Week 1: Lecture A

Course Intro & The Security Mindset

Tuesday, August 19, 2024

Reminders

- Be sure to join the course **Canvas** and **Piazza**
  - See links at top of course page
  - http://cs4440.eng.utah.edu
- Trouble accessing? See me after class!
  - Or email me at: snagy@cs.utah.edu

Today's Class

- **Welcome to CS 4440** 😃
- Course Overview
- The Security Mindset
  - Thinking like an attacker
  - Thinking as a defender
- Ethics and Academic Integrity

Course Staff

Course Instructor

Stefan Nagy
Assistant Professor, KSoC

Email: snagy@cs.utah.edu
Office: Merrill Eng. 3446

Teaching Assistants

Alishia Seo

Bella Miller

Ethan Quinlan

About Me

Stefan Nagy
Assistant Professor, KSoC

cs.utah.edu/~snagy
twitter.com/snagycs
@snagy@infosec.exchange

Co-founder and Co-director:

SSG UTAH SOFTWARE SECURITY GROUP
SCHOOL OF COMPUTING | THE UNIVERSITY OF UTAH

Places I've been:

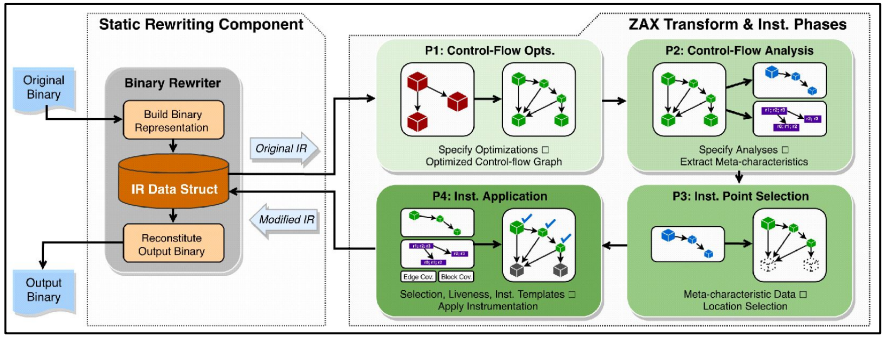
University of Utah, 2022–now

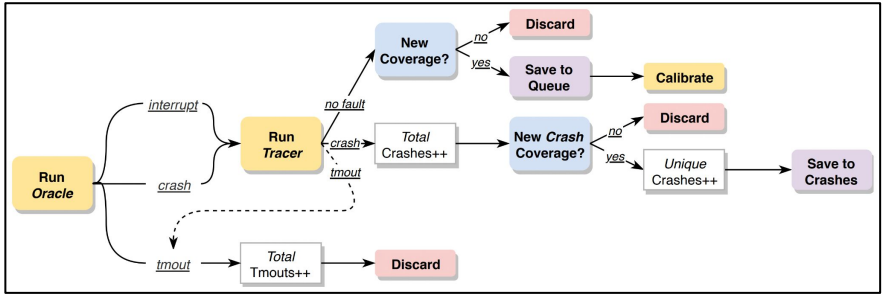
Virginia Tech, Ph.D. 2016–2022

Univ. of Illinois, B.S. 2012–2016

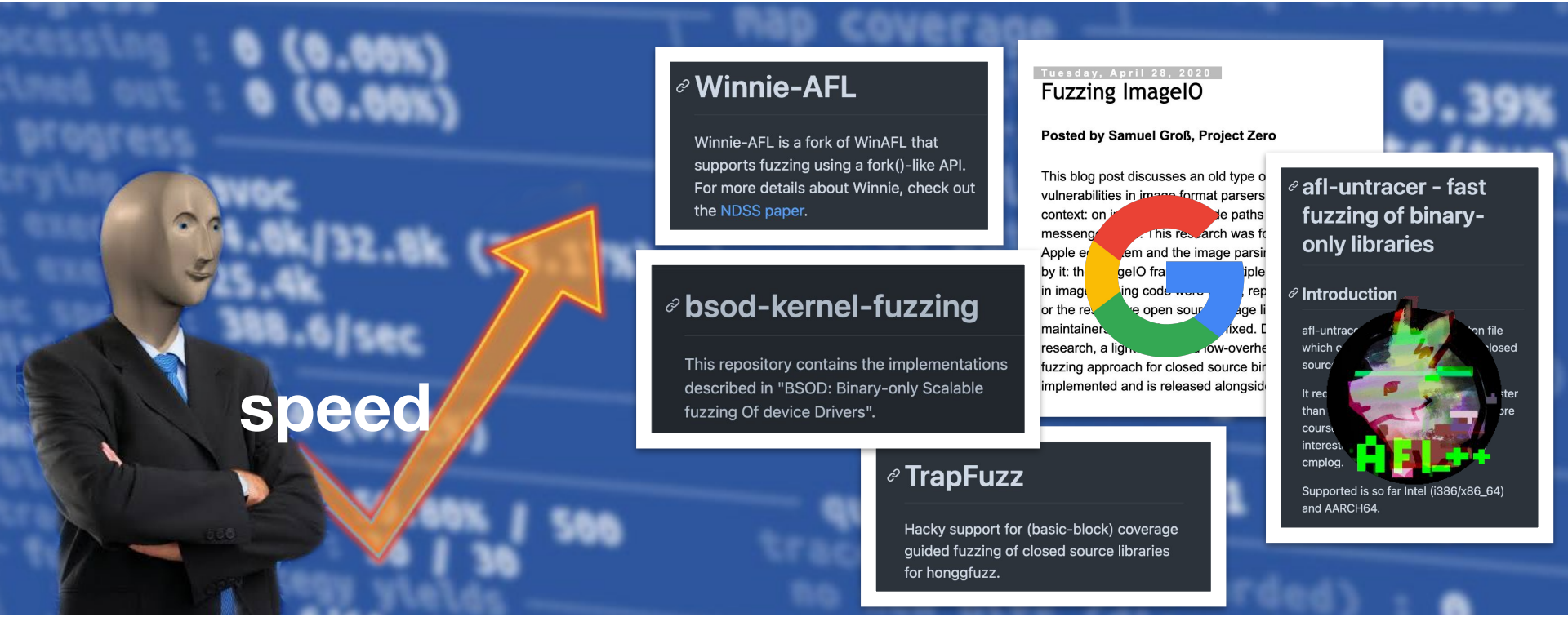
My Research: Fast Fuzzing

- Fastest techniques for fuzzing closed-source executable code
  - ZAFL (USENIX'21)
    - Make testing of closed-source code as fast as open-source
  - UnTracer (S&P'19), HeXcite (CCS'21)
    - Make testing of closed-source code faster than open-source





My Research: Fast Fuzzing



My Research: Extending Fuzzing's Reach

- Highly-configurable commodity software
  - Variability-induced bugs
  - Patch integrity vetting
  - Accelerating bug discovery

```
9     #ifdef CONFIG_INCR      // DISABLED
10    x = x + 1;
11    #endif
12    #ifdef CONFIG_DECR      // ENABLED
13    x = x - 1;
14    #endif
15    foo(x);
16 }
```

(3)

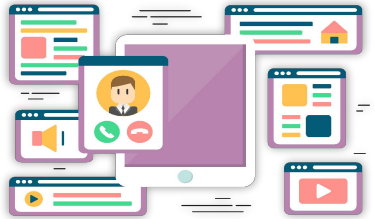
(4)→

My Research: Extending Fuzzing's Reach

- Highly-configurable commodity software
  - Variability-induced bugs
  - Patch integrity vetting
  - Accelerating bug discovery

```
 9    #ifdef CONFIG_INCR      // DISABLED
10    x = x + 1;
11    #endif
12    #ifdef CONFIG_DECR      // ENABLED
13    x = x - 1;
14    #endif
15    foo(x);
16  }
```

- Expanding auditing of opaque software
  - Testing binary analysis tools
  - Automated interface recovery
  - Closed-source operating systems



My Research Group

FUTURES^3

LAB FUTURE TECHNOLGY FOR USABLE, RELIABLE, & EFFICIENT SECURITY OF SOFTWARE & SYSTEMS

SCHOOL OF COMPUTING | THE UNIVERSITY OF UTAH | SALT LAKE CITY

Our work: systems and software security, binary analysis, fuzzing

The Utah Cybersecurity Club

utahsec

Come hack with us!

The Utah Cybersecurity Club

UtahSec Cybersecurity Club

Activities:

-Weekly Capture the flag Competitions

-Cybersecurity Skills Workshop

-Meeting with industry leaders

-Onsite visit to Zions Bank

ARCTIC WOLF

redo

Lucid

ZIONS BANK

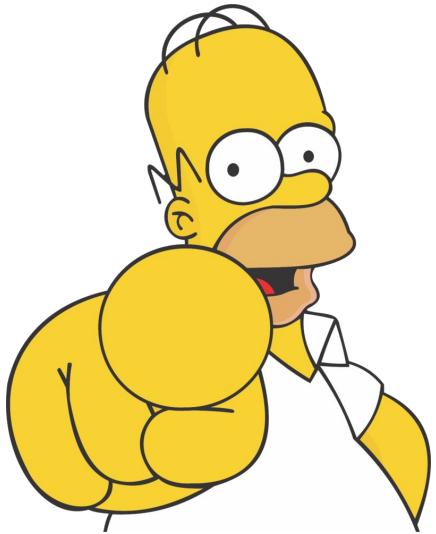
Meeting: Every Thursday
Time: 06:00 PM
Location: Meb 3115

DISCORD

Help us get to know you!

