ASES and Applied Cyber present

Safety and Security

Protecting your users, before and after an incident

January 13, 2023

Aditya Saligrama

Vice President, Stanford Applied Cyber
Teaching Assistant, INTLPOL 268 Hack Lab
saligrama@stanford.edu

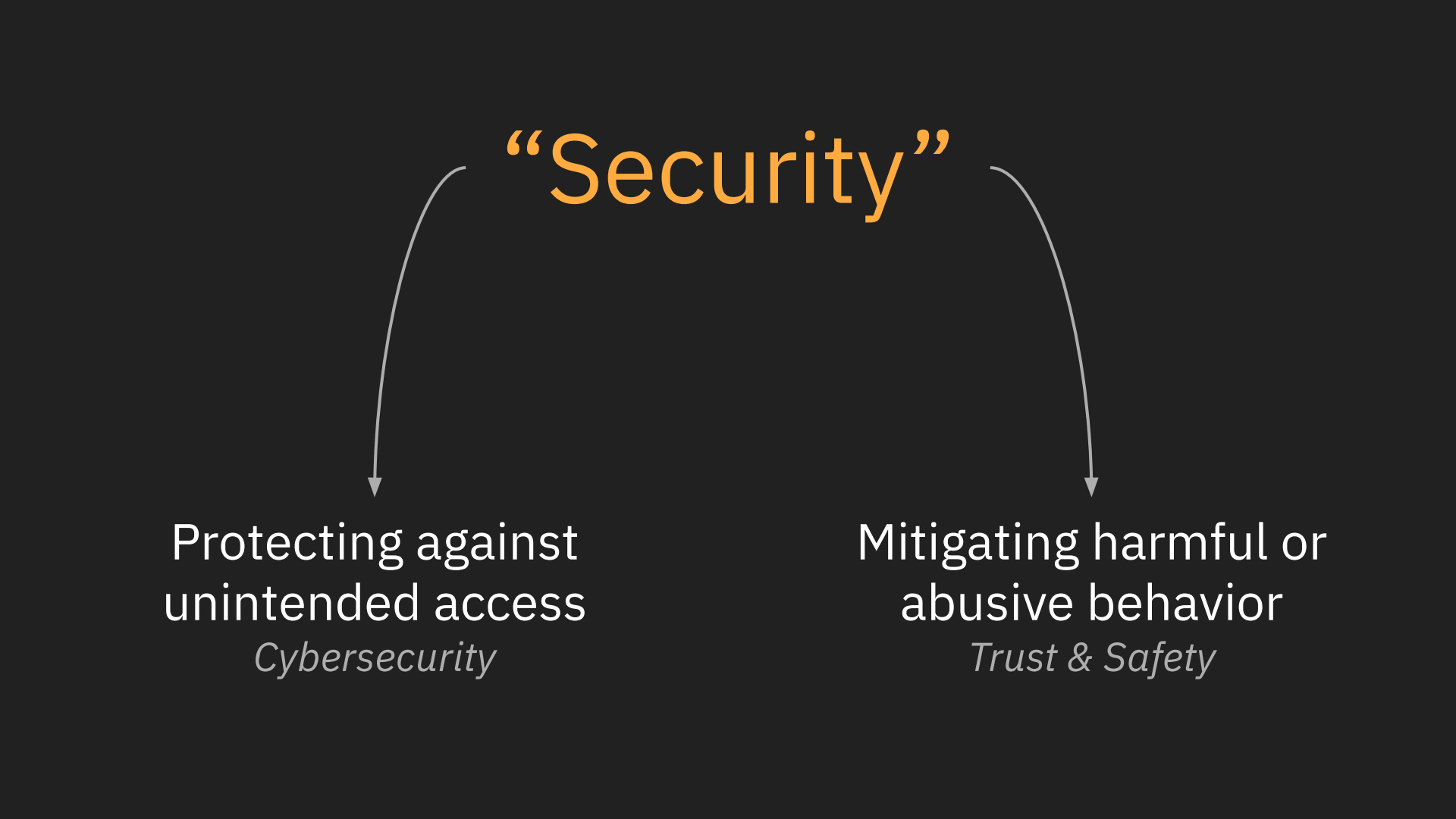
Miles McCain

Stanford Internet Observatory, Apple Privacy Engineering, CISA Election Security, CS 106S
mccain@stanford.edu

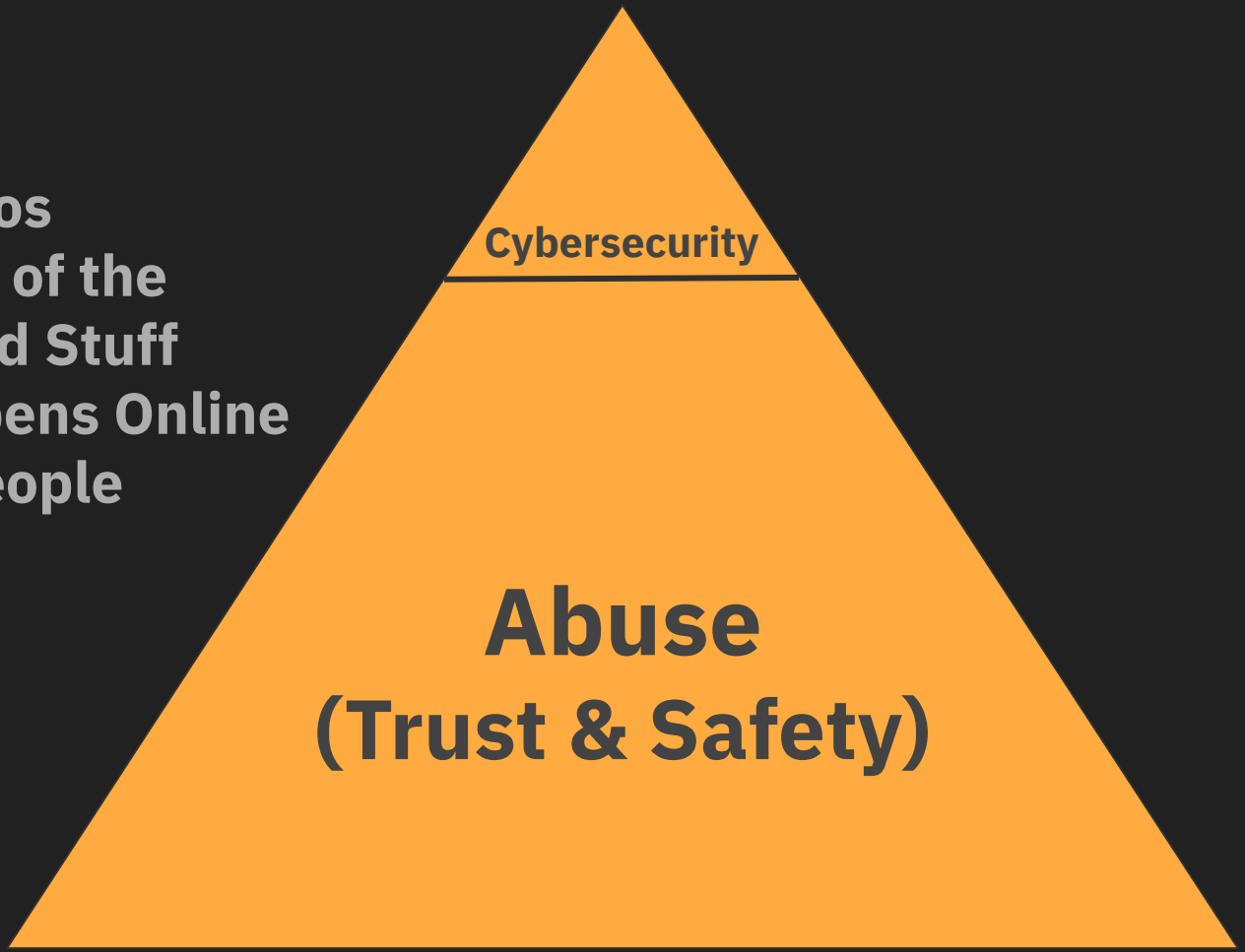
Growth without security is not sustainable growth

