

CSE 227 Computer Security

Stefan Savage, Fall 2021

Welcome

Course info

- Stefan Savage
 - Web: <http://www.cs.ucsd.edu/~savage>
 - E-mail: savage@cs.ucsd.edu
 - Office hours: Th 2-3pm (or by appt, or drop by)
CSE 3106
- Canvas: just points to the Web page
- Course Web pages (in progress)
 - <http://www.cse.ucsd.edu/classes/fa21/cse227-a/>

This is a class that is
perpetually “in progress”

You



Me



Background about me

- I work primarily on applied computer security research
- **Research**
 - I'm co-director of the **Center for Networked Systems** (CNS) on campus and the **Center for Evidence Based Security Research** (evidencebasedsecurity.org) w/UCSD and UCL.
 - Lots of work on security measurement, ecrime, security of cyberphysical systems (esp cars and planes)
- **Policy**
 - National Research Council's Cybersecurity Research group
 - Institute for Defense Analysis' ISAT advisory group
 - National Science Foundation CISE Advisory Committee
 - Way too much time in D.C.
 - I co-teach the graduate cybersecurity policy class in GPS and various grad topics security classes (last winter: Cyber & law, this winter: Ransomware)
- **Industry**
 - Asta Networks (defunct anti-DDoS company)
 - Netsift (UCSD-originated worm defense company) -> Cisco
 - A fair amount of consulting...



CSTB



How I got into security...

- Originally OS kernels...
and networking...



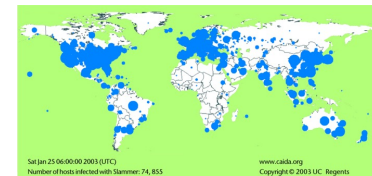
- Came to Security by accident
 - Misbehaving TCP receivers – think like a bad guy
 - DDoS traceback – in response to 2000 attacks

- Startup and... 

synchronicity (David Moore @ UCSD found indirect evidence of spoofed DoS attacks, hmmm... general analysis possible)

Startup was a huge failure, analysis technique was golden

- Code Red
 - Same technique allowed measuring worm outbreaks
 - Interest * opportunity snowballed...



What are we (UCSD) known for in security?

- Measurement
 - Cybercrime: Malware, spam, captcha solving, ad fraud, account abuse, etc
 - Attacks: worms, scanning, ddos, breaches
 - Defenses: threat intelligence, cyber hygiene
- Embedded security
 - Vulnerabilities in automobiles, voting machines, airplanes, credit card skimmers, medical devices
- Web security and PLsec
 - Cookies, information flow
- Intersection of crypto and security (i.e., when does crypto and its implementation really matter)
- Check out cryptosec.ucsd.edu (also sysnet.ucsd.edu)

Goals and non-goals of this course

- **Goals**

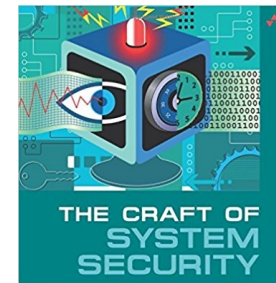
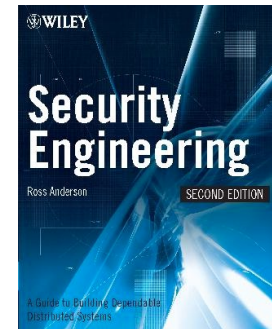
- Learn how to **read papers**; not just for content but **context** (why does this paper exist?)
- Explore range of current problems and tensions in modern computer security
- Understand how to identify security issues in your own work and how to address them
- Figure out if security is an area of interest for **you**
- Get feet wet in security research (research project)

- **Non-goals**

- Review of all standard security mechanisms
 - Read a textbook or take CSE127
- Significant examination of applied cryptography
 - Take one of our great crypto courses
- Deep dive into any one subarea of security

Readings

- There is **no textbook** for this class
 - We'll read a bunch of papers and occasionally from some books
- However, in general I recommend:
 - Security Engineering by Ross Anderson
<http://www.cl.cam.ac.uk/~rja14/book.html>
Free!
- For those who want some general “backup”, check out
 - *The Craft of System Security*
 - Authors: Sean W. Smith, John Marchesini



Topics we'll be covering

- Human factors/usability
- Measurement/analysis studies
- System design/implementation
 - Protection, small TCB, etc
- Information exposure
 - Privacy, anonymity, side & covert channels
- Software vulnerabilities & malware
 - Vulnerability research, viruses, botnets, defenses, etc
- I'm open to more topics... got any?