Help us get to know you!

- Throughout lecture we'll use **Poll Everywhere**
  - Use your laptop to send-in your responses
  - Share location—we're checking you're here!
  - Poll participation = **5%** of your grade
- To receive credit:
  - Log-in via your UMAIL (e.g., `u8675309@utah.edu`)
  - We've automatically registered you (**if not, see me**)
- Answer the following questions to give us some more info about you!



Experience with Programming Languages

(A) HTML / JavaScript

0%

(B) C / C++

0%

(C) Python

0%

(D) Assembly

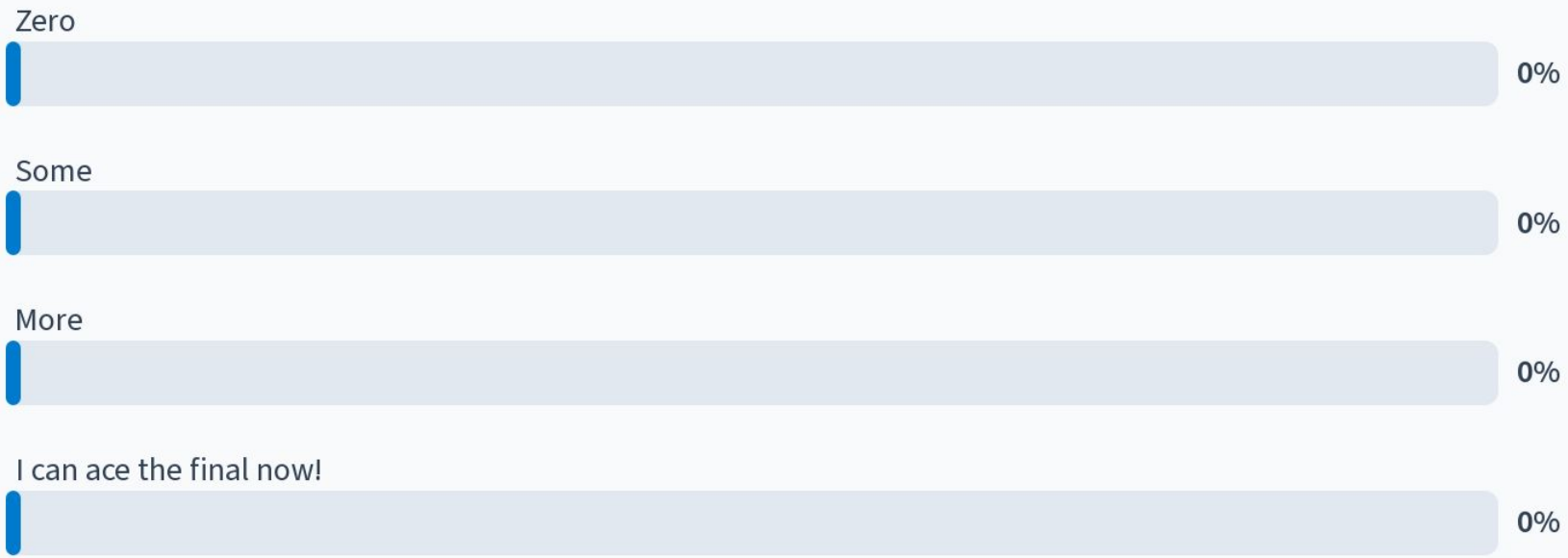
0%

(E) None of the above

0%

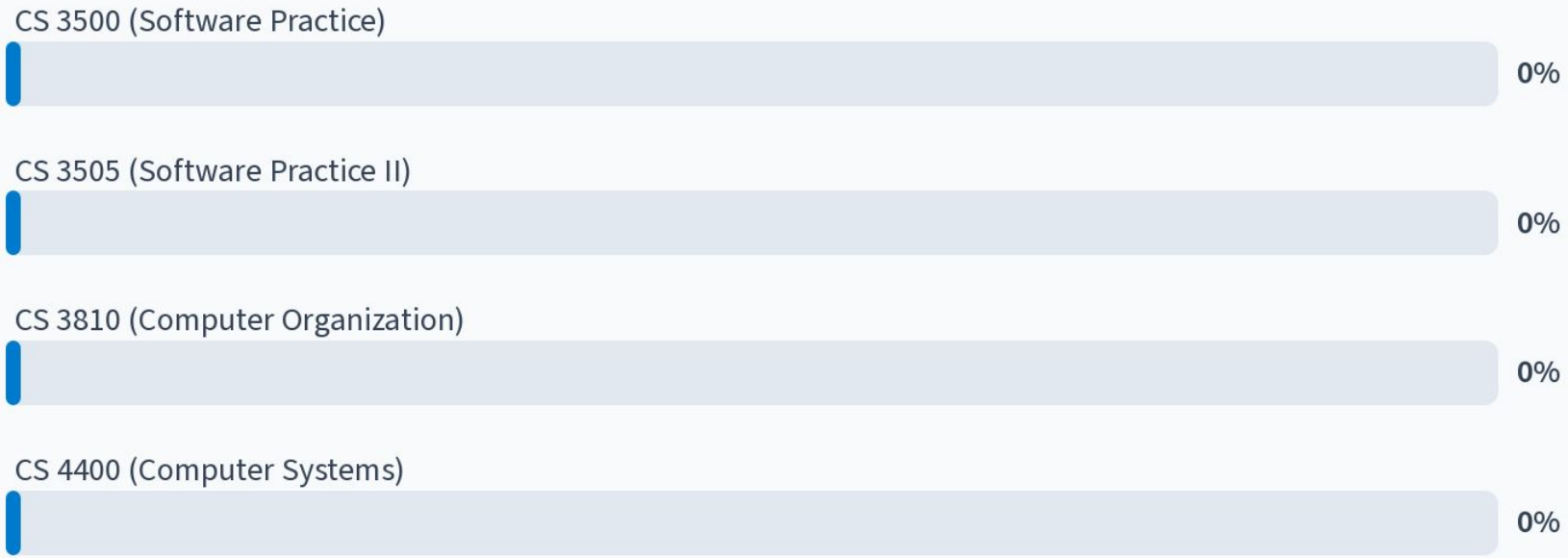
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Experience with Cybersecurity



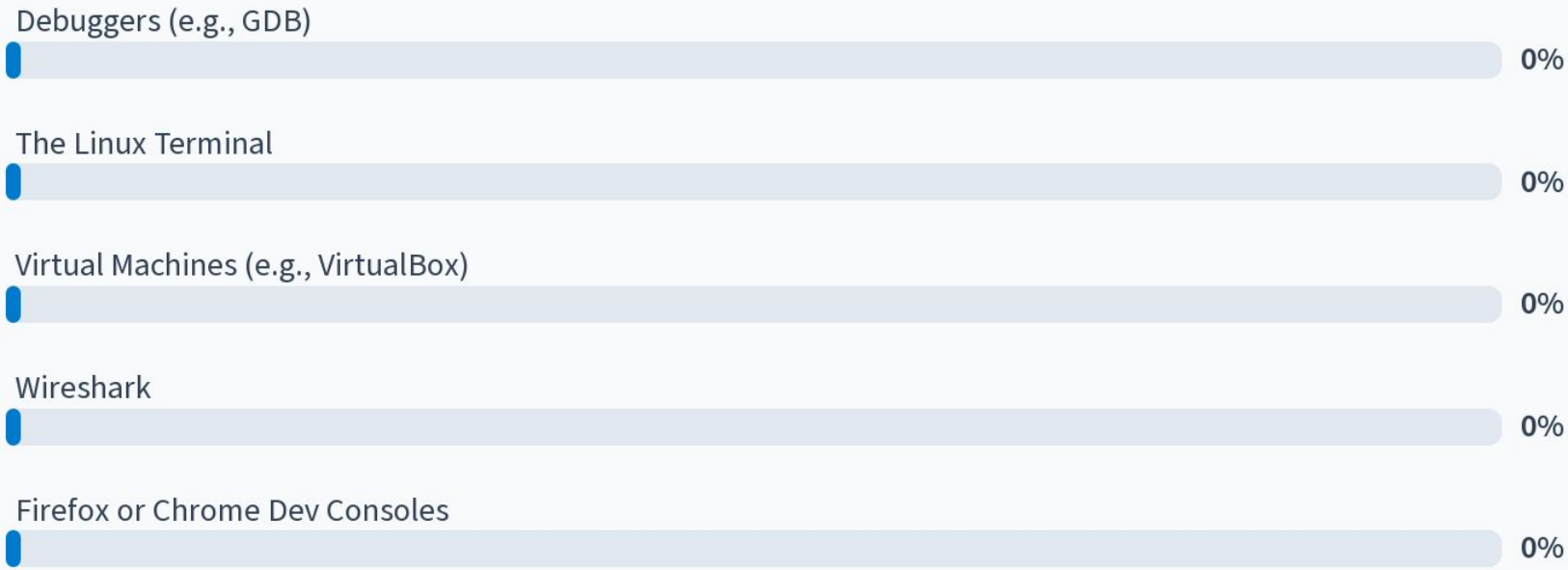
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Courses Previously Taken



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Experience with Tools



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Last Question

What do you hope to get out of this course?

