

# CSE 127: Introduction to Security

**Nadia Heninger**

UCSD

Winter 2023 Lecture 1

# Topics Covered

- The Security Mindset
  - Principles and threat modeling
- Systems/Software Security
  - Classic attacks and defenses on memory safety, isolation
- Web Security
  - Web architecture, web attacks, web defenses
- Network Security
  - Network protocols, network attacks, network defenses
- Cryptography
  - Public and private-key cryptography, TLS, PKI
- Privacy, Anonymity, Ethics, Legal Issues

# Course Goals

- Critical thinking
  - How to think like an attacker
  - How to reason about threats and risks
  - How to balance security costs and benefits

# Course Goals

- Critical thinking
  - How to think like an attacker
  - How to reason about threats and risks
  - How to balance security costs and benefits
- Technical skills
  - How to protect yourself
  - How to manage and defend systems
  - How to design and implement secure systems

# Course Goals

- Critical thinking
  - How to think like an attacker
  - How to reason about threats and risks
  - How to balance security costs and benefits
- Technical skills
  - How to protect yourself
  - How to manage and defend systems
  - How to design and implement secure systems
- Learn to be a security-conscious citizen

# Course Goals

- Critical thinking
  - How to think like an attacker
  - How to reason about threats and risks
  - How to balance security costs and benefits
- Technical skills
  - How to protect yourself
  - How to manage and defend systems
  - How to design and implement secure systems
- Learn to be a security-conscious citizen
- Learn to be a leet h4x0r

# Course Goals

- Critical thinking
  - How to think like an attacker
  - How to reason about threats and risks
  - How to balance security costs and benefits
- Technical skills
  - How to protect yourself
  - How to manage and defend systems
  - How to design and implement secure systems
- Learn to be a security-conscious citizen
- Learn to be a 1eet h4x0r, but an ethical one!

# Course Mechanics

40% Five projects

- Work in groups of two
- Do your own programming and writeup
- General discussion is encouraged



# Course Mechanics

40% Five projects

- Work in groups of two
- Do your own programming and writeup
- General discussion is encouraged

20% Midterm exam in class week 5

- In person
- Closed book with cheat sheet

# Course Mechanics

40% Five projects

- Work in groups of two
- Do your own programming and writeup
- General discussion is encouraged

20% Midterm exam in class week 5

- In person
- Closed book with cheat sheet

40% Final exam in scheduled exam time

- In person
- Closed book with cheat sheet

# Course Policies

## **Late days and extensions:**

- You have two late days to use as you wish
- Both you and your partner must have late days to use them

# Course Policies

## **Late days and extensions:**

- You have two late days to use as you wish
- Both you and your partner must have late days to use them

## **Regrade policy:**

- Regrades should be the exception not the norm
- Incorrect regrade request  $\implies$  negative points

# Course Policies

## **Late days and extensions:**

- You have two late days to use as you wish
- Both you and your partner must have late days to use them

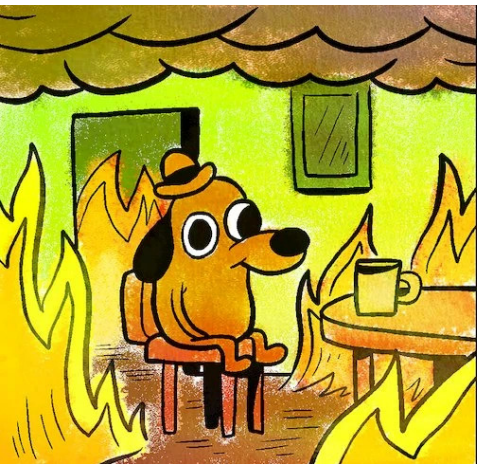
## **Regrade policy:**

- Regrades should be the exception not the norm
- Incorrect regrade request  $\implies$  negative points

## **Academic integrity:**

- UC San Diego policy:  
<https://academicintegrity.ucsd.edu>
- We have to report suspected cases, don't make it weird
- If you are not sure if something is cheating, ask

Talk to us, it's a weird time



# Course Resources

- No official textbook. Optional books:
  - *Security Engineering* by Ross Anderson
  - *Hacking: The Art of Exploitation* by Jon Erikson

# Course Resources

- No official textbook. Optional books:
  - *Security Engineering* by Ross Anderson
  - *Hacking: The Art of Exploitation* by Jon Erikson
- Assignments and readings on course web site.

