- **Stealth Measures:**
  - Intercept system calls responsible listing files, processes, etc.
  - Filter out the malware's files and processes to avoid being seen
- **Incredibly difficult to remove**
  - Can never guarantee system is clean

Sony BMG copy protection rootkit scandal

3 languages

Article Talk | Read Edit View history

From Wikipedia, the free encyclopedia

Screenshot of the Sony CD audio player, playing Switchfoot's fifth studio album *Nothing Is Sound*.

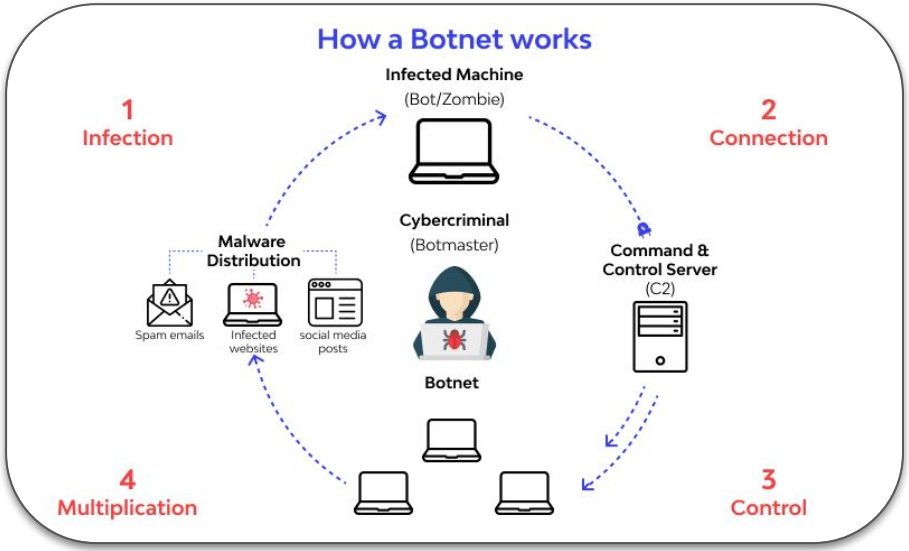
A scandal erupted in 2005 regarding Sony BMG's implementation of copy protection measures on about 22 million CDs. When inserted into a computer, the CDs installed one of two pieces of software that provided a form of digital rights management (DRM) by modifying the operating system to interfere with CD copying. Neither program could easily be uninstalled, and they created vulnerabilities that were exploited by unrelated malware. One of the programs would install and "phone home" with reports on the user's private listening habits, even if the user refused its end-user license agreement (EULA), while the other was not mentioned in the EULA at all. Both programs contained code from several pieces of copylefted free software in an apparent infringement of copyright, and configured the operating system to hide the software's existence, leading to both programs being classified as rootkits.

Sony BMG initially denied that the rootkits were harmful. It then released an uninstaller for one of the programs that merely made the program's files visible while also installing additional software that could not be easily removed, collected an email address from the user and introduced further security vulnerabilities.

Following public outcry, government investigations and class-action lawsuits in 2005 and 2006, Sony BMG partially addressed the scandal with consumer settlements, a recall of about 10% of the affected CDs and the suspension of CD copy-protection efforts in early 2007.

