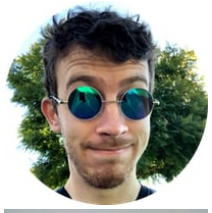
CSE 127:
Introduction to Security

Deian Stefan

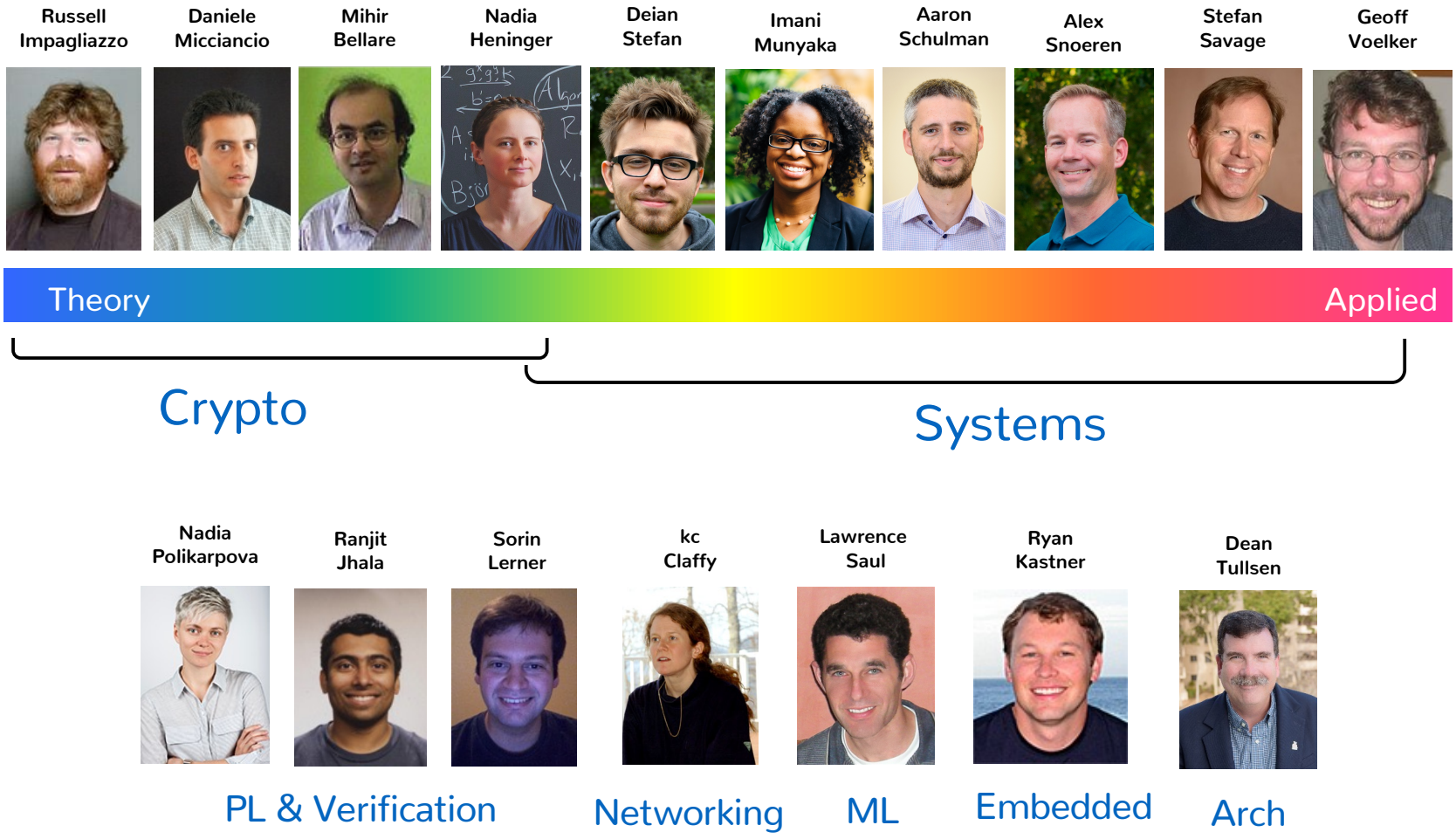
UCSD

Fall 2021 Lecture 1



- Instructor: Deian Stefan deian+cse127@cs.ucsd.edu
  - Office Hours: Mon 1:30-2:30pm
- TA: John Renner
  - Office Hours: Wed 1:30-2:30pm
- TA: Evan Johnson
  - Office Hours: Tue 3-4pm
- TA: David Thien
  - Office Hours: Thu 4-5pm
- TA: Zijie Zhao
  - Office Hours: Fri 10-11am

Many amazing folks at UCSD working on security



My group's research

Memory safety and sandboxing (MS-Wasm, RLBox, Swivel)

Practical verification for security (VeRA, IODINE, VeriWasm)

Bugfinding for browsers and runtime systems (Sys, SafeV8)

Constant-time programming (CT-Wasm, FaCT, CTFP)