<https://cseweb.ucsd.edu/classes/wi23/cse127-a/>



# Course Resources

- No official textbook. Optional books:
  - *Security Engineering* by Ross Anderson
  - *Hacking: The Art of Exploitation* by Jon Erikson
- Assignments and readings on course web site.  
<https://cseweb.ucsd.edu/classes/wi23/cse127-a/>
- Questions? Post to Piazza.

# Course Resources

- No official textbook. Optional books:
  - *Security Engineering* by Ross Anderson
  - *Hacking: The Art of Exploitation* by Jon Erikson
- Assignments and readings on course web site.  
<https://cseweb.ucsd.edu/classes/wi23/cse127-a/>
- Questions? Post to Piazza.
- Lectures, section, office hours:
  - In person by default
  - Podcasting is on for lectures and discussion
  - Two weeks of lecture mid-quarter will be on Zoom

# Ethics

We will be discussing and implementing real-world attacks.

Using some of these these techniques in the real world may be unethical, a violation of university policies, or a violation of federal law.

This includes the course assignment infrastructure (e.g., grading system).

# Ethics

We will be discussing and implementing real-world attacks.

Using some of these techniques in the real world may be unethical, a violation of university policies, or a violation of federal law.

This includes the course assignment infrastructure (e.g., grading system).

Be an ethical hacker

- Ethics requires you to refrain from doing harm
- Always respect human, privacy, property rights
- There are many legitimate hacking capture-the-flag competitions

## 18 U.S. CODE § 1030 - FRAUD AND RELATED ACTIVITY IN CONNECTION WITH COMPUTERS

Whoever intentionally accesses a computer without authorization or exceeds authorized access, and thereby obtains information from any protected computer...

# 18 U.S. CODE § 1030 - FRAUD AND RELATED ACTIVITY IN CONNECTION WITH COMPUTERS

Whoever intentionally accesses a computer without authorization or exceeds authorized access, and thereby obtains information from any protected computer...

The punishment for an offense...

- a fine under this title or imprisonment for not more than one year, or both...,
- a fine under this title or imprisonment for not more than 5 years, or both... if—
  - (i) the offense was committed for purposes of commercial advantage or private financial gain;
  - (ii) the offense was committed in furtherance of any criminal or tortious act...; or
  - (iii) the value of the information obtained exceeds \$5,000

## CFAA Cases

- In 2011, FBI prosecuted weev for exposing data of 114K AT&T iPad users
  - Criminal CFAA charge
  - Found guilty and sent to prison; appeals court overturned ruling in 2014 on venue grounds

# CFAA Cases

- In 2011, FBI prosecuted weev for exposing data of 114K AT&T iPad users
  - Criminal CFAA charge
  - Found guilty and sent to prison; appeals court overturned ruling in 2014 on venue grounds
- In 2011, Sony sued George Hotz for jailbreaking PlayStation 3
  - Civil CFAA and DMCA complaints
  - Settled out of court



# CFAA Cases

- In 2011, FBI prosecuted weev for exposing data of 114K AT&T iPad users
  - Criminal CFAA charge
  - Found guilty and sent to prison; appeals court overturned ruling in 2014 on venue grounds
- In 2011, Sony sued George Hotz for jailbreaking PlayStation 3
  - Civil CFAA and DMCA complaints
  - Settled out of court
- In 2011, FBI prosecuted Aaron Swartz for downloading academic articles on MIT network from JSTOR
  - Indicted for wire fraud and CFAA
  - Prosecution continued until his death in 2013