And no, I don’t mean the grade that you want 🙂

Course Overview

Course Goals

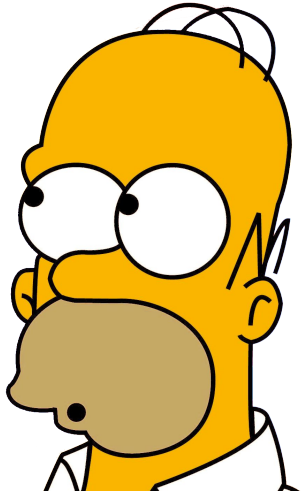
- **Critical Thinking**
  - How to think like an attacker
  - How to reason about threats and risks
  - How to balance security costs and benefits
- **Technical Skills**
  - How to protect yourself
  - How to manage and defend systems
  - How to design and program secure systems
- **Learning to be a security-conscious citizen**
- **Learning to be a** L337 H4X0R**... but an ethical one!**

Topics

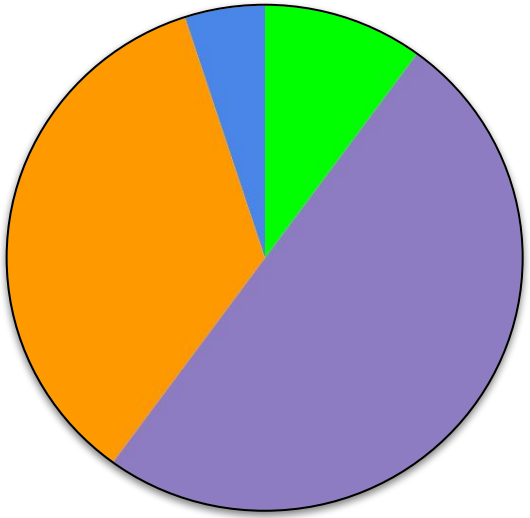
- Course Intro & The Security Mindset — Week 1
  - Principles, threat modeling, vulnerabilities, attacking versus defending; VM setup
- P1: Communications Security — Weeks 2–4
  - Public- and private-key crypto, digital signatures, authentication, hashes, secure channels
- P2: Application Security — Weeks 5–8
  - Memory protection, sandboxing, virtual machines, software exploitation, malware, testing
- P3: Web and Network Security — Weeks 9–12
  - IP, TCP, routing, net protocols, web architecture, web attacks, firewalls, intrusion detection
- P4: New Frontiers in Security — Weeks 13–15
  - Side channels, hardware, ML security, reverse engineering, election security, policy, ethics
- Course Wrap-up — Week 16
  - Careers in cybersecurity, the security ecosystem; the final exam

Common Concerns

- Attendance required? **Yes.**
  - Standard lecture format:
    - ~20 minutes of review
    - ~55 minutes of new material
- Textbook is required? **No.**
  - … but highly recommended!
  - We provide **6 free textbooks** on the site!
- Midterm exam? **No.** Final exam? **Yes.**
  - Covers entire course material
  - Review session as final lecture
  - Similar to in-class and quiz questions



Grading



- **10%** = weekly solo quizzes based on lectures
- **50%** = four Programming Projects (**12.5%** each)
- **35%** = Final Exam covering all course material
- **5%** = participation during lecture poll exercises

Lecture Quizzes (10%)

- Weekly exercises to be completed **individually**
  - Designed to test your understanding of the lectures
- Released on **Canvas** after Tuesday's lecture
  - You may work until the following **Monday by 11:59 PM**
  - Strict deadline—**late submissions will not be accepted**
- **Lowest score** will be **dropped** at no penalty

Programming Projects (50%)

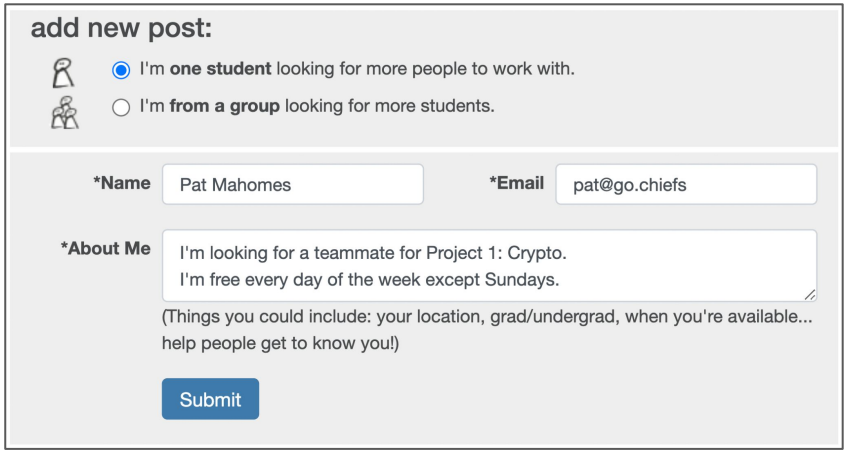
- Four projects completed in groups of **no more than two**
  - You can discuss your approaches with other groups
  - Must complete and submit **only within your group**
- **Topics:** Crypto, App security, Web security, Net security
- Where to find and submit?
  - Distributed via **course website** (we'll announce when)
  - Upload your work (**one** per team) as tarball to **Canvas**

Project Teams

- Can work in **teams of up to two**
  - Find teammates on **Piazza**
  - Post on 

- Why work with someone else?
  - Pair programming
  - Divide and conquer
  - Two sets of eyes to solve problems
  - Teaching others helps you learn more

- Yes, you are free to work solo…
  - But we encourage you to team up!



Project Lateness Policy

- Course staff constraints:
  - We want to return graded work promptly
  - Can't discuss solutions until all work graded
- Project lateness policy:
  - **10% penalty** for being late up to **two days past deadline**
  - **Will not accept after 48 hours** past the original deadline
  - Extensions made only under **extraordinary** circumstances
- **Please start early!** It is your responsibility to…
  - Turn in assignments ahead of the deadline
  - Ensure your submissions work as intended

Project Regrade Policy

- After grades posted, **regrade form** open for one week
  - We'll distribute regrade forms via **Piazza**
- Valid regrade requests:
  - You have verified your solution is correct (i.e., we made an error in grading)
- Requests that will be **rejected**:
  - My code crashed, but I've now fixed it
  - I am looking for more partial credit
  - I submitted late without an extension
  - I missed the regrade request deadline
- Your responsibility to stay atop of this!

Project Collaboration Policy

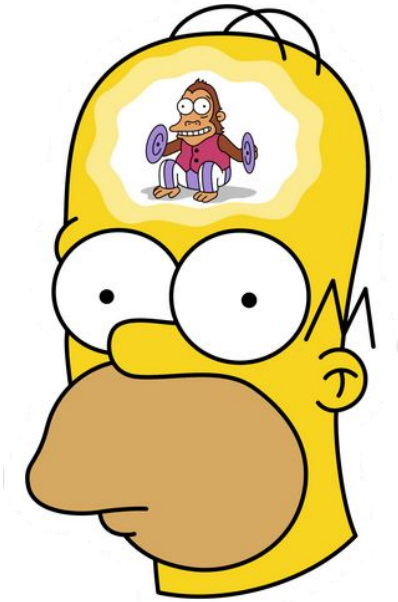
- We encourage you to help each other learn!
  - You may give or receive help on key **high-level concepts**
- However, **all code** must only be written by **you or your team**
- Cheating is when you give/receive an **unfair advantage**. Examples:
  - **Distributing your solutions** (e.g., to GitHub, Chegg, CourseHero) = **cheating**
  - **Copying code/solutions** (e.g., from GitHub, Google, another team) = **cheating**
  - **Copying code/solutions from AI tools** (e.g., CoPilot, GPT, Bard, etc.) = **cheating**
- Violations = misconduct sanctions. **Don't jeopardize your degree!**

Final Exam (35%)

- One exam covering all course material
- Questions similar to homework problems
- Final lecture will serve as a review session
- **Save the date:** **1–3PM** on **Tuesday, December 10**
  - Late exams only for conflicts with other finals

