

Web Hacking for Social Good

Why care about security?

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?***

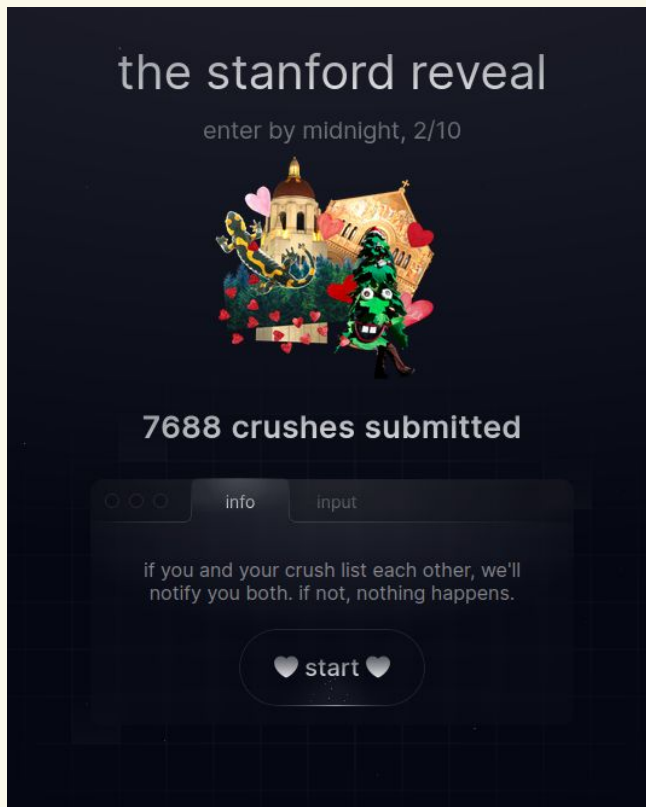
Case study: Stanford Link (2020)

The Stanford Daily

News • Campus Life

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

What's old is new again: Stanford Reveal (2023)



The Stanford Daily

Humor

Stanford Reveal pledges to leak only the “juiciest” crushes

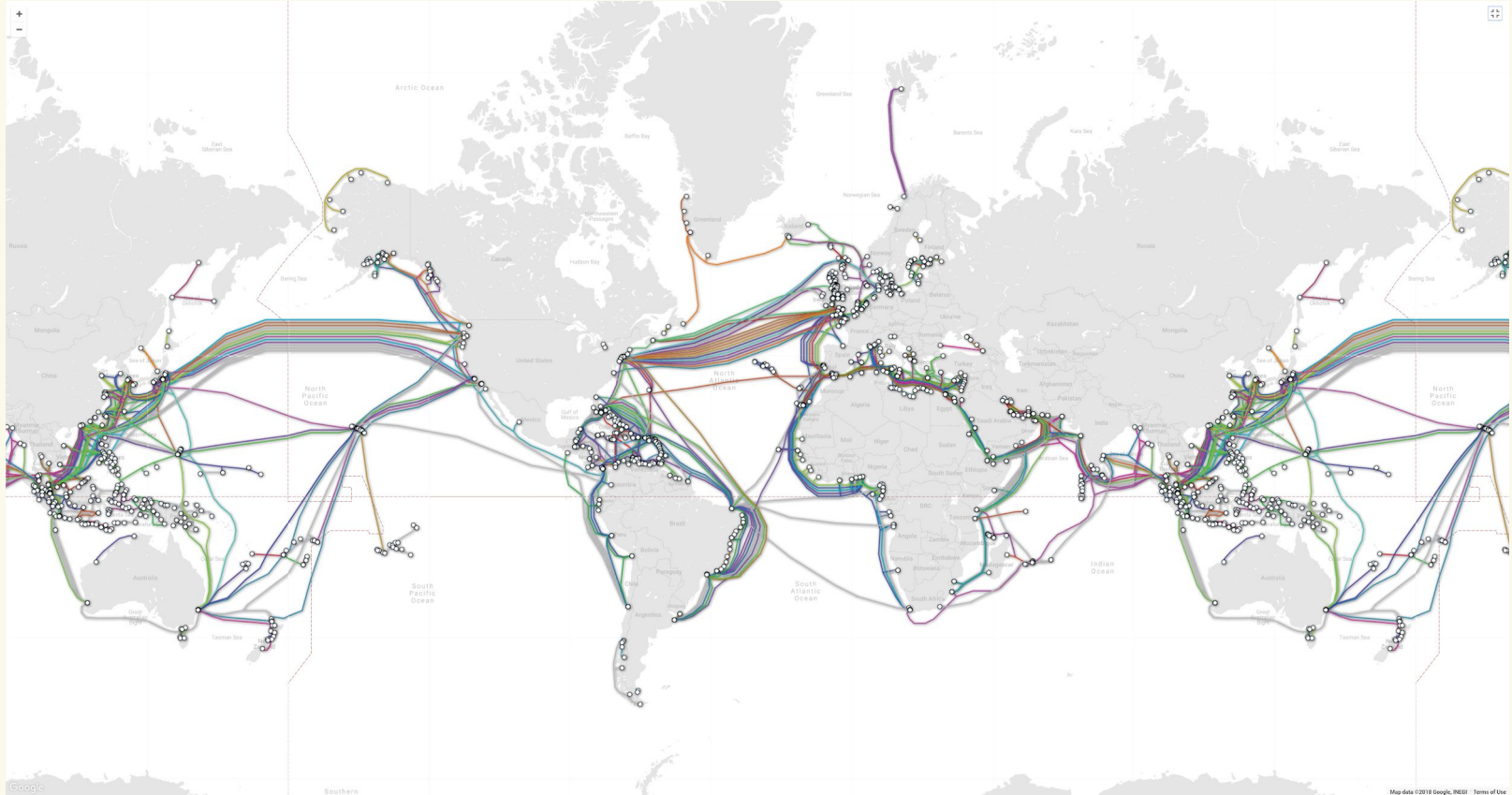
```
44 {  
45   "submittingUserFullName": "Aditya Saligrama",  
46   "user": "4yz2FPyYDgND8KhtVOQLCeeGsaq2",  
47   "submittingUserEmail": "akps@stanford.edu",  
48   "fullNames": []  
49 },
```

```
231 {  
232   "submittingUserEmail": "mccain@stanford.edu",  
233   "submittingUserFullName": "Robert Miles Redd McCain",  
234   "user": "N3Q9CkeKeJfKQz0hqt7qFbpanat1",  
235   "fullNames": [  
236     "isabelle levent"  
237   ]  
238 },
```