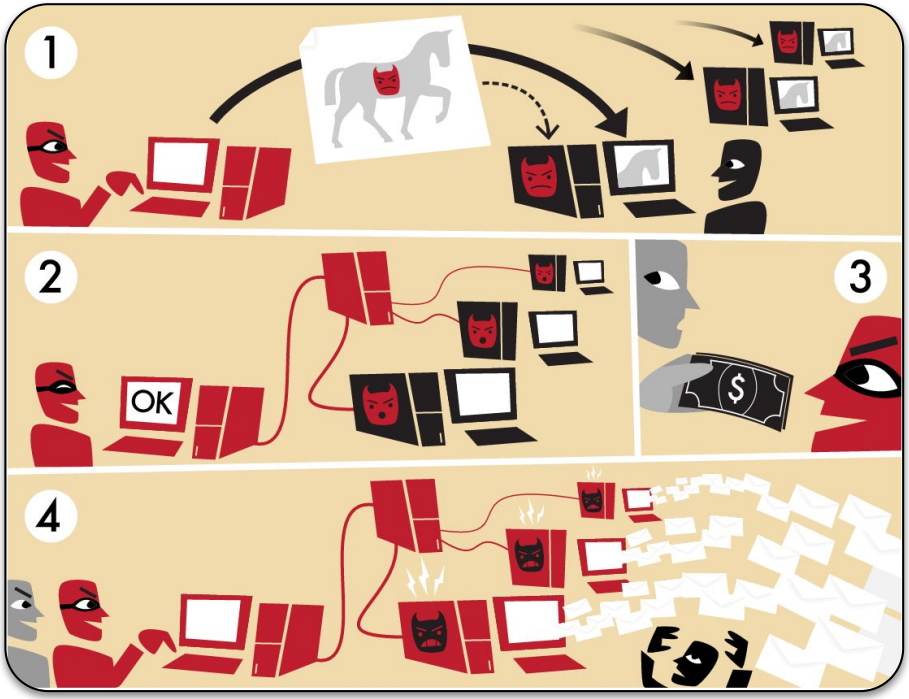
Bots and Botnets

- **Bot:** a victim system remotely under attacker control (e.g., rootkit)
- **Botnet:** a collection of bots
  - Often used for distributed cyber attacks
- **Command and Control Measures:**
  - **Centralized: single server** directs bots
    - Simple; easy to detect/disable
  - **Distributed:** bots direct **one another**
    - Complex; hard to detect/disable



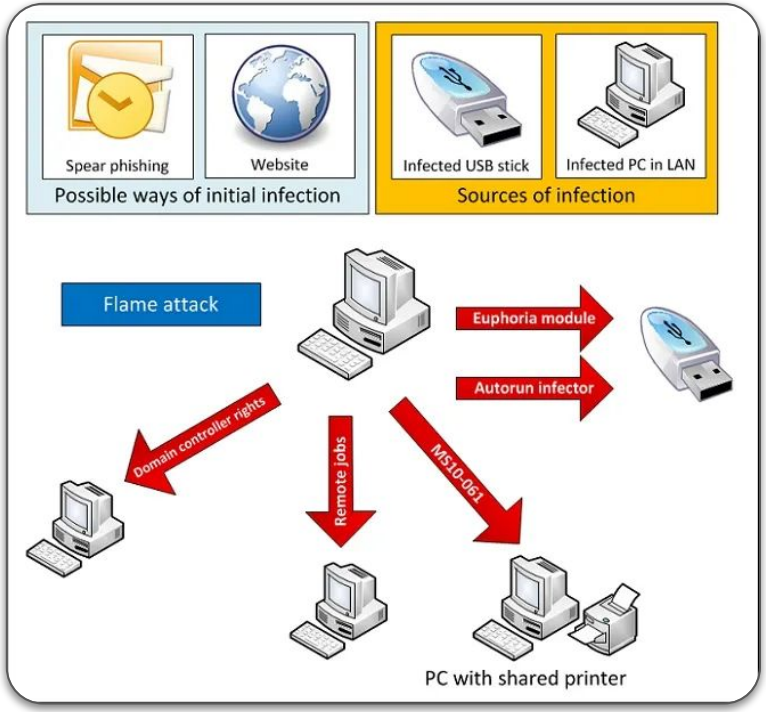
Famous Botnets

- **Mirai Botnet**
  - Propagated by exploiting default passwords in internet-connected household IoT devices
  - Used to DDOS targeted websites
- **Storm Botnet**
  - Propagated by email attachments
  - When infected, each bot spins up an email server and begins mass email spam campaign to propagate itself



