

# Beginner's guide to bug bounty

Rice COMP 427/541: Introduction to Computer Security

4/12/21

@kaoudis



## *outline*

- **What is a bug bounty?**
- **Rules of engagement and scope**
- **Example: Clickjacking in Twitter video player by *filedescriptor***
- **Example: Missing API permissions by *ryotak***
- **Example: Email bomb by *Akhil Kakkireni***
- **What could go wrong?**
- **More resources**



**Kelly Kaoudis**  
(she/her)

Tech Lead & Sr Software Engineer, Twitter Application Security  
@kaoudis



# Why would we want to let security researchers test others' computers without their permission?

- to protect public health
- to secure elections
- to make driving more safe
- to protect consumer privacy on the Internet



# Why would we want to let security researchers test others' computers without their permission?

- to protect public health
- to secure elections
- to make driving more safe
- **to protect consumer privacy on the Internet!**





## Twitter

Twitter helps you create and share ideas and information instantly, without barriers.

<https://twitter.com> · [@twittersecurity](#)

Reports resolved  
**1421**

Assets in scope  
**12**

Average bounty  
**\$560**

Submit report

### Bug Bounty Program

Launched on May 2014

Includes retesting ?

Bounty splitting enabled ?

[Policy](#) [Hacktivity](#) [Thanks](#) [Updates \(0\)](#) [Collaborators](#)

### Policy

#### Program Rules

Maintaining effective security is a community effort, and we are proud to have a vibrant group of independent security researchers who volunteer their time to help us spot potential issues. To recognize their efforts and the important role they play in keeping Twitter safe for everyone we offer a bounty for reporting certain qualifying security vulnerabilities. Please make sure you review the following program rules before you report a vulnerability. By participating in this program, you agree to be bound by these rules.

#### Rewards

### Response Efficiency

**8 hrs**

Average time to first response

**about 1 month**

Average time to bounty

**4 months**

Average time to resolution

**rules of engagement  
and scope**





# Rules of engagement

- stick to confidentiality agreements, NDAs
- stay in-scope where humanly possible
- “in good faith” and safe harbour
- professionalism



# Twitter in-scope

Scopes				
In Scope				
Domain	*.twitter.com	Critical	Eligible	
Domain	*.vine.co	Critical	Eligible	
Domain	*.periscope.tv	Critical	Eligible	
Domain	*.pscp.tv	Critical	Eligible	
Domain	*.twimg.com	Critical	Eligible	
Domain	gnip.com	Critical	Eligible	
Android: Play Store	com.twitter.android	Critical	Eligible	
iOS: App Store	com.atebits.Tweetie2	Critical	Eligible	
Domain	niche.co	Critical	Ineligible	
Domain	snappytv.com	Critical	Ineligible	
Domain	twitterflightschool.com	Medium	Ineligible	



# Twitter out-of-scope

Out of Scope	
Domain	<b>status.twitter.com</b> This is hosted by a third party, status.io.

The following issues are outside the scope of our vulnerability rewards program (either ineligible or false positives):

- Attacks requiring physical access to a user's device
- Any physical attacks against Twitter property or data centers
- Forms missing CSRF tokens (we require evidence of actual CSRF vulnerability)
- Logout CSRF
- Password and account recovery policies, such as reset link expiration or password complexity
- Invalid or missing SPF (Sender Policy Framework) records
- Content spoofing / text injection
- Issues related to software or protocols not under Twitter control
- Reports of spam ([see here for more info](#))
- Bypass of URL malware detection
- Vulnerabilities only affecting users of outdated or unpatched browsers and platforms
- Social engineering of Twitter staff or contractors
- Issues without clearly identified security impact, such as clickjacking on a static website, missing security headers, or descriptive error messages
- Issues that result in Denial of Service (DoS) to Twitter's servers at the network or application layer.
- Reports of broken hyperlinks from Twitter blog posts, press releases, or support articles to unclaimed Twitter Handles or to a location where it is not possible to cause the controlled contents to be downloaded to the victim's filesystem.
- Issues relating to unlocking client-side features in modified Twitter applications, rooted devices, or jailbroken devices.

# **example: email bomb**

***Akhil Kakkireni, 2017***

***<https://hackerone.com/reports/297359>***



trabajoduro\_2 submitted a report to **Twitter**.

Dec 12th (4 years ago)

Hi Team,

I have found a logical flaw(NOT DoS) in the website '<https://app.mopub.com/>'

1. Use Burp Suite and capture below request upon navigation to *Code integration*
2. Click on Send button after entering email address in the input field of 'Enter one or more email addresses and we'll send you links to the integration instructions for this ad unit.'

{code}

POST /web-client/api/ad-units/email-instructions HTTP/1.1

Host: app.mopub.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0

Accept: /

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

Accept-Encoding: gzip, deflate

Referer: [https://app.mopub.com/ad-unit?key=\[REDACTED\]&showIntegration=true](https://app.mopub.com/ad-unit?key=[REDACTED]&showIntegration=true)

Content-Type: application/json

x-csrfToken: [REDACTED]

Content-Length: 88

Cookie: \_ga=[REDACTED]; \_gid=[REDACTED]; csrfToken=[REDACTED]; mp\_mixpanel\_\_c=8; sessionId=[REDACTED];

mp\_c99579c4804fba6b8aeed7a911581652\_mixpanel=%7B%22distinct\_id%22%3A%20%22405f9ac1ce5749abb6092834819b3ec4%22%2C%22accountKey%22%3A%20%22748a6b56971b4bdf94ea73e4cc35e93f%22%2C%22accessLevel%22%3A%20%22member%22%2C%22%24initial\_referrer%22%3A%20%22https%3A%2F%2Fwww.mopub.com%2Fget-started%2F%22%2C%22%24initial\_referring\_domain%22%3A%20%22www.mopub.com%22%7D

Connection: close

```
{"addresses":["[REDACTED]@mailinator.com"],"key":"[REDACTED]"}
```

{code}

3. Send the captured request to INtruder and repeat the request in loop
4. Observe that email box is flooded with MoPub ad unit integration instructions

Remediation:

Rate limiting should be implemented

Regards

Akhil Kakkireni

**Impact**

E-mail bombs hack may create Denial of service (DoS) conditions against your e-mail software and even your network and Internet connection by taking up a large amount of bandwidth and, sometimes, requiring large amounts of storage space

2 attachments:

**F246270:** hackerone-Mopub.jpg

**F246271:** hackerone-Mopub2.jpg





mopub > MAX



Follow

**MoPub**

@mopub

MoPub has been acquired by AppLovin and our core Mediation and Marketplace solutions have been rolled into MAX, AppLovin's in-app mediation platform.