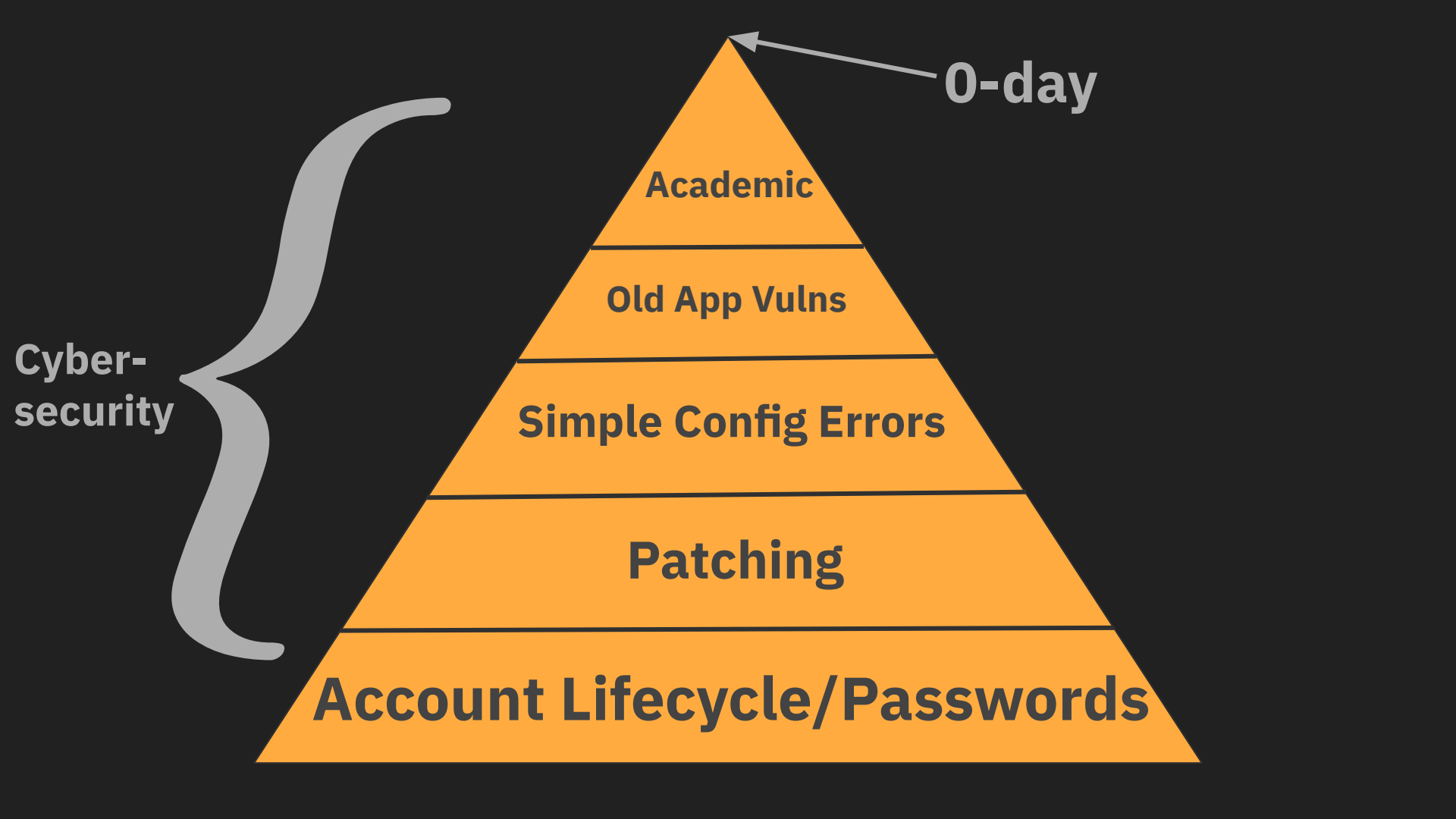
Growth without security is not sustainable growth

And security really isn’t that hard!


“Security”
Protecting against unintended access
Cybersecurity
Mitigating harmful or abusive behavior
Trust & Safety

The Stamos Hierarchy of the Actual Bad Stuff that Happens Online to Real People




0-day
Academic
Old App Vulns
Simple Config Errors
Patching
Account Lifecycle/Passwords
Cyber-
security

Cybersecurity

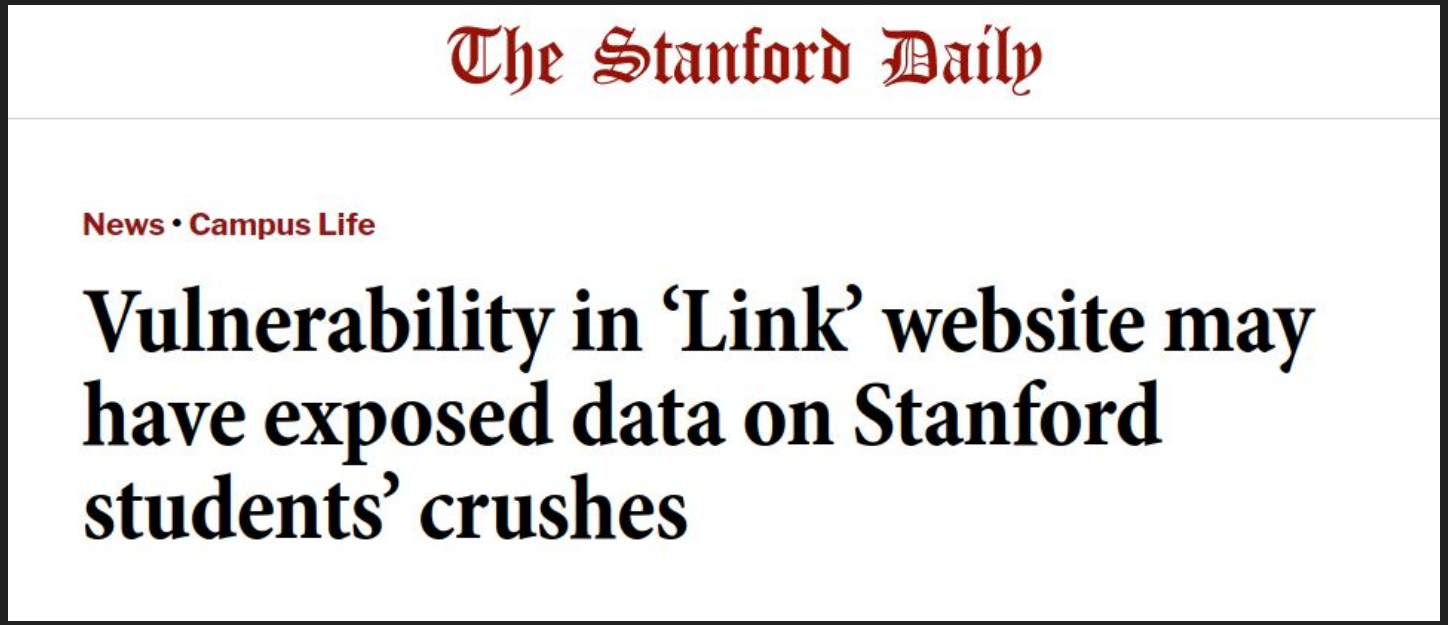
Case study: Stanford Link (2020)

- *Match with your crush if they like you back*
- *Website keeps you anonymous if they don't*

Case study: Stanford Link (2020)

- *Match with your crush if they like you back*
- *Website keeps you anonymous if they don't*
- ***What could go wrong?***

Case study: Stanford Link (2020)



The Stanford Daily

News • Campus Life

Vulnerability in 'Link' website may have exposed data on Stanford students' crushes

Case study: Stanford Link (2020)

Within days of its launch, hundreds of Stanford students signed up for Link, a website meant to connect users and their crushes. But in addition to violating University policy, the site was vulnerable to SQL injection, a kind of cyber attack, which may have compromised the data of many of them.

An anonymous individual emailed The Daily on Tuesday with what they claimed was user data from the site, attaching a spreadsheet that contained what appeared to be the names, email addresses and crushes submitted of around 100 users.

The individual also provided screenshots and a screen-recorded video depicting the alleged hack. The Daily confirmed the accuracy of a small sample of the data in the spreadsheet and video.

The site's creator, Ishan Gandhi '23, confirmed the existence of the vulnerability and said that as many as around 1,000 users' data could have been accessible at a given point in time. Asked if he thought that the alleged hacker had accessed user data, Gandhi said, "that's what it seems like, yes" but that "there's been no confirmation."

After being shown heavily redacted data, Gandhi began to walk back his previous statement. He wrote in an email after the interview that he was "genuinely skeptical" that the hacker had accessed the data because of "irregularities" in its format.

The Daily withheld reporting on the issue until Gandhi could secure the site. Gandhi says users' data is now secure and the site has since been taken offline.

Don’t lie about your security

Link wrote on Instagram that no one, not even members of Link’s team, could look at any of users’ data, with a bullet point saying “database encryption is industry-standard (Microsoft’s ‘SEAL’).”

But Gandhi told The Daily it was “technically feasible” for him — but only him — to access data in the “active” database, although such an action would have been against Link’s internal policies. According to Gandhi, those policies were written by himself and codified with his significant other.

He also said that Microsoft SEAL encryption was not actually in use on Link’s site, saying it was included as a comparison of Link’s encryption to industry-standard methods.

Gandhi said he can “categorically confirm that our databases are encrypted.” He said, however, that user data is split “across multiple databases” and that if a database is currently receiving new submissions from the website it may not have been encrypted.

“Encryption occurs once each database has hit an input limit,” Gandhi wrote, which lies around 1,000 entries, according to him. “If the database targeted was our ‘active’ database, … then it’s possible they were able to access unencrypted data.”

A page on Link’s website with their privacy policy stated that “no electronic transmission over the Internet or information storage technology can be guaranteed to be 100% secure” and that Link “cannot promise or guarantee that hackers, cybercriminals or other unauthorized third parties will not be able to defeat our security.”

Gandhi told The Daily that he believed Link had not violated their privacy policy.