Web security and privacy

Security foundations

We focus on real-world impact

Securing Firefox with WebAssembly

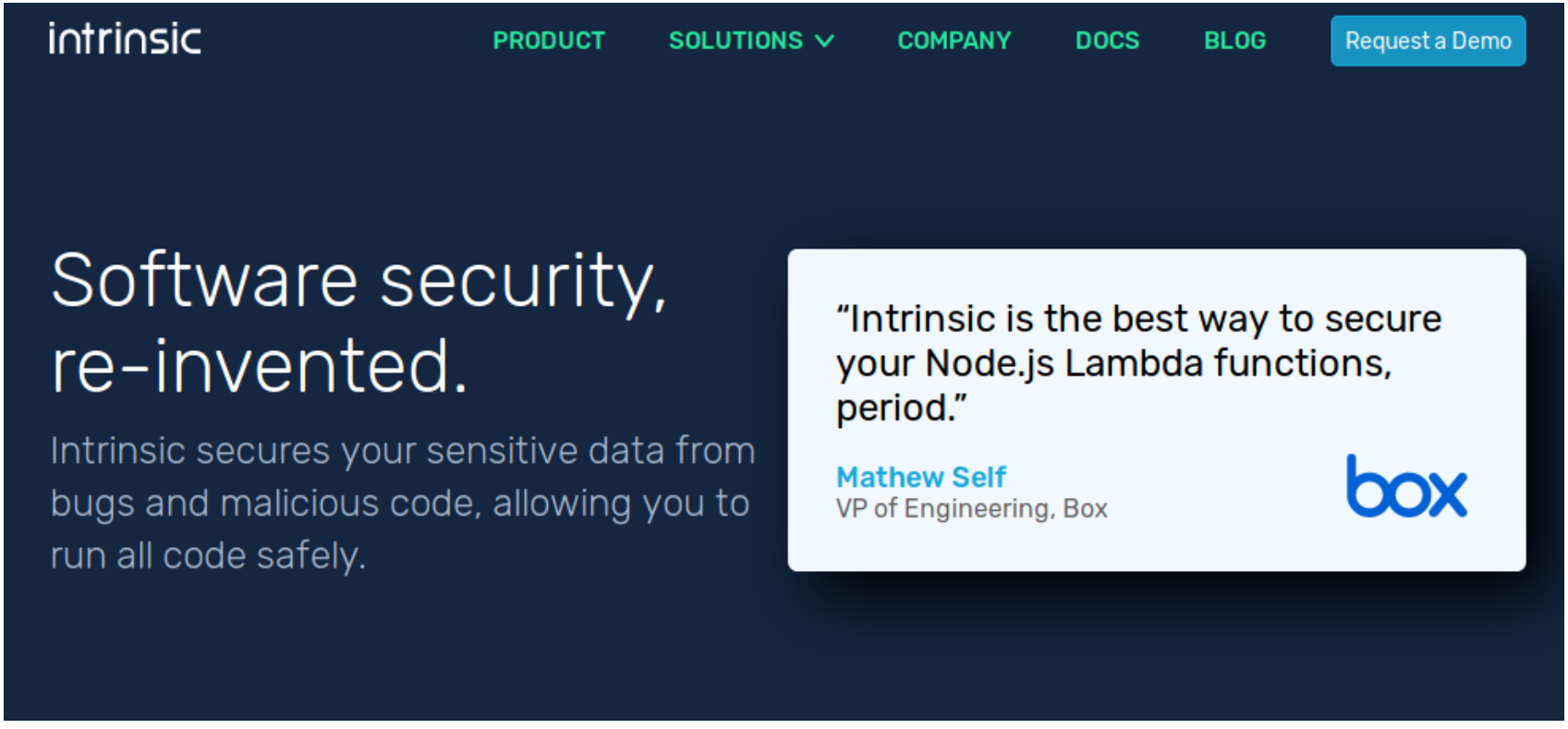
By Nathan Froyd

Posted on February 25, 2020 in Featured Article, Firefox, Rust, Security, and WebAssembly

Protecting the security and privacy of individuals is a central tenet of Mozilla's mission, and so we constantly endeavor to make our users safer online. With a complex and highly-optimized system like Firefox, memory safety is one of the biggest security challenges. Firefox is mostly written in C and C++. These languages are notoriously difficult to use safely, since any mistake can lead to complete compromise of the program. We work hard to find and eliminate memory hazards, but we're also evolving the Firefox codebase to address these attack vectors at a deeper level. Thus far, we've focused primarily on two techniques:

- Breaking code into multiple sandboxed processes with reduced privileges
- Rewriting code in a safe language like Rust

We focus on real-world impact



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 `leet h4x0r`, but an ethical one!

Course Mechanics

55% Eight projects

- Do your own programming and writeup
- General discussion is encouraged

Course Mechanics

55% Eight projects

- Do your own programming and writeup
- General discussion is encouraged

40% Exams

- Closed-book, cheat sheets OK
- Ressurection final

Course Mechanics

55% Eight projects

- Do your own programming and writeup
- General discussion is encouraged

40% Exams

- Closed-book, cheat sheets OK
- Ressurection final

5% Participation

- Ask/answer questions, make comments, generate discussion!

Course Mechanics

55% Eight projects

- Do your own programming and writeup
- General discussion is encouraged

40% Exams

- Closed-book, cheat sheets OK
- Ressurection final

5% Participation

- Ask/answer questions, make comments, generate discussion!

$\leq$ 10% Scribe notes

- Work in groups
- Our goal: use notes in future classes!

Course Policies

Early policy:

- Can turn in assigments 3 days early to get 10% of your grade extra credit
- No late days

Course Policies

Early policy:

- Can turn in assigments 3 days early to get 10% of your grade extra credit
- No late days

Regrade policy:

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

Course Policies

Early policy:

- Can turn in assigments 3 days early to get 10% of your grade extra credit
- No late days

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 Erikon

Course Resources

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

  `https://cse127.programming.systems`

Course Resources

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

  `https://cse127.programming.systems`
- Questions? Post to Piazza.

  `https://piazza.com/ucsd/fall2021/cse127`

Course Resources

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

  `https://cse127.programming.systems`
- Questions? Post to Piazza.

  `https://piazza.com/ucsd/fall2021/cse127`
- Lectures, section, office hours:
  - On this Zoom
  - Everything will be recorded and posted online

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 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-flags

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) overturned overly broad “exceeds authorized access” clause

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.


1
2
3
4
5
6
7
8
9
*
0
#

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?