📍 Global [🔗 mopub.com](https://mopub.com) 📅 Joined October 2010

**1,700** Following **131K** Followers

Tweets

Tweets & replies

Media

Likes

📌 Pinned Tweet



**MoPub** @mopub · Jan 1

This is our final tweet! MoPub is migrating to AppLovin's in-app mediation platform, MAX. Follow [@AppLovin](#) for updates & info on how to migrate to MAX before MoPub sunsets on March 31. Contact your AppLovin or MoPub teams for migration support.

[#developer](#) [#mobiledev](#) [#AppLovin](#)

💬 18

↻ 22

❤️ 172





trabajoduro\_2 submitted a report to Twitter.

Dec 12th (4 years ago)

Hi Team,

I have found a logical flaw(NOT DoS) in the website '<https://app.mopub.com/>'

- 1. Use Burp Suite and capture below request upon navigation to *Code integration*
- 2. Click on Send button after entering email address in the input field of 'Enter one or more email addresses and we'll send you links to the integration instructions for this ad unit.'

```
{code}
POST /web-client/api/ad-units/email-instructions HTTP/1.1
Host: app.mopub.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: /
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://app.mopub.com/ad-unit?key=[REDACTED]&showIntegration=true
Content-Type: application/json
x-csrfToken: [REDACTED]
Content-Length: 88
Cookie: _ga=[REDACTED]; _gid=[REDACTED]; csrfToken=[REDACTED]; mp_mixpanel__c=8; sessionId=[REDACTED];
mp_c99579c4804fba6b8aeed7a911581652_mixpanel=%7B%22distinct_id%22%3A%20%22405f9ac1ce5749abb6092834819b3ec4%22%2C%22accountKey%22%3A%20%22748a6b56971b4bdf94ea73e4cc35e93f%22%2C%22accessLevel%22%3A%20%22member%22%2C%22%24initial_referrer%22%3A%20%22https%3A%2F%2Fwww.mopub.com%2Fget-started%2F%22%2C%22%24initial_referring_domain%22%3A%20%22www.mopub.com%22%7D
Connection: close
```

```
{"addresses":["[REDACTED]@mailinator.com"],"key":"[REDACTED]"}
```

- 3. Send the captured request to INtruder and repeat the request in loop
- 4. Observe that email box is flooded with MoPub ad unit integration instructions

Remediation:

Rate limiting should be implemented

Regards

Akhil Kakkireni

Impact

E-mail bombs hack may create Denial of service (DoS) conditions against your e-mail software and even your network and Internet connection by taking up a large amount of bandwidth and, sometimes, requiring large amounts of storage space

2 attachments:

F246270: hackerone-Mopub.jpg

F246271: hackerone-Mopub2.jpg

clear repro steps



trabajoduro\_2 submitted a report to **Twitter**. Dec 12th (4 years ago)

Hi Team,

I have found a logical flaw(NOT DoS) in the website '<https://app.mopub.com/>'

1. Use Burp Suite and capture below request upon navigation to *Code integration*
2. Click on Send button after entering email address in the input field of 'Enter one or more email addresses and we'll send you links to the integration instructions for this ad unit.'

```
{code}
POST /web-client/api/ad-units/email-instructions HTTP/1.1
Host: app.mopub.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: /
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://app.mopub.com/ad-unit?key=[REDACTED]&showIntegration=true
Content-Type: application/json
x-csrfToken: [REDACTED]
Content-Length: 88
Cookie: _ga=[REDACTED]; _gid=[REDACTED]; csrfToken=[REDACTED]; mp_mixpanel__c=8; sessionId=[REDACTED];
mp_c99579c4804fba6b8aeed7a911581652_mixpanel=%7B%22distinct_id%22%3A%20%22405f9ac1ce5749abb6092834819b3ec4%22%2C%22accountKey%22%3A%20%22748a6b56971b4bdf94ea73e4cc35e93f%22%2C%22accessLevel%22%3A%20%22member%22%2C%22%24initial_referrer%22%3A%20%22https%3A%2F%2Fwww.mopub.com%2Fget-started%2F%22%2C%22%24initial_referring_domain%22%3A%20%22www.mopub.com%22%7D
Connection: close

{"addresses":["[REDACTED]@mailinator.com"],"key":"[REDACTED]"}
```

3. Send the captured request to INtruder and repeat the request in loop
4. Observe that email box is flooded with MoPub ad unit integration instructions

Remediation:  
Rate limiting should be implemented

Regards  
Akhil Kakkireni

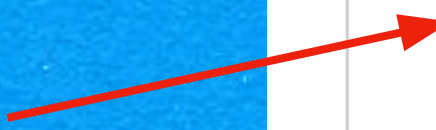
**Impact**

E-mail bombs hack may create Denial of service (DoS) conditions against your e-mail software and even your network and Internet connection by taking up a large amount of bandwidth and, sometimes, requiring large amounts of storage space

2 attachments:  
**F246270:** hackerone-Mopub.jpg  
**F246271:** hackerone-Mopub2.jpg



makes an attempt to estimate impact (good), but does not tie potential impact directly to Twitter users



E-mail bombs hack may create Denial of service (DoS) conditions against your e-mail software and even your network and Internet connection by taking up a large amount of bandwidth and, sometimes, requiring large amounts of storage space





trabajoduro\_2 submitted a report to Twitter.

Dec 12th (4 years ago)

Hi Team,

I have found a logical flaw(NOT DoS) in the website '<https://app.mopub.com/>'

- 1. Use Burp Suite and capture below request upon navigation to *Code integration*
- 2. Click on Send button after entering email address in the input field of 'Enter one or more email addresses and we'll send you links to the integration instructions for this ad unit.'

