Department of Electrical and Computer Engineering

ECE 7420: Computer Security

Instructor	Jonathan Anderson CSF-4123 jonathan.anderson@mun.ca
Office Hours	Mondays 1000–1100 and Wednesdays 1100–1200
Website	https://introsec.ca
Communication	Please email me using my `@mun.ca` address, as I check Brightspace/D2L/`online.mun.ca` infrequently.
Calendar	**ECE 7420: Computer Security** introduces students to key computer security concepts for applications, hosts, networks and the Web. Students will learn to employ the primitives provided by programming languages, cryptography, operating systems and network protocols for protecting engineered systems and their users.
Prerequisites	ECE 6500: Computer Architectures, ECE 6610: Communication Networks
Schedule	Lecture MTR 1300–1350 EN-2006 Lab F 1400–1700 CSF-2112
Credit value	3 credit-hours
Textbook	*Computer Security and the Internet: Tools and Jewels from Malware to Bitcoin*, van Oorschot, 2nd Ed. (2021).
Reference	*Security Engineering: A Guide to Building Dependable Distributed Systems*, Anderson, 3rd Ed. (2020).
Graduate Attributes	**KB:** A knowledge base for engineering, **PA:** Problem analysis, **Impacts:** Impact of engineering on society and the environment,

1. Course outcomes and graduate attributes

Upon successful completion of ECE 7420: Computer Security, students should be able to:

Learning outcome	Attr-Prof[1]	Assessment
explain how cryptography helps protect engineered systems	KB–D, Imp–D	assignments, case study, exams
explain mitigation techniques such as `W^X` and ASLR	KB–A, PA–A	assignments, labs, exams
design sandboxable software using language and OS primitives	PA–A, D–A	assignments, exams
evaluate potential authentication approaches for specified systems	KB–A, Imp–D	assignments, labs, exams
formulate appropriate threat models for engineered systems	PA–A, Imp–D	assignments, case study, exams
specify security policies using both local and network primitives	PA–A, D–D	assignments, exams
explain how protocols are used to overcome trust limitations	PA–A, Imp–D	assignments, case study, exams
demonstrate practical attacks against processes, hosts, networks, Web applications and their users.	Inv–D, T–D	assignments, labs

[1] Each course outcome is linked to a Graduate Attribute, which is an overarching expectation for engineers who graduate from our program. Each attribute is taught and/or assessed at an Introduced (I), Applied (A) or Developed (D) level.

2. Assessment

Your progression through ECE 7420: Computer Security will be assessed both individually and together with a lab partner. To pass ECE 7420, **you must pass the exam portion of the course**, and **you must complete all labs**. Exams will be closed-book: students may not bring written materials or electronic devices (including calculators or phones) to their seats.

Assignments (5)	15%	Individual work
Labs (6)	18%	Guided exploration of security tools and applications
Midterm test	25%	June 19th, during class time
Final exam	42%	

3. Major Topics

ECE 7420: Computer Security is structured in four modules. Each module builds on concepts from the previous module but takes a wider perspective, from a single program to a single host to networks to Web applications on the global Internet. You may need to **go back and refresh your memory** on prerequisite material.

This course will focus on computer security, but will necessarily include some discussion of cryptographic primitives. We will take a black-box approach to cryptography, exploring how a consumer of cryptographic primitives would see them within concerning ourselves about their implementation details. This level of detail is **insufficient for the safe use of cryptography**: we will chat more about why this is the case during the course.

3.1. Software security

The first module of the course will focus on security within a single virtual address space (a *process*). There will be a focus on the memory safety of programming languages, what it buys you and what it does not.

- review: virtual memory
- review: abstract flat-memory model of computation
- memory safety: program layout, memory safety violations, language-based security
- mitigation technologies: `W^X`, ASLR, ROP, CFI, fuzzing, RNGs…
- languages and type safety: bytecode-interpreted vs compiled languages and their security models

3.2. System security

The second module will consider *host security*: how operating systems provide protection for users and enforcement of systemic policies on individual computers.

- users, processes and authorization
- leaky abstractions
- authentication and other applications of cryptographic hash functions
- disk encryption: symmetric-key cryptography, disk encryption, cryptographic filesystems…
- compartmentalization: capabilities, system calls, app platforms…
- trusted execution: digital signatures, trusted boot…

3.3. Network security

The third module will explore security primitives, attacks and security-sensitive systems in networks of hosts.

- review: IP, UDP, TCP, ports and services
- attacks: DNS, service vulnerabilities, DDoS…

- services: Kerberos, firewalls, IDS, PKI, packet capture…
- protocols: end-to-end design, SSH, TLS, WPA, OTR…

3.4. Web security

The final module will consider pervasive problems in Web security and — to come full circle — how language and framework design can mitigate or protect against them.

- Authentication: passwords, certificates, OAuth and cookies
- Attacks against websites: SQL injection, CSRF, XSS
- Attacks against users: phishing, tracking…
- Censorship resistance

4. Academic Integrity and Professional Conduct

Students are expected to conduct themselves in all aspects of the course at the highest level of academic integrity. Any student found to commit academic misconduct will be dealt with according to the Faculty and University practices. More information is available at http://www.mun.ca/engineering/undergrad/academicintegrity.php. Students are encouraged to consult the Faculty of Engineering and Applied Science Student Code of Conduct and Memorial University's Code of Student Conduct.

Academic integrity means taking full responsibility for the academic work you submit in this course, so that I can evaluate you on the basis of **your own knowledge and effort**. When you submit work, you must **acknowledge sources** of both facts (references) and their presentation (authorship). It is an academic offence to claim work as original when it has been substantially derived from another source without attribution, whether that source is another person or a generative artificial intelligence tool. If GAI is permitted in a deliverable, you must reference any GAI tools you use and provide the sequence of prompts in an appendix.

5. Inclusion and Equity

Students who require accommodations are encouraged to contact the Glenn Roy Blundon Centre. The mission of the Blundon Centre is to provide and co-ordinate programs and services that enable students with disabilities to maximize their educational potential and to increase awareness of inclusive values among all members of the university community.

The university experience is enriched by the diversity of viewpoints, values, and backgrounds that each class participant possesses. In order for this course to encourage as much insightful and comprehensive discussion among class participants as possible, there is an expectation that dialogue will be collegial and respectful across disciplinary, cultural, and personal boundaries.

6. Student Assistance

Student Affairs and Services offers help and support in a variety of areas, both academic and personal. More information can be found at http://www.mun.ca/student.