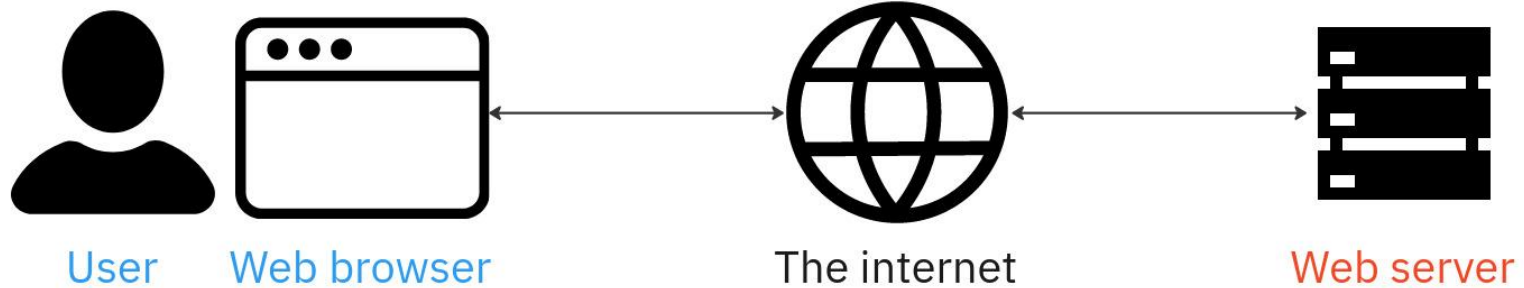
Why hack?

The fastest web crash course ever

How does the Internet work?



Our Internet abstraction



What language does the web speak?



How do we communicate with a web server?

HTTP

Hypertext Transport Protocol

HTTP: the missing language of the web



GET index.html



```
<!DOCTYPE html>  
<html>  
<body>
```

```
<h1>Hello World!</h1>
```

```
</body>  
</html>
```


HTTP protocol


GET / HTTP/1.0

Verb

Object (noun)

Protocol

HTTP requests



GET / HTTP/1.1

Host: stanford.edu

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:59.0)
Gecko/20100101 Firefox/59.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: close

Upgrade-Insecure-Requests: 1

The diagram shows an HTTP request structure. A yellow arrow labeled 'Request' points to the first line 'GET / HTTP/1.1'. Two orange arrows labeled 'Headers' point to the subsequent lines: 'Host: stanford.edu' and 'Upgrade-Insecure-Requests: 1'.

HTTP responses

HTTP/1.1 302 Found

Response Code

Date: Mon, 02 Apr 2018 02:37:56 GMT

Headers

Server: Apache

Location: <https://www.stanford.edu/>

Content-Length: 209

Connection: close

Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

Body

<html><head>

<title>302 Found</title>

</head><body>

<h1>Found</h1>

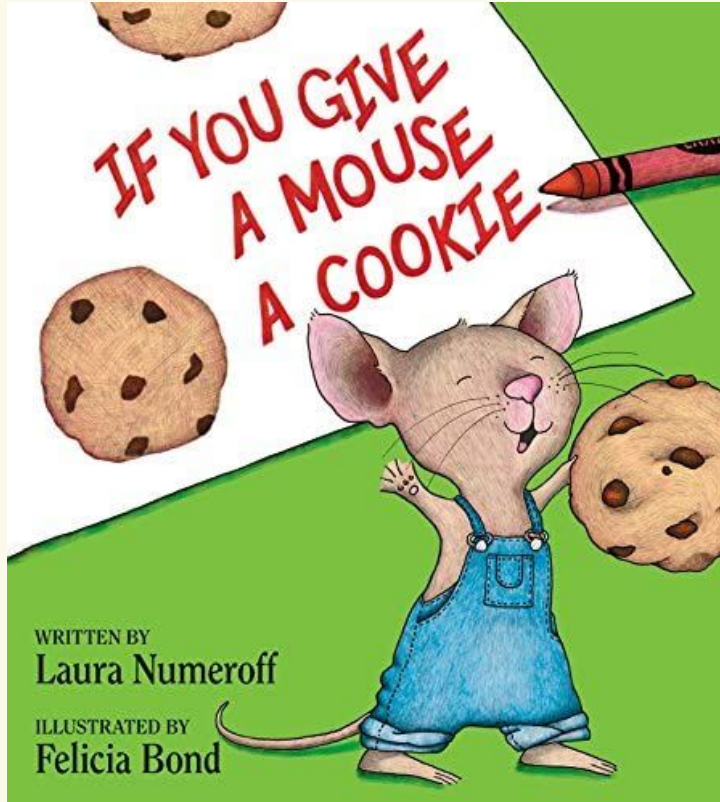
<p>The document has moved here.</p>

</body></html>

HTTP requests: GET and POST

- **GET**: Requests a specified resource
 - Should **only retrieve data**, without changing server state
- **POST**: Submits data to the specified resource
 - Often causes **changes in state** or side effects on the server

Session handling: how does a website remember?



- **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**
 - Future requests: **Cookie: session=session-id**

Demo: browser developer tools

Common insecure design patterns



CatShare

<https://catshare.saligrama.io>



We're a real startup!



