

Introduction to Computer Security



ECE568 – Lecture 1
Courtney Gibson, P.Eng.
University of Toronto ECE

Lab Assignment #7: Inheritance

1. Objectives

The objective of this assignment is to provide you with practice on the use of inheritance in C++ programming. This will be done in the context of re-implementing the simple student-marks database of Assignment 5 to allow the storage and retrieval of records of any type, not only of type `studentRecord`.

2. Problem Statement

In this assignment, you will implement a simple array-based database to store and retrieve records. In the first part of the assignment, you will implement two classes: `Record` and `DB`. The `Record` class will serve as a base class from which other types of record classes can be derived. The `DB` class will be used to create a database of `Record` objects. In the second part of the assignment, you will design and implement the class `studentRecord`, which is derived from the class `Record`. You will test your implementation with the `Driver` you wrote for Assignment 5. However, your implementation of `Record` and `DB` must work for any class that is derived from `Record`, even without any knowledge on your part of what the derived class does.

2.1 The Record Class

The `Record` class has fields to represent the number (`key`) of an individual. It also has the

BLIND SPOTS

THE PROBLEM WITH

ASSUMPTIONS

COME WITH

IS THAT THEY ALWAYS

~ Olivier Blanchard





Photos reveal NSA tampered with Cisco router prior to export

By JR Bookwalter, Routers & storage

Caught with hands in virtual cookie jar



TS//SI//NF) Left: Intercepted packages are opened carefully; Right: A "load station" implants a beacon



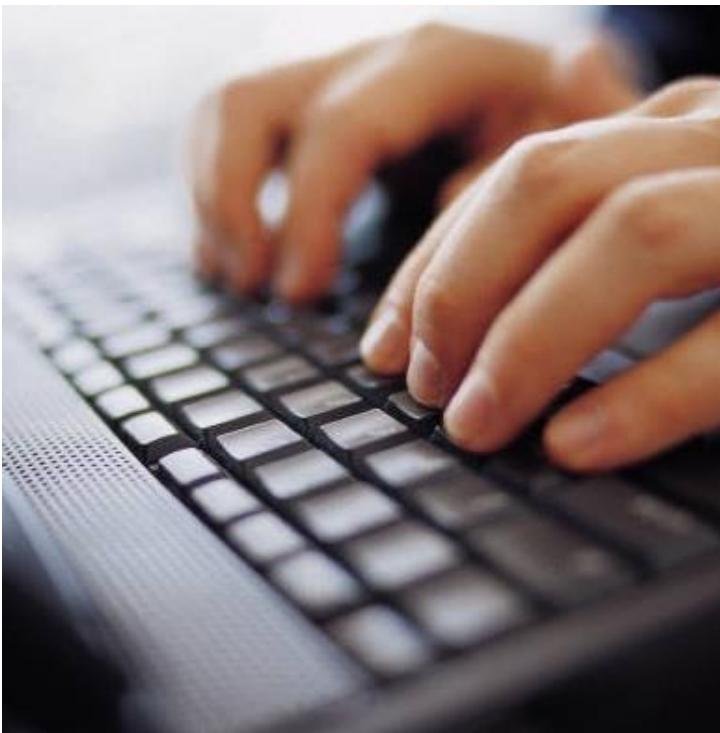
Motivation

How is this relevant to me?

Motivation: User

Software systems are ubiquitous in our daily lives

- Protect your system
- Protect your data
- Identity theft



The New York Times

Somebody's Watching: Hackers Breach Ring Home Security Cameras

Unnerved owners of the devices reported recent hacks in four states. The company reminded customers not to recycle passwords and user names.