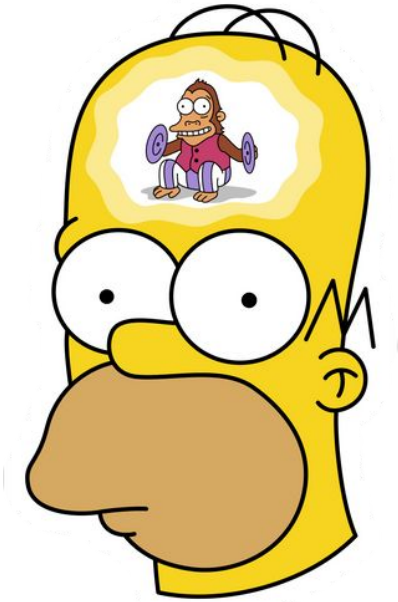
Participation (5%)

- **Lecture** participation via PollEverywhere:
  - **Three lecture absences allowed** at zero penalty
  - We'll track these internally—no need to notify us
  - Log-in as **your UMAIL** (e.g., `u8675309@utah.edu`)



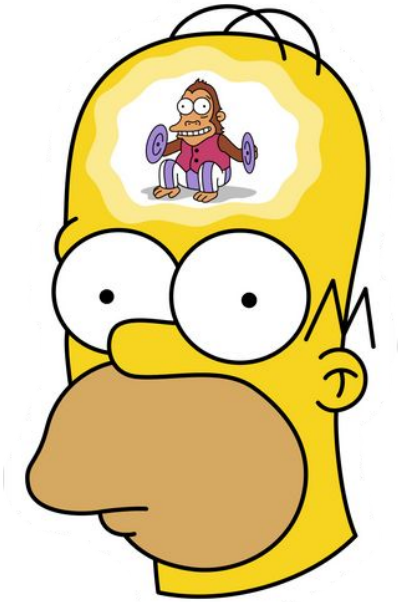
Participation (5%)

- **Lecture** participation via PollEverywhere:
  - **Three lecture absences allowed** at zero penalty
  - We'll track these internally—no need to notify us
  - Log-in as **your UMAIL** (e.g., `u8675309@utah.edu`)
- **Online** participation on course Piazza:
  - Make intellectual contributions to help others learn
  - Collaboration policies apply—**don't share your code!**



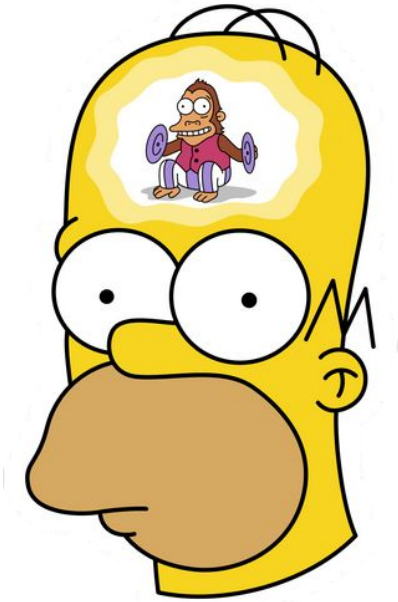
Participation (5%)

- **Lecture** participation via PollEverywhere:
  - **Three lecture absences allowed** at zero penalty
  - We'll track these internally—no need to notify us
  - Log-in as **your UMAIL** (e.g., `u8675309@utah.edu`)
- **Online** participation on course Piazza:
  - Make intellectual contributions to help others learn
  - Collaboration policies apply—**don't share your code!**
  - **New for Fall 2024:** top-10 answerers get **5pts extra credit**



Participation (5%)

- **Lecture** participation via PollEverywhere:
  - **Three lecture absences allowed** at zero penalty
  - We'll track these internally—no need to notify us
  - Log-in as **your UMAIL** (e.g., `u8675309@utah.edu`)
- **Online** participation on course Piazza:
  - Make intellectual contributions to help others learn
  - Collaboration policies apply—**don't share your code!**
  - **New for Fall 2024:** top-10 answerers get **5pts extra credit**
- How to **lose** points:
  - Frequently missing class, or not contributing online
  - Engaging in disruptive behavior or violating policies



Participation (5%)

- Lectures where attendance will NOT be graded:
  - Today's introductory lecture
  - **Week 2B** due to the football game
    - Hybrid streamed via Zoom
    - Details to come via **Piazza**
  - **Week 6B** and **Week 12B**
    - Instructor out of town
    - Guest lectures planned
  - **Week 14B** (final review lecture)

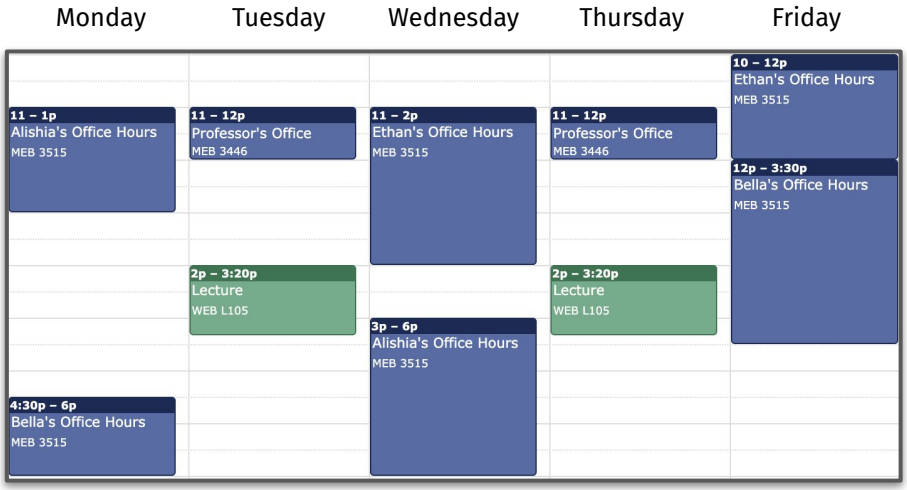
- Participation total = 23 lectures, with **three absences** dropped

Lectures

- Tuesdays and Thursdays
  - **2:00–3:20 PM** at **Warnock L105**
- Take notes!
  - Studies show most effective if hand-written 🙂
- Slides posted prior to each lecture
  - See “**Schedule**” on http://cs4440.eng.utah.edu
- Interrupt with questions, (relevant) stories
- **Not recorded—come to lectures!**

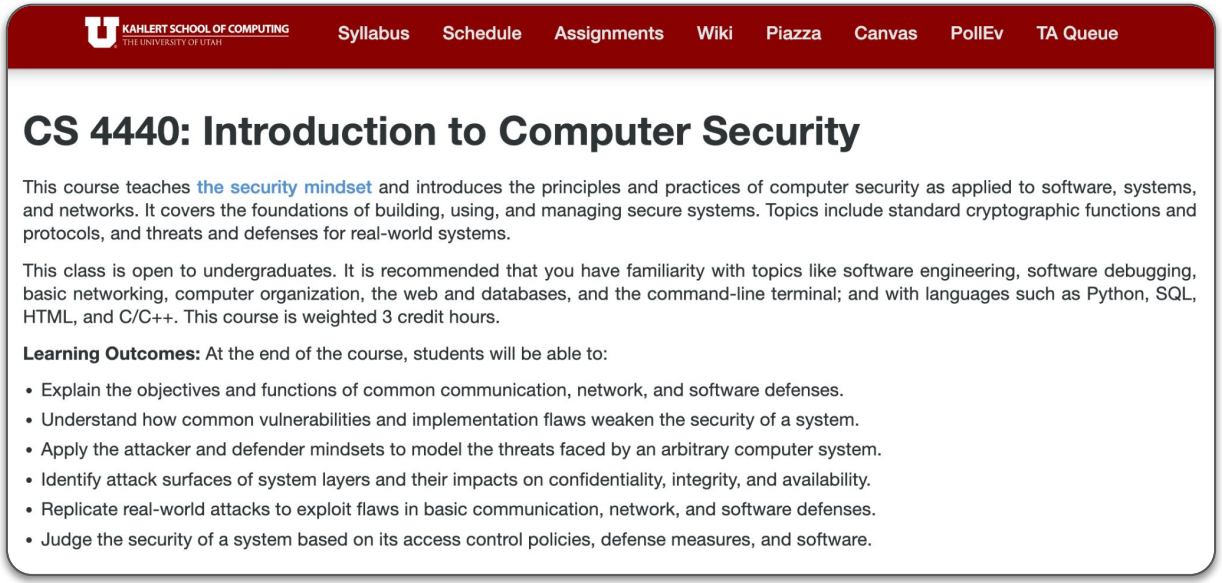
Office Hours

- TA office hours (**15 total hours**)
  - First-come/first-serve via **TA Queue**
  - Help with programming projects
- Professor's office hours
  - Help understanding lecture material
  - Administrative or grading issues
- Check the office hours calendar!
  - http://cs4440.eng.utah.edu
  - Cancellations announced via **Piazza**