The work in this class

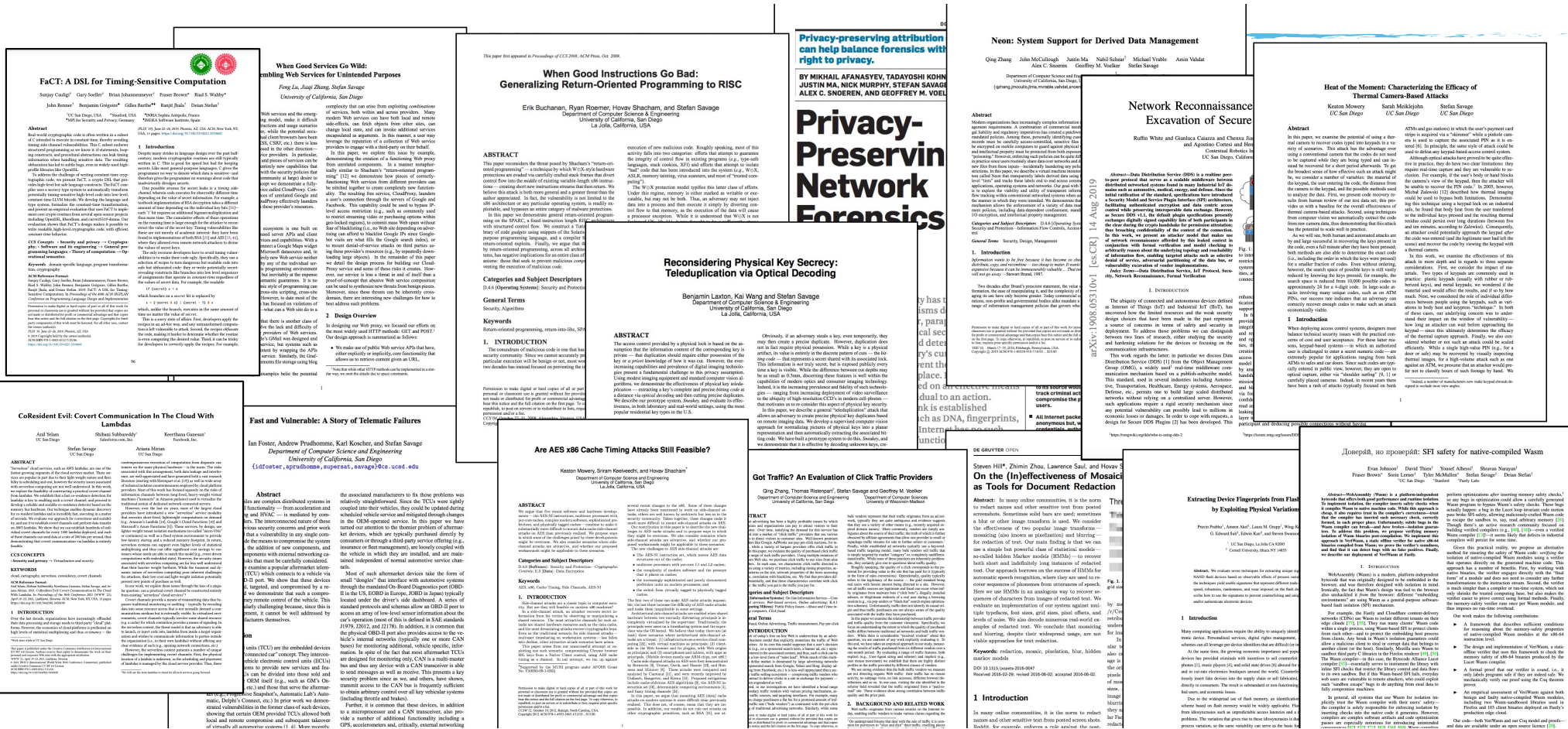
- Reading
 - Lots of it (~4 papers/week) and I expect you to be prepared (i.e., have read the papers) for class discussion
- Discussion in class
 - The papers and concepts we've covered
 - 15% of grade is participation
- Project
 - This is the purpose of the class (85% of grade)

