Announcements

- **Project 2: AppSec** released
  - **Deadline:** Thursday, October 19th by 11:59PM—two weeks from today!
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%
See Discord for meeting info!

www.utahsec.com
Questions?
Access Control
Permissions
Process Isolation
Isolating Applications
Isolating Applications

- Application
- Middleware
- Operating System
- Hypervisor
- Hardware
Isolating Applications

- Application
- Middleware
- Operating System
- Hypervisor
- Hardware
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:
  - Remove or Restrict Privileges
  - Only Necessary Systems and Applications
Read a file in directory D
Access Control

**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

D = Directory (or a file, if “-”)

drwxrwxr-x 2 cs4440 cs4440 bin

drwxrwxr-x 2 cs4440 cs4440 __pycache__

-rwxrwxr-x 1 cs4440 cs4440 shellcode.py
Implementing Access Control

D = Directory (or a file, if “-”)

First three = ???
Implementing Access Control

D = Directory (or a file, if “-”)

First three = owner’s permissions
Implementing Access Control

D = Directory (or a file, if “-”)

First three = owner’s permissions

Second three = ???
Implementing Access Control

D = Directory (or a file, if “-”)

First three = owner’s permissions

Second three = group’s permissions

drwxrwxr-x 2 cs4440 cs4440 bin
drwxrwxr-x 2 cs4440 cs4440 __pycache__
-rwxrwxr-x 1 cs4440 cs4440 shellcode.py
Implementing Access Control

D = Directory (or a file, if “-”)

First three = owner’s permissions

Second three = group’s permissions

Last three = ???
Implementing Access Control

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?
   - Answer: ??? ??? ???

2. Read and Write only for world?
   - Answer: ??? ??? ???

3. Execute only for group?
   - Answer: ??? ??? ???

4. Owner can read, write, & exec; Group can only exec; and all others have no permissions.
   - Answer: ??? ??? ???
More Permission Puzzles!

1. Read/Write/Exec for all but group?
   - Answer: `rwx --- rwx`

2. Read and Write only for world?
   - Answer: `--- --- rw-`

3. Execute only for group?
   - Answer: `--- --x ---`

4. Owner can read, write, & exec; Group can only exec; and all others have no permissions.
   - Answer: `rwx --x ---`
Goal: minimize damage by isolating every process
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!
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: Containers**

**Goal:** completely isolate the OS
Isolation Technique: Containers

**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...
Malware Infection

- **How** does malicious software get on victim computers in the first place?
How does malicious software get on victim computers in the first place?

- A local application is exploited to perform arbitrary code execution

```
$ wget malware.zip
$ unzip malware.zip
$ ./malware.bin
```
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

Gray-market Valuations of Security Exploits for Commodity Closed-source Software (per Zerodium)

- $1 million to $2 million
- Up to $500,000
- $100,000 to $250,000
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 else** does malicious software get on victim computers?
How else does malicious software get on victim computers?
- 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

---

**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 content to entice users into interacting with malware.

Malvertising is often just part of your everyday browsing, which sends your computer files to a server and then to malware it downloaded.
How else does malicious software get on victim computers?
Malware Infection

- How else does malicious software get on victim computers?
  - Social engineering attacks
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 does malicious software get on victim computers?
How else does malicious software get on victim computers?
- Malicious hardware plugged-in; automatically executes code
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†
†Network Security Group, Georgia Tech, Atlanta, GA
††Department of Computer Science, Stanford University

Abstract — We investigate how users will plug in USB drives without any knowledge of the drive’s owner. We perform a controlled experiment in a large university campus environment that finds users will connect the first drive they find with an estimated success rate of 45–98%. 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 responsibility for social tasks than is commonly assumed.

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

- How else does malicious software get on victim computers?
**Malware Infection**

- **How else** does malicious software get on victim computers?
  - Supply chain attacks; insider threats
Case Study: SolarWinds Breach

- **Idea:** infect software provider that serves major targets
Case Study: SolarWinds Breach

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

---

Partial customer listing:

- Axiom
- Ameritrade
- AT&T
- BellSouth Telecommunications
- Best Western Intl.
- Blue Cross Blue Shield
- Boeing
- Capital One
- Cerner
- CenturyLink
- Cisco
- Citrix
- Comcast
- Convergint
- Dell
- Digital Realty
- Equinix
- Ford
- General Dynamics
- Gillette Deutschland GmbH
- GTE
- H&R Block
- Harvard University
- Hertz Corporation
- ING Direct
- Intel
- Intuit
- JPMorgan Chase
- John Deere
- Johnson Controls
- Johnson & Johnson
- Kaiser Permanente
- KPMG
- Lloyds
- Lockheed Martin
- Lufthansa
- Mastercard
- Microsoft
- Motorola
- NTT
- Navy
- Neiman Marcus
- Nestle
- New York Power Authority
- New York Times
- Nielsen Media Research
- Nortel
- Northrop Grumman
- Oracle
- Orange
- Oracle
- PayPal
- PNC
- Pricewaterhouse Coopers
- Proctor & Gamble
- Purdue University
- Raytheon
- Renault
- Renault
- Rice University
- Rockwell Automation
- Royal Bank
- Royal Bank
- Royal Dutch Shell
- Rx Savings Solutions
- SAP
- San Francisco Intl. Airport
- SAP
- Siemens
- Smart City Networks
- Smith Barney
- Smithsonian Institution
- Starbucks
- St. Jude
- Stratasys
- Swisscom
- Toyota
- U.S. Army
- University of Arkansas
- University of California
- University of Chicago
- University of Colorado
- University of Connecticut
- University of Delaware
- University of Florida
- University of Georgia
- University of Hawaii
- University of Illinois
- University of Kansas
- University of Kentucky
- University of Massachusetts
- University of Maryland
- University of Minnesota
- University of Missouri
- University of Nebraska
- University of Nevada
- University of New Mexico
- University of North Carolina
- University of Notre Dame
- University of Oklahoma
- University of Oregon
- University of Pennsylvania
- University of Pittsburgh
- University of Texas
- University of Virginia
- US Dept. Of Defense
- US Postal Service
- US Secret Service
- Visa USA
- Volvo
- Williams Communications
- Yahoo

---

SolarWinds’ Customers

SolarWinds’ comprehensive products and services are used by more than 300,000 customers worldwide, including military, Fortune 500 companies, government agencies, and education institutions. Our customer list includes:

- More than 425 of the US Fortune 500
- All ten of the top ten US telecommunications companies
- All five branches of the US Military
- The US Pentagon, State Department, NASA, NSA, Postal Service, NOAA, Department of Justice, and the Office of the President of the United States
- All five of the top five US accounting firms
- Hundreds of universities and colleges worldwide
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**
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 “mastered” non-Windows OSs
  - Linux, MacOS, iOS aren’t as safe as you think!

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
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**)

---

Worm: Win32 Conficker

- Computers within a network that have weak passwords and without latest security updates/anti-virus software are infected with the worm.
- Computers that have unsecured/open shared folders without latest security update/anti-virus software are infected with the worm.
- Computer without a strong password, secured shared folder, latest security update or anti-virus software is infected with the worm.
- Computer with strong password, secured shared folder, latest security update and anti-virus software is protected from the worm.
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

---

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**

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 copyleft 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 for 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**

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
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
  - 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!**
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?
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?

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

Maintain constant vigilance!
Questions?
NO SCHOOL
FALL BREAK
Next time on CS 4440...

Intro to The Web, and Web Security