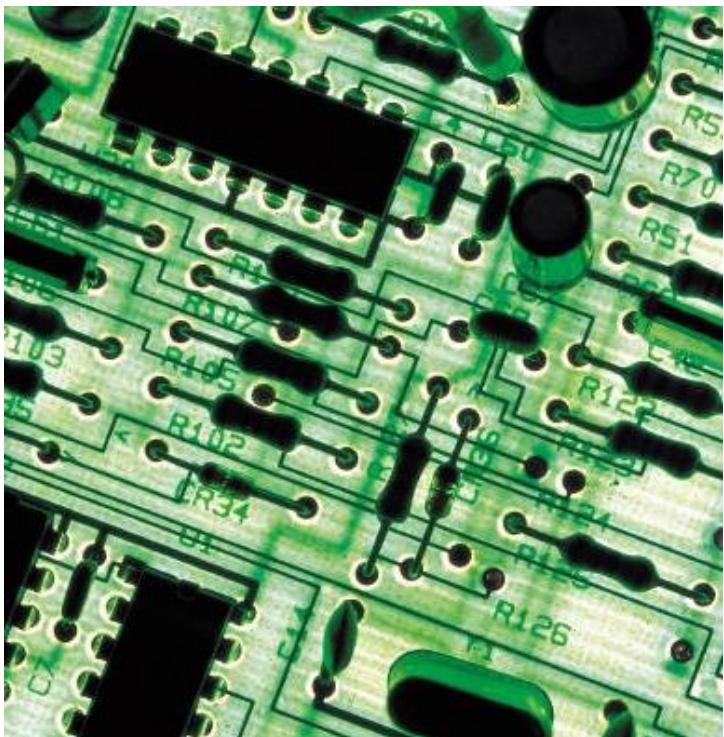


A family in Mississippi said a man hacked into a Ring home security camera in a bedroom shared by their daughters. Ashley LeMay

Motivation: Engineer

Security should be a daily consideration in your work

- Challenge your design assumptions
- Understand attackers and identify risks
- Defensive coding



SolarWinds hack officially blamed on Russia: What you need to know

The hack has compromised local, state and federal agencies as well as major tech companies.

▶ LISTEN • 08:26



Laura Hautala Jan. 5, 2021 4:32 p.m. PT



US Intelligence agencies have said Russia is responsible for a major hacking campaign striking federal agencies and major tech companies

Angela Lang/CNET

Motivation: Employee

Laws increasingly hold companies accountable for poor security practice

- Protect your company
- Protection of confidential data
- Regulatory requirements