No other homework, midterm, or final

Projects

- Some kind of **research project** in security
- Best in a group of 2-3
 - If you can't find partner(s) let me know and I will try to help
- Please try to form group by Oct 4th
 - Send me e-mail identifying your group members by then
- Initial project proposals due Oct 13th
 - One page
 - What you would like to do, why it is interesting, how you plan to do it, what the challenges/risks are (and/or what resources you need)
 - We will go over proposals and ok project or help you refine
- Ultimately 6 pages and short talk (10-15mins)
 - High standards: we've published well over a dozen papers from this class

Some class project alumni...



Generally speaking

- Most projects will fall into the category of:
 - **Analysis**: evaluate the security of a system of interest
 - **Attack**: identify some new attack/vulnerability, develop/test it and discuss the possible ramifications, mitigations, etc
 - **Measurement**: measure some aspect of adversarial behavior (real or potential), user behavior, code behavior, characterize it, explore its limits, etc
 - **Design/Implementation**: design and/or build a new system that addresses a problem in a new way

Things to think about...

- Pick good **problems**
 - Why is this problem interesting or will become interesting?
 - New technology, new approach, new question, new usage, etc?
 - Look at what others are doing:
 - Academic conferences: USENIX Security, ACM CCS, IEEE S&P, NDSS, UIST, DIMVA, RAID, LEET, WOOT, PETS, eCrime -- others
 - Non-academic conferences: BlackHat, Defcon, HITB, ShmooCon, INFILTRATE, various blogs
- Pick approaches to problems that are **achievable**
 - What resources would you need to investigate the problem?
- Think about how to **evaluate** your work

Random ideas

- On the class Web page)
 - This is not a list you must pick from!
 - Just examples to give you ideas and make sure you understand how broad the scope is

Example:

Seme random projects I'm interested in

- Privacy issues with Zoom status
 - Zoom (and other systems) provide status and notification APIs (i.e., that someone is available or busy) and even callbacks for when they get free
 - Can one infer, from that data, who is in meetings with whom across campus?
- Seeing through virtual backgrounds
 - “virtual greenscreen” depends on abstraction of status background
 - Real-background is static and esp around moving edges
 - How much can you discern of real background?
- Implementation of implicit second factor auth using Smartwatch
 - Accelerometer identifies password typing at same time password entered
- System for identifying popular libraries (and function signatures) in unknown binary for reverse engineering (integrate with Ghidra)
- Electrical system watermarking to establish where audio/video was recorded(see: <https://github.com/bellingcat/open-questions>, also check out PinDrop paper)

Resources for projects

- Data: blacklists, DNS data (zone files), spam, malware
 - Computing: servers, network bandwidth
 - Equipment: telescope, camera, lots of low-level stuff in the embedded lab (scopes, probes, etc),
 - Lots of experience in doing unusual things
-
- If there is something you need to do your project and you're not sure how to get it – ask.

Questions about project?

What is security?

- Merriam-Webster online dictionary:

Function: *noun*

- 1** : the quality or state of being [secure](#) : as **a** : freedom from danger : [SAFETY](#) **b** : freedom from fear or anxiety **c** : freedom from the prospect of being laid off
<job security>
- 2** **a** : something given, deposited, or pledged to make certain the fulfillment of an obligation **b** : [SURETY](#)
- 3** : an instrument of investment in the form of a document (as a stock certificate or bond) providing evidence of its ownership
- 4** **a** : something that [secures](#) : [PROTECTION](#) **b** (1) : measures taken to guard against espionage or sabotage, crime, attack, or escape (2) : an organization or department whose task is security

Computer security?

