CPEN 442 – Introduction to Cybersecurity

Module 1

Introduction to Cybersecurity

This material in these slides is largely taken from the "CS458: Computer Security and Privacy" course at the University of Waterloo, and it has been originally designed by Profs. Ian Goldberg and Urs Hengartner, with contributions of other instructors.

Course Goals

- The primary goal is to be able to identify security issues in various aspects of computing, including:
  - Software
  - Operating systems
  - Networks
  - Internet applications
- The secondary goal is to use this ability to design systems that are more protective of security (and privacy)

Module Outline

Part 1: Fundamental Concepts

1. Cast of characters
2. What is cybersecurity?
3. Terminology
4. Types of attacks
5. Types of adversaries
6. Methods of defense

Part 2: Cybersecurity Design Principles

1. Simplicity
2. Failsafe defaults
3. Complete mediation
4. Open design
5. Separation of privileges
6. Least privilege
7. Least common mechanism
8. Ease of use

CPEN 442 – Introduction to Cybersecurity

Module 1 – Introduction

Part 1 – Fundamental Concepts

The cast of characters

- When talking about cryptography, but also more generally about cybersecurity, there is a recurrent cast of characters:

Alice **Bob** **Carol** **Dave**

Legitimate/honest users

Malicious CA?

Trent
The Trusted Third-Party (TTP)
or Certificate Authority (CA)

Eve
The Eavesdropper
(passive adversary)

Mallory
Man-in-the-Middle (MitM)
(active adversary)

If you're curious, check the Wikipedia page for Alice and Bob

What is Cybersecurity?

- Cybersecurity = computer security

"the combined art, science, and engineering practice of protecting computer-related assets from unauthorized actions and their consequences […]"

van Oorschot, "Tools and Jewels"

- Computer security usually protects from *intentional* misuse.
- There is also *unintentional* damage, but this is not the focus of the course.

The Objectives of Cybersecurity

Which one do you think is the hardest to get?

- The general goals of cybersecurity are:
  - **C**onfidentiality: the information can only be read by authorized parties.
    - Usually achieved via cryptography (encryption)
  - **I**ntegrity: the information can only be modified by authorized parties.
  - **A**vailability: information, services, and computing resources should remain accessible for authorized parties.
- Easy to remember as the CIA acronym.
- These appear, for example, in the NIST Handbook on Computer Security
- Another important concept to remember is:
  - **A**uthenticity/authentication: being able to verify that someone or something is genuine.

What is privacy?

- In this course, we will focus on security, and not privacy
- However, these concepts are related
- There are many definitions of privacy, a useful one is:

  "informational self-determination"

- This means you get to control information about you:
  - Who gets to see it…
  - Who gets to use it…
  - What they can use it for…
  - Who they can give it to…
  - …
- Not the main focus of this course, but it will come up and you need to be able to "distinguish" between security and privacy

Example: confidentiality, integrity, availability, and privacy

Identify whether the following compromise confidentiality, integrity, availability, and/or privacy

1. A hacker breaks into your phone and steals the pictures in your camera folder.
2. A parent installs spyware on their kid's laptop without their kid's consent.
3. A wifi access point is modified so that it replaces URLs with ad websites.
4. Google collects your location information when you access Google Maps.

- **Confidentiality**: the information can only be read by authorized parties.
- **Integrity**: the information can only be modified by authorized parties.
- **Availability**: information, services, and computing resources should remain accessible for authorized parties.
- **Privacy**: control who can see, user, share, etc., your personal information

Some terminology

- Assets: something of value that we want to protect
  - data, hardware, software, computing resources, a network, etc.
- Vulnerabilities:
  - weaknesses in a system that may be able to be exploited to cause a loss or harm
- Threats:
  - the loss or harm that might happen to a system
- Security policy: specifies the design intent of a system's rules (what's allowed and what's not)
  - A security policy is violated if it moves to an unauthorized state
- Attack: an action or steps which exploit a vulnerability to execute a threat.
  - Successful execution would case a security violation
  - Attacker or adversary is the party that executes the attack
- Defenses or security controls: protect against attacks

Do not stress, these are just so that we're all on the same page

Types of attacks

- There are four major categories of attacks:
  1. Interception: an unauthorized party gains access to (confidential) information
  2. Interruption: service is made unavailable for legitimate users
  3. Modification: an unauthorized party alters information
  4. Fabrication: create illegitimate information
- When designing a system, we need to state the adversary/threat model
  - Objectives (what does the adversary want to do?)
  - Methods (what attack techniques does the adversary use?)
  - Capabilities (computing resources, knowledge, opportunities, etc.)
  - Funding level (these affects the methods and capabilities)
  - Outsider vs insider
  - …
- Whom do we want to prevent from doing what?

Can you defend against any possible threat?

Example: types of attacks

- **Paper-based voting**: is the system susceptible to each type of attack? Provide a realistic attack scenario for that case.
  - Interception: wireless camera to spy on voters? someone taking a picture of their vote?
  - Interruption: users called and given the wrong location for voting, or they are told voting is on a different date
  - Modification: the official counting the votes marks an extra option, making the vote not valid
  - Fabrication: double-voting, the ballot box was not empty at the start of voting
- **Internet voting**: each user gets mailed a letter with a URL and a unique code for voting. Is the system susceptible to each type of attack? Provide a realistic attack scenario for that case. (State your assumptions.)

- **Interception**: an unauthorized party gains access to (confidential) information
- **Interruption**: service is made unavailable for legitimate users
- **Modification**: an unauthorized party alters information
- **Fabrication**: create illegitimate information

Who are the adversaries?

