General Information

Welcome to ECE 568!

This course covers principles of computer systems security. We will start by examining a variety of security vulnerabilities in general software applications, and provides you with the knowledge to identify these weaknesses in your own work, understand how these vulnerabilities could be exploited, and then discusses techniques you can use to help defend against such attacks. We will then cover the basic elements of modern cryptography, and the security of operating systems, networks and web/cloud applications.

Lectures will be delivered synchronously, and you are highly-encouraged to attend. Opportunities will be provided to interact, ask questions and confirm your understanding of the material. If you are unable to attend the live lectures, recordings will be available via Quercus.

Topics Covered

We will cover the following topics in class; slides will be posted prior to the lectures to assist your note-taking:

- Section 1: Introduction
  - Course introduction
  - Quick review: Operating Systems, 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.
• Section 2: Software Code Vulnerabilities
  Injection-type attacks: buffer overflows, format strings, double-free, Return-Oriented Programming 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.

• Section 3: Cryptography
  Encryption: Symmetric (block/stream ciphers) and Public Key
  Secure key exchange
  Hashes and Signatures: MAC, HMAC, bcrypt, Merkle trees, TOTP/HOTP
  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.

• 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 secure sessions and end-user data.
  • Ability to identify and demonstrate 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.
• **Section 5: Access Control**
  Access control models
  Hardware Security Modules (HSMs)
  Side-Channels and Covert Channels
  Attacks on embedded/IoT 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.

• **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.

• **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 function and propagate, and the risks these may pose to your systems.

• **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.

**Instructor**

Prof. Courtney Gibson, P.Eng.
Department of Electrical and Computer Engineering
courtney.gibson@utoronto.ca
Office hours by appointment
Course Website

Information on ECE568, including important announcements, lecture slides, recordings, assignments and course marks can be found on Quercus (https://q.utoronto.ca/). Please visit the website on a regular basis for up-to-date information.

Textbook

There is no required textbook for the course. However, the instructor will provide a list of reference books in class and lecture slides on the course web site. Also, various other resources will be available on the web site.

Timetable

The timetable for the course is shown below. You are highly encouraged to attend the weekly lecture (three hours) each week; recordings will be posted after the lecture for those unable to attend. The Teaching Assistants will be on-line and answering questions live during the designated lab times, and will be answering questions asynchronously through the course forums outside of those hours.

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Monday</th>
<th>6:10pm – 9:00pm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labs</td>
<td>Tuesday (P0101)</td>
<td>3pm – 6pm</td>
</tr>
<tr>
<td></td>
<td>Tuesday (P0102)</td>
<td>3pm – 6pm</td>
</tr>
</tbody>
</table>

Labs

The labs consist of a number of programming exercises that will take a substantial amount of your time – but students who have taken this course in past years consistently report that these labs have taught them a tremendous amount of useful material. Please take them seriously, and ask for help early if you need it.

IMPORTANT: Some of the lab assignments will have us doing things that we are not normally “supposed” to be able to do (e.g., attacking applications with invalid input, and changing how they run); the nature of what we are doing means that different operating systems, processors and system security policies may effect our “attacks” and how they run. The TAs will test your lab on the ECF lab workstations (p___.ecf.utoronto.ca). For those labs that require a compiler, you should be using the C compiler installed in /usr/bin/gcc. You may do the labs on your own machines - but, if you do, it is your responsibility to make sure that your solutions work on the ECF computers. My suggestion is to SSH into the machines from home, if you prefer to work remotely, rather than discovering at the last minute that your personal operating system or computer configuration is different than the lab machines. PLEASE NOTE that you should not use remote.ecf.utoronto.ca for your labs: it is running an O/S that will respond differently than the O/S on the lab workstations. If you do not have an ECF account, please email me and I will sort things out for you.
The TAs will be using automated scripts to aid them in grading the labs: as a result, it is important that you follow the submission instructions for each lab carefully. You are encouraged to include documentation for your labs (not exceeding 1 page, please no essays!). If your labs do not work completely, the TAs may use this documentation to assign part marks. Lab attendance is not required. Labs may be done individually or in groups of two students.