- Most of computer science is about providing *functionality*:
 - UX/UI
 - Software Architecture
 - Algorithms
 - Operating Systems/Networking/Databases
 - Compilers/PL
 - Microarchitecture
 - VLSI/CAD
- Computer security is *not* about functionality
- It is about how the embodiment of functionality behaves *in the presence of an adversary*
- Holistic property
 - “Software security is about integrating security practices into the way you build software, not integrating security features into your code” – Gary McGraw

History: two competing philosophies

- **Binary** model [secure vs insecure]
 - Traditional crypto and trustworthy systems
 - Assume adversary limitations X and define security policy Y
 - If Y cannot be violated without needing X then system is secure, else insecure
 - You know people are invoking some version of this model if they say “proof of security”, “secure by design” “trustworthy systems”
- **Risk management** model. [more secure vs less secure]
 - Most commercial software development
(and much real-world security... e.g., terrorism)
 - Try to minimize biggest risks and threats
 - Improve security where most cost effective (expected value)
 - You know people are in this model if they use the words “risk”, “mitigation”, “defenses”, “resilience”, etc.

Classic example (binary model): perfect substitution cipher

$$\begin{array}{ccccccc} p_1 & p_2 & p_3 & \dots & p_n \\ \oplus & \underline{b_1} & \underline{b_2} & \underline{b_3} & \dots & \underline{b_n} \\ c_1 & c_2 & c_3 & \dots & c_n \end{array}$$

- Invited by combination of Vernam & Mauborgne (~1919)
- Choose a string of **random** bits the same length as the plaintext, XOR them to obtain the ciphertext.
- **Perfect Secrecy** (proved by Claude Shannon)
 - ♦ Probability that a given message is encoded in the ciphertext is **unaltered** by knowledge of the ciphertext
 - ♦ Proof: Give me any plaintext message and any ciphertext and I can construct a key that will produce the ciphertext from the plaintext. Zero information in ciphertext

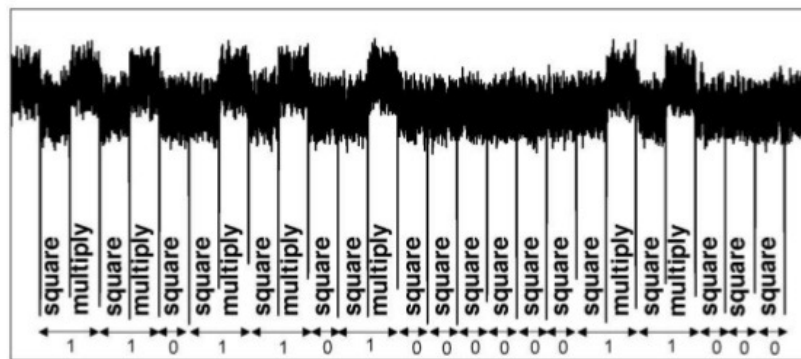
Classic example (risk management): Concrete barricades

- Prevent incursion by car bombers



Problems with the binary model: Abstract design != Concrete artifact

- Many assumptions are **brittle** in real systems
 - Real artifacts fragile, imperfect, have bugs/limitations
 - Don't do precisely what spec says or documentation says
 - E.g., what is an integer?
 - Large gap between abstraction and implementation
 - Example: secret key in chip used to decrypt data; key leaks via the current the chip draws for different operations