October 14, 2022

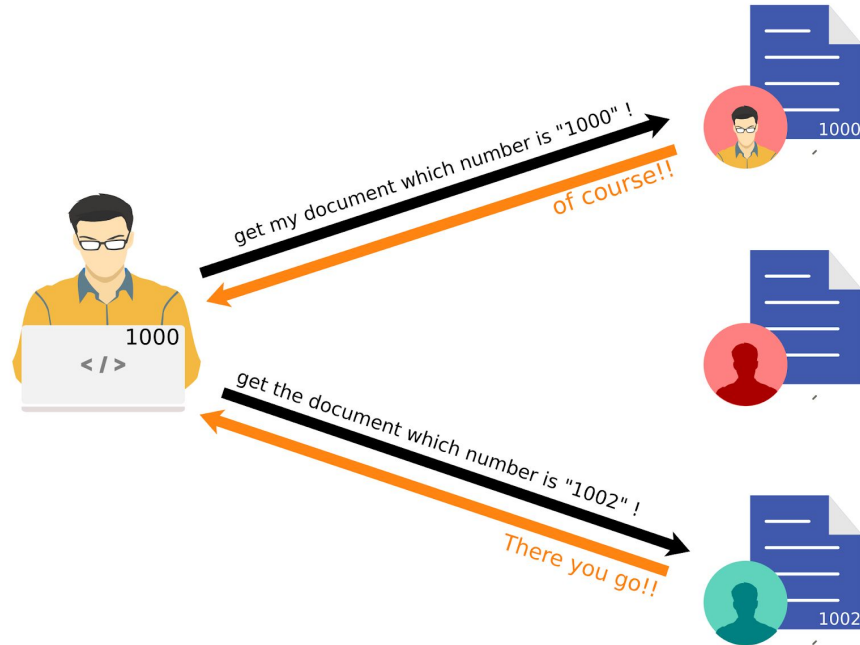
Vulnerabilities

- Insecure Direct Object Reference (IDOR)
- Cross Site Scripting (XSS)
- Improper Session Handling
- Database vulnerabilities: Firebase

Insecure Direct Object Reference

Insecure Direct Object Reference (IDOR)

Or: asking the server for the resources you want

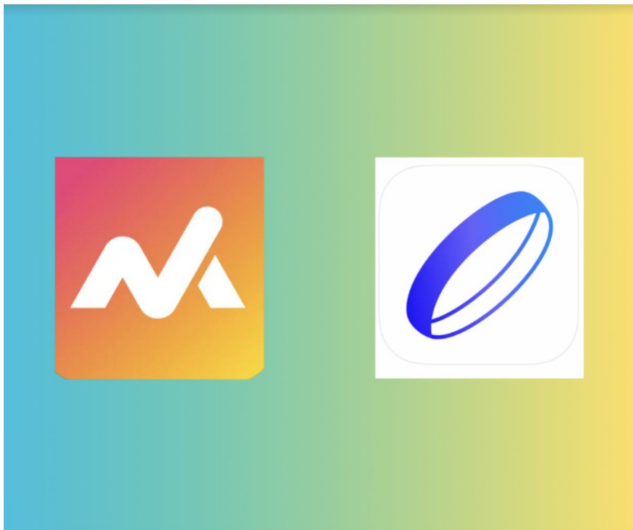


IDOR case study I: Wristband (2023)

The Stanford Daily

News • Campus Life

Stanford party apps hit the scene



Mixer and Wristband both launched this year at Stanford. (Graphic: ANANYA UDAYGIRI/The Stanford Daily)

By Ananya Udaygiri and Joseph Shull
Oct. 24, 2023, 11:42 p.m.

Wristband: an app for finding and getting into public and private events

Vulnerability disclosure, unauthorized read and write to sensitive data

-- Wristband



ⓧ Aditya Saligrama <saligrama@stanford.edu>

Thursday, October 26, 2023 at 4:49 PM

To: contact@wristband.events;



+1 more

Moreover, since your event IDs are sequentially ordered, anyone can use the share URL functionality to access private events; this is an issue even if row-level security is enabled. For example, <https://wristband.events/event/269> is a private event that can be accessed by enumerating event IDs starting from 1.

TRY IT!

- The CatShare team has a website <https://catshare.saligrama.io/> that **stores personal information**
- There's an endpoint <https://catshare.saligrama.io/user> to access this info
 - e.g. <https://catshare.saligrama.io/user?id=0>
- CatShare claims this is secure and only accessible to admins
- **Prove CatShare wrong**

IDOR case study II: 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

IDOR case study II: Stanford Marriage Pact (2020)

<https://mp.com/554d417a3bc9fbcba653c0097c6f3710>

554d417a3bc9fbcba653c0097c6f3710

MD5

saligrama@stanford.edu

cdenicol@stanford.edu

MD5

29d2223b196d87e8e9292308c074e593

<https://mp.com/29d2223b196d87e8e9292308c074e593>

Avoiding IDOR

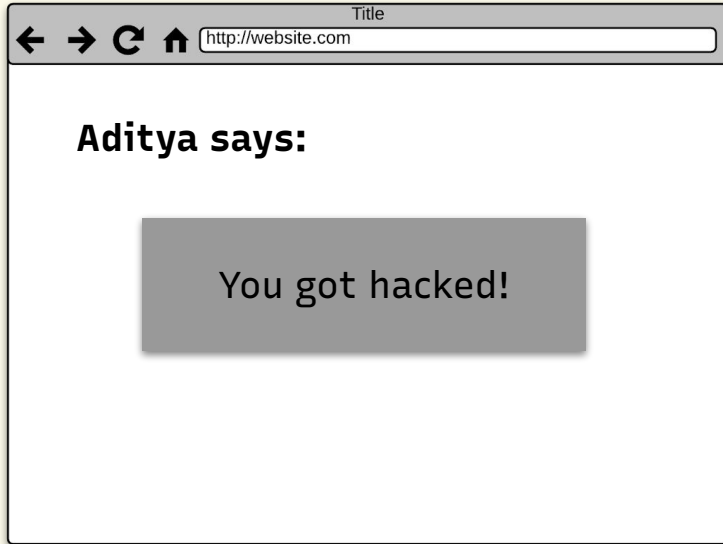
- Ensure that a user is **allowed to access a resource** before returning it
- If not possible (e.g. cloud storage buckets), then make resource URIs **random and unpredictable**. Avoid:
 - Automatically incrementing resource IDs
 - Hashing a **guessable property** such as usernames, phone numbers, or emails
- Instead: **use random identifiers** such as UUIDs