# CFAA Cases

- In 2011, FBI prosecuted weev for exposing data of 114K AT&T iPad users
  - Criminal CFAA charge
  - Found guilty and sent to prison; appeals court overturned ruling in 2014 on venue grounds
- In 2011, Sony sued George Hotz for jailbreaking PlayStation 3
  - Civil CFAA and DMCA complaints
  - Settled out of court
- In 2011, FBI prosecuted Aaron Swartz for downloading academic articles on MIT network from JSTOR
  - Indicted for wire fraud and CFAA
  - Prosecution continued until his death in 2013
- In 2021, Van Buren was charged with exceeding authorized access under CFAA
  - Police officer misused license plate database
  - Supreme court ruling (6-3) ruled that authorized access for improper purposes is not "exceeding authorized access"

My work: Cryptographic systems security





# Crypto shocker: four of every 1,000 public keys provide no security (updated)

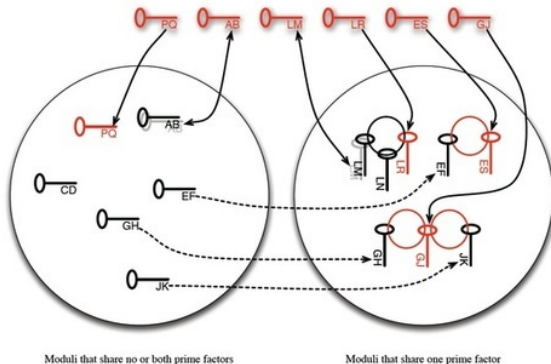
Almost 27,000 certificates used to protect webmail, e-commerce, and other ...

by Dan Goodin - Feb 15 2012, 7:00am EST

Share

Tweet

68



Keys that share one prime factor are vulnerable to cracking by anyone. Keys that share both prime factors can be broken by the other holder.

# Researchers reveal a method the NSA may use to spy on Web traffic

By Sean Spósito | October 21, 2015 | Updated: October 21, 2015 5:05pm



1



# My work: Mathematical cryptography

```
p = random_prime(2^512); q = random_prime(2^512)
N = p*q
```

$$a = p - (p \% 2^{86})$$

```
sage: hex(a)
```

[illegible]
$$X = 2^{86}$$

```
M = matrix([X^2, 2*X*a, a^2], [0, X, a], [0, 0, N])
```

```
B = M.LLL()
```

$$f = B[0][0]*x^2/X^2+B[0][1]*x/X+B[0][2]$$

```
sage: f.factor()[0]
```

$$(x - 2775338500016599864377709 \bullet 1)$$

ISSN: 1077-5229/000165000642373700

What is security?



What makes it different from robustness?



# What makes it different from robustness?



“Computer security studies how systems behave in the presence of *an adversary*.”

*\*Actively tries to cause the system to misbehave.*

# The Security Mindset

- Thinking like an attacker
  - Understand techniques for circumventing security
  - Look for ways security can break, not why it won't

# The Security Mindset

- Thinking like an attacker
  - Understand techniques for circumventing security
  - Look for ways security can break, not why it won't
- Thinking like a defender
  - Know what you're defending, and against whom.
  - Weigh benefits vs. costs:  
No system is ever completely secure.
  - Rational paranoia  
Don't build bridges to sustain bombings

# Thinking like an Attacker

- Look for weakest links
- Identify assumptions that security depends on  
Are they false?

# Thinking like an Attacker

- Look for weakest links
- Identify assumptions that security depends on  
Are they false?
- Think outside the box

# Thinking like an Attacker

- Look for weakest links
- Identify assumptions that security depends on  
Are they false?
- Think outside the box  
Not constrained by system designer's worldview!

# Thinking like an Attacker

- Look for weakest links
- Identify assumptions that security depends on  
Are they false?
- Think outside the box  
Not constrained by system designer's worldview!

Start practicing: When you interact with a system, think about what it means to be secure, and how it might be exploited.







## Exercise

How would you break into the CSE building?