{code}

POST /web-client/api/ad-units/email-instructions HTTP/1.1

Host: app.mopub.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0

Accept: /

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

Accept-Encoding: gzip, deflate

Referer: [https://app.mopub.com/ad-unit?key=\[redacted\]&showIntegration=true](https://app.mopub.com/ad-unit?key=[redacted]&showIntegration=true)

Content-Type: application/json

x-csrfToken: [redacted]

Content-Length: 88

Cookie: \_ga=[redacted]; \_gid=[redacted]; csrfToken=[redacted]; mp\_mixpanel\_\_c=8; sessionId=[redacted];

mp\_c99579c4804fba6b8aeed7a911581652\_mixpanel=%7B%22distinct\_id%22%3A%20%22405f9ac1ce5749abb6092834819b3ec4%22%2C%22accountKey%22%3A%20%22748a6b56971b4bdf94ea73e4cc35e93f%22%2C%22accessLevel%22%3A%20%22member%22%2C%22%24initial\_referrer%22%3A%20%22https%3A%2F%2Fwww.mopub.com%2Fget-

started%2F%22%2C%22%24initial\_referring\_domain%22%3A%20%22www.mopub.com%22%7D

Connection: close

{\"addresses\": [\"[redacted]@mailinator.com\"], \"key\": \"[redacted]\"}

{code}

- 3. Send the captured request to INtruder and repeat the request in loop

- 4. Observe that email box is flooded with MoPub ad unit integration instructions

Remediation:

Rate limiting should be implemented

Regards

Akhil Kakkireni

**Impact**

E-mail bombs hack may create Denial of service (DoS) conditions against your e-mail software and even your network and Internet connection by taking up a large amount of bandwidth and, sometimes, requiring large amounts of storage space

2 attachments:

F246270: hackerone-Mopub.jpg

F246271: hackerone-Mopub2.jpg

actual customer impact





HackerOne, Inc. [US] | https://hackerone.com/twitter/reports/new

Hacktivity Directory Inbox

### Intruder attack 3

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
107	null	200	<input type="checkbox"/>	<input type="checkbox"/>	433	
108	null	200	<input type="checkbox"/>	<input type="checkbox"/>	433	
109	null	200	<input type="checkbox"/>	<input type="checkbox"/>	433	
110	null	200	<input type="checkbox"/>	<input type="checkbox"/>	433	
111	null	200	<input type="checkbox"/>	<input type="checkbox"/>	433	
112	null	200	<input type="checkbox"/>	<input type="checkbox"/>	433	
113	null	200	<input type="checkbox"/>	<input type="checkbox"/>	433	
114	null	200	<input type="checkbox"/>	<input type="checkbox"/>	433	
115	null	200	<input type="checkbox"/>	<input type="checkbox"/>	433	
116	null	200	<input type="checkbox"/>	<input type="checkbox"/>	433	
117	null	200	<input type="checkbox"/>	<input type="checkbox"/>	433	
118	null	200	<input type="checkbox"/>	<input type="checkbox"/>	433	
119	null	200	<input type="checkbox"/>	<input type="checkbox"/>	433	
120	null	200	<input type="checkbox"/>	<input type="checkbox"/>	433	

By clicking 120

2:22 AM 12/13/2017



Browser tabs: Zim, Twi, em: #87, #27, imp, Rat, Wh, #17, #17, Sub, #22, #22, #50, Twi, x, hov, BT Bur

Address bar: Secure | https://www.mailinator.com/v2/inbox.jsp?zone=public&query=akhilkakkireni#/inboxpane

MAILINATOR Log in Sign up

akhilkakkireni privacy-level: public query: akhilkakkireni@mailinator.com

Public Inboxes @mailinator

akhilkakkireni	50
----------------	----

Using Mailinator for QA? [Upgrade Now](#)

Donate Bitcoin: 15QPnuQVLJU2Z9iU5KEzML6egYvs726o5Z

Links monetized by ClickRouter

<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	moments ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	moments ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	moments ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	moments ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	moments ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	minute ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	minute ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	minute ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	minute ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	2 minutes ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	2 minutes ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	2 minutes ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	2 minutes ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	2 minutes ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	2 minutes ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	2 minutes ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	3 minutes ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	3 minutes ago
<input type="checkbox"/>	no-reply	MoPub ad unit integration instructions	3 minutes ago

PRICING | FAQ | API DOCS | BLOG

Copyright 2017 Manybrain, Inc. All Right Reserved. [Terms and Conditions](#) and [Privacy Policy](#) [support@manybrain.com](mailto:support@manybrain.com)

Taskbar: Windows, IE, File Explorer, Chrome, Firefox, Outlook, Skype, Word, OneDrive, Mail, Taskbar, System tray: 2:23 AM 12/13/2017



# starting points

- Disclaimer: not the process of email bomb vuln reporter; some tactics and tools they could have used...
- [Burp Community Edition](#) (free!)
- Portswigger (Burp maker) [intruder tutorial](#)
- How'd they find this thing in the first place? Probably [recon](#)
  - Collect live subdomains in scope ("wide recon"): [subfinder](#), [amass](#), [masscan](#), Google dorks...
  - Is anything interesting hosted there ("narrow recon")? [httprobe](#), [dirsearch](#), [wfuuzz](#), [and more](#)
  - and [even more](#) (and [more!](#))



Burp Suite Professional v2021.9.1 - Temporary Project  
 Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

**Tasks** New scan New live task Pause Settings Help Refresh

Filter Running Paused Finished Live task Scan Intruder attack Search...

**Issue activity** Filter High Medium Low Info Certain Firm Tentative Search...

#	Task	Time	Action	Issue type
29	2	10:57:36 3 Apr 2022	Issue found	Strict transport security not enforced
27	2	10:57:36 3 Apr 2022	Issue found	Strict transport security not enforced
24	2	10:57:36 3 Apr 2022	Issue found	Strict transport security not enforced
23	2	10:57:36 3 Apr 2022	Issue found	Strict transport security not enforced
21	2	10:57:36 3 Apr 2022	Issue found	Strict transport security not enforced
20	2	10:57:36 3 Apr 2022	Issue found	Strict transport security not enforced
18	2	10:57:36 3 Apr 2022	Issue found	Strict transport security not enforced
11	2	10:57:36 3 Apr 2022	Issue found	Strict transport security not enforced
9	2	10:57:36 3 Apr 2022	Issue found	Strict transport security not enforced
1	2	10:57:31 3 Apr 2022	Issue found	Strict transport security not enforced
6	2	10:57:36 3 Apr 2022	Issue found	Email addresses disclosed
15	2	10:57:36 3 Apr 2022	Issue found	Cross-domain script include
4	2	10:57:32 3 Apr 2022	Issue found	Cross-domain script include
19	2	10:57:36 3 Apr 2022	Issue found	Cross-domain Referer leakage
28	2	10:57:36 3 Apr 2022	Issue found	Cookie without HttpOnly flag set

**Event log** Filter Critical Error Info Debug Search...

Time	Type	Source	Message
10:57:33 3 Apr 2022	Info	Proxy	api2.branch.io is using HTTP/2
10:57:33 3 Apr 2022	Info	Proxy	app.link is using HTTP/2
10:57:33 3 Apr 2022	Info	Proxy	accounts.google.com is using HTTP/2
10:57:32 3 Apr 2022	Info	Proxy	api.twitter.com is using HTTP/2
10:57:31 3 Apr 2022	Info	Proxy	abs.twimg.com is using HTTP/2
10:57:30 3 Apr 2022	Info	Proxy	www.googleapis.com is using HTTP/2
10:57:30 3 Apr 2022	Info	Proxy	twitter.com is using HTTP/2
10:57:00 3 Apr 2022	Info	Suite	This version of Burp Suite was released over three months ago
10:56:59 3 Apr 2022	Info	Proxy	Proxy service started on 127.0.0.1:8080