Cross-Site Scripting

Cross-Site Scripting (XSS)

- XSS attacks enable attackers to hijack your website to **run JavaScript code** on other users' browsers
- They occur when **user input is not properly sanitized and displayed**, allowing it to execute as code

Cross-Site Scripting (XSS)



GET /myfeed

```
<!DOCTYPE html>
<html>
<body>

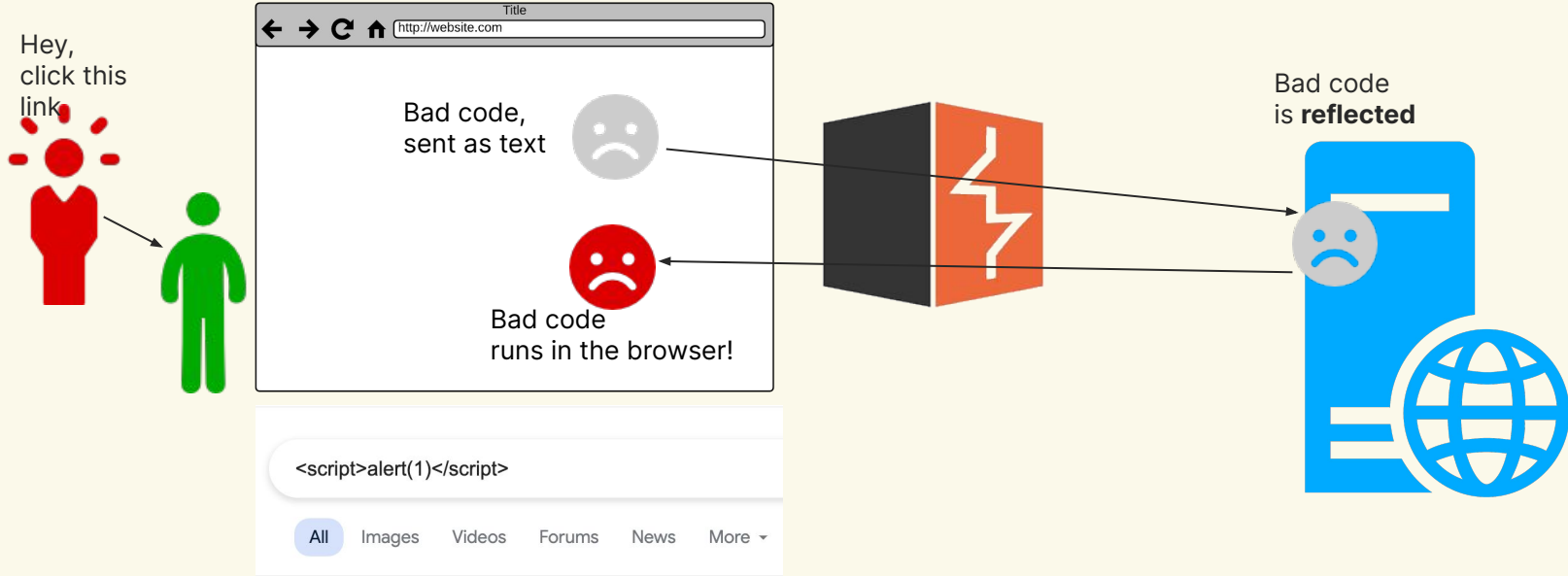
<b>Aditya says:</b>
<script>
    alert("You got hacked!");
</script>

</body>
</html>
```



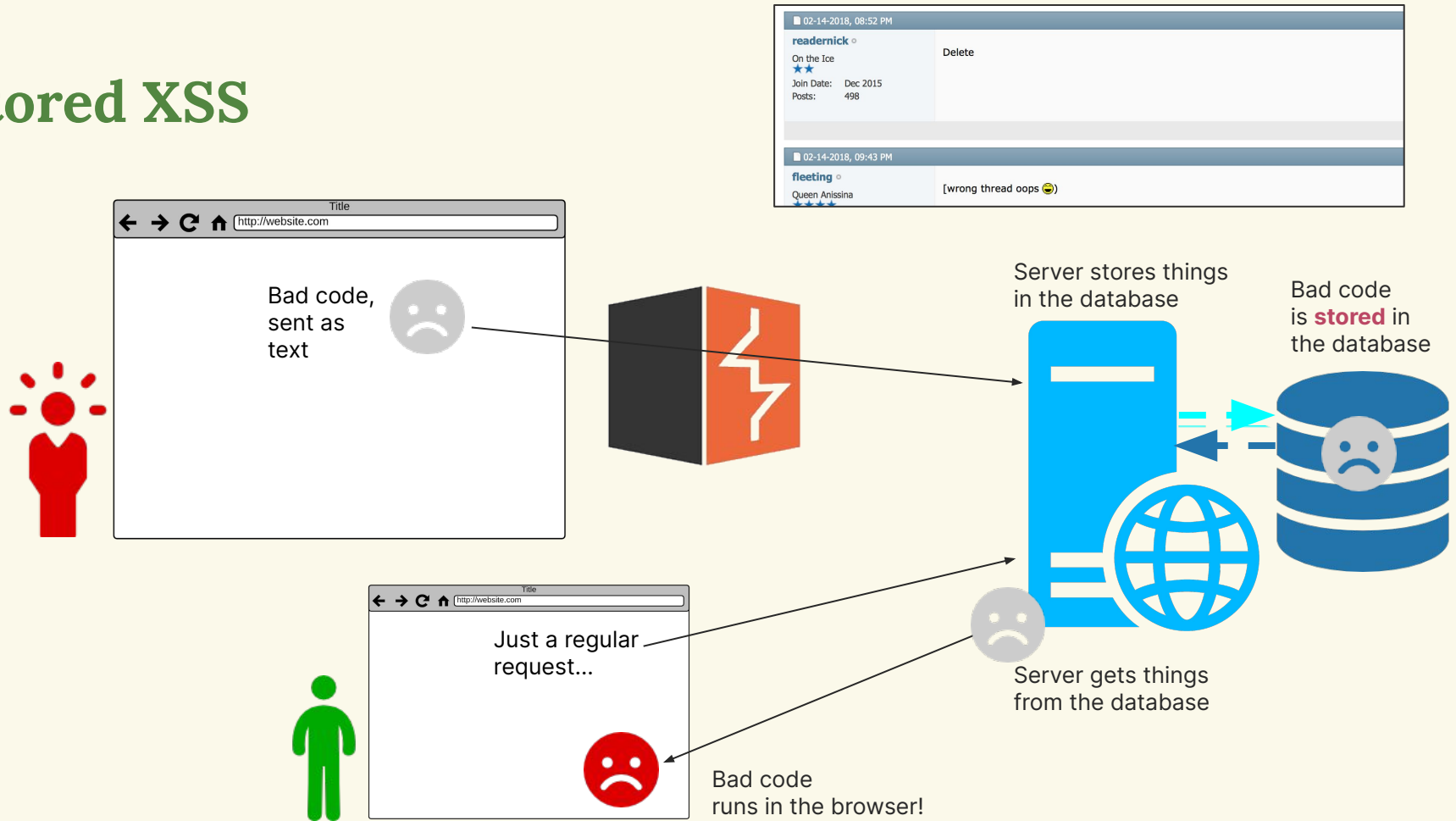
[https://vulnerable.website/search?query=<script>alert\("pwned"\)</script>](https://vulnerable.website/search?query=<script>alert("pwned")</script>)