1. There was no encryption on Link’s site
2. Encryption could have prevented this access
3. Why even mention Microsoft SEAL?
   It’s the wrong kind of encryption for this use
4. Databases do not work this way

Aside: don't send official-looking spam

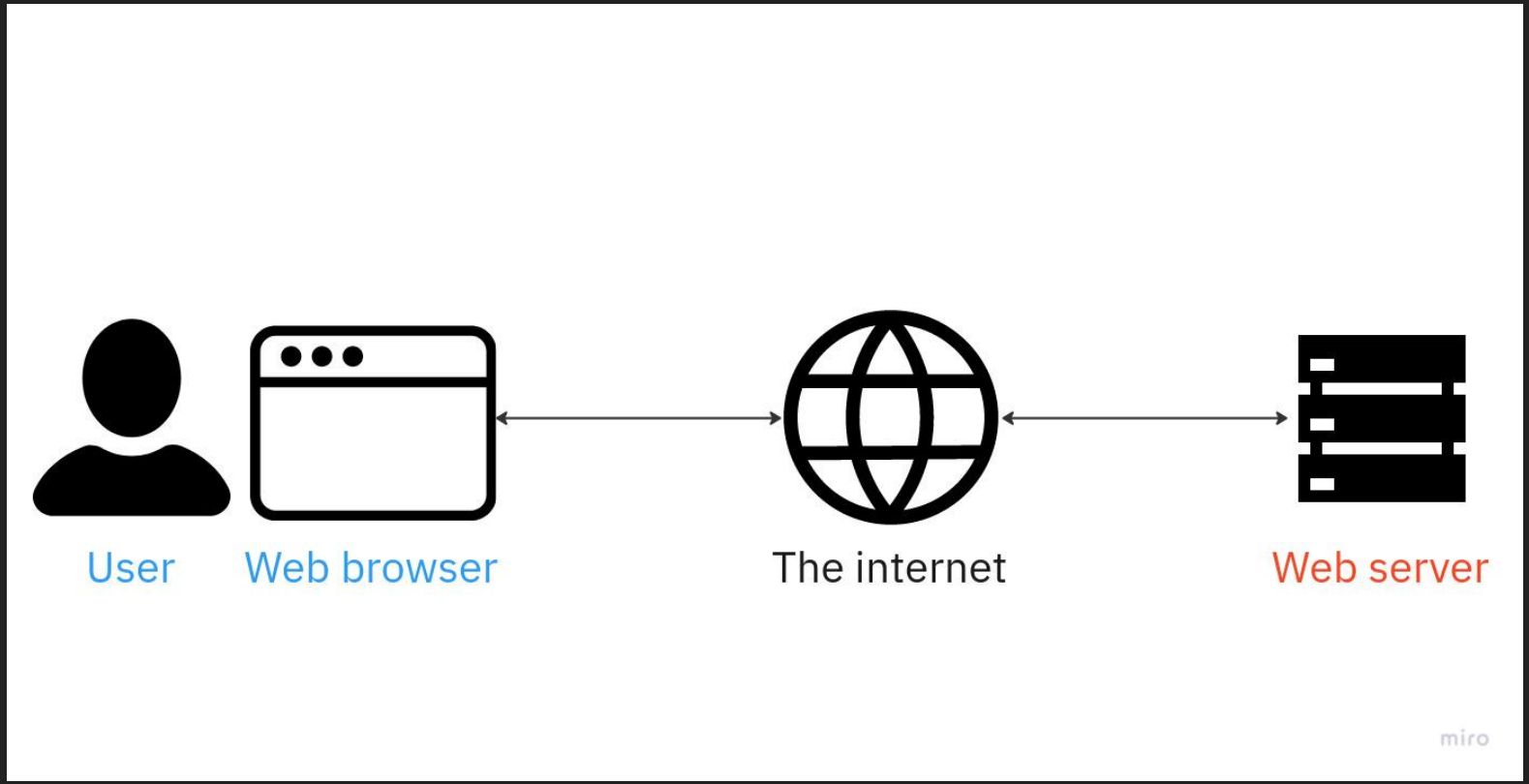
Link sent emails to students that appeared to come from "info@stanford.com." University spokesperson E.J. Miranda wrote in a statement to The Daily that it is a violation of University policy for anyone other than University officials to use the address. Miranda also wrote that the University was reviewing the incident and that infractions by students of University policies like the Computer and Network Usage Policy and the Fundamental Stanford are referred to Student Affairs for review.

"It was spoofing," Miranda wrote to The Daily in a statement about the use of the address. Spoofing involves the changing of email metadata to make a message appear to be from an address that the sender doesn't actually own.

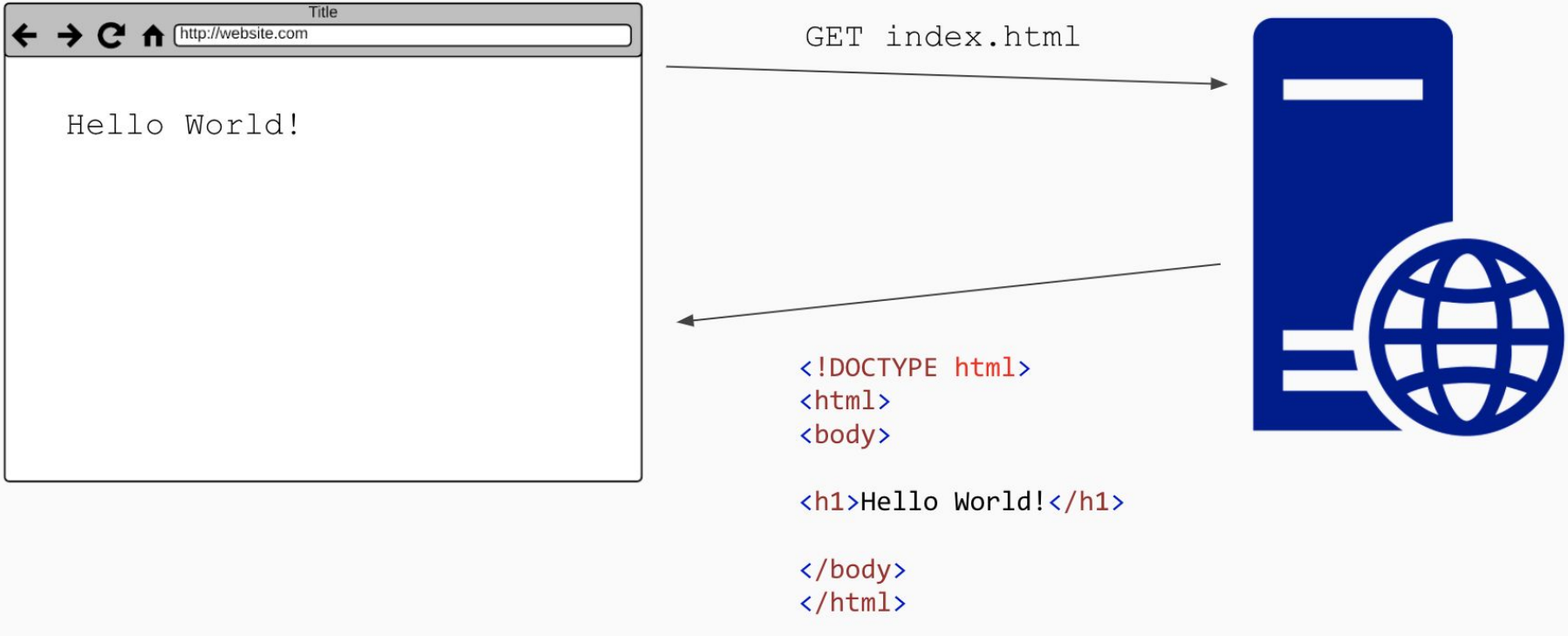
Asked about spoofing, Gandhi replied, "I'm not quite sure what you're referring to" and later told The Daily that "info@stanford.com is ours -- I'm not 100% on whether I'm happy to disclose how I got it."

