ECE 8813: Advanced Computer Security

General Information:

Digital technologies impact every aspect of our daily lives. As such, good cybersecurity practices are now essential for protecting not only our national's critical infrastructure, but also for ensuring the privacy of our data, the secrecy of our communications, and the everyday functioning of global economies. The goal of the course is to help spur engaging discussions on selected research topics in computer security/privacy that propose solutions for building resilient and secure digital ecosystems. Unlike other areas of computer science (e.g., software development), where practitioners can leverage simplifying assumptions to quickly complete a task derived from an external need, cybersecurity practitioners need to be aware of, and repeatedly question, the validity of any simplifying assumptions to either prove a system's security or find exploitable weaknesses. As such, in this specific field, being able to ask the right questions and finding the right problems to solve is perhaps just as important as being able to solve them.

This course is research-oriented and structured around research papers that students must read before each session. Students will also work on a course project that involves validating or extending ideas covered in one or more topics discussed in class. Potential projects will be suggested, but students are encouraged to work on topics that they are passionate about. Students will be required to read the papers assigned during the semester and be able to competently discuss the material in class. Each student will be responsible for jointly leading (with the instructor) a discussion based on the assigned paper(s) for the week. The instructor will provide a comprehensive overview of the topic. Each student will be responsible for submitting a constructive written critique of the main paper(s) assigned each week. The critique will follow a standard format that critically evaluates the paper(s) (e.g., questioning the assumptions, questioning whether the experiments met the stated objectives, discussing flaws or omissions in the methodology and/or findings, suggesting areas for improvement, etc). The overall goal is to assess the student's understanding and critical thinking.

Logistics:

Class location: Van Leer | Room C240
Class date and time: T/TH 12:30 - 1:45 pm

Instructors Office Location: Klaus 3360
Email:<fabian> at ece.gatech.edu, subject line preamble: [ECE8813:]
Presentation review slots: M/W 2-4 pm and by appointment on Fridays

Prerequisites

This is a graduate course, therefore, students are expected to have undergraduate-level background on computer architecture, operating systems, and networking. ECE 4115 or equivalent is also strongly advised as a prerequisite. In addition, familiarity with low-level systems

programming (e.g., C and assembly) will be necessary for understanding the details of some of the assigned readings, and will be helpful in completing in-class exercises.

Materials:

This is a research intensive course intended for PhD students. There is no required textbook for this course. Instead, we will study research papers from various computer security venues. Topics to be covered include (but are not limited to), software security attacks (e.g, memory disclosure exploits) and defenses, computer forensics techniques (e.g, recovery of forensic information from memory snapshots), securing the software supply chains, directed vulnerability discovery (e.g, fault isolation and fuzzing), software and hardware-based side channel attacks, protocol reverse engineering, malware classification and provenance, transparent and privacy-preserving systems.

Learning Objectives

Given that the course assignments and project will focus on *applied* systems topics, students will be assessed in several ways (e.g., whether they can apply learned knowledge in a previously demonstrated way, can apply learned knowledge in contexts not seen before, can independently close any knowledge gaps when completing a task, or can demonstrate mastery by being able to teach others). Upon successful completion of the course, students will:

- Be able to understand fundamental notions within certain subfields of computer security, and engage in critical discussions thereof.
- Be able to study cutting-edge research papers and provide constructive feedback based on the scientific merit, novelty, and thoroughness of the work.
- Be able to identify threats, vulnerabilities, attacks (and associated countermeasures) to critical cyber infrastructure.
- Have a better appreciation for concerns about security and privacy, and be better able to analyze proposals to protect user privacy.
- Be able to describe technical computer security concepts to their peers.
- Be able to propose new ideas to relevant research problems and to prototype their solutions.

Assignments & Grading

There will be mini-projects (with 1 week deadlines each) related to topics in the course. These are programming assignments involving network traffic and binary analysis (x86). Students are expected to be comfortable with rapid prototyping, and be comfortable working on semi-structured programming problems. In addition, students are expected to work in small teams on a half-semester long research project approved by the instructor. While some project ideas will be suggested, students are encouraged to pursue projects that align with their own graduate research — as long as that research has applicable security or privacy components.

Breakdown:

20% for programming assignments and in-class activities.
35% for paper presentation and weekly paper reviews.
45% for research project (including intermediate and final write-ups and the in-class project presentation).

Project Requirements:

- The project must contain **original** research. In the case of replicability studies (i.e., validating an approach given in prior work), the research contribution should cover (at minimum), a comparative empirical analysis as well as an and in-depth assessment of the appropriateness of the original conclusions as well as actionable insights. I also expect an evaluation on a new dataset not covered in the original paper. For new ideas, the report must be workshop-ready quality, including a clear description of the motivation, approach, preliminary results, summary of findings and limitations.
- All project reports must be typeset in LaTeX and the authors must provide a link to a private GitHub repository containing instructions on how to replicate their findings.
- Project proposals will be due roughly 6 weeks in to the semester. Projects must be approved by the Professor. You will be required to work in small groups of two. A status update will be required two weeks before the final writeup is due.
- Each team must prepare a final presentation and report on the outcomes of their research project during the final instruction days of class. The in-class presentations will be peer-reviewed (by other students in the class).

Paper Review Format and Requirements for Presenters

A description covering guidance when conducting your reviews on papers and on presenting your lessons on can be found here.

Honor Code

Students are expected to abide by the Georgia Tech Academic Honor Code. Honest and ethical behavior is expected at all times. All incidents of suspected dishonesty will be reported to and handled by the Office of Student Integrity. You will have to do all assignments individually unless explicitly told otherwise. You may discuss with classmates but you may not copy any solution (or any part of a solution).