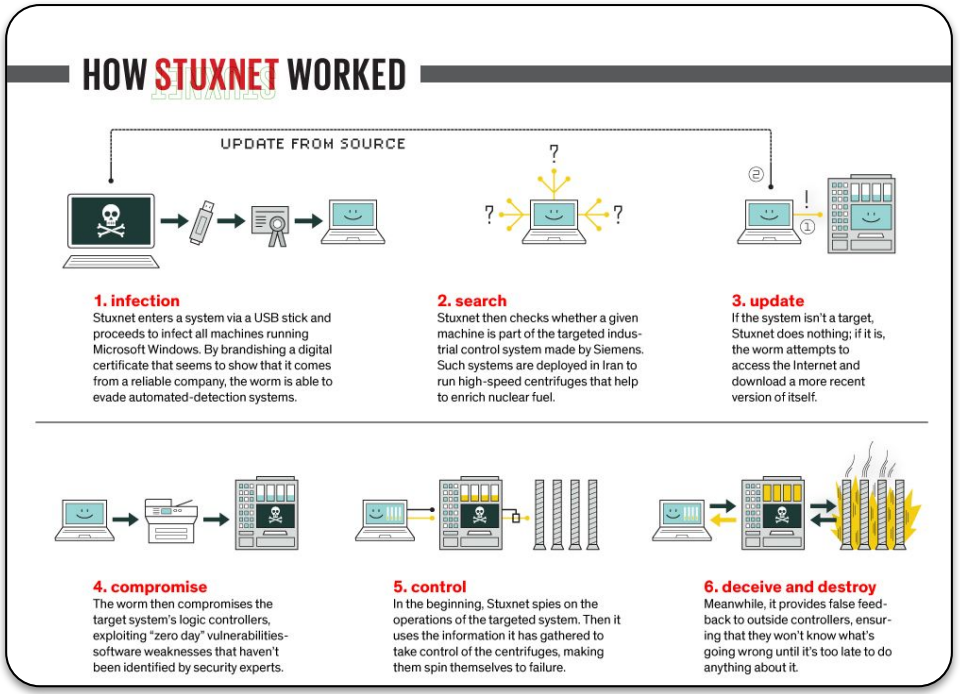
Advanced Persistent Threats (APTs)

- **Combined Threats**
  - Typically a rootkit, spyware, combined with other capabilities
- Extremely sophisticated, stealthy, and target-specific
  - **Insanely complex exploit chains**
- Believed to be developed by **nation-state cyber threat** actors
  - E.g., the NSA, CIA, Mossad, GRU



The Stuxnet APT

- Believed to be developed by USA (NSA) and Israel (Mossad)
- Sophisticated malware designed to infect, destroy ICS computers
  - **Primary target:** uranium enrichment at Iran's Natanz nuclear plant
  - **Payload 1:** make uranium centrifuge spin up so fast that it self-destructs
  - **Payload 2:** feed operators fake data that appears everything is fine
- https://darknetdiaries.com/episode/29/



Summary: Major Malware Types

- **Virus**
  - Self-replicating software that infects other programs, mutates itself to avoid detection
- **Worm**
  - Self-replicating software that spreads over networks to infect programs on other systems
- **Trojans**
  - Appears to perform desirable function, but does something malicious behind the scenes
- **Rootkit**
  - Malware that uses stealth to achieve persistent presence on a machine
- **Botnet**
  - A network of compromised, “Zombie” or “bot” computers that do a botmaster’s bidding

Questions?

Detecting and Preventing Malware

Detection

- **Anti-virus software**
  - Software for detecting, eliminate malware
  - E.g., Malwarebytes, Avast, McAfee, Symantec
- **Signature-based** anti-virus:
  - Track identifying strings (like a fingerprint)
  - Difficult against mutating viruses
- **Heuristic-based** anti-virus:
  - Analyze program behavior, identify unusual patterns
  - E.g. network access, file deletion, modify boot sector

Are you currently running antivirus software on your laptop?

Yes!

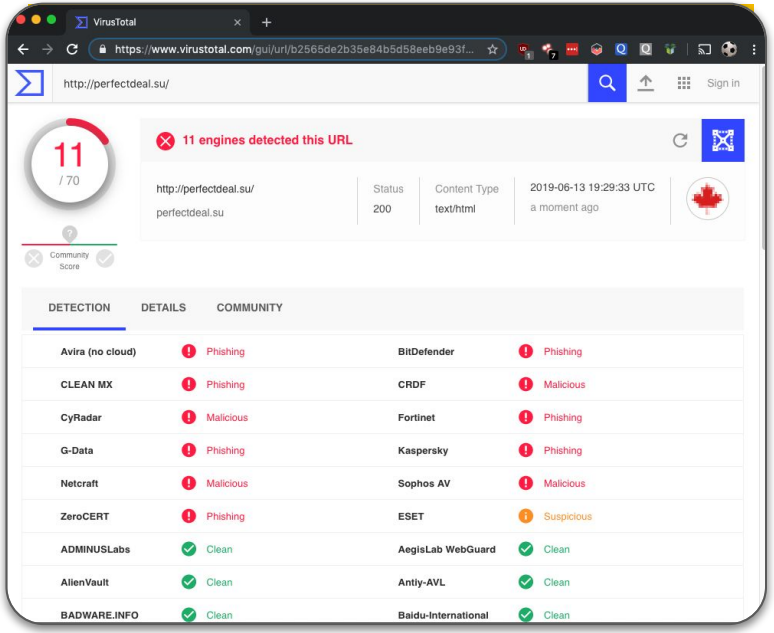
0%

No :(

0%

Detection

- **No anti-virus is perfect!**
  - A constant cat and mouse game
  - Heuristics, signatures need constant updating
- See for yourself: www.virustotal.com
- **Solution:** use **layered defense** approach
  - Use a firewall, anti-virus, sandboxing, etc.
  - **Note:** running multiple AVs may cause issues
    - They may detect and delete one another!



