CMSC 414 — Computer and Network Security
Sections 0101 & 0301

Dr. Michael Marsh

Fall 2017

Course Summary

An introduction to the topic of security in the context of computer systems and networks. Identify, analyze, and solve network-related security problems in computer systems. Fundamentals of number theory, authentication, and encryption technologies, as well as the practical problems that have to be solved in order to make those technologies workable in a networked environment, particularly in the wide-area Internet environment.

1 Policy and Academic Honesty

1. During this course, you will learn how to compromise systems. It is essential that you **use this knowledge responsibly and ethically**. Obey all laws and University policies, as well as common courtesy, when interacting with any system (computer, network, data, etc.) for which you are not the exclusive owner and user, or for which you do not have explicit permission to perform an analysis or attack. As a simple rule, if it isn't yours, and on your machine, you should not be attempting to break it. If you break any laws or invade anyone's privacy using knowledge gained in this course, do not expect the University or the course staff to provide you with any defense.
2. *Turn in all work for the course.* If you do not at least make a good-faith effort on all graded assignments, **we reserve the right to fail you for the course**. The assignments are designed to help you master the material, and are an essential part of the course.
3. *Turn assigments in **on time**.* Unless we have agreed on an extension in advance, you will *not* receive credit for work that is turned in after the *day* and *time* it is due. The only exception is for excused absences as defined by the university (Section V-1.00(G) of the Consolidated USMH & UMCP Policies and Procedures Manual.

   > Any student who needs to be excused for an absence from a single lecture, recitation, or lab due to a medically necessitated absence shall:
   >
   > (a) Make a reasonable attempt to inform the instructor of his/her illness prior to the class.
   >
   > (b) Upon returning to the class, present their instructor with a self-signed note attesting to the date of their illness. Each note must contain an acknowledgment by the student that the information provided is true and correct. Providing false information to University officials is prohibited under Part 9(i) of the Code of Student Conduct (V-1.00(B) University of Maryland Code of Student Conduct) and may result in disciplinary action.
4. *Do not miss exams.* As with assignments, you will not receive credit for missed exams unless previously arranged or covered by the university policy cited above.

5. *The punt box rule:* On any problem you turn in (homework or exam), if you clearly mark a rectangular box with an "X", we will not grade the problem and you will receive 1/10 of the points for the problem. All punt points aer added, and rounded up. This is to stop you from guessing on exams.

6. *Please read and understand the UMCP code on academic integrity* (Section III-1.00(A) of the Consolidated USMH & UMCP Policies and Procedures Manual). *Do not violate it.* The whole point of taking a class is to learn the material; violating the academic integrity code means you are robbing *yourself* of this opportunity, wasting both your time and money. You are also wasting the time of the instructors.

   Working together on projects is encouraged. Working together on homework is acceptable, within limits. Namely, the work you turn in *must be your own*. How to address a problem is fair game for discussion with your classmates, the actual code is not. No collaboration or communication is permitted for exams.

   > *An example for clarification:* Suppose Alice and Bob are working on a programming project/homework. It is fine for Alice and Bob to discuss their proposed solutions, work on a whiteboard together, and even ask questions on the forum. Once they figure something out, they can also answer specific questions on the forum. However, they should not post complete solutions (or code snippets), unless specified by the instructor or the TAs.
   >
   > After discussing their solutions, Alice and Bob go off and write up their work/code up their project. This level of cooperation is allowed and encouraged.
   >
   > However, if Alice or Bob had simply copied code or text from each other, their effort would be deemed dishonest. They should not use "old versions" of the other's code, or steal throwaway code from a temporary directory or a dustbin, or "look at the other's screens" while typing in their solution.

   All University course-related policies can be found at `http://www.ugst.umd.edu/courserelatedpolicies.html`. In particular, *any student eligible for and requesting reasonable academic accommodations due to a disability is requested to provide, to the instructor in office hours, a letter of accommodation from the Office of Disability Support Services (DSS) within the first TWO weeks of the semester.*

2 Logistics

This syllabus covers two of the three sections for Fall 2017. These sections will cover the same material and will have the same assignments. There will be in-class graded work, and the rooms are at or very close to capacity, so please attend the section for which you are registered. It is possible that we will be able to accommodate some (rare) exceptions to this, but do not expect that to be the case.

You should bring a laptop and a TurningPoint-enabled device to every class session. The laptop should be configured with python and bash, though if you have the course VM installed, that provides an acceptable programming environment. The TurningPoint-enabled device may be a clicker or any device with the mobile version of TurningPoint installed. The software is site-licensed by the university, and is freely installable for all students. If you already have a physical clicker, we will only be using the multiple-choice functionality.

