MEMORIAL UNIVERSITY

Faculty of Engineering and Applied Science

Course Outline
ENGI 9807

Spring 2022–23

Engineering 9807: Computer Security

Instructor	Jonathan Anderson `jonathan.anderson@mun.ca` CSF-4123	**TA(s)**	TBA
Office Hours	Mondays 16:00–17:00		
Website	`https://introsec.ca`		
Communication	In-person or e-mail communications are preferred. Brightspace (`https://online.mun.ca/d2l/home/478849`) will only be used to record grades.		
Calendar entry	**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.		
Schedule	Lecture MTR 13:00–13:50 EN1000 Lab Thu 14:00–17:00 CSF 2112		
Credit value	3 credit-hours	**Lab hours**	Eight 3-hour sessions per semester
Textbook	Van Oorschot, Paul. ***Computer Security and the Internet: Tools and Jewels***, Second Edition. Springer. 2021.		
References	Anderson, R.J. ***Security Engineering: A Guide to Building Dependable Distributed Systems***, 2nd Ed. Wiley. 2008.		

1 Assessment

Assignments (5)	20%	Independent exploration of security topics (greater depth than undergraduate version)
Labs (6–8)	16%	Guided exploration of security tools
Project	14%	Independent exploration or validation of a system, attack or defensive technique
Midterm test	15%	June 18 (tentative)
Final exam	35%	

To pass Engineering 9807, **you must pass the exam portion of the course**. Exams will be closed-book: students may not bring written materials or electronic devices (including calculators or phones) to their seats.

2 Major Topics

Engineering 9807 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.

2.1 Software security

The first module of the course will focus on security within a single virtual address space (***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

2.2 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…

2.3 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…

2.4 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

3 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 at `http://www.mun.ca/engineering/undergrad/academicintegrity.php` and Memorial University's Code of Student Conduct at `http://www.mun.ca/student/conduct`.

4 Inclusion and Equity

Students who require accommodations are encouraged to contact the Glenn Roy Blundon Centre, `http://www.mun.ca/blundon/about/index.php`. 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.

5 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`.