Tutorials
There are no tutorials in this course. Please make use of the TAs and office hours with the instructor.

Notice of video recording and sharing
At times during this course, some interactions including your participation, may be recorded on video and will be available to students in the course for viewing remotely and after each session.

Course videos and materials belong to the instructors, the University, and/or other source depending on the specific facts of each situation, and are protected by copyright. You are permitted to download session videos and materials for your own academic use, but you should not copy, share, or use them for any other purpose without the explicit permission of the instructor.

For questions about recording and use of videos in which you appear please contact your instructor.

Course Policies
Plagiarism (of code or of written material) will not be tolerated. You and your lab partner are jointly responsible for ensuring that your submitted lab work is original work. If for some valid reason you are unable to submit the lab on time, please provide an explanation and appropriate documentation.

Re-grading
Everybody makes mistakes, including TAs and the instructor! I will ensure you receive fair evaluation for your work in this course. If you feel that there has been a grading mistake, you can request a reevaluation of your work within three weeks of the test or lab results being returned. You should submit a short note explaining which questions are in error and why you think you deserve a regrade. A TA or the instructor will regrade the entire assignment, and fix any marking mistakes. (It is not guaranteed that you will receive a higher mark: it’s unlikely, but you may end up with a lower grade on your assignment if it’s determined that the marking key was not properly applied.)
Important Dates

The release date and due dates for the labs and assignments are shown below. A handout for each lab will be available from the course website, and no hard copies will be provided in class. **Labs are due by no later than 11:59pm EST/EDT on Friday of the week they are due.**

<table>
<thead>
<tr>
<th>Week 1 – Jan 11-15</th>
<th>Course orientation: no labs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 2 – Jan 18-22</td>
<td>Lab #1 released.</td>
</tr>
<tr>
<td>Week 3 – Jan 25-29</td>
<td>Lab hours begin.</td>
</tr>
<tr>
<td>Week 4 – Feb 1-5</td>
<td><strong>Lab #1 due.</strong></td>
</tr>
<tr>
<td>Week 5 – Feb 8-12</td>
<td>Lab #2 released.</td>
</tr>
<tr>
<td>Feb 15-19</td>
<td><strong>Reading Week: No lectures or labs.</strong></td>
</tr>
<tr>
<td>Week 6 – Feb 22-26</td>
<td><strong>Lab #2 due.</strong> Lab #3 released.</td>
</tr>
<tr>
<td>Week 7 – Mar 1-5</td>
<td>Midterm.</td>
</tr>
<tr>
<td>Week 8 – Mar 8-12</td>
<td><strong>Lab #3 due.</strong></td>
</tr>
<tr>
<td>Week 9 – Mar 15-19</td>
<td>Lab #4 released.</td>
</tr>
<tr>
<td>Week 10 – Mar 22-26</td>
<td></td>
</tr>
<tr>
<td>Week 11 – Mar 29 – Apr 2</td>
<td><strong>Lab #4 due.</strong></td>
</tr>
<tr>
<td>Week 12 – Apr 5-9</td>
<td></td>
</tr>
<tr>
<td>Week 13 – Apr 12-16</td>
<td>Last week: final exam review.</td>
</tr>
</tbody>
</table>

Marking and Evaluation

There will be one mid-term exam during the course, and a final exam during the final exam period. The midterm and final will both be delivered via Quercus, with accommodations to recognize both timezones and individual needs. The exams will be made available for a 24-hour window, and you may take test at any time during that period. You will be given an extra 30 minutes buffer on each exam, to deal with any Internet connectivity issues, etc.. The details for the exact timing of the midterm and final exam will be provided in class and on the web site. Copies of previous midterms and final exams will be provided as a study aid.

The composition of the final mark is as follows:

- Labs: 25%
- Mid-Term Exam: 35% (90 minute exam + 30 minutes buffer)
- Final Exam: 40% (2.5 hour exam + 30 minutes buffer)
- Calculator Type: 4 (none)
- Exam Type: C (single reference sheet, both sides)