

# **CS 458: Computer Security and Privacy**

Michael Noukhovitch

Spring 2016, University of Waterloo

Notes written from Erinn Atawater's lectures.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Security . . . . .	3
1.2	Privacy . . . . .	3
1.3	Terminology . . . . .	3
1.4	Types of Defence . . . . .	3
1.5	Methods of Defence . . . . .	4
<b>2</b>	<b>Program Security</b>	<b>4</b>
2.1	Flaws, faults, and failures . . . . .	4
2.1.1	Defintions . . . . .	4
2.1.2	Unexpected Behaviour . . . . .	4
2.2	Unintentional Security Flaws . . . . .	4
2.2.1	Types of Flaws . . . . .	4
2.3	Buffer Overflow . . . . .	4
2.3.1	Definition . . . . .	4

# 1 Introduction

## 1.1 Security

Security can be defined as:

**confidentiality** access to systems is limited to authorized

**integrity** getting the correct data

**availability** system is there when you want it

## 1.2 Privacy

There are many definitions but we will stick to **informational self-determination**, where you control the information about you

## 1.3 Terminology

**assets** things we want to protect

**vulnerabilities** weaknesses in a system that can be exploited

**threats** loss or harm that may befall a system

- interception
- interruption
- modification
- fabrication

**threat model** set of threats to defend against (who/what)

**attack** an action which exploits a vulnerability to execute a threat

**control** removing or reducing a vulnerability

## 1.4 Types of Defence

Defend against an attack:

- **prevent** stop the attack from happening
- **deter** make the attack more difficult
- **deflect** make it less attractive for attacker
- **recover** mitigate effects of the attack

Make sure that defence is correct with principles:

- **easiest penetration** system is only as strong as weakest link
- **adequate protection** don't spend more on defence than the value of the system

## 1.5 Methods of Defence

- Software controls: passwords, virus scanner ...
- Hardware controls: fingerprint reader, smart token ...
- Physical controls: locks, guards, backups ...
- Policies: teaching employees, password changing rules

## 2 Program Security

### 2.1 Flaws, faults, and failures

#### 2.1.1 Definitions

**flaw** problem with a program

**fault** a potential error inside the logic

**failure** an actual error visible by the user

#### 2.1.2 Unexpected Behaviour

A spec will list the things a program will do but an implementation may have additional behaviour. This can cause issues as these behaviours might not be tested and would be hard to test.

### 2.2 Unintentional Security Flaws

#### 2.2.1 Types of Flaws

- **intentional**
  - **malicious**: inserted to attack system
  - **nonmalicious**: intentional features meant to be in the system but can cause issues
- most flaws are **unintentional**

### 2.3 Buffer Overflow

#### 2.3.1 Definition

Most common exploited type of security flaw when program reads or writes past the bounds of the memory that it should use.

**Example 2.1.** Basic buffer overflow

```
#define LINELEN 1024

char buffer[LINELEN];
strcpy(buffer, argv[1]);
```