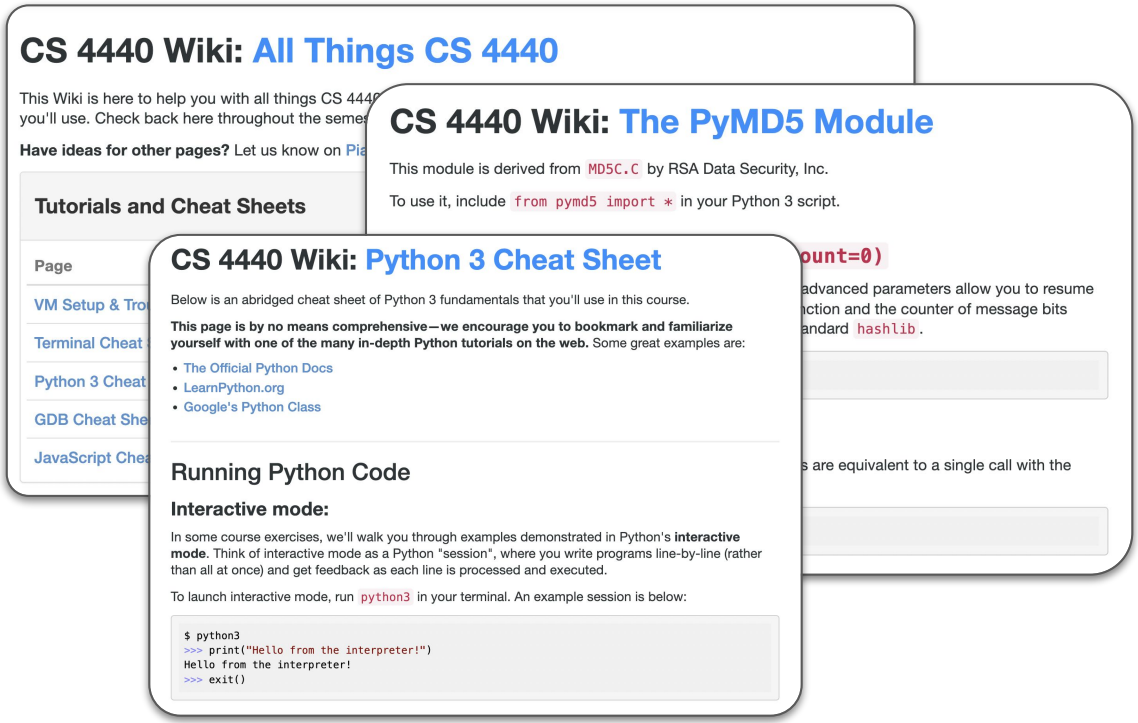
Communication

- **Course website:** your go-to resource for all things CS 4440
  - http://cs4440.eng.utah.edu



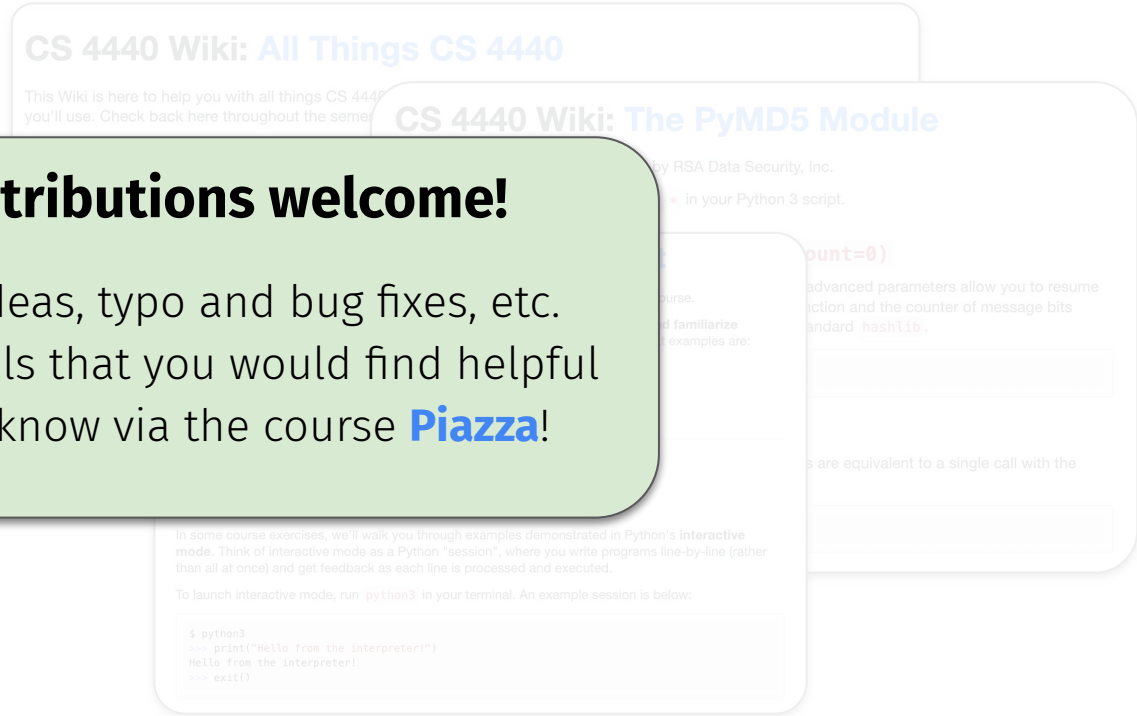
Changes in Fall 2023: The CS 4440 Wiki

- Our aim is to lower the overall learning curve
- Resources to help you:
  - Tutorials
  - Cheat Sheets
  - Software documentation
- **Fall 2024: more pages!**
  - HTML, SQL basics
  - Wireshark, Scapy
  - **Coming soon!**



Changes in Fall 2023: The CS 4440 Wiki

- Our aim is to lower the overall learning curve
- Resources to h
  - Tutorials
  - Cheat Sheets
  - Software docu
- **Fall 2024: more p**
  - HTML, SQL basics
  - Wireshark, Scapy
  - **Coming soon!**

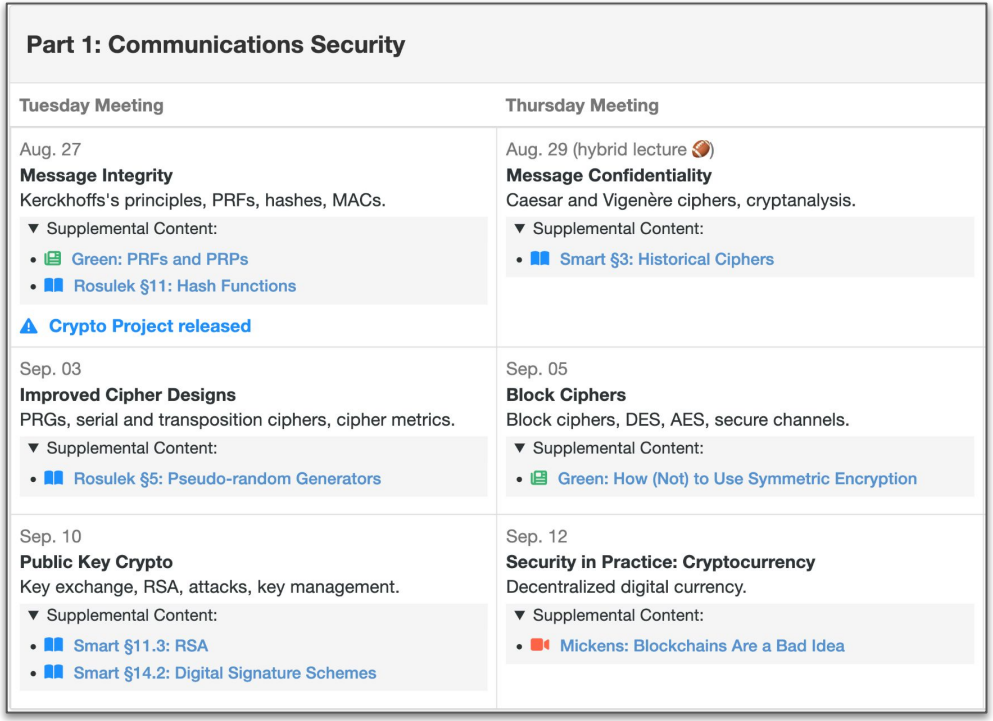


Contributions welcome!

- Page ideas, typo and bug fixes, etc.
- Tutorials that you would find helpful
- Let us know via the course **Piazza**!