Reflected XSS



[https://vulnerable.website/search?query=<script>alert\("pwned"\)</script>](https://vulnerable.website/search?query=<script>alert('pwned')</script>)

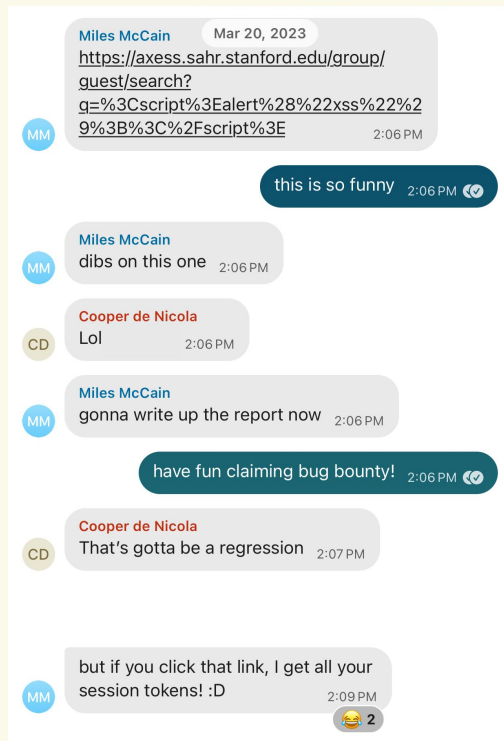
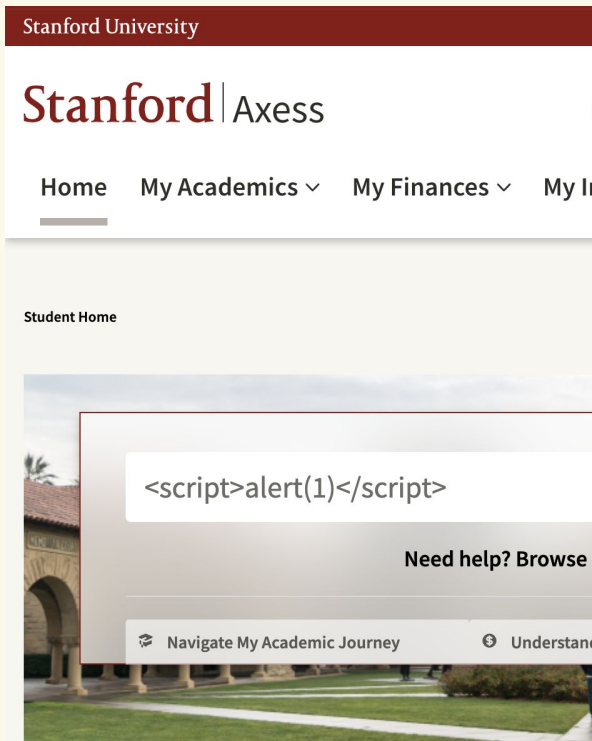
Stored XSS



TRY IT!

- After our last data breach, we at CatShare want to make our customers feel like we care about them
- We added an endpoint <https://catshare.saligrama.io/hello> that takes a user's name and greets them kindly. Ya know, to show we care
 - e.g. <https://catshare.saligrama.io/hello?name=User1>
- We think this is harmless and will only build customer trust. **Show us our mistake.**

XSS in Stanford Axess (2023)



Found and disclosed in March 2023, awarded **\$1000** by the **Stanford bug bounty**.

Remediated January 2024.