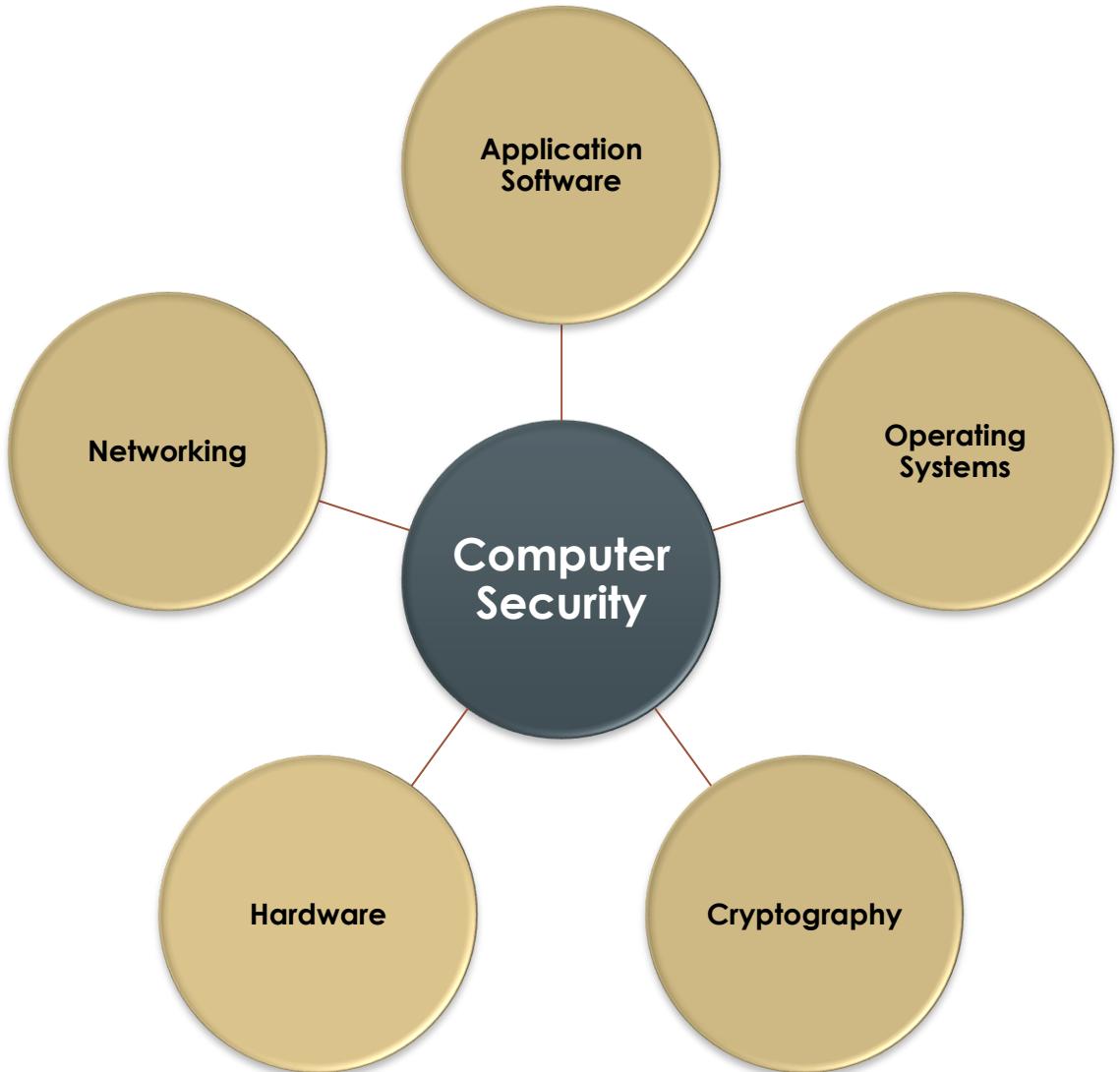




While previous generations of attacks targeted technology such as networks and servers and exploited vulnerabilities in software, attackers have now evolved to target human inadequacies and weaknesses . . . These targeted and personalized attacks are difficult to prevent because they leverage human vulnerabilities and human trust.

We believe it is reasonable to assume, if an advanced attacker targets your company, that a breach is inevitable.

Kevin Mandia, October 4, 2001
*Before the Permanent Select Committee
on Intelligence, U.S. House of Representatives*





Computer Security

Rules and Assumptions

Rules

Like the programs that enforce them, computer systems have rules. Some rules are **explicit** and well thought out; others are **implicit** and based on assumptions.



Assumptions

Security risks occur
when our
assumptions turn out
to be false

- Data
- Input
- User behaviour



Computer Security

Computer security is about understanding a system really well, and questioning the implicit rules

- A **reliable** system does what it is supposed to do
- A **secure** system does what it is supposed to do, and nothing else.



Computer Security

Why is it hard?

Security is a Negative Goal

Our job is to ensure that something **cannot** happen: much harder to measure / verify.

- **Positive goal:** Alice can read the file
- **Negative goal:** Bob cannot read the file

Problem: In what ways might Bob try to access the file? (Not an easy question to answer.)

Identifying the Weakest Link

- Programmers are often not trained to consider their adversaries
- The weakest part of the system will be exploited
- Expect the unexpected





Instructor

Background, Contact Info

Courtney Gibson

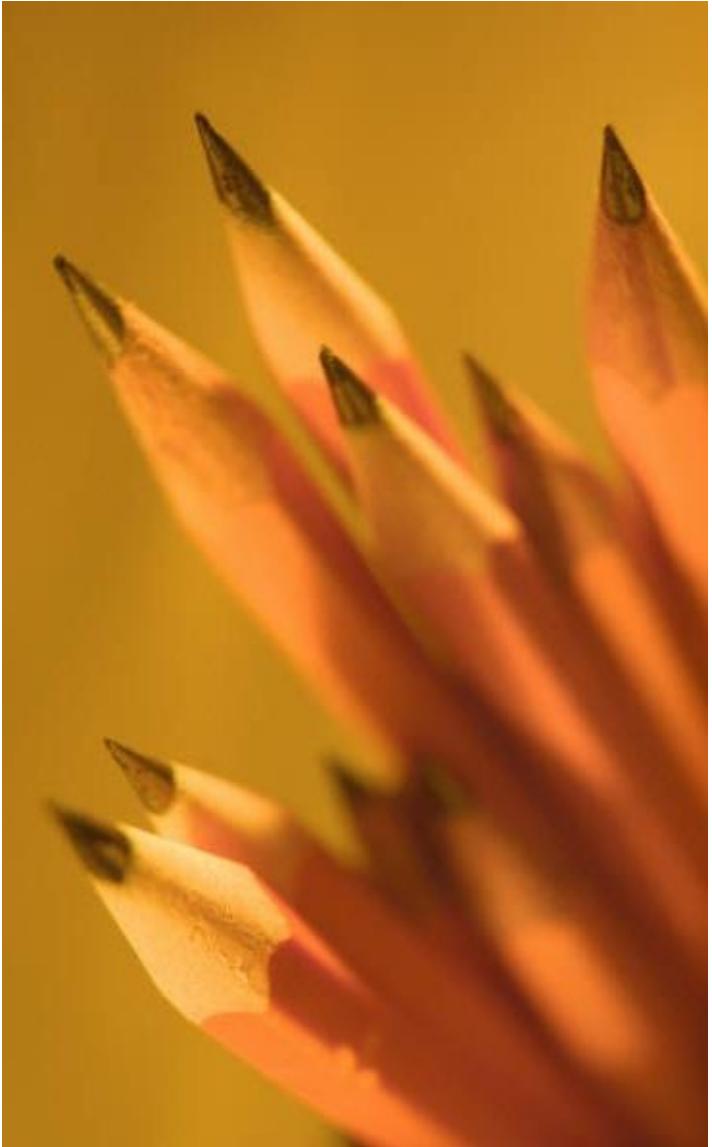


UofT Electrical and Computer Engineering
P. Eng., Adjunct Lecturer



BioConnect
Chief Technology Officer
Chief Information Security Officer

email: courtney.gibson@utoronto.ca
office hours by request



Course Outline

Major Topics

Course Page

<https://q.utoronto.ca/>



- Syllabus
- Lecture Slides
- Assignments
- Past Tests
- Announcements
- Midterm + Final