New for Fall 2024: Supplemental Content

- To further help you learn, we've provided **supplemental content** relevant to every lecture topic
  - Short blog posts
  - Free textbook chapters
  - Fun podcasts or videos
- **Totally optional**—not required
  - … though we do recommend them as additional resources to lectures!
- To access, click the drop-down "▸" button beside each lecture



Part 1: Communications Security

Tuesday Meeting	Thursday Meeting
Aug. 27 **Message Integrity** Kerckhoffs's principles, PRFs, hashes, MACs. ▼ Supplemental Content: • Green: PRFs and PRPs • Rosulek §11: Hash Functions ⚠ Crypto Project released	Aug. 29 (hybrid lecture 🏈) **Message Confidentiality** Caesar and Vigenère ciphers, cryptanalysis. ▼ Supplemental Content: • Smart §3: Historical Ciphers
Sep. 03 **Improved Cipher Designs** PRGs, serial and transposition ciphers, cipher metrics. ▼ Supplemental Content: • Rosulek §5: Pseudo-random Generators	Sep. 05 **Block Ciphers** Block ciphers, DES, AES, secure channels. ▼ Supplemental Content: • Green: How (Not) to Use Symmetric Encryption
Sep. 10 **Public Key Crypto** Key exchange, RSA, attacks, key management. ▼ Supplemental Content: • Smart §11.3: RSA • Smart §14.2: Digital Signature Schemes	Sep. 12 **Security in Practice: Cryptocurrency** Decentralized digital currency. ▼ Supplemental Content: • Mickens: Blockchains Are a Bad Idea

New for Fall 2024: Free Online Textbooks

- We now make available several **freely-distributed textbooks** via the **Wiki**
  - Some textbook chapters are referenced as lecture-relevant Supplemental Content
  - Also **totally optional**—they are meant only as additional resources to help you learn

Recommended Textbooks

Textbook	Author(s)
An Introduction to Computer Networks	Peter L Dordal
Computer Networks: A Systems Approach	Bruce Davie, Larry Peterson
Computer Systems Security: Planning for Success	Ryan Tolboom
Cryptography: An Introduction	Nigel Smart
Software Security: Principles, Policies, and Protection	Mathias Payer
The Joy of Cryptography	Mike Rosulek

Summary

Course website ..... wiki, assignments, schedule, slides, office hours

Piazza ..... questions, discussion, announcements

PollEverywhere ..... lecture participation

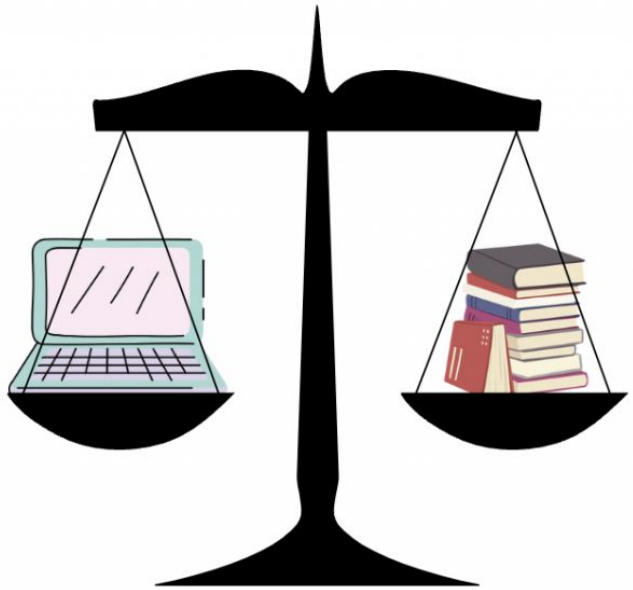
Canvas ..... quizzes, project submission, course gradebook

Instructor email (snagy@cs.utah.edu**)** ..... administrative issues

Questions?

The Security Mindset

What does Computer Science impact?



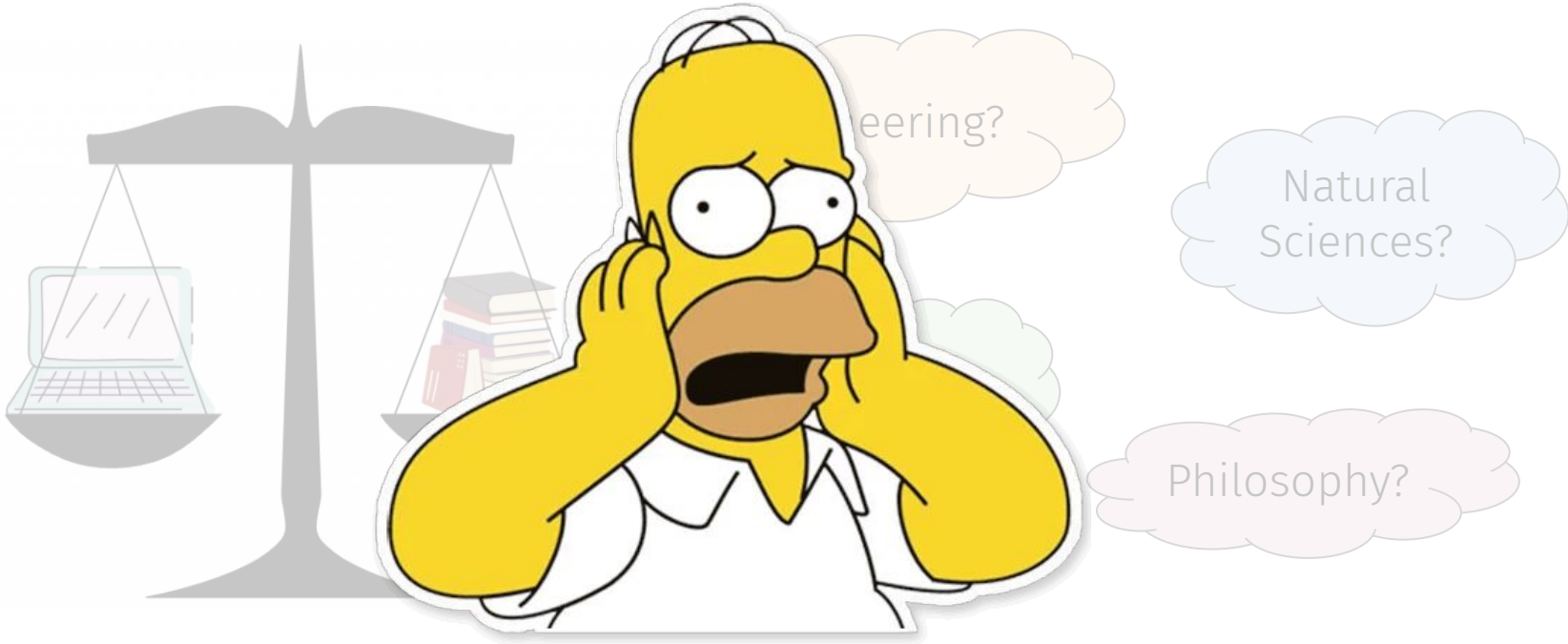
Engineering?

Natural Sciences?

Math?

Philosophy?

What does Computer Security impact?



Computers Nowadays

- … are like wheelbarrows of orangutans
  - Think of every app, user, file as an orangutan
- What could go wrong?

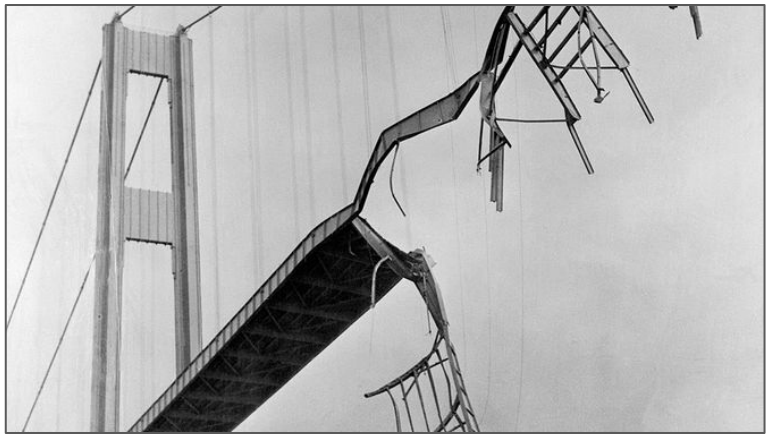
Computers Nowadays

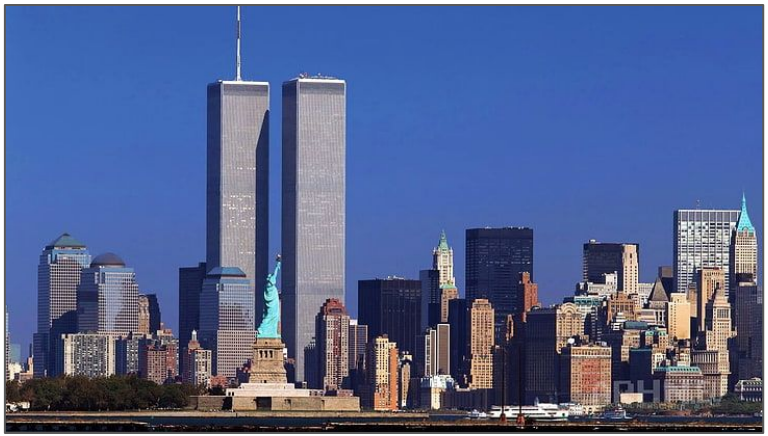
- … are like wheelbarrows of orangutans
  - Think of every app, user, file as an orangutan
- What could go wrong?
  - One might try to **throw another one off**
  - One is probably trying **to spy on another**
  - One **will bite you** and **steal your credit card**

Computers Nowadays

- … are like wheelbarrows of orangutans
  - Think of every app, user, file as an orangutan
- What could go wrong?
  - One might try to **throw another one off**
  - One is probably trying **to spy on another**
  - One **will bite you** and **steal your credit card**
- **Call to action:** let's adjust our thinking based on the possibility of such attacks
  - How we design new systems
  - How we permit user interaction
  - How we store sensitive information

What's the difference?