Attacks on session handling

Improper session handling

Cookie itself is insecure

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

Cookie not checked for authorization

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

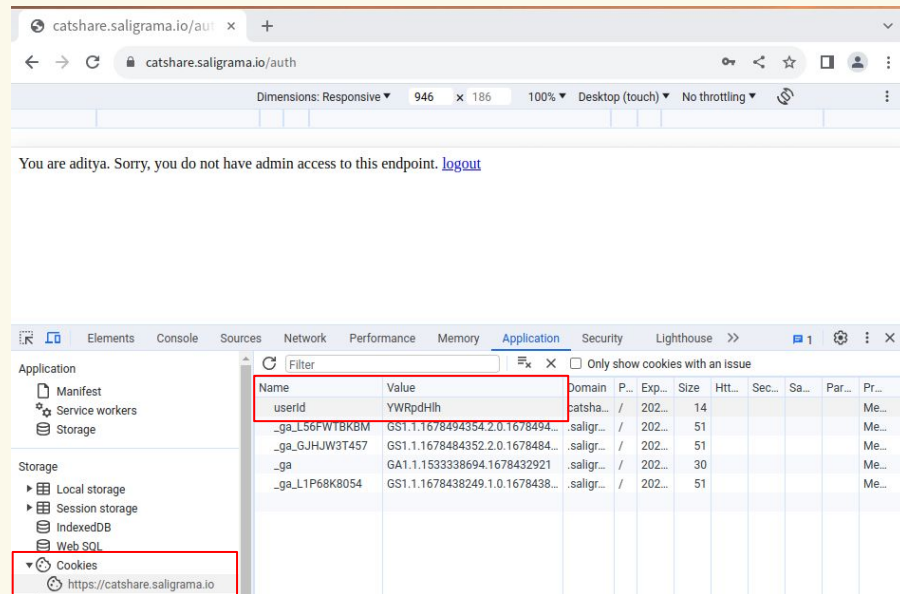
TRY IT!

- CatShare added an **admin view** to <https://catshare.saligrama.io/login> for admins to **view user data**
- Log in using **stanford:stanford**
- Can you become **admin** and view the user data?

TRY IT!

TOOLS/REFERENCE

- Cookie is in **Base64** format
 - Transforms data into a mix of letters and numbers.
 - Doesn't actually secure or encrypt data; it's just a **different way to show it**.
 - Use <https://kk.lol> to encode/decode
- Your browser's **Developer Tools**
 - Accessible from **Inspect Element**



What to look for is in **red** (logged in as **aditya** here)

- <https://catshare.saligrama.io/login>
 - Login with **stanford:stanford**

Session handling case study: Kontra (2022)



Session handling case study: Kontra (2022)

The screenshot displays two browser tabs side-by-side, both showing network traffic analysis. The left tab, titled "Request", shows an HTTP GET request to a Twitter API endpoint. A red arrow points from the text "Tweedledum's UUID" to the value "ae697870-3e71-4a0e-bd5e-5ea501a62dd0" in the URL. Another red arrow points from the text "Tweedledee's Authorization header" to the "Authorization:" field in the request headers. The right tab, titled "Response", shows the corresponding JSON response. Two red arrows point from the text "Exposure of Tweedledum's votes" to the "interestcount" fields in the JSON objects, which contain the values "13" and "17".

Request Tab:

```
GET /prod/users/ae697870-3e71-4a0e-bd5e-5ea501a62dd0/topics/interests?interested=true&pageCount=40&pageNumber=0 HTTP/2
Host: api.dissonantchat.com
Accept: application/json
Accept-Encoding: gzip, deflate
User-Agent: kontra/1.0.0 CFNetwork/1240.0.4 Darwin/20.6.0
Accept-Language: en-us
Authorization: eyJraWQ1OjIzdDekEka2RnazVsdnVGMDXzV2ElaXEtYjRmOFJpaDhlDfczEM4K2oP5IsImFsZyI6IjMxMTU2In0.eYJzdWI1OiI1YjY1MzOS1lODU0L3Q3ZCt0kTWkyIWMwRmRlbnNjg3NTMiLCJlbWpfbF92ZXJpZm1lZCI6dHJlZWsiY3ZvdG90cmVRINicklkljoiIjM2NmZBM2DMNjQzM5S090WEZLTk0OTEbdkdZTJlZmJkNDMyIlwiiaXNzIjoiaHR0cHMhXCB9CL2NvZ25pdGdtbmRwLnVzLCQ3ME15HWw6B25hdHM0Y2VpKX91cy13XNOlTJfMOlwcTViYWw3IiwiaWZmZWVubG9pICZybmFnZSI6IHN3ZWkvbkVkbGVkZWJsLChnaXZlbG9uYVwiIljoivohVldlZWRSZSIsImF1ZC16Ijdhb2VtOGVqdWVmncGgzY3BvbHJtdC4mJykyIiwiaWZkZlbnRfaWQ1OjI3ZTN3ZSE5MC83MTg2LTQ2NDMtOTcNS1h2TG2gdzQGMzNzYlCl90b2tlbg19lc2UioiJpZCIsImF1dGhfZlZSI6MTY1MTM2MTC4M1w1ZkhwaixojXUNjNXMyZmZyLChpVXQ1OjE2ZTEZnEjE3ODIsImZhbmVseV9uYVwiIljoirRFRfu_ZNQ9qalNdubLfUFTQdpda0ccQoz6p-tVyZajpsj7Fe3zbvmpaaB87FvhqFYDzg90-ykATAEL-KIJNINO0iXGMmdada3TYxrBP8PA_zzt1uMOBJ0z-vGu632vg5pIndMXtkf461B86gpBa5K1RJPHdbhiT2su_g7I45DMxdPVDeogR65jTde1wbRrtBNMPTklypA6N4KKYwzw-5qxJNEBF7qJqeyghg6xxz-pw8Hws2npL4lmZNPEqagd5PE9vXanNWIEaxRFrWVD0C-ed3CyUYToT87LMExJlsvp93ZFqw
```

