Northeastern University - Seattle
CS 5770 Software Vulnerabilities and Security
Spring 2017

Time: Thursdays from 6:00-9:00pm at **the Cornish College, LUI 202 classroom**
(Lab part of the class from 8:00-9:00pm)

Instructor: Tamara Bonaci (t.bonaci@neu)
Office hours: By appointment

TA: Rahul Mondal
Office hours: TBD

Course website: https://piazza.com/northeastern/spring2017/cs5770/home
Course assignments and dropbox: TBD
Course discussion board: https://piazza.com/northeastern/spring2017/cs5770

Course Overview:

Cyber and cyber-physical technologies have an ever-increasing role in our lives. As they are becoming more complex and ubiquitous, the need to ensure their safety, security and reliability is increasing as well. Security is a discipline dedicated to protecting cyber and cyber-physical systems, as well as their users from adversarial actions. This course is a foundational graduate security course, providing an introduction to tools, concepts and ideas of modern security research.

In this course, we will start by defining security goals and objectives. We will then briefly cover some of the most important cryptographic primitives. Throughout the remainder of the semester, we will talk about common software programming, configuration and design mistakes, and how to avoid them. We will focus on several central security themes, including: system, software security, network, and web security.

The goals of this course are to:

- Examine major vulnerability classes that can be introduced in various software domains and levels of the software stack
- Understand effective techniques for defending against exploitation in situ
- Understand approaches for detecting the presence of vulnerabilities during development and deployment
- Gain hands-on experience in attacking and defending vulnerable software

Course Progression:

The following is a preliminary class progression covering the 15 weeks of the course (January 12- April 20). It is subject to changes.

Week 1: Course overview. Introduction to security, its goals and objectives.

Week 2: Cryptography – Introduction and symmetric encryption.

Week 3: Cryptography – Block ciphers. Public key cryptosystems.

Week 4: Cryptography – Hash functions and message authentication codes.

Week 5: Cryptography. Digital signatures. Key management.

Week 6: System security – Users and privileges.

Week 7: System security – Shells, races and sandboxing. **Midterm.**

Week 8: Software security – Introduction and memory corruption.

Week 9: Software security – Memory corruption.

Week 10: Software security – Vulnerability and malware analysis.

Week 11: Network security – Link, network and transport layer attacks.

Week 12: Web security – SSL/TLS. Basic Web security model.

Week 13: Web security – Web application security (XSS, SQL injection, session management).

Week 14: Privacy and anonymity.

Week 15: Final project presentations.

Prerequisites:

In this course, you will be expected to have maturity in mathematics of computer science and in computer engineering. This means that you should:

- Have a solid understanding of data structures and algorithms,
- Be comfortable writing programs from scratch, and
- Be comfortable in a command-line Unix development environment.

You should also have a good understanding of computer architecture, operating systems, and computer networks. Most importantly, you should be eager to challenge yourself and learn more!

About the Course:

The course will consist of ***in-class activates/quizzes, homework assignments, lab assignments, and a project.***

In-class activities/quizzes: In-class activities are just that – activities done in class. Occasionally, worksheets will be handed out at the beginning of the class, and will be used to review class material and facilitate discussion. Please be sure to write your name and the date of each activity when you turn it in, since in-class activities will be graded on a scale 0-2, where:

- 0 means missed or irrelevant in-class activity,
- 1 means relevant answers submitted, and
- 2 means good and interesting answers submitted, and/or interesting discussion in class.

There may theoretically be an in-class activity every week, but we will take **seven best scores** when determining your grade.

Homework assignments: There will be **five** homework assignments in this course, and those will be a mix of written questions and simulation problems.

While simulation parts of the homework may be designed with a specific programming language/tool in mind, you are welcome to code them up using any software tool you prefer. **You should, however, submit all your code and simulation models with your homework.**

Lab: Labs are an important part of this course, as they are expected to give you a more practical, hands-on experience with some important security concepts. There will be up to ***five lab assignments*** through the quarter, and you will have at least two lab sessions to work on those assignments. You are encouraged to work in groups of two persons, but if you prefer, you can work on those assignments individually.

Each lab will be graded based upon deliverables, which will be defined in each individual lab assignment.

Project: The final component of this course is a project, and its goal is to give you a deeper understanding of how to think about, and how to solve a real-life problem from a security perspective.

For the project, you will be able to choose a topic related to any area of security and privacy (including those not directly covered in this course). You can work on the project either individually, or in groups of up to three persons. When working in a group, your end result should reflect the fact that it is a multi-person effort.

Your work on the project will consists of several milestones:

- Project proposal,
- Progress report,
- Final report, and
- Project presentation.

Grading:

Your grade in this course will be based on **midterm, in-class activates, homework assignments, labs, and a project**. The expected grade breakdown is:

- Midterm – 15%
- In-class activities – 10%
- Homework – 25%
- Lab: 20%
- Project – 30%

Course Material:

There is no official textbook for this course. Instead, we will rely on lectures and readings. Additionally, you might find these books useful.

- N. Daswani, C. Kern, and A. Kesavan, *Foundations of Security, What Every Programmer Needs to Know,* Apress, 2007
- C. Kaufman, R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World, Prentice Hall, 2002*
- D. Stinson*, Cryptography Theory and Practice, Third Edition,* CRC Press, 2006
- W. Stallings, *Cryptography and Network Security, Principles and Practice, 5th Edition,* Prentice Hall, 2006
- B. Schneier, *Applied Cryptography, Protocols, Algorithms and Source Code in C,* Wiley, 1996
- Menezes, P. Van Oorschot, S. Vanstone, *Handbook of Applied Cryptography* (available online)

Course Policies:

Collaboration: In this course, we want you to learn from each other. Therefore, you are allowed (and encouraged) to talk to your classmates and other students about all course assignments. You may also consult outside reference materials, or the instructor. However, all material that you decide to turn in should reflect your own understanding of the subject matter at the time of writing. If you work with someone else on any assignment, please include their names on the material that you turn in.

Assignment Turn-in: All assignments (including homework assignments, lab

deliverables and project) should be submitted in a PDF form to the course dropbox. Please, ***do not use*** email for assignment submissions.

Late Assignment Turn-in: All assignments are due **by 11:59pm on their assigned date,** but we understand that you may have to sometimes turn them in late. The grading penalty is 20% of the grade that you would otherwise receive for each day, or part of the day, that you are late. No submissions will be accepted after 5 days.

Ethics: This course will cover some sensitive material that includes information on how to exploit vulnerable software. Attack-oriented work must be restricted to the computing resources provided. Alternatively, students can perform this work using personal resources as long as other computing resources are not affected. In particular, attacks performed against University resources, or the open Internet are expressly prohibited. Students should also be familiar with the University Appropriate Use policy.