The fastest web crash course ever

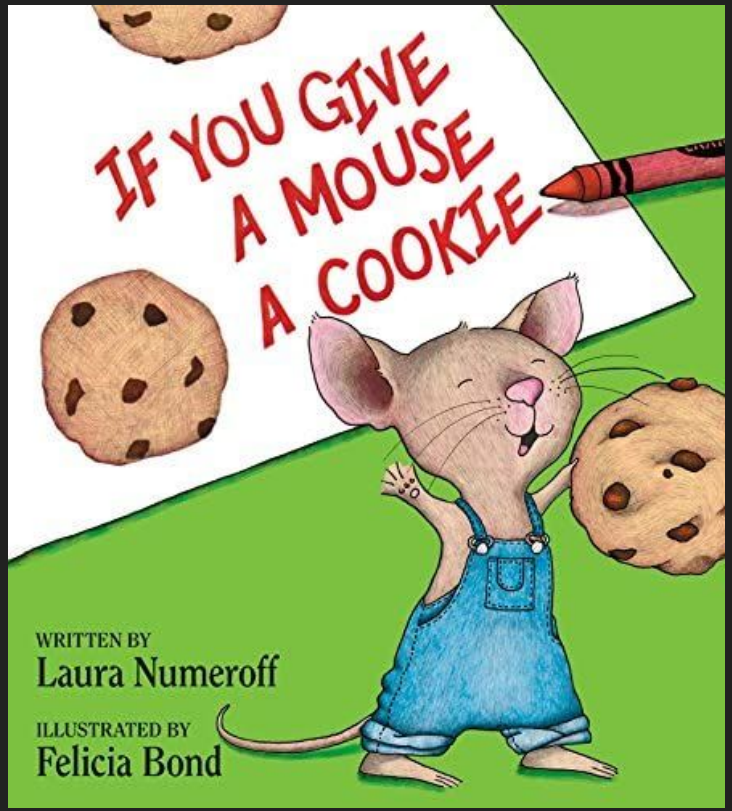
Our Internet Abstraction



HTTP: the missing language of the web



Session Handling: *How does a website remember?*

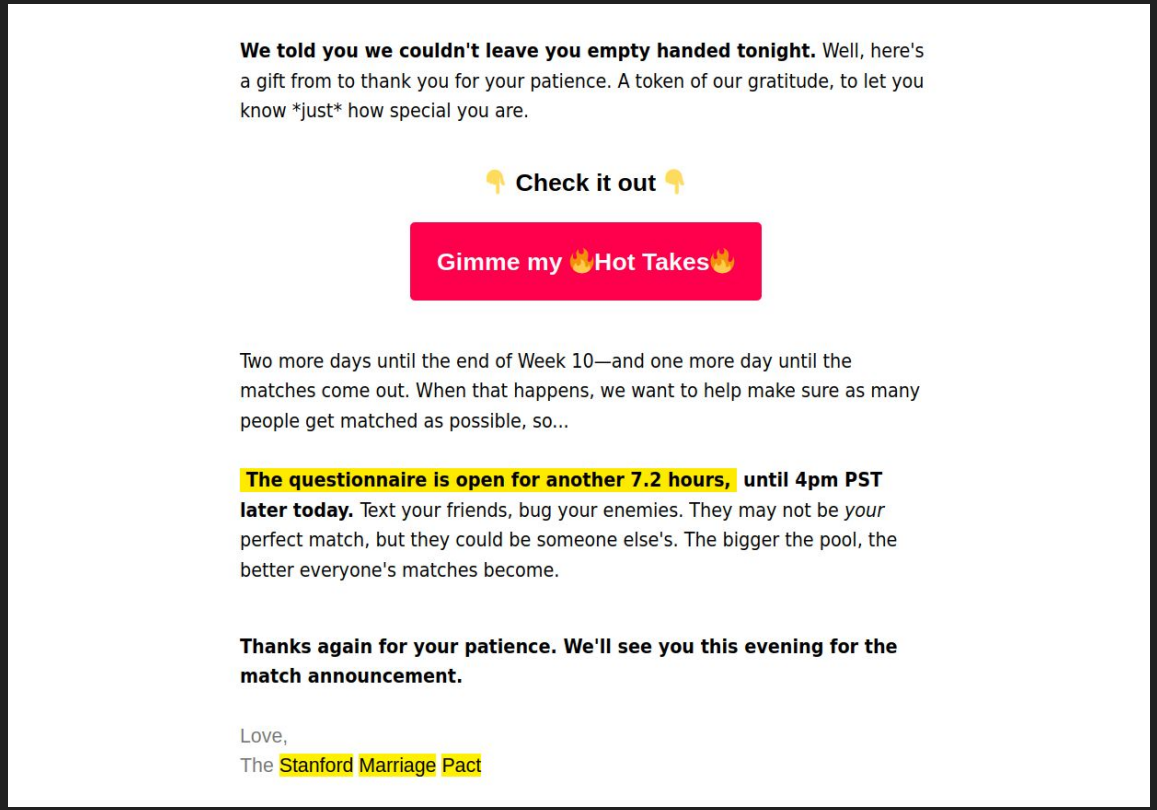


- ***Cookies!***
- Cookies enable web servers to store stateful information in your browser
- Authentication cookies are used to authenticate that a user is logged in, and with which account
  - On login: `Set-Cookie: session=session-id`

Common insecure design patterns

And ways to avoid and mitigate them

Case study: Stanford Marriage Pact (2020)



We told you we couldn't leave you empty handed tonight. Well, here's a gift from to thank you for your patience. A token of our gratitude, to let you know *just* how special you are.

👇 **Check it out** 👇

Gimme my 🔥Hot Takes🔥

Two more days until the end of Week 10—and one more day until the matches come out. When that happens, we want to help make sure as many people get matched as possible, so...

The questionnaire is open for another 7.2 hours, until 4pm PST later today. Text your friends, bug your enemies. They may not be *your* perfect match, but they could be someone else's. The bigger the pool, the better everyone's matches become.

Thanks again for your patience. We'll see you this evening for the match announcement.

Love,
The Stanford Marriage Pact

Case study: Stanford Marriage Pact (2020)

https://mp.com/**29d2223b196d87e8e9292308c074e593**

MD5

29d2223b196d87e8e9292308c074e593

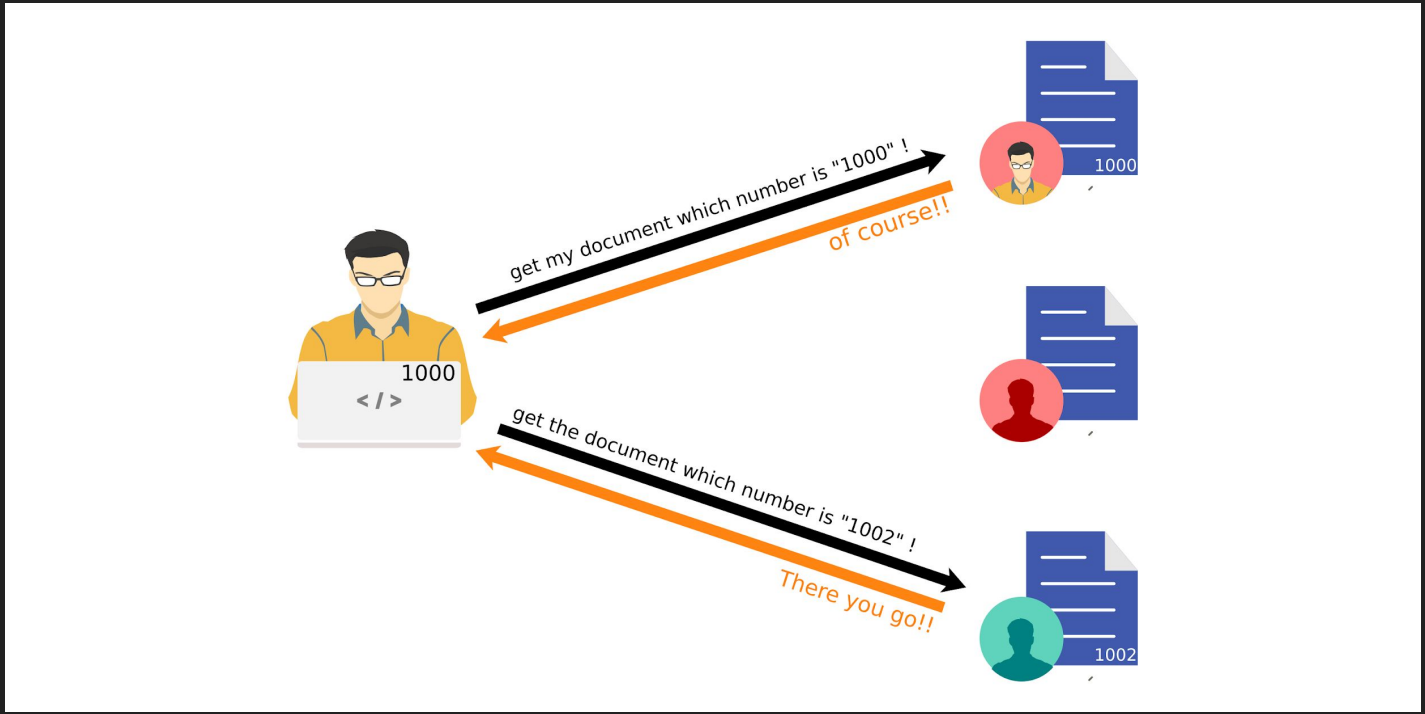
saligrama@stanford.edu

mccain@stanford.edu → MD5 → **65af214d836bb936fd32c5c11f93c70d**

https://mp.com/**65af214d836bb936fd32c5c11f93c70d**

Insecure Direct Object Reference (IDOR)