**Advisory** Request Response

**Strict transport security not enforced**

Issue: **Strict transport security not enforced**  
 Severity: **Low**  
 Confidence: **Certain**  
 Host: **https://api.twitter.com**  
 Path: **/1.1/guest/activate.json**

**Issue description**  
 The application fails to prevent users from connecting to it over unencrypted connections. An attacker able to modify a legitimate user's network traffic could bypass the application's use of SSL/TLS encryption, and use the application as a platform for attacks against its users. This attack is performed by rewriting HTTPS links as HTTP, so that if a targeted user follows a link to the site from an HTTP page, their browser never attempts to use an encrypted connection. The sslstrip tool automates this process.

To exploit this vulnerability, an attacker must be suitably positioned to intercept and modify the

```

cat staging-apps.txt
https://staging.example.com
https://staging.admin.example.com
https://staging.crm.example.com
https://api-staging.example.com
https://internal.example.com
https://build-app.example.com
https://demo.example.com
https://preprod.backend-api.example.com
  
```

```
nuclei -t amazon-mww-secret-leak.yaml -l staging-apps.txt
```



projectdiscovery.io

```

[WRN] Use with caution. You are responsible for your actions
[WRN] Developers assume no liability and are not responsible for any misuse or damage.
[INF] Loading templates...
[INF] [amazon-mww-secret-leak] Amazon MWS Auth Token leak (@puzzlepeaches) [medium]
[INF] Using 1 rules (1 templates, 0 workflows)
[amazon-mww-secret-leak] [http] [medium] https://internal.example.com
[amazon-mww-secret-leak] [http] [medium] https://build-app.example.com
[amazon-mww-secret-leak] [http] [medium] https://staging.admin.example.com
  
```

<https://github.com/projectdiscovery/nuclei>



# Dupes :(

<b>CVE-2018-6389</b>	18th Sep 2020	Rejected - duplicate, already reported
<b>DOS on main domain</b>	18th Sep 2020	Rejected - duplicate, already reported
<b>Subdomain Takeover</b>	2nd Sep 2020	Rejected - duplicate, already reported
<b>Unauthenticated Popular Filters on Jira</b>	25th Aug 2020	Rejected - duplicate, already reported
<b>Access to video call without authorization &amp; user enumeration</b>	19th Aug 2020	Rejected - duplicate, already reported
<b>Subdomain Takeover</b>	19th Aug 2020	Rejected - duplicate, already reported
<b>Vulnerability Report</b>	14th Aug 2020	Rejected - duplicate, already reported
<b>Vulnerability Report #3</b>	10th Aug 2020	Rejected - duplicate, already reported
<b>Vulnerability Report</b>	10th Aug 2020	Rejected - duplicate, already reported
<b>Vulnerability Report</b>	7th Aug 2020	Rejected - duplicate, already reported

<https://safaras.medium.com/tired-of-duplicates-in-bug-bounty-b34d786fe6a4>



dagmd · il y a 2 a

You can always ask them to invite you to the original report as well. Then you can verify yourself that your report is a duplicate.

As a rule of thumb:

- if your scanner found it, it's probably a duplicate. Other researchers use the the same tools
- if it's something easy/quick to find for you, it's probably a duplicate.
- if you invested many hours in it or wrote custom scripts to exploit it, you should be more suspicious.

Also, programs by non-tech companies probably have a higher duplicate rate, as the take longer to solve the vulnerabilities.



2



Répondre Partager ...



# Pentesting vs bug hunting

- **pentesting**
  - likely “first” on a target, closer to private program bug hunting
  - smaller scope can mean less need for wide recon
  - access to documentation (sometimes)
  - grey or white box testing (sometimes)
  - salaried or contracted through consulting company by engagement



# Pentesting vs bug hunting

- **pentesting**
  - likely “first” on a target, closer to private program bug hunting
  - smaller scope can mean less need for wide recon
  - access to documentation (sometimes)
  - grey or white box testing (sometimes)
  - salaried or contracted through consulting company by engagement
- **general-purpose bug hunting**
  - low-hanging fruit likely gone (dupes)
  - payment not guaranteed
  - more typically black box
  - wider scope generally





# Pentesting vs bug hunting

- **pentesting**
  - likely “first” on a target, closer to private program bug hunting
  - smaller scope can mean less need for wide recon
  - access to documentation (sometimes)
  - grey or white box testing (sometimes)
  - paid by contract (salary or engagement)
- **general-purpose bug hunting**
  - low-hanging fruit likely gone (dupes)
  - payment not guaranteed
  - more typically black box
  - wider scope generally
- **both**
  - stick to scope
  - occasional use of common tooling
  - automation, custom tooling
  - good writing & communication skills!



**questions before we continue?**

# **example: missing API permissions**

*by ryotak, 2020*

*<https://hackerone.com/reports/1032468>*



ryotak submitted a report to [Twitter](#).

Nov 12th (about 1 year ago)



### Summary:

Twitter released [Fleet](#) yesterday. This feature is working with few APIs, and these APIs are missing permission checks.

### Description:

In `/fleets/v1/create` of `https://api.twitter.com`, there is no check to whether if the application has permission to write to the account. `/fleets/v1/delete` has also this issue.

### Steps To Reproduce:

1. Install [twurl](#).
2. Authenticate as a read-only application.
3. Execute following command: `twurl /fleets/v1/create -X POST --header 'Content-Type: application/json' -d '{"text":"Hey yo"}'`
4. A fleet with `Hey yo` text will be created.