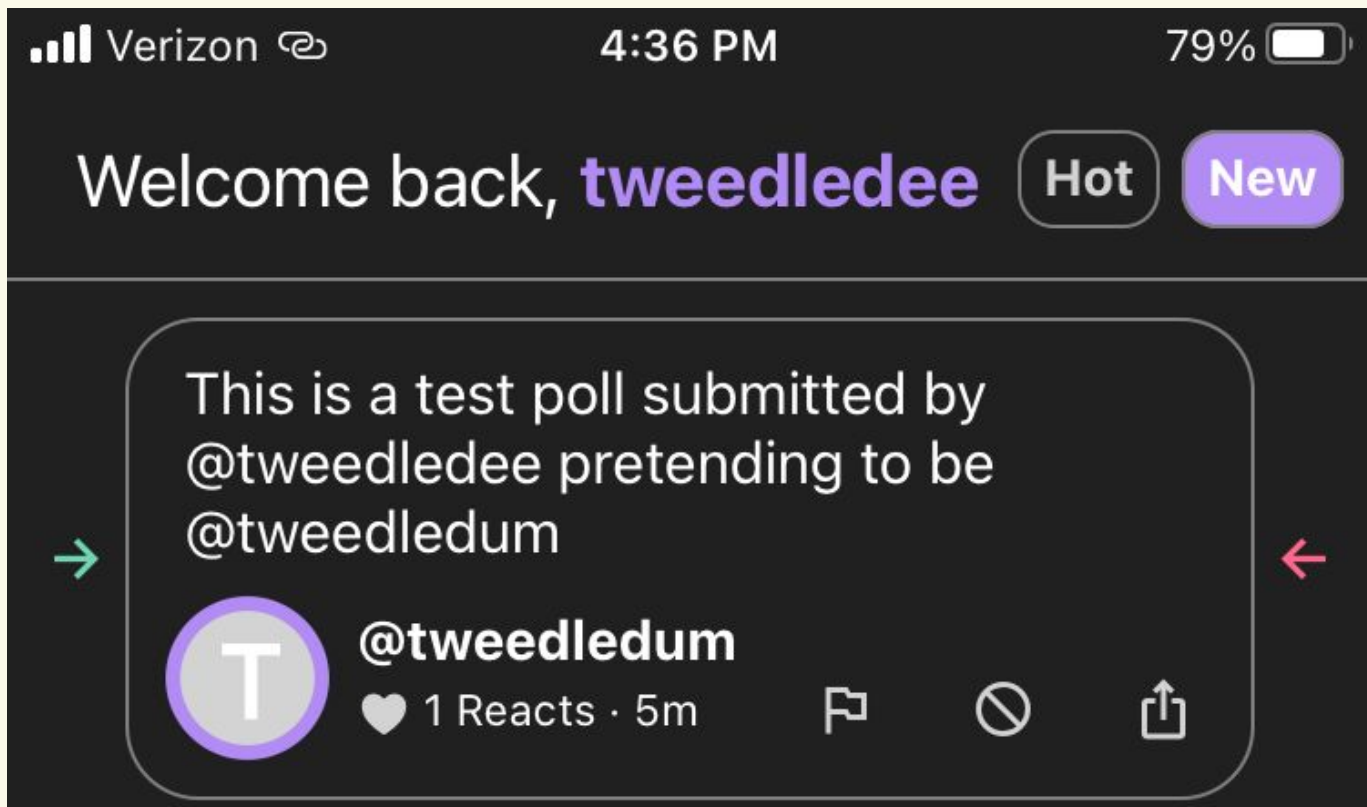
Response Tab:

```
{
  "userid": "ae697870-3e71-4a0e-bd5e-5ea501a62dd0",
  "topicid": "c4d7a095-d25f-4ce7-b795-0c48baaac899",
  "interestlevel": "No",
  "createdate": "2022-04-30T23:44:09.221Z",
  "topicdescription": "Will China and the US go to war before 2030?",
  "topicowner": null,
  "topicownerusername": null,
  "topiccreatedate": "2022-04-23T06:42:24.928Z",
  "interestcount": "13",
  "yescnt": "1",
  "nocnt": "12",
  "commentcontent": null
},
{
  "userid": "ae697870-3e71-4a0e-bd5e-5ea501a62dd0",
  "topicid": "a58cf98b-clf7-400c-81f8-d3f54ce503a3",
  "interestlevel": "Yes",
  "createdate": "2022-04-30T23:44:08.027Z",
  "topicdescription": "Can an atheist and a deeply Christian person be happily married?",
  "topicowner": null,
  "topicownerusername": null,
  "topiccreatedate": "2022-04-23T06:42:52.386Z",
  "interestcount": "17",
  "yescnt": "4",
  "nocnt": "13",
  "commentcontent": null
},
{
  "userid": "ae697870-3e71-4a0e-bd5e-5ea501a62dd0",
  "topicid": "18dde812-4b78-4525-ab18-3be2caebbdadd",
  "interestlevel": "Yes",
  "createdate": "2022-04-30T23:44:05.807Z",
  "topicdescription": "Do you see your political opponents as evil?",
  "topicowner": null,
  "topicownerusername": null,
  "topiccreatedate": "2022-04-23T06:44:32.964Z",
  "interestcount": "13",
  "yescnt": "1",
  "nocnt": "12",
  "commentcontent": null
},
{
  "userid": "ae697870-3e71-4a0e-bd5e-5ea501a62dd0",
  "topicid": "a2f3cefe-6bf7-46c4-aedb-f43417f1b681",
  "interestlevel": "No",
  "createdate": "2022-04-30T23:44:04.683Z",
  "topicdescription": "Are man buns ever okay?",
  "topicowner": null,
  "commentcontent": null
}
```

Session handling case study: Kontra (2022)

[illegible]

Session handling case study: Kontra (2022)



Avoiding improper session handling

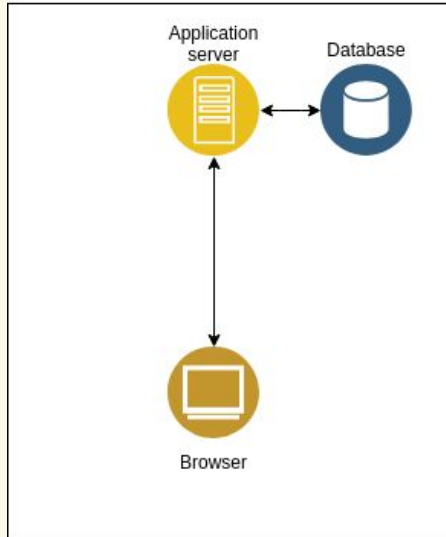
Before taking a sensitive action:

- Check the user **is who they say they are**
- And that they are **allowed to perform the action**

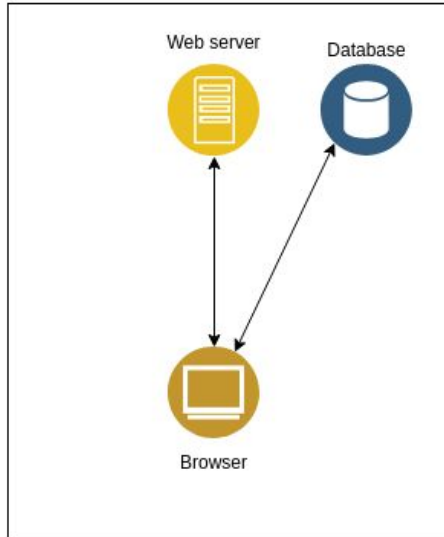
Database vulnerabilities

Misconfigured Firebase security rules

Traditional web application



Firebase web application



*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

Firestore 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.

Firestore 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

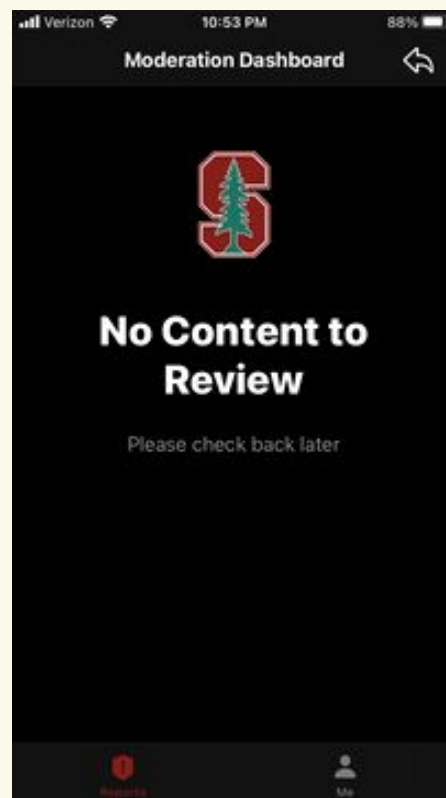
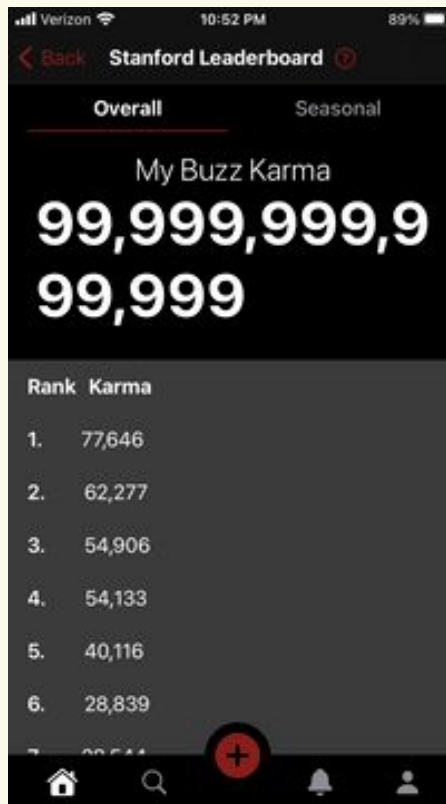
Firestore case study: Fizz (2021)

```
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



Wrapping up

Nothing is 100% secure

It happens to the best of us

saligrama.io/blog/hack-lab-got-hacked

Aditya Saligrama

Portfolio

Blog

Notes

Photography

Resume



Flipping the script: when a hacking class gets hacked

October 12, 2022

1351 words

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.

The community can help!

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

Disclosing vulnerabilities ethically



Client Name

Date

What we did

-

Findings & areas for improvement

-

Areas for further investigation

-

Heads up

A consult does not constitute an exhaustive security evaluation of your app. Rather, it represents a good starting point for the evolution of your service with the benefit of a security-informed perspective.

Looking ahead

Please tell your friends to visit the security clinic! You're also welcome to schedule another visit down the line. If you have any feedback, please email contact@securityclinic.org.

<https://securityclinic.org>

Bug bounty programs



Bug bounty programs incentivize the community to responsibly disclose security vulnerabilities to the vendor, in exchange for an (often monetary) **reward**.

Potential legal consequences to ethical hacking

November 22, 2021

Via E-Mail

Cooper Barry deNicola
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.

Security courses at Stanford

- **INTLPOL 268**: Hack Lab
- **CS 155**: Computer and Network Security
- **CS 152**: Trust and Safety Engineering
- **CS 255**: Cryptography
- **CS 153**: Applied Security at Scale
- **INTLPOL 268D**: Online Open Source Investigation
- **CS 40**: *Cloud Infrastructure and Scalable Application Deployment*

Security fun and hijinks @ Stanford Applied Cyber



applied-cyber.stanford.edu

A student's dream: hacking (then fixing) Gradescope's autograder

February 28, 2023
2630 words

Seven judges graded the submissions. The top scorers were “cody3,” “aray4” and “cody2.”

Two of those handles came from Cody Ho, a student at Stanford University studying computer science with a focus on A.I. He entered the contest five times, during which he got the chatbot to tell him about a fake place named after a real historical figure and describe the online tax filing requirement codified in the 28th constitutional amendment (which doesn't exist).



Credits

CatShare source code

<https://github.com/saligrama/catshare-serverless>

Other materials

- *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
- *Stanford Link, Fizz, Stanford Reveal articles* – The Stanford Daily
- *CatShare* – Cooper de Nicola, Aditya Saligrama, George Hosono