Or: asking the server for the resources you want



Avoiding IDOR

- Ensure that a user is allowed to access a resource before returning it

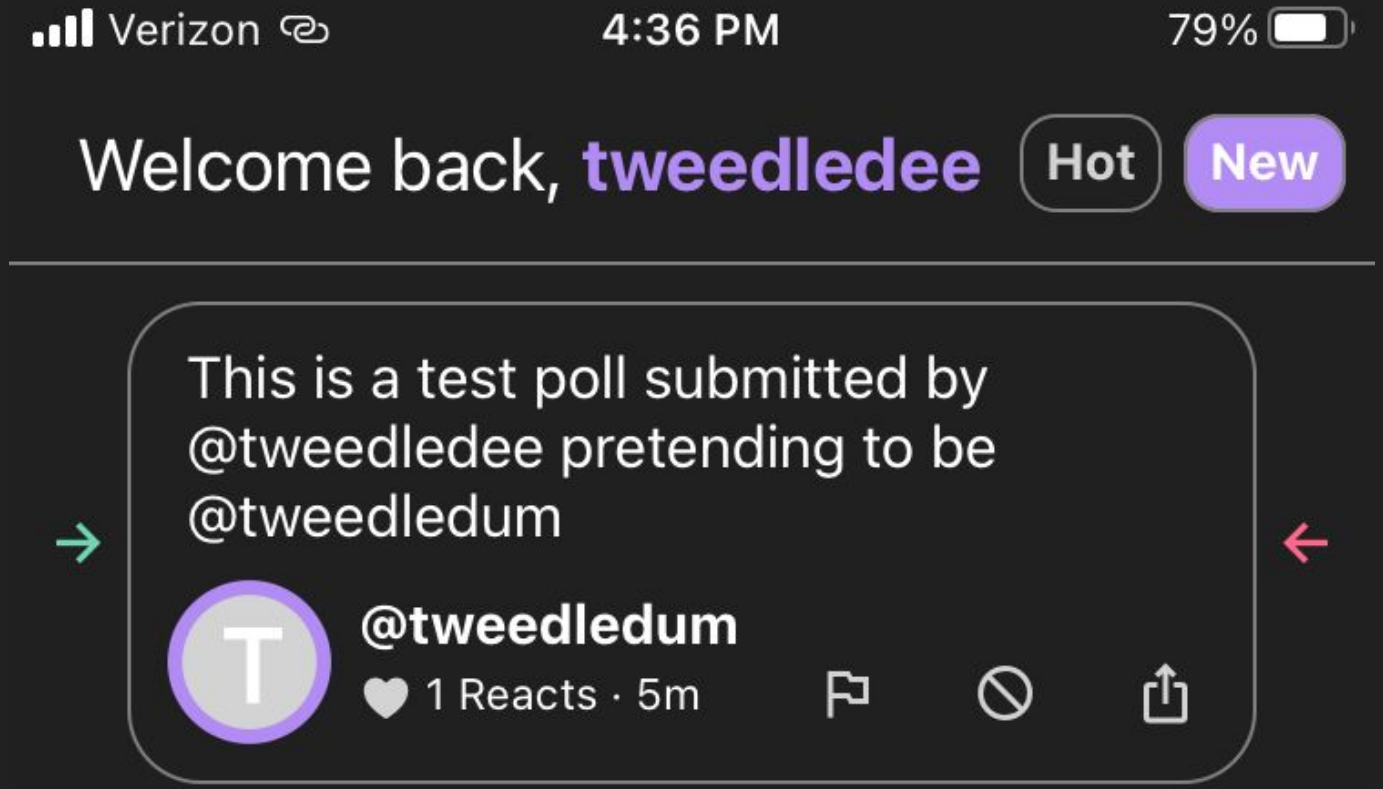
Avoiding IDOR

- Ensure that a user is allowed to access a resource before returning it

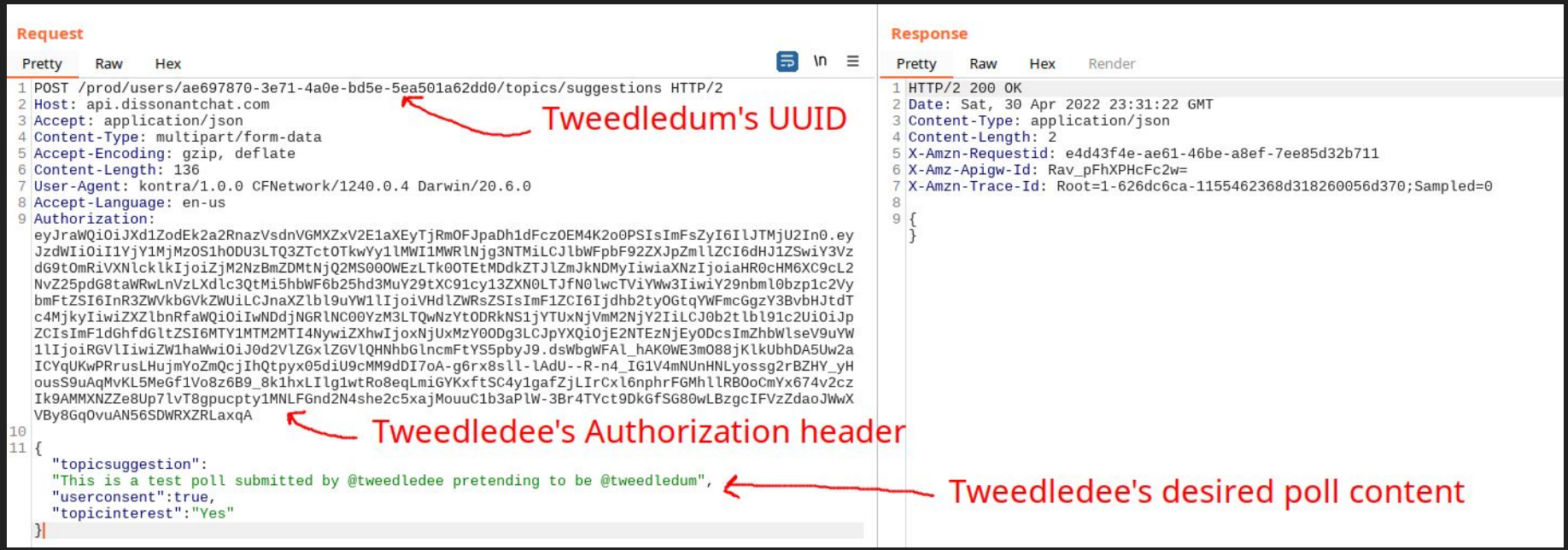
- If not possible (e.g. cloud storage buckets), then make resource URIs random and unpredictable. Avoid:
  - Automatically incrementing resource IDs
  - Hashing a guessable property such as usernames, phone numbers, or emails

- Instead: use random identifiers such as UUIDs

Case study: Kontra (2022)



Case study: Kontra (2022)



Improper session handling

Cookie not checked for authorization

- Use your own account to
  - Impersonate someone else
  - Escalate privileges to admin

Cookie itself is insecure

- Can modify cookie to access another's account
  - e.g. become admin

Consequences are IDOR-like, even when resource IDs are randomized

Avoiding improper session handling

Before taking a sensitive action:

Check the user is who they say they are

Case study: Fizz (2021)

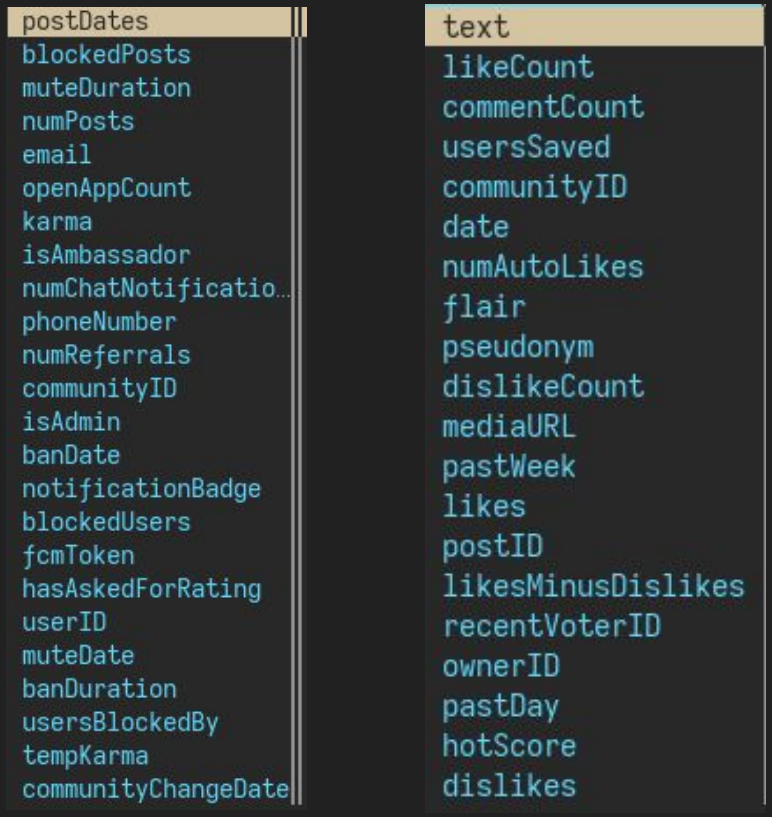
Opinions

Opinion | Fizz previously compromised its users' privacy. It may do so again.

Fizz had a large data vulnerability discovered last fall. Their response raises questions about the app today.
(Graphic: JOYCE CHEN/The Stanford Daily)

Opinion by Joyce Chen
Nov. 1, 2022, 10:00 p.m.

Case study: Fizz (2021)



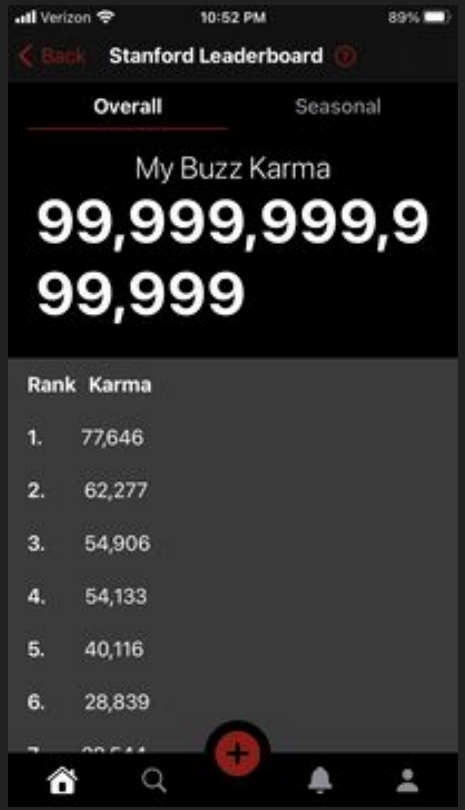
Case study: Fizz (2021)