### Supporting Material/References:

Video F1075380: 2020-11-12\_21-28-47.mp4 2.96 MiB

[Zoom in](#) [Zoom out](#) [Copy](#) [Download](#)



### Impact

The read-only application can publish fleets without getting Write permission. This issue has a similar impact to [#434763](#)

1 attachment:

F1075380: 2020-11-12\_21-28-47.mp4



# Twitter rolls out Stories, aka 'Fleets,' to all users; will also test a Clubhouse rival

Sarah Perez @sarahintampa / 6:00 AM MST • November 17, 2020

 Comment

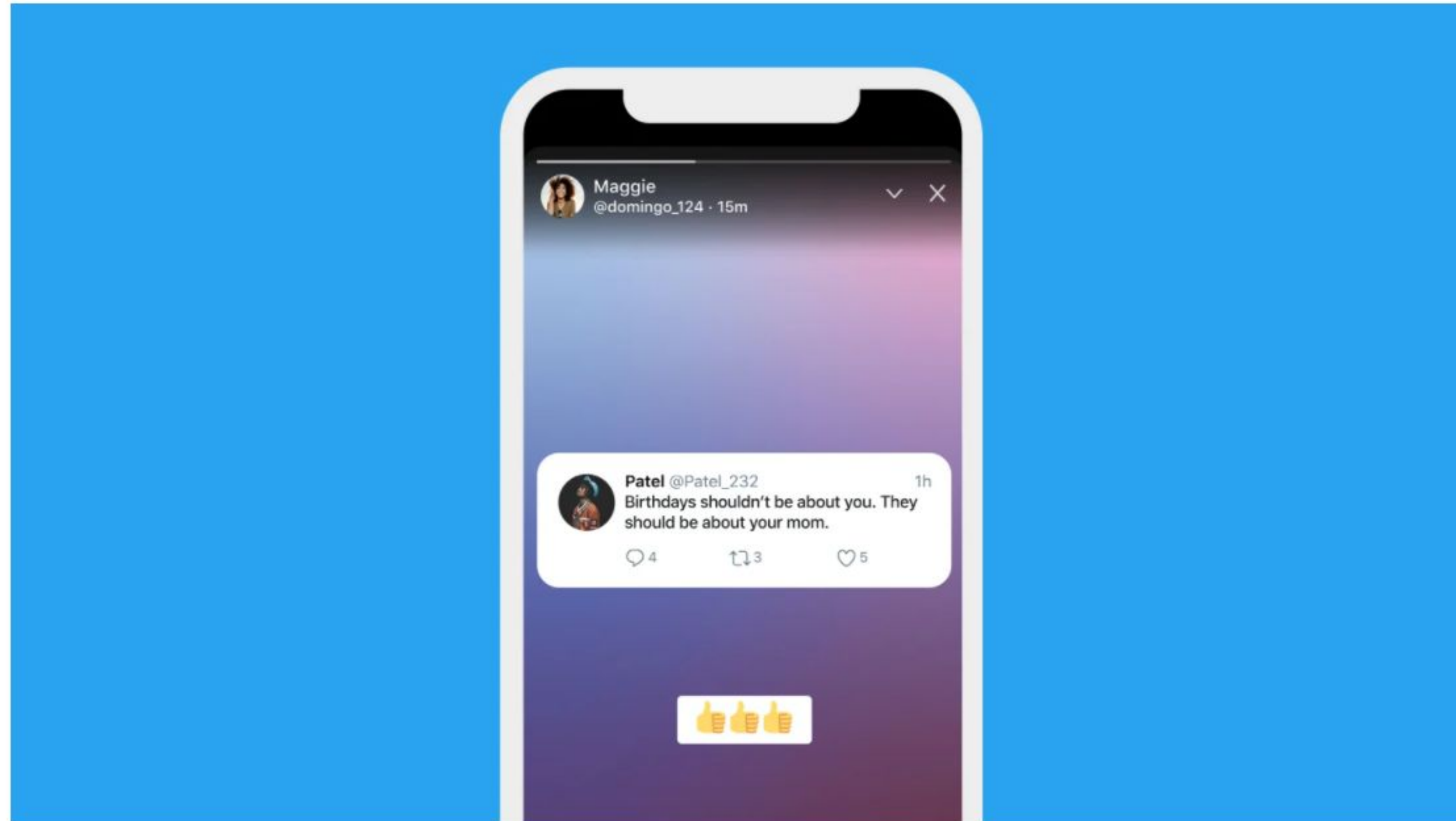


 Image Credits: Twitter

Twitter this morning is launching its own version of Stories — aka “Fleets” — to its global user base. The product, which allows users to post ephemeral content that disappears in 24 hours, had already rolled out to select markets, including Brazil, India, Italy, South Korea and, most recently, Japan. The company, in a press briefing on Monday, also revealed its plans to test an audio-based social networking feature similar to the controversial app Clubhouse.



ryotak submitted a report to **Twitter**.

Nov 12th (about 1 year ago)

### Summary:

Twitter released **Fleet** yesterday. This feature is working with few APIs, and these APIs are missing permission checks.

### Description:

In `/fleets/v1/create` of `https://api.twitter.com`, there is no check to whether if the application has permission to write to the account. `/fleets/v1/delete` has also this issue.

### Steps To Reproduce:

1. Install `twurl`.
2. Authenticate as a read-only application.
3. Execute following command: `twurl /fleets/v1/create -X POST --header 'Content-Type: application/json' -d '{"text":"Hey yo"}'`
4. A fleet with `Hey yo` text will be created.

### Supporting Material/References:

Video F1075380: 2020-11-12\_21-28-47.mp4 2.96 MiB

[Zoom in](#) [Zoom out](#) [Copy](#) [Download](#)



### Impact

The read-only application can publish fleets without getting Write permission. This issue has a similar impact to [#434763](#)

1 attachment:

F1075380: 2020-11-12\_21-28-47.mp4

problem tl;dr



ryotak submitted a report to [Twitter](#).

Nov 12th (about 1 year ago)

### Summary:

Twitter released [Fleet](#) yesterday. This feature is working with few APIs, and these APIs are missing permission checks.

### Description:

In `/fleets/v1/create` of `https://api.twitter.com`, there is no check to whether if the application has permission to write to the account. `/fleets/v1/delete` has also this issue.

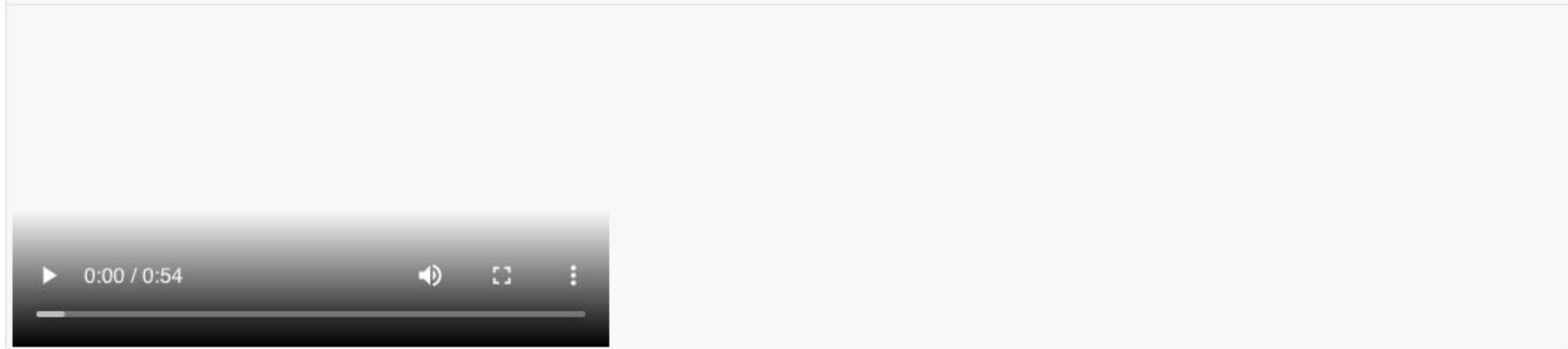
### Steps To Reproduce:

1. Install [twurl](#).
2. Authenticate as a read-only application.
3. Execute following command: `twurl /fleets/v1/create -X POST --header 'Content-Type: application/json' -d '{"text":"Hey yo"}'`
4. A fleet with `Hey yo` text will be created.