Reliability does not equal **Security**

Meet the Adversary

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

- The adversary…
- a.k.a. the attacker
- a.k.a. the bad guy
- An intelligence that actively tries to cause the system to misbehave.

Know thine Enemy

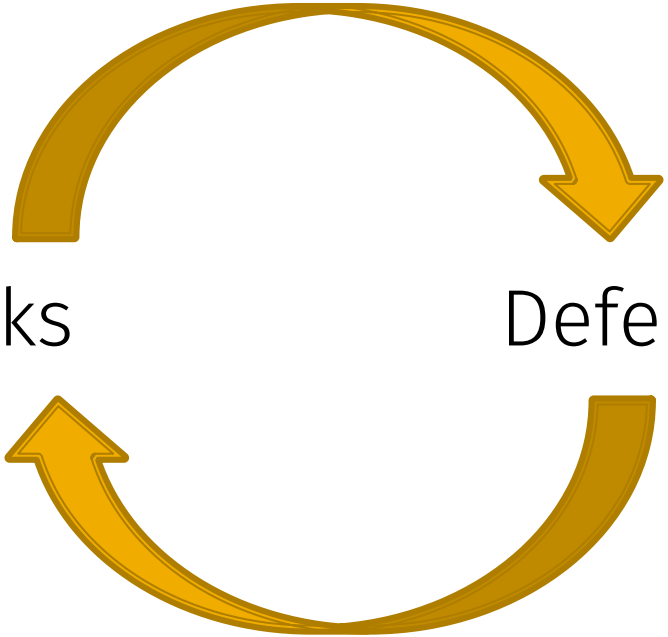
- Motives?
  - Disruption
  - Espionage
  - Money
- Capabilities?
  - Denial of service
  - Code execution
- Degree of access?
  - Physical access
  - Root privileges

The Security Mindset

- Thinking like a defender
  - Know what you're defending, and against whom
  - Weigh benefits vs. costs:
    No system is ever completely secure.
  - Embrace "rational paranoia"

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

High-level Approaches

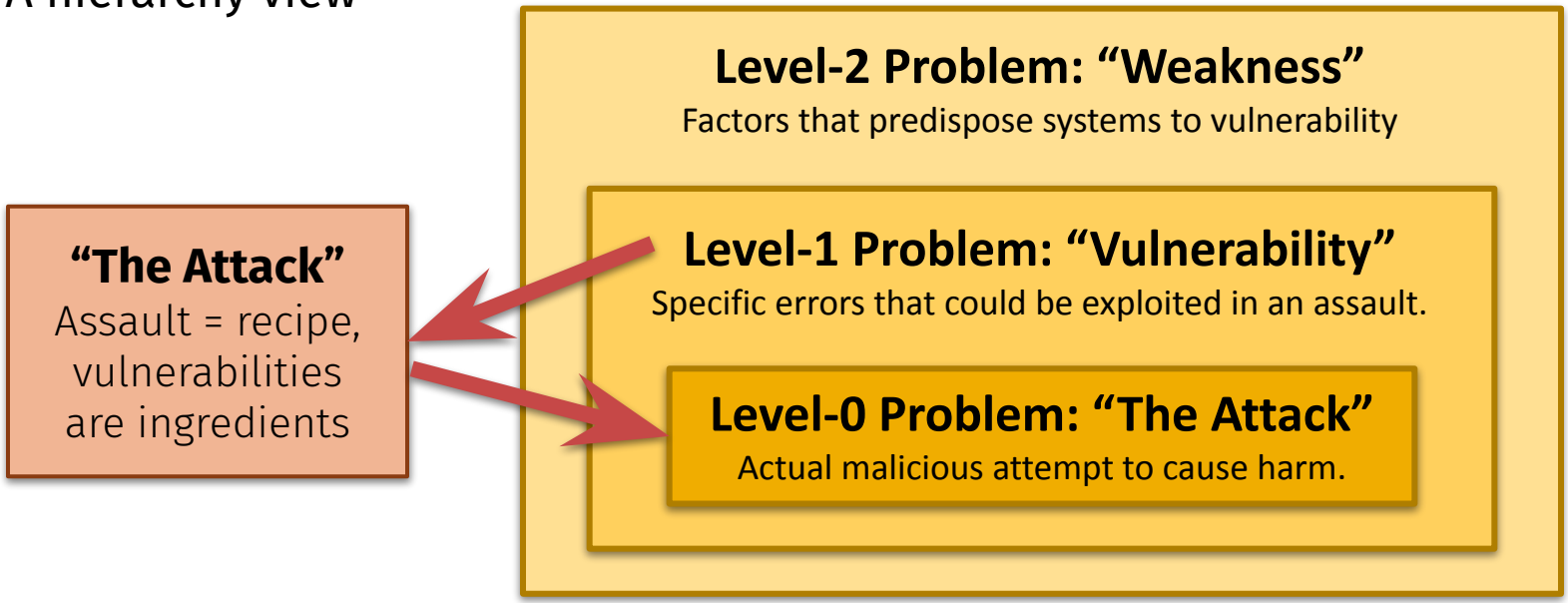


Why study attacks?

- Identify vulnerabilities so they can be fixed
- Create incentives for vendors to be careful
- Learn about **new classes** of threats
  - Determine what we need to defend against
  - Help designers build stronger systems
  - Help users more accurately evaluate risk

“Insecurity”

- A hierarchy view



Thinking like an Attacker

- Look for the weakest links
  - What is easiest to attack
- Identify assumptions that the security depends on
  - Are any assumptions false?
  - Can you render them false?
- **Think outside the box!**
  - Don't be constrained by the system designer's worldview

Practice thinking like an attacker:

For every system you interact with, think about what it means for it to be secure, and **imagine how it could be exploited**

Exercise

- What are some security systems that you interact with in everyday life?

Exercise

- What are some security systems that you interact with in everyday life?
  - Breaking into a University professor's office after hours to alter your grade?

Thinking as a Defender

- Security policy
  - What are we trying to protect?
  - What properties are we trying to enforce?
- Threat model
  - Who are the attackers? Capabilities? Motivations?
  - 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?

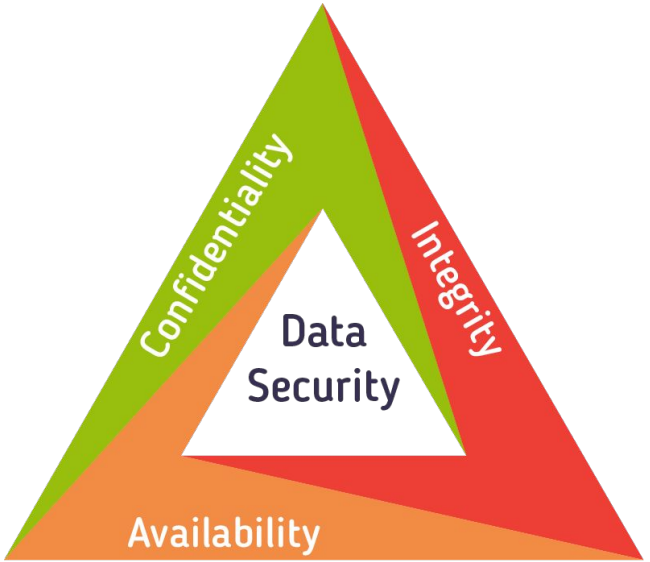
The challenge is to think rationally and rigorously about risk.

Rational paranoia.

Security Policies

- What assets are we trying to protect?

- What properties are we trying to enforce?
  - **C**onfidentiality
  - **I**ntegrity
  - **A**vailability
  - **P**rivacy
  - **A**uthenticity



Threat Models

- Who are our adversaries?
  - Motives?
  - Capabilities?
  - Level of access?
- What kinds of attacks must we prevent?
  - Think like the attacker!
- Limits: kinds of attacks we should ignore?
  - Unrealistic versus unlikely

Security through... obscurity?

Common mistake:

- Trying to convince yourself the system is secure since attacker won't know *X*

Security through... obscurity?

Common mistake:

- Trying to convince yourself the system is secure since attacker won't know *X*

Better approach:

- Limit the assumptions that the security of your system depends upon
- Assume the attacker knows everything but a *small* bit of data (e.g., a key)

Assessing Risk

- Remember: *Rational* paranoia

- What would security breaches cost us?
  - Direct: money, intellectual property, safety
  - Indirect: reputation, future business, well being

- How likely are these costs?
  - Probability of attacks?
  - Probability of success?

Countermeasures

- Technical countermeasures
  - Bug fixes, more crypto, re-architecting, etc.
- Nontechnical countermeasures
  - Law, policy (government, institutional)
  - Procedures, training, auditing, incentives, etc.

Costs of Security

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

Exercises

Should you lock your house/room door?