Course Outline

Section 1: Introduction

- Course introduction
- Quick review: O/S, assembly language, software tools
- Basic principles of computer security
- Confidentiality, Integrity, Availability

Primary Learning Goals:

- Ability to explain the fundamental goals of Confidentiality, Integrity and Availability, and how they each can be applied to creating secure product designs.
- Ability to use basic software debugging tools (e.g., gdb) to analyze program stack behaviour.

Course Outline

Section 2: Software Code Vulnerabilities

- Injection-type attacks: buffer overflows, format strings, double-free, ROP, etc.
- Library attacks: encapsulation, etc.
- Defenses and good programming practice

Primary Learning Goals:

- Ability to explain and execute the most common attacks against software vulnerabilities, in order to locate design and implementation flaws.
- Ability to use appropriate tools and best-practice coding techniques to protect against these types of attacks in your own work.

Course Outline

Section 3: Cryptography

- Encryption: Symmetric (block/stream ciphers) and Public Key
- Secure key exchange
- Hashes and Signatures: MAC, HMAC, etc.
- Secure Communication Protocols: SSL, VPN

Primary Learning Goals:

- Ability to select appropriate encryption techniques to meet both the Data Confidentiality requirements and the performance requirements of your software.
- Ability to securely exchange cryptographic keys over insecure network channels.
- Ability to use hash-based tools to provide Data Integrity and Data Authentication, through the application of MACs, HMACs and digital signatures.

Course Outline

Section 4: Web Security

- Web authentication, cookies
- Web attacks: Cross-Site Scripting (XSS), CSRF, etc.
- SQL Injection
- Cloud security

Primary Learning Goals:

- Ability to safely use web cookies to secure sessions and end-user data.
- Ability to execute the most common web-based attacks, in order to locate design and implementation flaws.
- Ability to use appropriate tools and best-practice coding techniques to protect against these types of attacks in your own work.

Course Outline

Section 5: Access Control

- Access control models
- Hardware Security Modules (HSMs)
- Side-Channels and Covert Channels
- Attacks on IoT/embedded systems

Primary Learning Goals:

- Ability to identify potential sources of side- and covert-channels that could be used to leak information out of both hardware- and software-based systems.
- Ability to identify and describe potential injection vulnerabilities, and how they might compromise the security of IoT/embedded systems.

Course Outline

Section 6: Network Security

- Network-layer security risks: BGP, ARP, DNS, TCP/IP, etc.

Primary Learning Goals:

- Ability to identify how vulnerabilities in many standard networking protocols could be exploited by a third-party to pose risks to confidentiality, integrity and availability of your software systems.
- Ability to use appropriate tools and best-practice techniques to protect your own systems and designs.

Course Outline

Section 7: Malware

- Viruses and worms
- Rootkits and bootkits

Primary Learning Goals:

- Ability to identify the mechanisms through which computer viruses, worms, rootkits and bootkits functions and propagate, and the risks these may pose to your systems.

Course Outline

Section 8: Physical Security

- Physical security system design and vulnerabilities

Primary Learning Goal:

- Ability to explain the most common vulnerabilities and defenses in physical security systems, and relate this to design challenges in digital security designs.

Optional Texts

Security in Computing

- Pfleeger and Pfleeger

Computer Security: Principles & Practice

- Stallings and Brown

Applied Cryptography

- Bruce Schneier

Marking Scheme

- Labs: 25%
- Midterm: 35%
- Final Exam: 40%

The final is “Type C”
(single reference sheet),
no calculator.



Plagiarism

All labs, assignments and tests are to be completed with your original work.

Anything submitted for credit must be something that you (and your lab partner) produced.



What to Expect

Course Covers a Lot of Material

- OS, Networking
- Mathematics of Cryptography

Where You'll Spend Your Time

- Four Labs
- Most of the work will be in the labs: course focuses on practical aspects of security

Labs

Lab 1: Identification of Vulnerabilities, Construction of Attacks

- You will be given some vulnerable programs.
- Your job is to construct attacks that will let you hijack the programs and spawn a command shell.

Labs

Labs 2-3: Network and Web Security

- ~~You will use SSL to write code to securely communicate between a client and a server.~~ (← Lab 2 under revision)
- You will be given a web application. You will craft a series of attacks that exploit vulnerabilities in the application's design.

Lab 4: Single Sign-On