Preliminary Syllabus for I533/B649: The Design and Analysis of Secure Protocols and Systems & Information Assurance Spring 2015

Instructor: Steven Myers
email: samyers@indiana.edu

November 9, 2015

Office Hours: Lindley Hall 301e, by appointment.

Motivation: With the increasing number of people, businesses, governments and organizations using the Internet, the Cloud, the IoT and information technology for a growing number of their operations, the need to ensure that these systems are secure is becoming more and more essential. While understanding the specific security aspects of different technologies such as firewalls, operating systems, cryptography, secure development, etc. are essential to designing secure systems, it is not sufficient. In particular, a systems designer must be able to integrate these technologies in appropriate ways to ensure the resulting system is secure. Many people improperly believe that if they include enough security technology the resulting system will magically be secure, without giving thought to what problems these technologies both solve and create.

In this course, we will cover the design and analysis of secure systems including identifying security goals and risks, risk management, risk analysis and treat modeling, defense in depth, integrating different technologies to achieve security goals, developing security protocols & policies, implementing security protocols and secure coding. In short, how does one take all of the security technologies, secure development practices and protocols and integrate them into a functioning whole? To do this, we also need to understand how most of the modern attacks function, including writing zero-day attacks. For it is difficult to understand how to protect a system, if you don't understand the major attacks that are currently being deployed.

In this course, we will consider some real world scenarios that have many security requirements. We will require the use of many security technologies and protocols, determine the requirements necessary to solve the security requirements, and then consider how to design systems and protocols to achieve these requirements. In particular, we will start with defining some vague security requirements for a system to be implemented, and then consider the development process from the highest levels of planning hardware and software requirements, staff security policy, etc.. to the lowest level of writing secure code, and then finally the testing process.

Prerequisites: A reasonable programming background is necessary. A course in operating systems, networking and computer architecture are helpful but not necessary. You are not required to know any particular language, but rather *it is assumed you can pick up new languages if needed for the course.* In particular, you will be programming in C, assembly language, shell scripts and potentially other languages, as well as using related tools during this course. Basic knowledge on cryptographic primitives such as what symmetric-key and public-key encryption, digital signatures, and cryptographic hashes is also assumed.

Textbooks: The course textbooks are:

- *Threat Modeling: Designing for Security*, but Adam Shostack, Wiley 2014.
- *Computer Security, Principles and Practice Third Edition*, by William Stallings and Lawrie Brown, Prentice Hall, 2014.

Even though the above are the course text and readings will be assigned from them, they are far from perfect resources. Therefore, there will be readings and other resources given throughout the term from other sources. Further, since we can only delve in to the many different security topics at a fairly low depth, the following texts go in to some of the topics of this course in significantly more detail. Thus, students interested in further reading may find the following texts interesting.

- Writing Secure Code, 2nd Edition, by Michael Howard and David LeBlanc, Microsoft Press.
- Computer Security: Art & Science, by Matt Bishop. Addison Wesley
- Introduction to Computer Security, by Math Bishop, Addison Wesley
- Security Engineering (Second Edition), by Ross Anderson, Wiley. First Edition is available here: `http://www.cl.cam.ac.uk/~rja14/book.html`
- Secure Programming with Static Analysis, Brian Chess and Jacob West, Addison Wesley.
- Metasploit: The Penetration Tester's Guide, by David Kennedy, Jim O'Gorman, Devon Kearns, and Mati Aharoni, No Starch Press.
- The Basics of Hacking and Penetration Testing, Second Edition: Ethical Hacking and Penetration Testing Made Easy, Patrick Engebretson, No Starch Press.
- The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics, John Sammons, No Starch Press.

High-level Syllabus:

> No battle plan ever survives contact with the enemy.
> –Helmuth von Moltke

Below is a rough syllabus of the topics I intend to cover. However, this is only a guideline, and we will most likely deviate from it. I will make updates to this section frequently on Canvas, so please check there for a more specific timeline and future readings.

Topic #	Description
1	Security & privacy Goals, principles & excuses costs of security & insecurity. Risk management
2	Risk analysis: threat & adversary modeling (high-level)
3	Review of cryptographic primitives
4	Passwords & authentication Mechanisms
5	Access control in operating and data management systems
6	OS memory protection, stack call frames, link library tables
7	Buffer overflows, shell code, and related attacks
8	Penetration testing
9	Logging and auditing
10	Secure coding practices
11	Software Security, OS hardening, patching, CVEs and other strategies.
12	Forensics
13	Privacy technologies
14	True randomness for cryptography

Individual Assignments: There will be several individual assignments (3-5) that allow students to show mastery of technical parts of the course. These are independent projects, and you will not be able to complete them in lab hours.

Lab Assignments: Most weeks your lab will have an assignment where students will need to demonstrate their ability to perform certain tasks in the lab, or commenced in lab and finished on the students own time. Unless specifically stated otherwise, each individual assignment will have the same weighting. Lab assignments are due at the next regularly scheduled lab session, unless otherwise stated.

Readings: Every week there will be readings assigned, based on our expected progress. These readings will be announced in class, and on Canvas. **The expectation is that students will arrive to class having done the assigned readings, and students will be called upon in class to lead the discussion of reading**. There may also be quizzes given to asses students on readings. These quizzes will be given with no prior warning (other than this one).

Just-in-time (JIT) Quizzes: The instructor may use just-in-time quizzes to ensure that students are reading material before class, and following material discussed in class. In this case, you will be notified of an online quiz to be performed through Canvas before the next class. All JIT quizzes will have the same value.

Final Exam: There will be a final exam for this course. It will be at the registrar scheduled time for this course. **While the final exam is only worth 20% of your grade, you must achieve a grade of 45% on it to pass the course.**

Grading:

- Final Exam 30%
- Mid Term Exam 15%
- Weekly Labs 20%
- Assignments 30%
- Reading, Just-in-time quizzes, and class participation 5%

Emailing The Instructor If you email the instructor, please ensure that 'I533' or 'B649' appears in the topic. If it does not, there is no guarantee that your email gets past my spam filter, gets read in a timely manner, or gets read at all.

Canvas The instructor and the AIs will be using Canvas to post alternate readings, a listing of the next few weeks topics and readings, grades, and other course management topics. Please make sure you are set up to receive emails from Canvas.