There are many possibilities; roughly in order of strength:

- Murphy
- Amateurs
- "Script kiddies"
- Crackers/Hackers
- Organized crime (groups)
- Cyber-terrorists or politically-motivated adversaries
- Foreign intelligence (government-funded agencies)
- …

Methods of Defense

- How can we defend against a threat?
  - Prevent it: avoid the attack.
  - Deter it: make the attack harder or more expensive.
  - Deflect it: make yourself less attractive to the attacker.
  - Detect it: notice that the attack is occurring or has occurred.
  - Recover from it: mitigate the effects of the attack.

Example: methods of defense

- Threat: your car may get stolen, how can you…

Prevent it: avoid the attack.
Deter it: make the attack harder or more expensive.
Deflect it: make yourself less attractive to the attacker.
Detect it: notice that the attack is occurring or has occurred.
Recover from it: mitigate the effects of the attack.

Example: methods of defense

- Threat: your car my get stolen, how can you…
  - Prevent it: not have a car
  - Deter it: wheel clamp, immobilizer, …
  - Deflect it: hide valuables, add car alarm sticker, …
  - Detect it: car alarm, location tracker, …
  - Recover from it: insurance

Prevent it: avoid the attack.
Deter it: make the attack harder or more expensive.
Deflect it: make yourself less attractive to the attacker.
Detect it: notice that the attack is occurring or has occurred.
Recover from it: mitigate the effects of the attack.

How secure should we make a system?

- Principle of easiest penetration
  - "A system is only as strong as its weakest link"
  - The attacker will go after whatever part of the system is easiest for them, not for you
  - In order to build secure systems, we need to learn how to think like an attacker!!

How would you get secret information from the CRA?

- Principle of adequate protection
  - "Security is economics"
  - Don't spend $100,000 to protect a system that can only cause $1,000 in damage

The weakest link

Methods of Defense

- There are many methods to protect our assets:
  - Cryptography: provides confidentiality, integrity, and authentication over insecure channels, protects confidentiality and integrity of stored data, etc. (see Modules 2 – 4)
  - Software controls: defenses implemented as software, e.g., passwords, virus scanners, firewalls, etc. (Modules 5 and 6)
  - Hardware controls: using specific hardware to protect the system, e.g., fingerprint readers, smart tokens, some firewalls, trusted execution environments, etc.
  - Physical controls: protection of the hardware itself, e.g., locks, guards, off-site backups, etc.
  - Policies and procedures: non-technical means of protection, e.g., UBC privacy rules to comply with FIPPA, rules about choosing passwords, training, etc.

CPEN 442 – Introduction to Cybersecurity

Module 1 – Introduction

Part 2 – Cybersecurity Design Principles

Cybersecurity Design Principles

- The security of a system has to be part of its design early on
  - Hard to retrofit security, see Windows 95/98
- There is not one complete checklist one can follow to ensure a system is secure.
- However, there are useful and widely applicable design principles.
- The following is a list of eight design principles, taken from Saltzer and Schroeder https://web.mit.edu/Saltzer/www/publications/protection/Basic.html (Section I.A.3)
- A similar (and longer) list can be found on van Oorschot https://www.scs.carleton.ca/~paulv/toolsjewels/TJrev1/ch1-rev1.pdf (Section 1.7)

Trusted System Design Principles

1. Simplicity (economy of mechanism): keep the design of the protection mechanism as simple and straightforward as possible
   - Simpler design are easier to check for errors (e.g., smaller code implementations)
   - Minimize the attack surface

2. Fail-safe defaults (Permission-based / Default deny): access decisions based on permission rather than exclusion.
   - Explain who we allow to use the system (allowlists), rather than who we do not (denylists)
   - The default should be to not provide access to the system
   - If we forget to give someone permission to do something, we will notice soon enough

Trusted System Design Principles

3. Complete mediation: every access to every object must be checked for authority.

4. Open design: the design of a system should not be secret
   - Avoid "security by obscurity": the security of a system should not rely on the secrecy of its design details or the attacker's ignorance:
     - Assume the adversary knows all the system details
     - We can have secret keys or passwords, but not secret algorithms
   - If you decide to publish it: open-source code can be checked for errors by many people.

Trusted System Design Principles

5. Separation of privileges: if possible, split up access privileges to different parts of the system
   - Two or more people or processes must cooperate in order to get access
   - E.g., an adversary compromising one part of the system only can still not gain access
   - A system that requires two keys to unlock it is more robust and flexible than one that allows access only with a single key

6. Least privilege: every program and user of the system should operate using the least set of privileges necessary to complete their job
   - Only grant access to sensitive information if it is essential for the job

Trusted System Design Principles

7. Least common mechanism: minimize the amount of mechanism common to more than one user and depended on by all users
   - e.g., if you want some service available to all users, it is better to provide a library so that each user can decide if they want to use it or not, than having a single version in the OS for everyone
   - Genetic diversity is good! (we will see this in Module 4)
   - a shared mechanism (with shared variables) can be used as a covert channel (intentionally) or could be a side channel (unintentionally)

8. Ease of use (psychological acceptability): the human interface must be designed for ease of use
   - If using the system is annoying, the users will not use it or will use it incorrectly
   - The users are trying to do "things", not "secure things"

Recap

- Goals:
  - Identify security issues
  - Know how to prevent/defend against them
- What is security?
  - Confidentiality, integrity, availability, (authentication),
- What is privacy?
  - Informational self-determination
- Adversary models
  - Types of attacks
  - Who is the adversary?
- Defending
  - Cryptography, software controls, hardware controls, physical controls, policies and procedures

- Cybersecurity design principles:

1. Simplicity
2. Failsafe defaults
3. Complete mediation
4. Open design
5. Separation of privileges
6. Least privilege
7. Least common mechanism
8. Ease of use