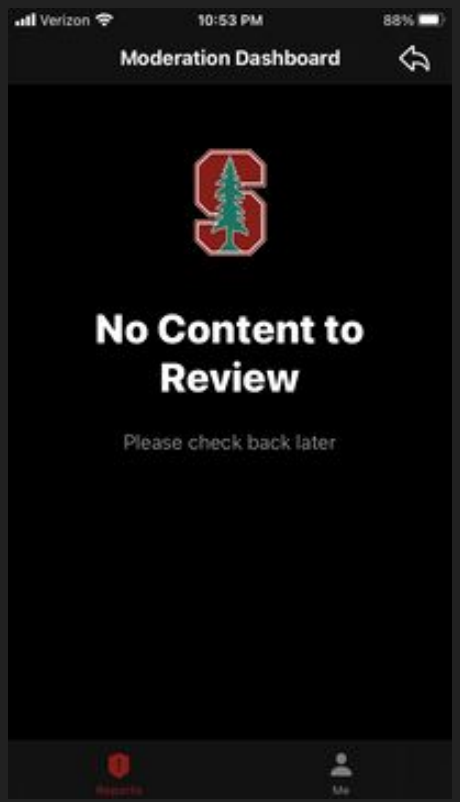
postDates
blockedPosts
muteDuration
numPosts
email
openAppCount
karma
isAmbassador
numChatNotificatio...
phoneNumber
numReferrals
communityID
isAdmin
banDate
notificationBadge
blockedUsers
fcmToken
hasAskedForRating
userID
muteDate
banDuration
usersBlockedBy
tempKarma
communityChangeDate

Users

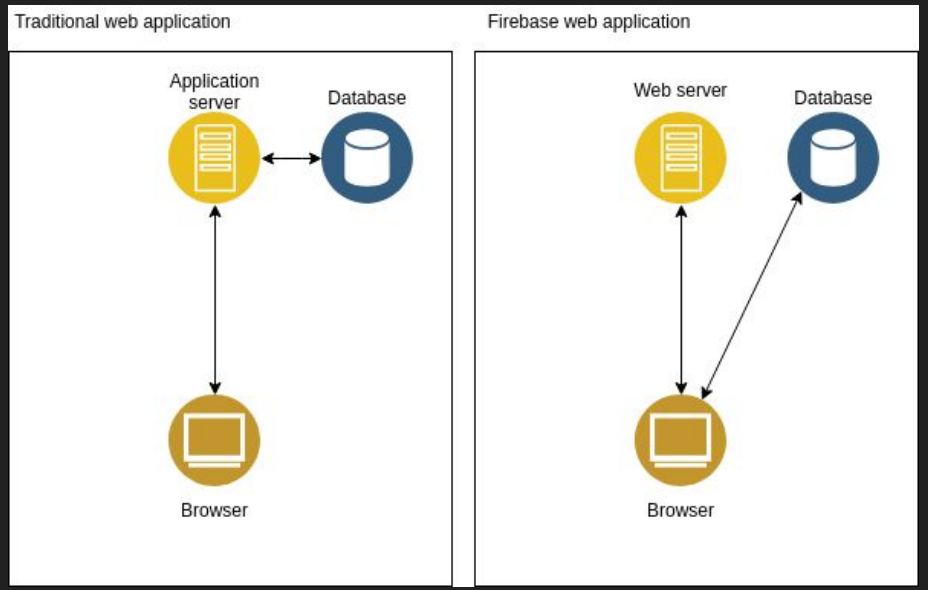
text
likeCount
commentCount
usersSaved
communityID
date
numAutoLikes
flair
pseudonym
dislikeCount
mediaURL
pastWeek
likes
postID
likesMinusDislikes
recentVoterID
ownerID
pastDay
hotScore
dislikes

Posts





Misconfigured Firebase security rules



Clients can directly access the database (including malicious clients!)

- Database is in charge of validating user access to data
- Poor validation (e.g. misconfigured rules) -> unauthorized data access

Avoiding Firebase misconfigurations

- A little harder: Google documentation on good rules is confusing
- Set up unit tests for your rules

Unsanitized user input

- Always assume user input can be malicious
- If user input gets misinterpreted as code, bad things happen!
  - Cross-Site Scripting (XSS)
  - SQL injection
- Using modern

On your own time:
catshare.saligrama.io

Security takeaways

Don’t reinvent the wheel

- Modern frameworks abstract away raw code and data handling
  - Helps avoid user input-related vulnerabilities

Don’t:

- Roll your own cryptography/auth
- Write your own SQL w/ user input
- Modify HTML DOM raw w/ user input

Do:

- Use well-tested frameworks
  - And as much of their native functionality as you can
- Use managed cloud services!!!

Build with a security mindset

- When building a product/feature, consider:
  - How can this be abused?
  - What can I add to prevent that abuse vector?

Build with a security mindset

- When building a product/feature, consider:
  - How can this be abused?
  - What can I add to prevent that abuse vector?

- There’s no one-size-fits-all approach to security
  - Our advice helps you avoid common mistakes

Build with a security mindset

- When building a product/feature, consider:
  - How can this be abused?
  - What can I add to prevent that abuse vector?

- There’s no one-size-fits-all approach to security
  - Our advice helps you avoid common mistakes

- These concerns apply to all tech products, whether B2C or B2B

It happens to the best of us

Aditya's Blog

Thoughts, guides and fun from a security/systems enthusiast @ Stanford

Flipping the script: when a hacking class gets hacked

October 12, 2022 1316 words No tag

This morning, an EternalBlue-vulnerable machine used for testing for Stanford's Hack Lab course accidentally given a public IP address on Google Cloud was unsurprisingly pwned and used to launch further EternalBlue scanning against other public web hosts.

This blog post describes our course's infrastructure setup (including why we had that testing box in the first place), how we discovered and remediated the incident, and how we used the incident as a way to teach students about incident response and public disclosure.

Let the community help you

A vulnerability disclosure policy is intended to give ethical hackers clear guidelines for submitting potentially unknown and harmful security vulnerabilities to organizations.

Vulnerability Disclosure Policy Resources

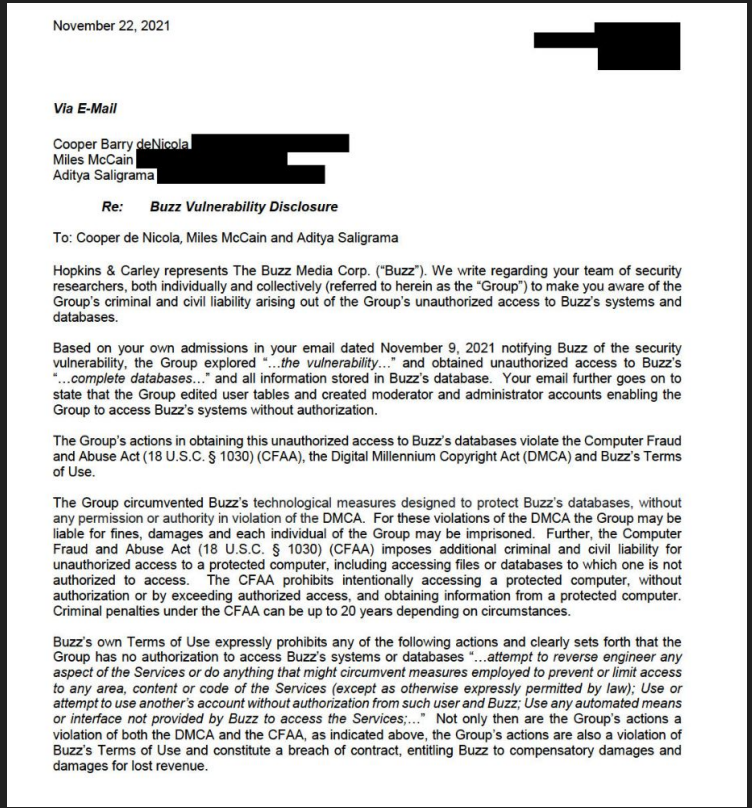
DHS Template: https://cyber.dhs.gov/bod/20-01/vdp-template/

DoJ Framework: https://www.justice.gov/criminal-ccips/page/file/983996/download

HackerOne: https://www.hackerone.com/blog/What-Vulnerability-Disclosure-Policy-and-Why-You-Need-One

Example Safe Harbor: https://github.com/cybertransparency/vdp-terms

Please don’t do this



November 22, 2021

[redacted]

Via E-Mail

Cooper Barry deNicola [redacted]
Miles McCain
Aditya Saligrama

Re: Buzz Vulnerability Disclosure

To: Cooper de Nicola, Miles McCain and Aditya Saligrama

Hopkins & Carley represents The Buzz Media Corp. (“Buzz”). We write regarding your team of security researchers, both individually and collectively (referred to herein as the “Group”) to make you aware of the Group’s criminal and civil liability arising out of the Group’s unauthorized access to Buzz’s systems and databases.

Based on your own admissions in your email dated November 9, 2021 notifying Buzz of the security vulnerability, the Group explored “*…the vulnerability…*” and obtained unauthorized access to Buzz’s “*…complete databases…*” and all information stored in Buzz’s database. Your email further goes on to state that the Group edited user tables and created moderator and administrator accounts enabling the Group to access Buzz’s systems without authorization.

The Group’s actions in obtaining this unauthorized access to Buzz’s databases violate the Computer Fraud and Abuse Act (18 U.S.C. § 1030) (CFAA), the Digital Millennium Copyright Act (DMCA) and Buzz’s Terms of Use.