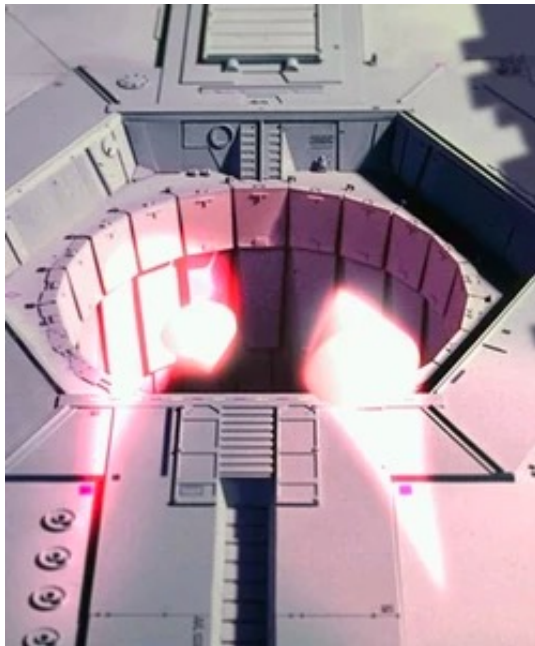


Courtesy
Oswald

Problems with the binary model: security evolution

- As engineers, we often delude ourselves into thinking that we understand our own creations
 - or that we can create complex systems to do only what we meant them to do
- But ... nobody knows how these systems really work
 - Complexity of computer systems is approaching complexity of biological organisms
 - 3 billion base pairs in human genome
 - 10+ billion transistors in modern CPUs
- Complex systems co-evolve with attacks against them
 - How we use systems, how we depend on them and how they might be attacked – all change over time
 - Systems deemed secure today may not be resilient to new threats

Problems with the risk management model:
One vulnerability can matter...



Problems with the risk management model: You never win

- Creates arms race – forced co-evolution



- The best you can hope for is **stalemate**

Problems with the risk management model: How to measure

- Its fine to say security is a spectrum, but how to **evaluate** risk or reward?
 - How many units of security does your anti-virus product give you?
- Big question: how do we measure security?
 - How is this different from car safety?
 - Or drug safety?

Key meta issues in Security

- Policy
- Assets, Risks & Threats
- Value
- Protection
- Deterrence

Policy

- What **is** a bad thing?
- Remarkably tricky to define for known threats
 - The software on your computer likely has 100s of security options... How should you set them?
 - What might be a good security policy for who gets to access faculty salary data?
- Even harder for unknown threats
 - SPAM
- Should a **highly privileged** user have more rights on a system or less?

Assets, Risks & threats

- Assets
 - What you want to protect
- Threats
 - Actions likely to cause damage, harm or loss
 - Includes both kinds of attacks (e.g., virus, social engineering) and kinds of attackers (e.g., script kiddie vs state sponsored actor)
 - Need to reason about requirements of each threat (what capabilities does the attacker need) and what it enables (what harm might come? What motivations might drive such a threat)
- Risk
 - What is the potential likelihood of a something bad happening (i.e., what threats are likely)
- These tend to be well formalized in some communities (e.g. finance sector) and less in others (e.g. energy sector)
- We'll talk more about threat models next class...

Value

- What is the cost if the bad thing happens?
- What is the cost of preventing the bad thing?
- Example: credit card fraud
 - Who pays if someone steals your credit card # and buys a TV with it?
- Example: Permissive Action Links for nuclear weapons
 - <http://www.cs.columbia.edu/~smb/nsam-16o/pal.html>



Protection

- The mechanisms used to protect resources against threats
 - This is most of academic and industrial computer security
- Many classes of protections
 - Cryptographic protection of data
 - Software guards
 - Communication guards
 - User interface design (protect user against own limitations)
- Can be either proactive or reactive

Deterrence

- There is some non-zero expectation that there is a future cost to doing a bad thing
 - i.e. going to jail, having a missile hit your house, having your assets seized, etc
 - Criminal cost-benefit: $M_b + P_b > O_{cp} + O_{cm}P_aP_c$ [Clark&Davis 95]
 - M_b : Monetary benefit
 - P_b : Psychological benefit
 - O_{cp} : Cost of committing crime
 - O_{cm} : Monetary cost of conviction
 - P_a : Probability of getting caught
 - P_c : Probability of conviction
- Need meaningful forensic capabilities
 - Audit actions, assign identity to evidence, etc
 - Must be cost effective relative to positive incentives

That's it for today

- For next time, read
 - Low-level Software Security by Example
 - Exploit Programming: From Buffer Overflows to “Weird Machines”
- Write down questions you have as you read these papers
- Also, spend a few minutes looking into the background of the authors and the citations and ask yourself:
 - Why are they writing this paper? Why them? Why then? Why there?