### Supporting Material/References:

Video F1075380: 2020-11-12\_21-28-47.mp4 2.96 MiB

[Zoom in](#) [Zoom out](#) [Copy](#) [Download](#)



### Impact

The read-only application can publish fleets without getting Write permission. This issue has a similar impact to [#434763](#)

1 attachment:

F1075380: 2020-11-12\_21-28-47.mp4

clear impact



**“The read-only application can publish Fleets without getting Write permission”**



**“When a **Twitter user** authorizes a read-only application on **their account**, at time of writing the **read-only app can publish or delete Fleets without the Write permission** since Fleets APIs lack application permissions checks”**





ryotak submitted a report to [Twitter](#).

Nov 12th (about 1 year ago)

### Summary:

Twitter released [Fleet](#) yesterday. This feature is working with few APIs, and these APIs are missing permission checks.

### Description:

In `/fleets/v1/create` of `https://api.twitter.com`, there is no check to whether if the application has permission to write to the account. `/fleets/v1/delete` has also this issue.

### Steps To Reproduce:

1. Install [twurl](#).
2. Authenticate as a read-only application.
3. Execute following command: `twurl /fleets/v1/create -X POST --header 'Content-Type: application/json' -d '{"text":"Hey yo"}'`
4. A fleet with `Hey yo` text will be created.

### Supporting Material/References:

Video F1075380: 2020-11-12\_21-28-47.mp4 2.96 MiB

[Zoom in](#) [Zoom out](#) [Copy](#) [Download](#)



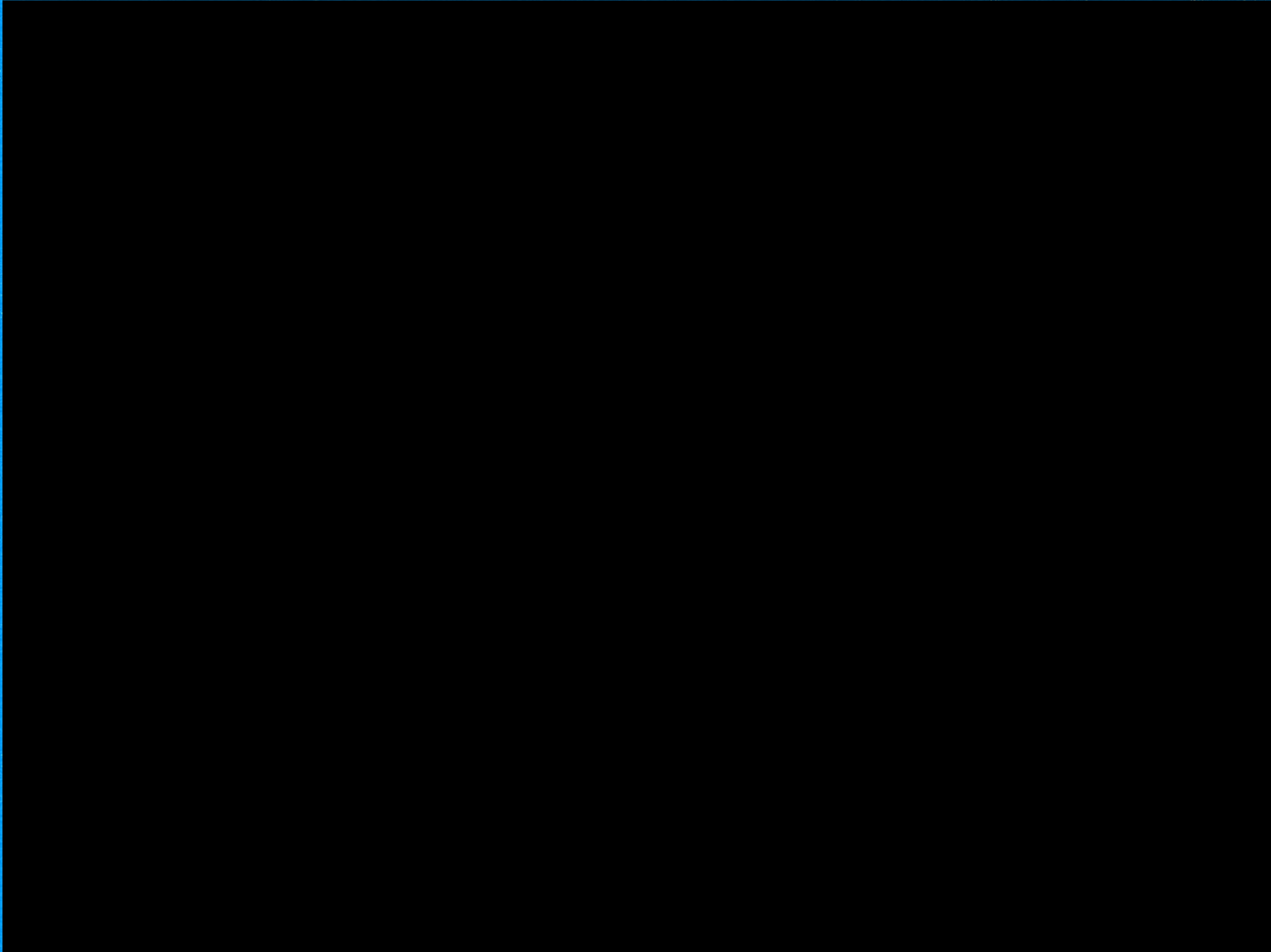
### Impact

The read-only application can publish fleets without getting Write permission. This issue has a similar impact to [#434763](#)

1 attachment:

F1075380: 2020-11-12\_21-28-47.mp4

clear repro  
steps





# reporter found & acked similar-feeling issue

## Impact

The read-only application can publish fleets without getting Write permission. This issue has a similar impact to [#434763](#)

71

#434763

**Incorrect details on OAuth permissions screen allows DMs to be read without permission**Share:     

### SUMMARY BY TWITTER



The reporter discovered that when a select set of applications are authenticated using a PIN or non-intended OAuth flow, the permission dialog that is shown may not show the permissions that the application has. We do not believe anyone was misled by the permissions that these applications had or that their data was unintentionally accessed by the Twitter for iPhone or Twitter for Google TV applications as those applications use other authentication flows. To our knowledge, there was not a breach of anyone's information due to this issue. There are no actions people need to take at this time.