2.1 Class Times

Section 0101 MW 3:30–4:45
Location: Edward St. John 0215

Section 0301 MW 2:00–3:15
Location: Edward St. John 2212

2.2 Grading

Component	Percentage of Final Grade
In-class assignments	5
Preliminary at-home assignments	5
Reinforcing at-home assignments/projects	40
In-class exams (2)	30
Final exam (cumulative)	20

3 Course Materials

There is no assigned text for this course. Each assigned reading will be posted to the course webpage, and will be freely available online. Many of these will come from Ross Anderson's *Security Engineering*, second edition. This is also available for purchase, if you like to have hardcopies of textbooks. Other readings will predominantly be classic papers in security.

You might find *RTFM: Red Team Field Manual* useful and interesting. It retails for $9. The ISBN is 978-1494295509, and it is available from popular online booksellers. Dr. Marsh has a copy that you are welcome to peruse before deciding whether it is a worthwhile purchase.

A number of blogs and news sites also frequently have excellent articles related to security:

Cryptographers

- Bruce Schneier: `https://www.schneier.com/`
- Matt Blaze: `http://www.crypto.com/blog`

Penetration Testers

- Nikhil "SamratAshok" Mittal: `http://www.labofapenetrationtester.com/`
- Egor Homakov: `https://sakurity.com/blog/`

News Sites

- LWN's Security Page: `https://lwn.net/Security/`
- The Register's Security Page: `http://www.theregister.co.uk/security/`

4 Assignments and Projects

4.1 In-Class Assignments

This course will emphasize active-learning techniques. Part of this will be Clicker multiple-choice questions. These questions will contribute a small portion of your final grade in the course (see 2.2). Your in-class grade will be based on the questions from all class sessions for which you *do not* have an *excused absence*. In addition, for the questions on which you are graded, the lowest 20% of responses will be dropped. **You must have a Clicker, either a physical device or the app on your phone/tablet/laptop.** The app is now site-licensed to the University, and you can find it at `https://umd.service-now.com/itsc?id=service&service=3c08fdd7370a8600fd771f9543990eea&t=so`.

Due to the in-class assignments, it is essential that you come to class *every session*. If you cannot attend, you *must* get an *excused absence* and have it on record in ELMS.

4.2 At-Home Assigments

According to University guidelines, you should expect to spend roughly 5 hours per week outside of lectures on course work for a class with 2.5 hours per week of lecture time. This will be split into roughly the following categories:

Background Reading Lectures will assume that you have read the background material in advance, so that we can get into details and have an interactive discussion. These readings will be posted on the course webpage well in advance.

Preliminary Assignments In order to ensure that you have read the material and help you solidify some of the major points, there will be short preliminary assignments before the material is covered in lecture. These will be a relatively small fraction of your final grade.

Reinforcing Assignments After lecture, more in-depth assignments will be posted to help you master the material. These will predominantly be in the earlier part of the semester, before the projects begin, but smaller assignments will also be given while you are working on the projects.

Projects The final category of graded assignments is the projects. There will be two of these, which are closely related. The first is a "Build It" project, in which you will design and develop a secure system. The second is a "Break It" project, in which you will evaluate and attempt to find vulnerabilities in a classmate's "Built It" submission.

Exam Preparation There will be two in-class exams and a cumulative final. As these exams approach, the other at-home assignments will be reduced so that you have adequate time to prepare. Ideally, reviewing the assignments should suffice as preparation.

5 Technical Background

This course will focus on **C** and **Unix**, the former because it is still the language in which much of the world's computing infrastructure is built and the latter because it provides an excellent environment in which to work with C programs at our level of detail. We will also make considerable use of the **gdb** debugger. These should already be familiar to you, though you may wish to refresh your memory.

Some in-class exercises will be in **python** or **bash**, so you should make sure that you are comfortable programming in these. In-class programming assignments will typically be group efforts and will not be graded, but will be an important part of mastering the material.

Programming assignments will be done in a virtual machine, so that everyone has the same environment. Programs will be tested in this same VM, so it is very much to your advantage to ensure that your assignments and projects work properly in this environment. No credit will be given for "works on my machine" if it does not work in the VM. **VirtualBox** is the recommended hypervisor for the VM.

Some of the programming assignments will require you to use the **git** source code revision control system. This is installed on the VM, and is available for any operating system you are likely to be using. We will host remote repositories for you, which in some instances will be how you submit your work.

6 Topics

The following topics are subject to change, but will be close to the final list covered:

1. Security background and definitions
2. Software security (attacks and defenses)
   (a) Buffer overflows
   (b) Viruses
   (c) SQL injection
   (d) Web-based attacks
3. Cryptographic primitives
4. Applications of cryptography
   (a) Digital currency
   (b) Anonymous communication
5. Network security
   (a) Networking background
   (b) Attacks across all layers
   (c) Network control (firewalls, VPNs)
   (d) Censorship and censorship resistance
6. Economic incentives and underground economies