Master Course Description for EE-468

Title: Computer, Network, and Embedded Security
Credits: 4

EE 468: Computer, Network, and Embedded Security Fundamental principles of software and embedded system security and their application to network, web, and embedded systems. Introduction to the practical tools used for software security, cryptography, and protocols that enable its application to network and system security. Prerequisite: EE 205 or EE 215; CSE 373; CSE 374.

Coordinator: Radha Poovendran, Professor, Electrical and Computer Engineering

Goals: To develop an understanding of the fundamental principles of software and embedded system security and their application to network, web, and embedded systems. This course will serve as an introduction to the practical tools in software security, cryptography, and protocols that enable its application to network and system security. It will cover topics including penetration testing, buffer overflows, memory safety, SQL injection attacks, encryption (symmetric, public-key), digital signatures, secure authentication, key management, cryptographic hashing, and web security. In the embedded systems side, it will cover physically unclonable Functions (PUFs), hardware Trojans, and Controller Area Network (CAN) protocols for automotive applications. We will study, analyze and implement *attacks* on earlier buggy versions of Transmission Control Protocol (TCP), Domain Name Server (DNS), and Dynamic Host Configuration Protocol (DHCP) standards. The course also covers the IEEE/ACM Code of Ethics in the context of security.

Learning Objectives: At the end of this course, students will be able to:

1. *Describe* the basic cryptographic primitives, authentication protocols and why they work, and what are the common design errors.
2. *Design and analyze* existing and emerging CAN protocol standards (or other languages such as, C, C++, Python, MATLAB).
3. *Describe and analyze* authentication protocols for two party communications.

4. *Design and analyze* algorithms for PUFs for security and efficiency.
5. *Design and analyze* spam filtering and the false alarm rate reduction.
6. *Describe* the ethical issues related to the misuse of computer security in the framework or IEEE/ACM Code of Ethics.
7. *Describe and analyze* buffer overflow attacks.
8. *Describe attacks* on Transmission Control Protocol (TCP), Domain Name Server (DNS), Dynamic Host Configuration Protocol (DHCP) Standards.
9. *Describe* cross scripting attacks on the websites.
10. *Describe and analyze* the vulnerabilities in the older SSL and TSL Standards
11. *Understand and explain* the X509 Certificate standard format and its applications
12. *Describe* TOR standard and its applications in the context of anonymous communication networks
13. *Design and implement* software programs within the context of security applications.

Textbook: None. Instructor's notes will be provided.

Reference Texts:

1. Ross Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, Wiley, 2020.
2. C. Kern, A. Kesavan, N. Daswani, *Foundations of Security: What Every Programmer Needs to Know*, Dreamtech Press, 2007.
3. C. Kaufman, R. Perlman, M. Speciner, *Network Security: Private Communication in a Public World,* Prentice Hall, 2002.
3. D. Stinson, *Cryptography Theory and Practice*, 3rd edition, Chapman & Hall/CRC, 2006.
4. W. Stallings, *Cryptography and Network Security,* 4th edition, Prentice Hall, 2005.

Prerequisites by Topic:

1. Mature knowledge of computer programming, and the C language (CSE 374).
2. Data structures and algorithms (CSE 373).

3. Basic electronic circuits (EE 205 or EE 215)

Topics:

1. Introduction to building secure systems.
2. Buffer overflows, memory safety attacks and defenses.
3. Software security and testing.
4. Web Security and cross scripting attacks.
5. Network background and attacks on TCP, DNS, DHCP.
6. Intro to cryptography and cryptanalysis.
7. Hash functions.
8. Public key encryptions.
9. Digital signatures (RSA, El-Gamal, DSA).
10. Physically Unclonable Functions (PUFs).
11. Hardware Trojans and Detection Algorithms.
12. CAN Protocols and their vulnerabilities.
13. IEEE code of ethics.

Course Structure: The class meets for four lectures a week, each consisting of 50 minutes. There will be weekly or bi-weekly programming assignments. There will be two team projects with two to three members in each team. Course includes one midterm and one final exam, as well as short quizzes as needed.

Computer Resources: The course assignments and course projects will require students to be familiar with one or more of the following languages: C, Java, Python. Students are expected to use their personal laptops, but they may use the ECE department computers as needed.

Grading: 20% Homework, 40% projects, 10% midterm, 20% final exam, 5% in-class activity, 5% for filling the end of the course evaluation forms.

ABET Student Outcome Coverage: This course addresses the following outcomes: H = high relevance, M = medium relevance, L = low relevance to course.

(1) *An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.* **(H)** The course uses mathematical tools. Students must identify, design, and implement suitable software algorithms. Students must be able to identify vulnerabilities of a given protocol against specific attacks and develop mitigating defenses against the attacks. Engineering judgment is developed through the understanding the limitations and advantages of a given software code, cryptographic algorithm or security protocol in the context of PUFs, CAN and hardware Trojans. Throughout the course we emphasize the need to use sound design principles instead of relying on mathematics only. Towards this direction, software codes and security protocols that were mathematically correct (using formal analysis) but had design flaws are discussed. Assignments require students to analyze related software codes and security protocols with weaknesses. The homework involves solving software engineering problems identified by the assignments and exemplified by class discussion. The homework and projects challenge the students to identify the issues and formulate, and implement their individual solutions and show that they work.

(2) *An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors.* **(M)** The project challenges the students to develop, design and implement different cryptographic algorithms.

(3) *An ability to communicate effectively with a range of audiences.* **(M)** Students are required to write up their simulations in an engineering format. The ability to communicate effectively in writing is a portion of the grade received on homework and projects. Students are required to give a short presentation on a selected security topic to the class (depending on the instructor).

(4) *An ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts.* **(H)** The course covers security vulnerabilities in software, websites, and embedded hardware systems and their societal implications, enabling the students to recognize the ethical dilemmas that they may face in their professions. Impact of good

software and network security practices is emphasized. We discuss the impact of design and implementation of insecure protocols and the way they can be exploited in the context of software, websites, and embedded systems. Main focus here will be to show how to design protocols that are resilient to common security attacks such as buffer overflow, spoofing on CAN, replay attacks on web authentication protocols.

(5) *An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives.* **(H)** The two course projects are conducted in teams of up to 3 members and constitute a significant portion of their grade.

(7) *An ability to acquire and apply new knowledge as needed, using appropriate learning strategies.* **(H)** The course emphasizes the need for evolving current secure system designs as new threats emerge and security assumptions are weakened. Further, pointers to security websites and articles are provided in order to enhance personal knowledge in this developing area.

Prepared By: Radha Poovendran, Scott Hauck
Last revised: May 10th 2020