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

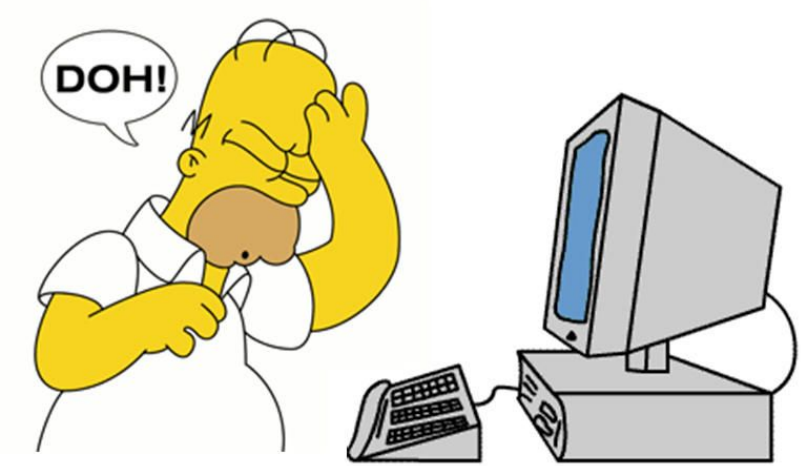
Exercises

Using a credit card safely?

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

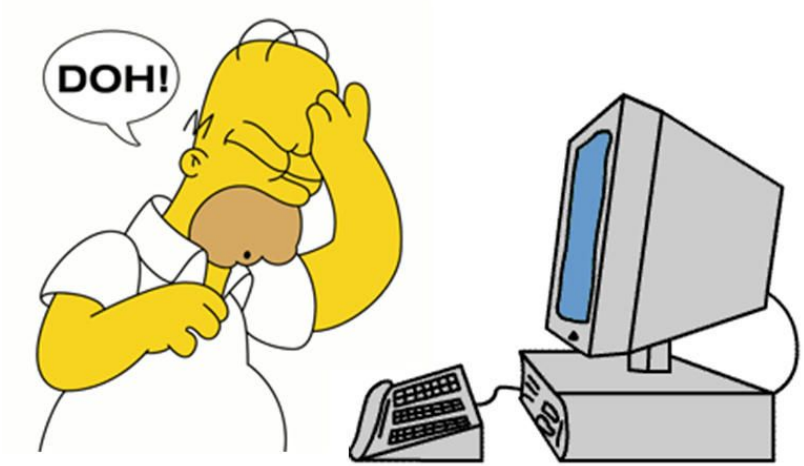
Secure Design

- Common mistake:
  - Trying to convince yourself that the system is secure as-is



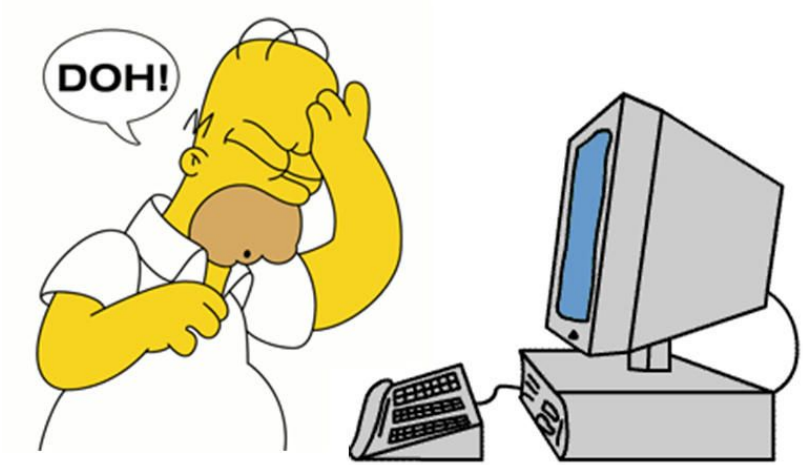
Secure Design

- Common mistake:
  - Trying to convince yourself that the system is secure as-is
- Better approach:
  - Identify the weaknesses of your design and focus on correcting them



Secure Design

- Common mistake:
  - Trying to convince yourself that the system is secure as-is
- Better approach:
  - Identify the weaknesses of your design and focus on correcting them
- Secure design is a process
  - Must be practiced continuously
  - Very difficult to be retrofitted

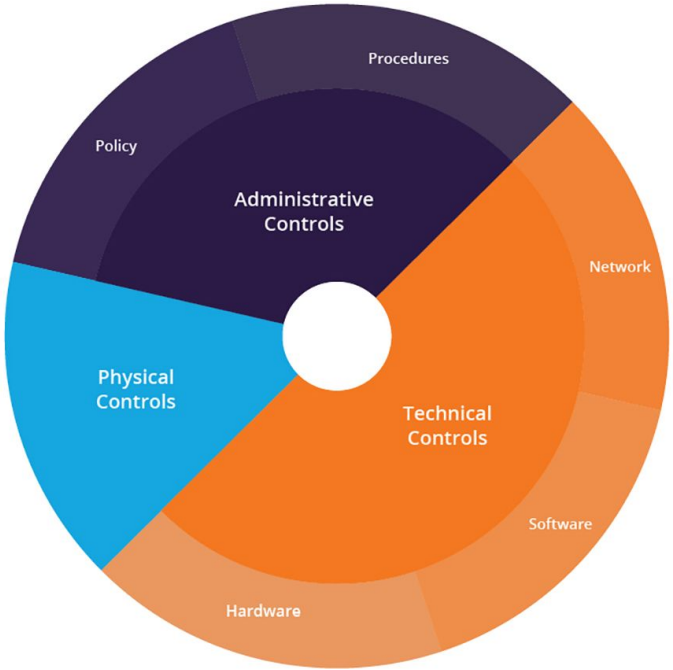


Where to Focus Defenses

- Trusted components (aka Trusted Computing Base)
  - Parts that must function correctly for the system to be secure.
- Attack surface
  - Parts of the system exposed to the attacker
- **Complexity versus security** are inversely related

Other Principles

- Defense-in-Depth
  - Multiple layers of safeguards
  - Physical, technical, administrative
- Diversity
  - More moving parts = harder to attack
  - Conversely, harder to secure
- Maintainability
  - Minimize maintainer workload
  - Make fixes easy/fast to deploy

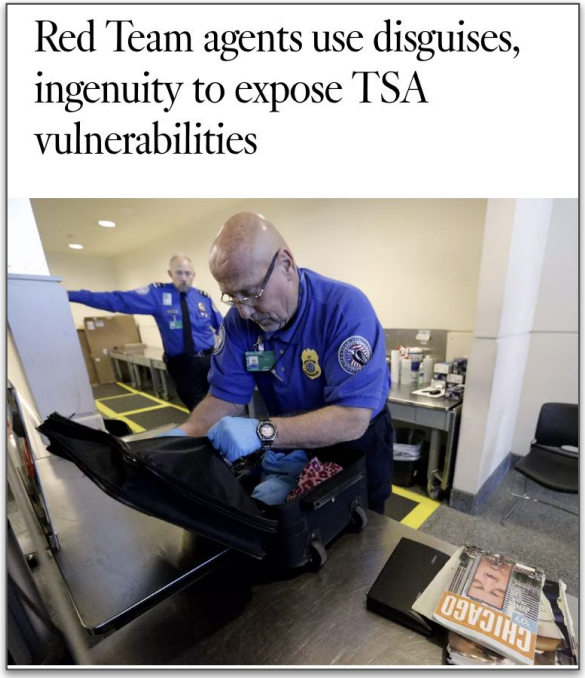


Exercise

- Preventing cheating on the CS 4440 Final Exam?

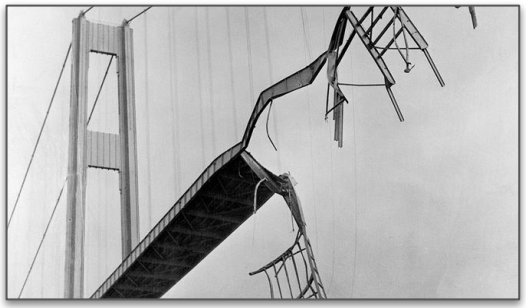
Security Testing

- Testing against requirements
  - What you have done up until now?
  - What are the correct requirements?
- Adversarial testing (my work)
  - Black-box testing
  - White-box testing
  - Gray-box testing
- Example: airport security



Learning from Failures

- Time-honored engineering practice
  - Especially important in security
- Identifying causes of failures
  - Where, how, why
  - First step of fixing
- What can failure teach us?
  - New kinds of attacks
  - New kinds of defenses



Questions?

A Note on Ethics...

Laws and Ethics

- **Don't be evil!**
  - Ethics requires you to refrain from doing harm
  - Always respect privacy and property rights
  - Otherwise, you will fail the course (and worse)
- Federal/state laws criminalize computer intrusion, wiretapping, or other abuse
  - Computer Fraud and Abuse Act (CFAA)
  - You can be sued or go to jail
- University policies prohibit tampering with campus or other systems
  - You can/will be **disciplined** and even **expelled**

Questions?

Next time on CS 4440...

Python Tutorial and Course VM Setup

Bring your laptops, and pre-download your VM image!