The Group circumvented Buzz’s technological measures designed to protect Buzz’s databases, without any permission or authority in violation of the DMCA. For these violations of the DMCA the Group may be liable for fines, damages and each individual of the Group may be imprisoned. Further, the Computer Fraud and Abuse Act (18 U.S.C. § 1030) (CFAA) imposes additional criminal and civil liability for unauthorized access to a protected computer, including accessing files or databases to which one is not authorized to access. The CFAA prohibits intentionally accessing a protected computer, without authorization or by exceeding authorized access, and obtaining information from a protected computer. Criminal penalties under the CFAA can be up to 20 years depending on circumstances.

Buzz’s own Terms of Use expressly prohibits any of the following actions and clearly sets forth that the Group has no authorization to access Buzz’s systems or databases “*…attempt to reverse engineer any aspect of the Services or do anything that might circumvent measures employed to prevent or limit access to any area, content or code of the Services (except as otherwise expressly permitted by law); Use or attempt to use another’s account without authorization from such user and Buzz; Use any automated means or interface not provided by Buzz to access the Services;…*” Not only then are the Group’s actions a violation of both the DMCA and the CFAA, as indicated above, the Group’s actions are also a violation of Buzz’s Terms of Use and constitute a breach of contract, entitling Buzz to compensatory damages and damages for lost revenue.

Trust & Safety

Copyright violation

Terrorist content

Doxxing

Spam

Suicide and self-harm content

Mis/disinformation

Hate speech

Solicitation of minors (grooming)

Explicit threats of violence

Sexual extortion (“sextortion”)

Child sexual abuse material (CSAM)

White supremacist content

Glorification of violence

Human trafficking

Images of animal cruelty

Criminal organizations

Non-consensual intimate imagery (“revenge porn”)

Violent imagery

Images of animal cruelty

Stalking

Case study
Ridesharing platform

- Your ridesharing platform offers a $5 credit on users' first ride
- You see very strong growth of DAUs, but retention and conversion is non-existent
- You observe that many new rides happen along the same route

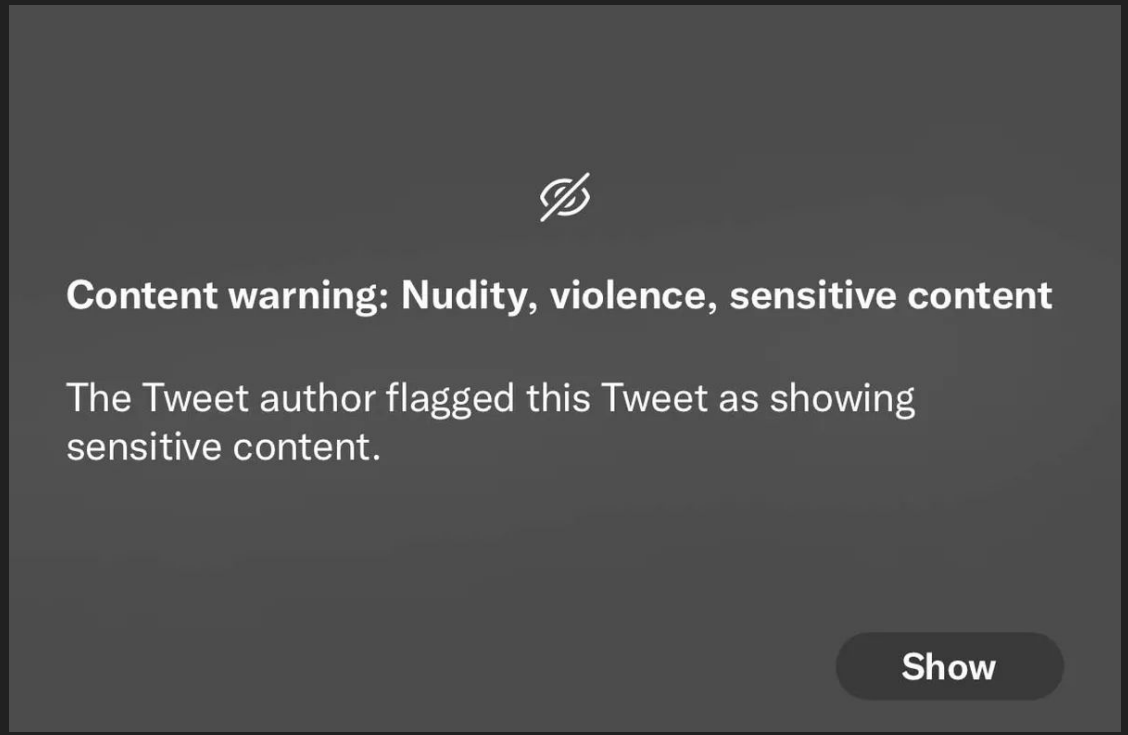
Case study

Ridesharing platform

What are some potential mitigations?

Case study
Video sharing platform

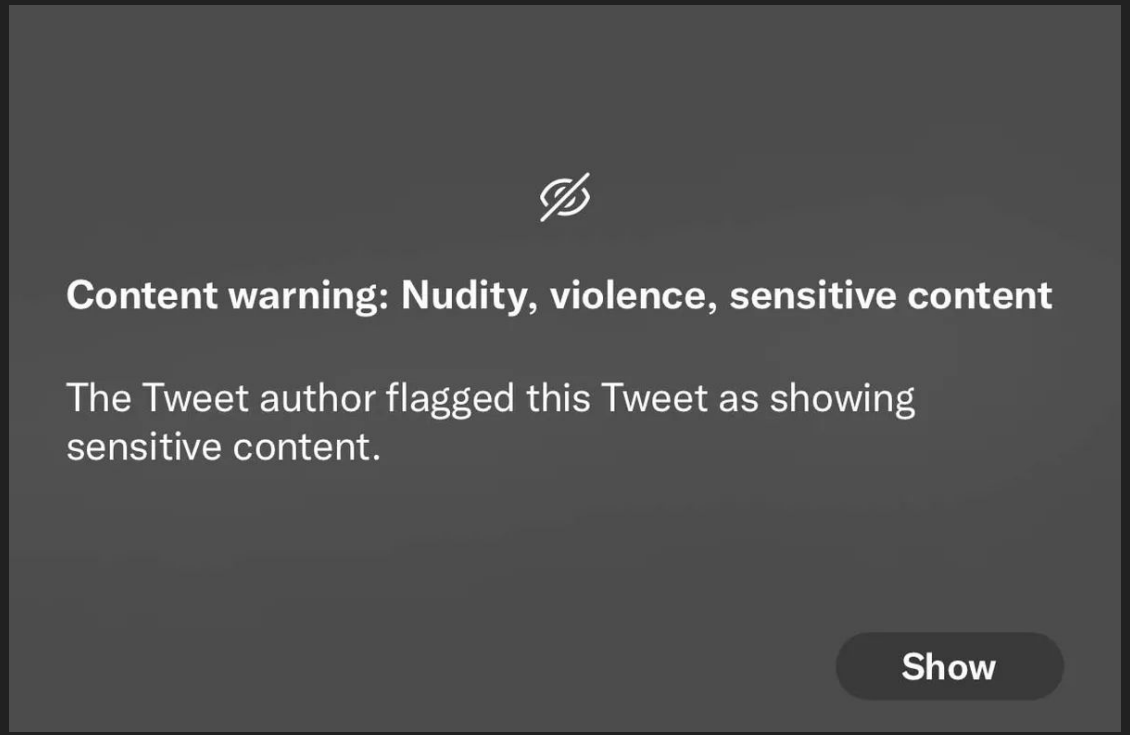
- Your revolutionary video sharing platform is gaining popularity around the world
- Users begin to upload highly graphic content — though not illegal under U.S. law
- You worry this content will hurt your brand



Case study

Video sharing platform

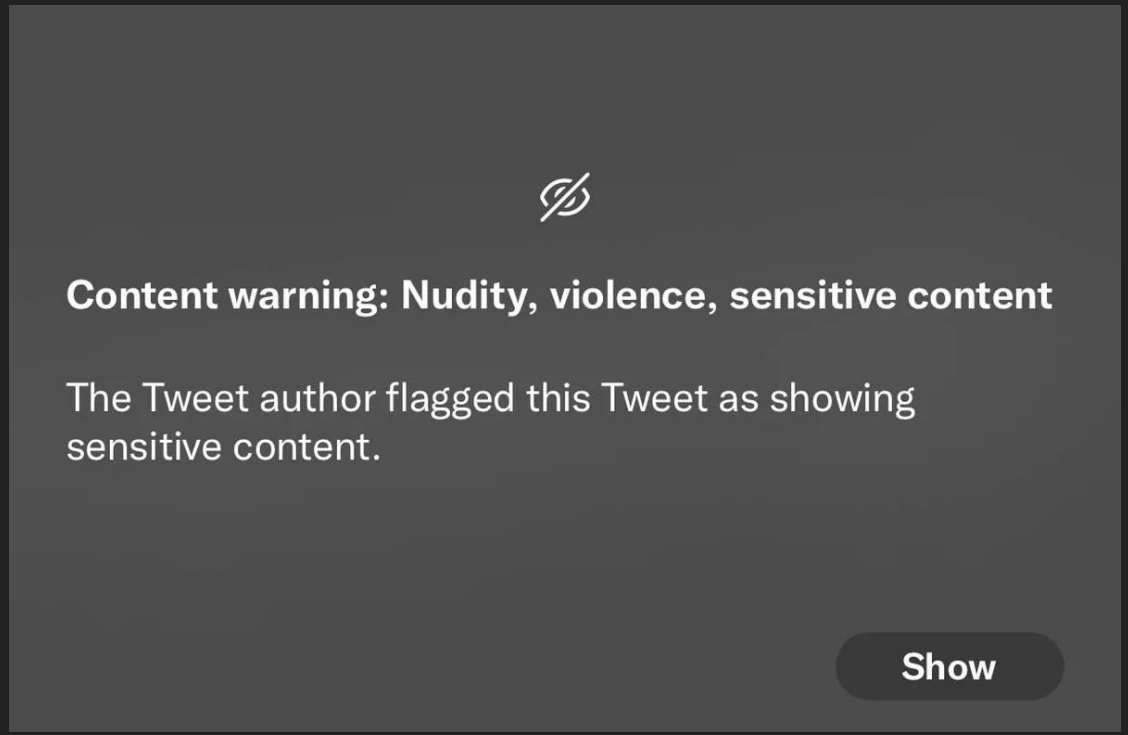
What are some potential mitigations?