# starting points

- Disclaimer: not process of API permissions vuln reporter; some tactics and tools they could have used...
- general understanding of how permissions work for other (Twitter) HTTP API endpoints
- **twurl**: OAuthed curl wrapper for interacting with API endpoints
- read other disclosed reports from program, potentially determine likely payout and also issues that would be potentially considered too similar / duplicates
- Portswigger **OAuth tutorial**, **access control tutorial**
- ryotak's writeup (in Japanese): <https://blog.ryotak.me/post/twitter-privesc/>

# **example: video player clickjacking**

*by filedescriptor, 2015 (disclosed 2018)*

*<https://hackerone.com/reports/85624>*



filedescrptor submitted a report to **Twitter**.

Aug 30th (7 years ago)

Hi,

I would like to report an issue where player card is vulnerable to clickjacking in certain browsers. This may result in something similar to XSS worm and many other critical damages.

### Details

Twitter Player Card allows a website to embed a custom player(html) into an iframe in a tweet. There are currently 2-3 security features in place to defend clickjacking on Twitter:

1. `X-Frame-Options: SAMEORIGIN` covering the whole twitter.com domain
2. `Content-Security-Policy: frame-ancestors 'self'` ditto
3. JS-based frame-buster in some pages (but not all)

For (1), SAMEORIGIN only checks if the embedded frame is on the same origin of the top window. For example, attacker can do something like twitter.com -> attacker.com -> twitter.com to evade it. More details can be seen from here: [https://bugzilla.mozilla.org/show\\_bug.cgi?id=725490](https://bugzilla.mozilla.org/show_bug.cgi?id=725490)

For (2), this is the only way to correctly prevent framing from other websites (it performs the check against the ancestor list). However this is a CSP2 directive so not all browsers support it. For example, Safari and IE do not support it.

For (3), using the sandbox attribute of iframe can disable JS of a frame, hence anti-frame-buster

Since Player Card is shown on a Tweet (on twitter.com), attacker can embed an iframe which embeds a Twitter page so that attacker can overlay it with "bait" content to lure victims to click on it.

The impact is huge because of the following facts:

- The card displays directly on the user's timeline, making the attack less suspicious to normal clickjacking
- The click is very subtle that victims cannot notice what's happened behind the scene
- Wormable because attacker can make victims tweet arbitrary content to spread it
- Can perform click-based critical actions, like follow, retweet, favorite... etc
- If sent as promoted tweet, it can target even more victims, also player is directly expanded

### Repo step

1. Clone the Player Card started bundle here: <https://github.com/twitterdev/cards>
2. Change the card's property `twitter:player` to a custom HTML file
3. In the HTML file, embed iframe to a Twitter page, e.g. `<iframe src="//twitter.com"></iframe>`
4. Post the link in a Tweet (make sure the domain is white-listed)
5. Expand the tweet in Safari or IE, it will show that a Twitter page is embedded

Documentation of Player Card: <https://dev.twitter.com/cards/types/player>

### PoC

<https://twitter.com/AttackerCanvas/status/637859735501279232> (Open with Safari or IE)

Video demo: <https://vimeo.com/137725491> (password: click)

The PoC demonstrates how the attack can be conducted. There will be a fake video to lure victims to click to play it. After clicking the victim will automatically post a tweet with content "Pwn3d!".





**Jonathan Cipriano**

@joncipriano



The dusk and dawn light in [@DeathValleyNPS](#) is amazing. I visit almost every year for [#photography](#).



youtube.com

Death Valley Wilderness: Wilderness Light

Follow the course of light through the Death Valley Wilderness and observe the obvious and subtle ...

10:49 AM · Feb 22, 2016



40



Reply



Copy link to Tweet

[Read 5 replies](#)



# clickjacking?

- in researcher's own words:



filedescrptor posted a comment.

Updated Sep 3rd (7 years ago)

Here is a more delicate and stealthy PoC with maximum damage:

<https://twitter.com/AttackerCanvas/status/639465287629144064>

Video demo: <https://vimeo.com/138212863> (password: app)

Victim will see it as a normal embedded YouTube video, and it can be interacted properly (click to play). Victims will definitely not notice anything. Behind the scene the victim is actually clicking on the Authorize button on a fake app with full read, write permission, so that attacker can gain long-term control over the victim. Besides it is easy to fake an legitimate app (like Twitter for iPhone), so even victims notice there is a new app that has access to his/her account he/she would not be able to distinguish it.