Learning Accommodations

Whenever needed, the instructor will make accommodations for students with documented disabilities. These accommodations must be arranged in advance and in accordance with the Office of Disability Services.

Academic Honor Code

Students are expected to abide with Georgia Tech's academic honor code.

Questionnaire

Please fill out the Background and Paper Selection Questionnaire.

List of Papers (subject to change)

The list of papers we will cover is given below. The order of presentation and assigned co-leads will be set after the first week of class.

Date	Topic	Main Paper	Additional Readings	Assigned Co-leads
8/24	**Traffic Analysis**	***Active Exercise 1 Assigned***		
8/29	**Software Security**	Silent Bugs Matter: A Case Study of Compiler-Induced Security Bugs, Xu et al., 2023	The Correctness-Security Gap in Compiler Optimizations, D'Silva et al., 2015	Yufei Du,?
8/31		*Active Exercise 1 in-class discussion and enhancements*		
9/5	**Fuzzing**	datAFLow: Toward a data-flow-guided fuzzer, Herrera et al, 2022	Evaluating Fuzz Testing, Klee et al, 2018.	Kevin Valakuzhy,?
	Network Security / Measurement	Network Detection of Interactive SSH Imposters using Deep Learning, Piet el al., 2023	Timing Analysis of Keystrokes and Timing Attacks on SSH, Song et al, 2001.	

Date	Topic	Main Paper	Additional Readings	Assigned Co-leads
	Side Channels	Hertzbleed: Turning Power Side-Channel Attacks into Remote Timing Attacks on x86, Wang et al, 2022.	Power Side-Channel Attack Analysis: A review of 20 years of study for the layman, Randolph and Diehl, 2020	
	Applications of Machine Learning	How Machine Learning is Solving the Binary Function Similarity Problem, Marcelli et al, 2022.	Dos and Don'ts of Machine Learning in Computer Security, Arp et al, 2022.	
	Detecting Audio Deep Fakes	Who Are You (I really wanna know)? Detecting Audio Deepfakes through vocal tract reconstruction, Blue et al, 2022.	Breaking Security-Critical Voice Authentication, Kassis et al, 2023	
	Adversarial ML	The Space of Adversarial Strategies, Sheatsley et al, 2022.	Malware Makeover: Breaking ML-based Static Analysis by Modifying Executable Bytes, Lucas et al, 2021	
	Privacy	Arana: Discovering and Characterizing Password Guessing Attacks in Practice, Islam et al, 2023.	The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords, Bonneau, 2012.	
	Software Security	The Sound of Silence: Mining Security Vulnerabilities from Secret Integration Channels in Open-Source Projects, Ramsauer et al, 2020	TRACER: Finding Patches for Open Source Software Vulnerabilities, Xu et al, 2021	

Date	Topic	Main Paper	Additional Readings	Assigned Co-leads
	Privacy	Analyzing Leakage of Personally Identifiable Information in Language Models, Lukas et al, 2023.	Membership inference attacks from first principles, Carlini et al, 2022.	
	Network Security	Helping Hands: Measuring the impact of a large threat intelligence sharing community, Bouwman, 2022.	A Different Cup of TI? The Added Value of Commercial Threat Intelligence, Bouwman et al., 2020	
	Privacy	Asleep at the Keyboard? Accessing the Security of GitHub's Copilot's Code Contributions, Pearce et al, 2022	CodexLeaks: Privacy Leaks from Code Generation Language Models in Github Copilot, Niu et al, 2023.	
	Forensics	SymLM: Predicting Function Names in Stripped Binaries via Context-Sensitive Execution-Aware Code Embeddings, Jin et al, 2022	SoK: All you ever wanted to know about x86/x64 binary disassembly but were afraid to ask, Pang et al, 2021.	
	Software Security	FloatZone: Accelerating Memory Error Detection using Floating Point Unit, Gorter et al, 2023.	Lightweight Bounds Checking, Hasabnis et al., 2012	
	Privacy	Timeless Timing Attacks and Preload Defenses in Tor's DNS Cache, Dahlberg et al, 2023.	Timeless Timing Attacks: Exploiting Concurrency to Leak Secrets over Remote Connections, Van Goethem et al, 2020	

Date	Topic	Main Paper	Additional Readings	Assigned Co-leads
	Authentication	Might I Get Pwned: A Second Generation Compromised Credential Checking Service, Islam et al, 2022	Detecting Stuffing of a User's Credential a Her Own Accounts, Wang and Reiter, 2020.	
	Hardware Security / Moral Dilemmas Casestudy	Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses, Halperin et al, 2008	Cybersecurity Concerns and Medical Devices: Lessons from a pacemaker advisory, Kramer and Fu, 2017.	
	Network Security	The Journey to the Center of the Cookie Ecosystem: Unraveling Actors' Roles and Relationships, Sanchez-Rola et al, 2021.	The Web never forgets: Persistent Tracking mechanisms in the wild, Acar et al, 2014.	
	Side Channels	Checking Passwords on Leaky Computers: A side channel analysis of Chrome's Password Leak Detect Protocol, Kwong et al., 2023	Enhancing Privacy and Trust in Electronic Communities, Huberman et al, 1999	
	Security and Usability	How do Developers Really Feel About Bug Fixing? Directions for Automatic Program Repair, Winter et al, 2023	How Developers Diagnose Potential Security Vulnerabilities with a Static Analysis Tool, Smith et al, 2019.	

Deliverables

Information regarding due dates for various activities will be posted on Canvas.