Case study
Video sharing platform

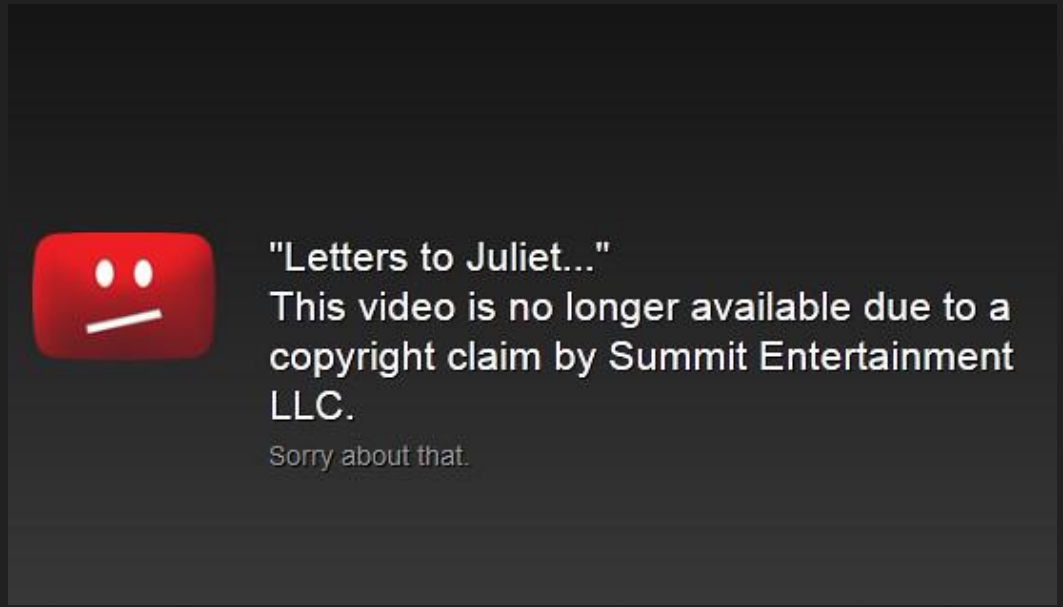
Complications:

- Your app is popular in Ukraine, and hosts media documenting Russian war crimes
- Your mitigation deleted some of this evidence
- You have received a threat from the Russian Government to take down all Ukraine-related content — or else



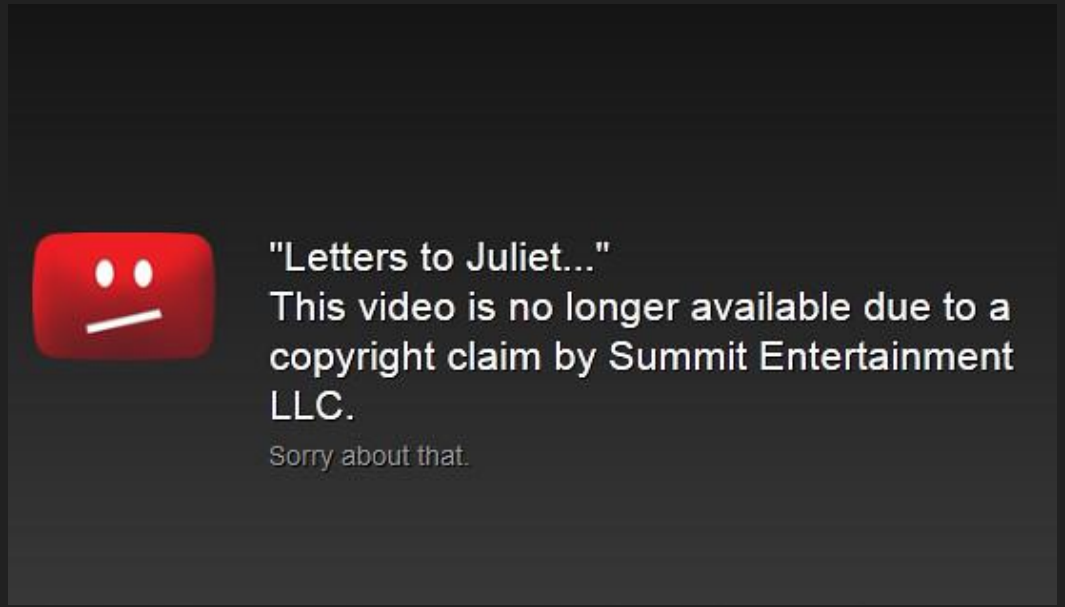
Case study
Consumer social

- Your revolutionary new social media platform has a copyright abuse problem
- Someone keeps posting copy-pasted articles and film clips
- You've received a cease & desist from a major media conglomerate



Case study
Consumer social

What are potential mitigations?



Let’s play with a real moderation API: perspectiveapi.com

Proactive measures

Reactive measures

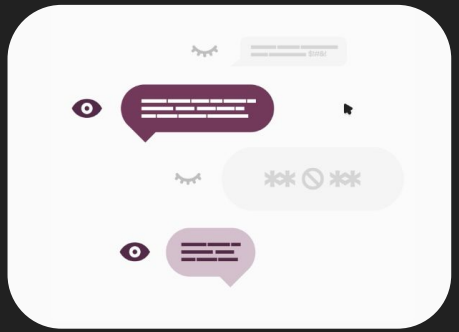
Proactive measures

Automated content safety APIs, e.g.,

- PhotoDNA for CSAM
- Perspective for hate speech
- Google Cloud Vision APIs for gore, sexual content, violent content, etc.

Design-level considerations, e.g.,

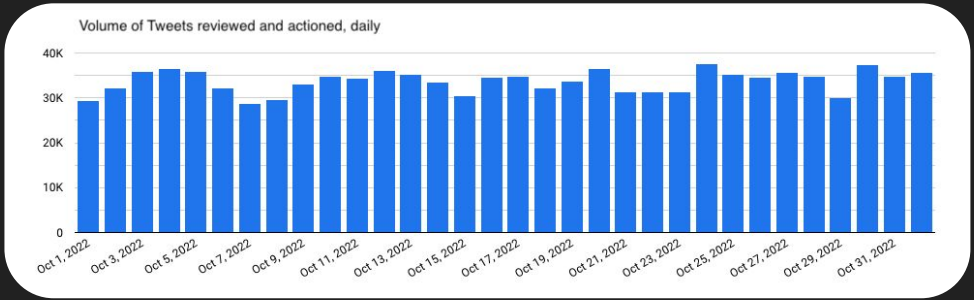
- Give users agency over what they see (blocking, muting, etc.)
- Be mindful of opportunities for algorithmic manipulation
- Have clear content guidelines



Reactive measures

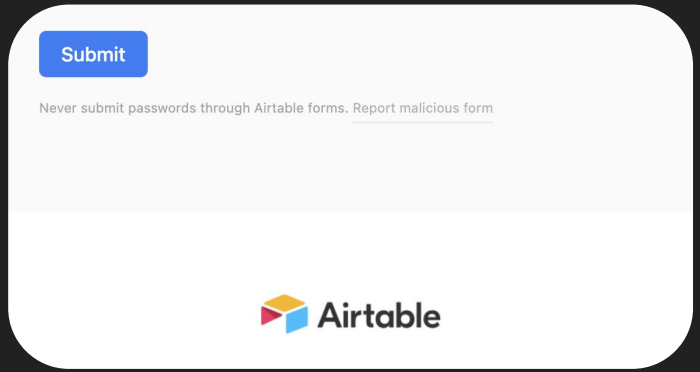
Have visibility into your platform

- Periodically review a random sample of activity
- Monitor user activity for anomalies
- Be mindful of cultural differences and norms



Let your users be your eyes + ears

- Add the ability to report bad content, even when you think it's unnecessary



Wrapping up

Nothing is 100% secure

You *are* a target

Don’t wait and see;
be proactive!

Data brings responsibility and risk

Talk to us!

Additional Security Resources

*Security 101 for SaaS Startups (**please read this one**):*
https://github.com/forter/security-101-for-saas-startups

Credits

- *Stamos's Hierarchy, Web Crash Course* – Alex Stamos, INTLPOL 268 Hack Lab
- *Web Crash Course, IDOR/XSS/Session Handling Slides, Marriage Pact IDOR Case Study* – Cooper de Nicola, CS 106S Coding for Social Good
- *Stanford Link, Fizz articles* – The Stanford Daily
- *Firebase web app vs. Traditional web app graphic* – Iosiro Security
- *CatShare* – Cooper de Nicola, Aditya Saligrama, George Hosono