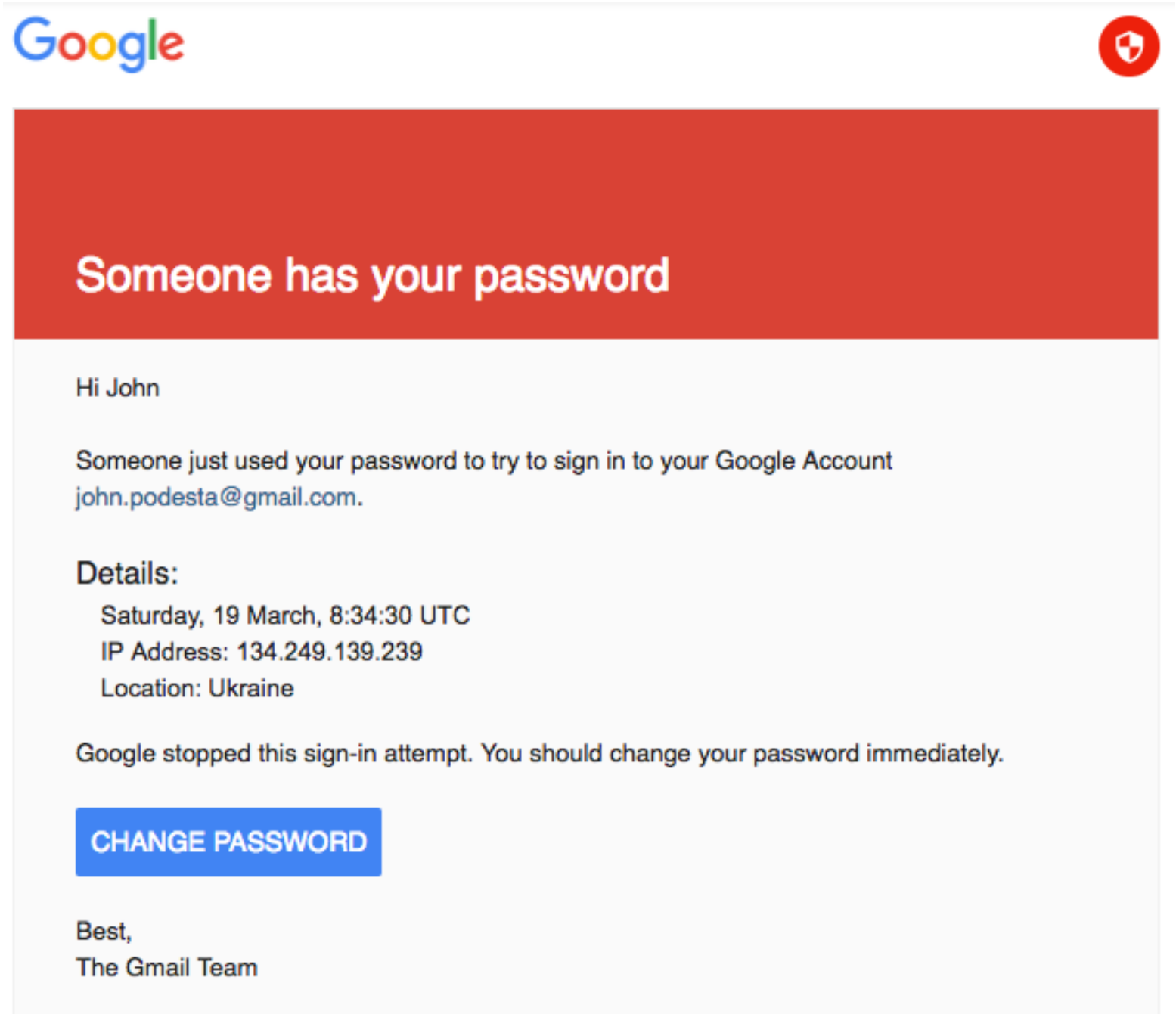
Example of Threat Modeling

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



Who is John Podesta?