## Exercise

How would you identify who was at a protest?

## Exercise

How would you steal my email password?

## Exercise

What security systems do you interact with?

# Thinking like a Defender

- Security policy
  - What are we trying to protect?
  - What properties are we trying to enforce?
- Threat model
  - Who are the attackers? Capabilities? Motivation?
  - What kind of attack are we trying to prevent?
- Risk assessment
  - What are the weaknesses of the system?
  - What will successful attacks cost us?
  - How likely?
- Countermeasures
  - Costs vs. benefits?
  - Technical vs. nontechnical?

# Security Policies

- What *assets* are we trying to protect?
  - Password (hashes)
  - Emails
  - Browsing history
- What properties are we trying to enforce?
  - Confidentiality
  - Integrity
  - Availability
  - Privacy
  - Authenticity



# Threat Models

- Who are our adversaries?
  - Motives?
  - Capabilities?
- What kinds of attacks do we need to prevent?  
(Think like the attacker!)
- Limits: What kinds of attacks we should ignore?

# Example of Threat Modeling

Threat	Ex-girlfriend/boyfriend breaking into your email account and publicly releasing your correspondence with the My Little Pony fan club	Organized criminals breaking into your email account and sending spam using your identity	The Mossad doing Mossad things with your email account
Solution	Strong passwords	Strong passwords + common sense (don't click on unsolicited herbal Viagra ads that result in keyloggers and sorrow)	Magical amulets? Fake your own death, move into a submarine? YOU'RE STILL GONNA BE MOSSAD'ED UPON

Figure 1: Threat models

James Mickens "This World of Ours"

# Example of Threat Modeling



## Someone has your password

Hi John

Someone just used your password to try to sign in to your Google Account  
[john.podesta@gmail.com](mailto:john.podesta@gmail.com).

### Details:

Saturday, 19 March, 8:34:30 UTC

IP Address: 134.249.139.239

Location: Ukraine

Google stopped this sign-in attempt. You should change your password immediately.

[CHANGE PASSWORD](#)

Best,  
The Gmail Team

Who is John Podesta?

# Assessing Risk

Remember: *Controlled paranoia*

- What would security breaches cost us?
  - Direct costs: Money, property, safety, ...
  - Indirect costs: Reputation, future business, well being, ...
- How likely are these costs?
  - Probability of attacks?
  - Probability of success?

# Countermeasures

- Technical countermeasures
- Nontechnical countermeasures  
Law, policy (government, institutional), procedures, training, auditing, incentives, etc.



# Security Costs

- No security mechanism is free
  - Direct costs:  
Design, implementation, enforcement, false positives
  - Indirect costs:  
Lost productivity, added complexity
- Challenge is to rationally weigh costs vs. risk
  - Human psychology makes reasoning about high cost/low probability events hard



# Exercise

Should you lock your door?

- Assets?
- Adversaries?
- Risk assessment?
- Countermeasures?
- Costs/benefits?

# Exercise

Should you use automatic software updates?

- Assets?
- Adversaries?
- Risk assessment?
- Countermeasures?
- Costs/benefits?

# Exercise

Should we protect the CSE bear?

- Assets?
- Adversaries?
- Risk assessment?
- Countermeasures?
- Costs/benefits?

# Secure Design

- Common mistake:  
Convince yourself that the system is secure
- Better approach:  
Identify *weaknesses* of design, focus on correcting them  
Formally prove that design is secure (soon)
- Secure design is a **process**  
Must be practiced continuously  
Retrofitting security is super hard

# Where to focus defenses

- *Trusted components*  
Parts that must function correctly for the system to be secure.
- *Attack surface*  
Parts of the system exposed to the attacker

## Security Principles

- Simplicity, open design, and maintainability
- Privilege separation and least privilege
- Defense-in-depth and diversity
- Complete mediation and fail-safe

## Exercise

Preventing cheating on an online exam?

## Exercise

Preventing you from stealing my password?

Next lecture: Buffer overflows!