Other Defenses

- **Tripwired Hashes**
  - Keep hash of known system files
  - **Then what?**

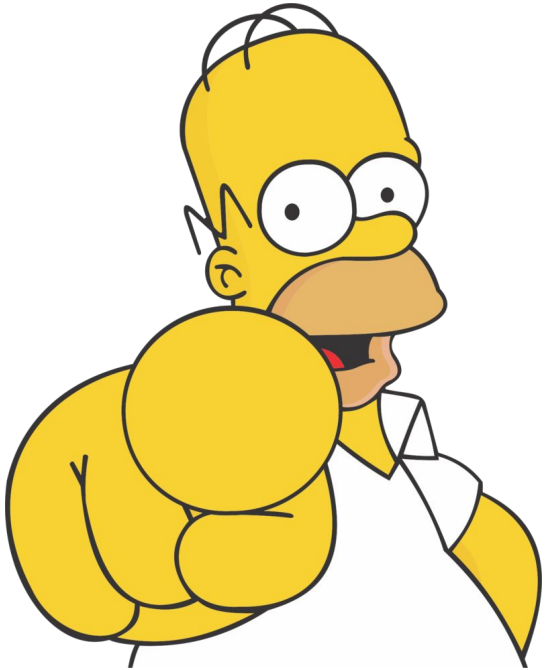
Other Defenses

- **Tripwired Hashes**
  - Keep hash of known system files
  - Periodically re-hash and check
    - **If hash changes, file tampered**

Other Defenses

- **Tripwired Hashes**
  - Keep hash of known system files
  - Periodically re-hash and check
    - **If hash changes, file tampered**

- Be a **security-conscious** citizen
  - Strong passwords, 2-factor authentication
  - Do not access suspicious files or websites
    - **Use your intuition: if it seems too good to be true, it probably is!**
  - Keep software updated and use anti-virus
  - **Teach others!**



Food for Thought

- **Using malware for good?**
  - E.g., would it be ethical to use a worm to patch a ubiquitous security vulnerability?
  - E.g., installing firewalls to censor websites we think are against the common good?

Food for Thought

- **Using malware for good?**
  - E.g., would it be ethical to use a worm to patch a ubiquitous security vulnerability?
  - E.g., installing firewalls to censor websites we think are against the common good?

- **Implications of sophisticated malware on public, international policy?**
  - E.g., intercepting everyone's phone records to find a handful of terrorists?
  - E.g., not disclosing critical vulnerabilities so as to stockpile cyberweapons?

Food for Thought

- **Using malware for good?**
  - E.g., would it be ethical to use a worm to patch a ubiquitous security vulnerability?
  - E.g., installing firewalls to censor websites we think are against the common good?
- **Implications of sophisticated malware on public, international policy?**
  - E.g., intercepting everyone's phone records to find a handful of terrorists?
  - E.g., not disclosing critical vulnerabilities so as to stockpile cyberweapons?
- **What if the hardware itself has been backdoored?**
  - "Reflections on Trusting Trust": Ken Thompson's 1983 Turing Award lecture
  - "A2: Analog Malicious Hardware": Matthew Hicks et al. in 2016 IEEE S&P

Food for Thought

- Using malware for good?
  - E.g., would it be ethical to use a worm to patch a ubiquitous security vulnerability?
  - E.g., installing firewalls to censor websites we think are against the common good?
- [illegible]
- [illegible] hardware itself has been backdoored?
  - "Reflections on Trusting Trust": Ken Thompson's 1983 Turing Award lecture
  - "A2: Analog Malicious Hardware": Matthew Hicks et al. in 2016 IEEE S&P

Maintain constant vigilance!

Questions?


NO SCHOOL
FALL BREAK

Next time on CS 4440...

Intro to The Web, and Web Security