Assessing Risk

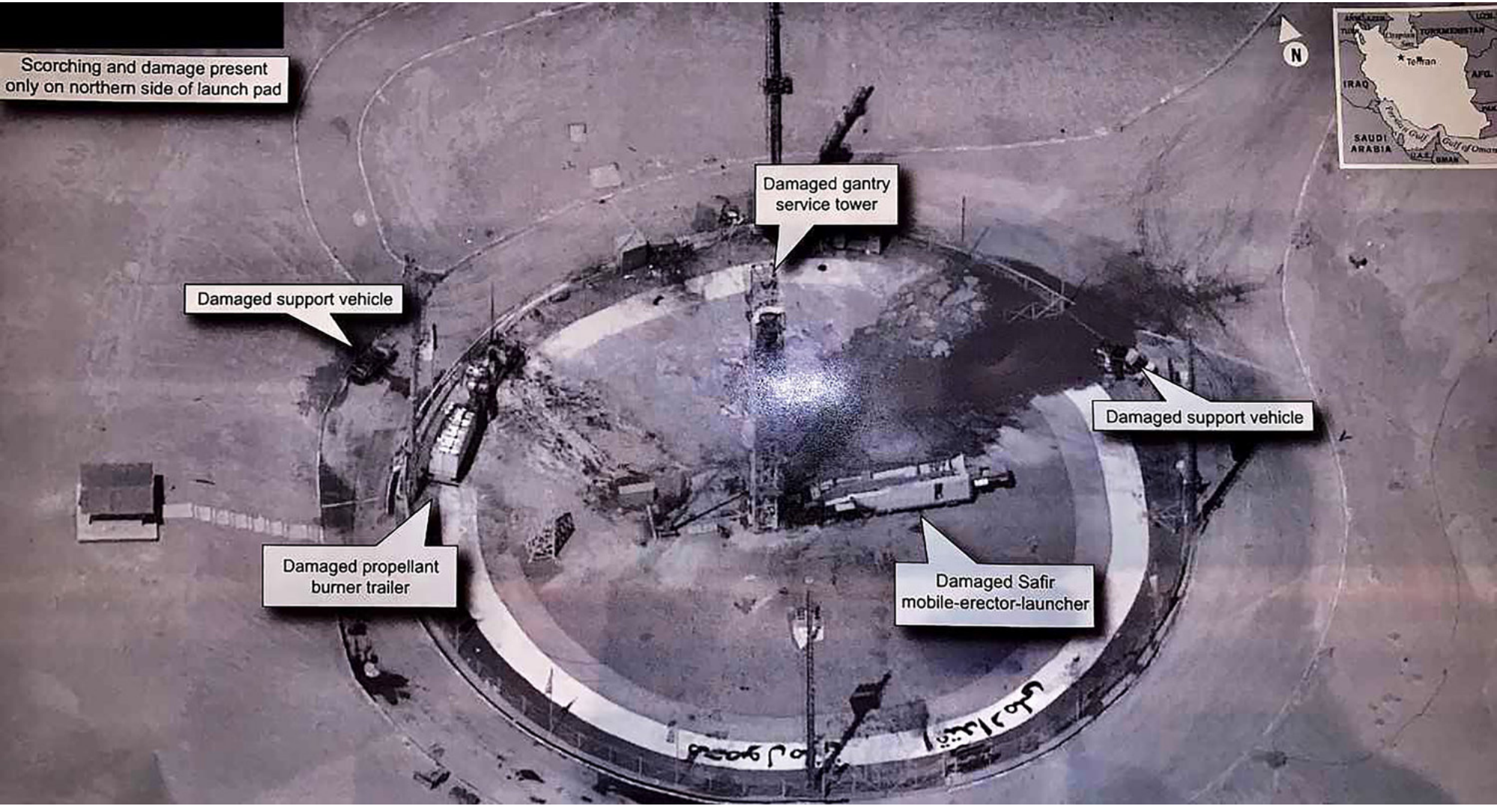
Remember: *Controlled paranoia*

- 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.

How do we protect classified satellites?



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!