poc: <https://vimeo.com/138212863>

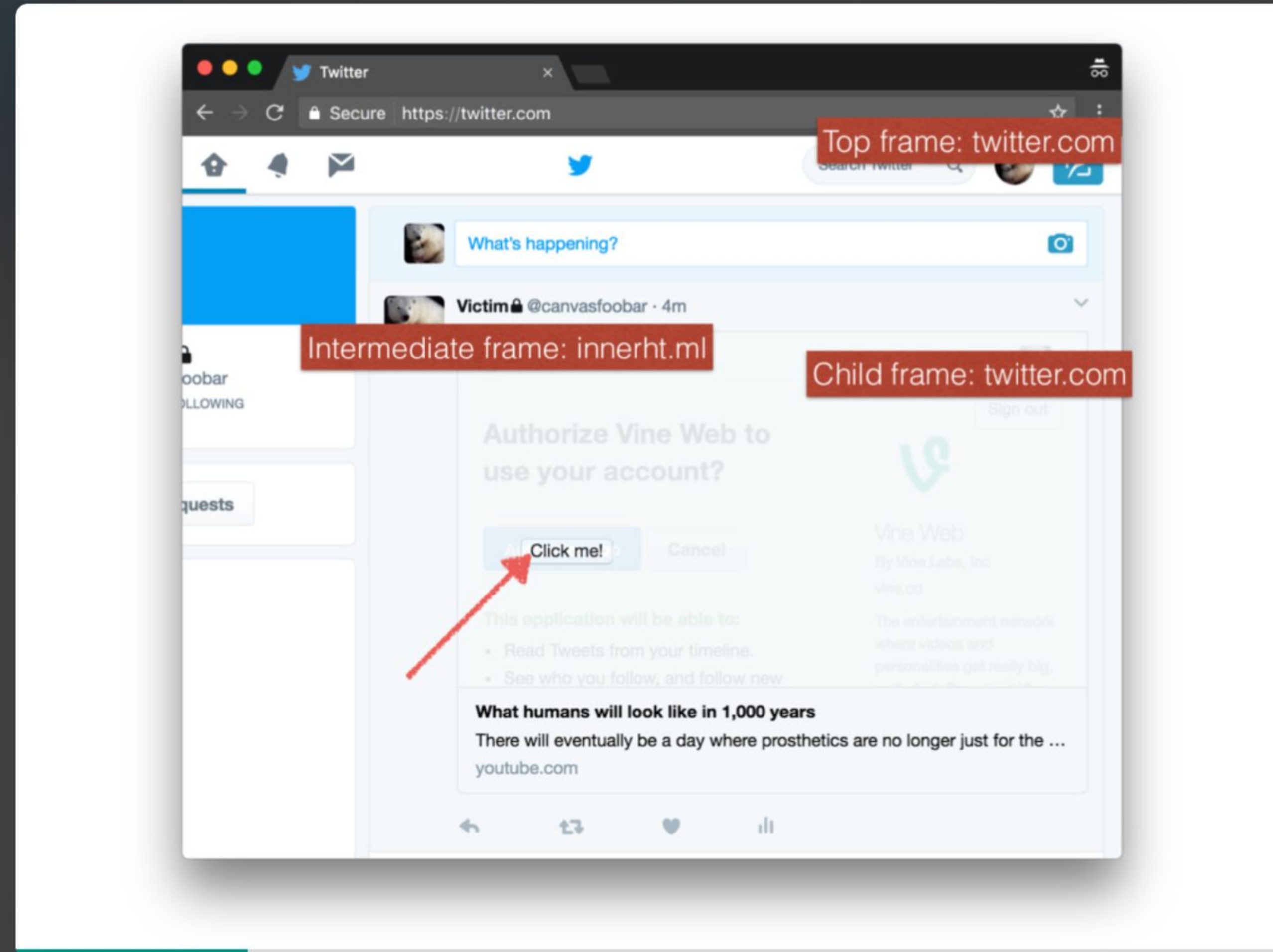
- **impact**: app can Tweet from victim account without consent

The PoC demonstrates how the attack can be conducted. There will be a fake video to lure victims to click to play it. After clicking the victim will automatically post a tweet with content "Pwn3d!".





# Exploiting the unexploitable with lesser known browser tricks



tl;dr of problem

TIMELINE

filedescriptor submitted a report to **Twitter**. Aug 30th (7 years ago)

Hi.

I would like to report an issue where player card is vulnerable to clickjacking in certain browsers. This may result in something similar to XSS worm and many other critical damages.

**Details**

Twitter Player Card allows a website to embed a custom player(html) into an iframe in a tweet. There are currently 2-3 security features in place to defend clickjacking on Twitter:

1. `X-Frame-Options: SAMEORIGIN` covering the whole twitter.com domain
2. `Content-Security-Policy: frame-ancestors 'self'` ditto
3. JS-based frame-buster in some pages (but not all)

For (1), SAMEORIGIN only checks if the embedded frame is on the same origin of the top window. For example, attacker can do something like twitter.com -> attacker.com -> twitter.com to evade it. More details can be seen from here: [https://bugzilla.mozilla.org/show\\_bug.cgi?id=725490](https://bugzilla.mozilla.org/show_bug.cgi?id=725490)

For (2), this is the only way to correctly prevent framing from other websites (it performs the check against the ancestor list). However this is a CSP2 directive so not all browsers support it. For example, Safari and IE do not support it.

For (3), using the sandbox attribute of iframe can disable JS of a frame, hence anti-frame-buster

Since Player Card is shown on a Tweet (on twitter.com), attacker can embed an iframe which embeds a Twitter page so that attacker can overlay it with "bait" content to lure victims to click on it.

The impact is huge because of the following facts:

- The card displays directly on the user's timeline, making the attack less suspicious to normal clickjacking
- The click is very subtle that victims cannot notice what's happened behind the scene
- Wormable because attacker can make victims tweet arbitrary content to spread it
- Can perform click-based critical actions, like follow, retweet, favorite... etc
- If sent as promoted tweet, it can target even more victims, also player is directly expanded

**Repo step**

1. Clone the Player Card started bundle here: <https://github.com/twitterdev/cards>
2. Change the card's property `twitter:player` to a custom HTML file
3. In the HTML file, embed iframe to a Twitter page, e.g. `<iframe src="//twitter.com"></iframe>`
4. Post the link in a Tweet (make sure the domain is white-listed)
5. Expand the tweet in Safari or IE, it will show that a Twitter page is embedded

Documentation of Player Card: <https://dev.twitter.com/cards/types/player>

**PoC**

<https://twitter.com/AttackerCanvas/status/637859735501279232> (Open with Safari or IE)

Video demo: <https://vimeo.com/137725491> (password: click)

The PoC demonstrates how the attack can be conducted. There will be a fake video to lure victims to click to play it. After clicking the victim will automatically post a tweet with content "Pwn3d!".



TIMELINE

filedescriptor submitted a report to **Twitter**. Aug 30th (7 years ago)

Hi,

I would like to report an issue where player card is vulnerable to clickjacking in certain browsers. This may result in something similar to XSS worm and many other critical damages.

**Details**

Twitter Player Card allows a website to embed a custom player(html) into an iframe in a tweet. There are currently 2-3 security features in place to defend clickjacking on Twitter:

1. `X-Frame-Options: SAMEORIGIN` covering the whole twitter.com domain
2. `Content-Security-Policy: frame-ancestors 'self'` ditto
3. JS-based frame-buster in some pages (but not all)

For (1), SAMEORIGIN only checks if the embedded frame is on the same origin of the top window. For example, attacker can do something like `twitter.com -> attacker.com -> twitter.com` to evade it. More details can be seen from here: [https://bugzilla.mozilla.org/show\\_bug.cgi?id=725490](https://bugzilla.mozilla.org/show_bug.cgi?id=725490)

For (2), this is the only way to correctly prevent framing from other websites (it performs the check against the ancestor list). However this is a CSP2 directive so not all browsers support it. For example, Safari and IE do not support it.

For (3), using the sandbox attribute of iframe can disable JS of a frame, hence anti-frame-buster

Since Player Card is shown on a Tweet (on twitter.com), attacker can embed an iframe which embeds a Twitter page so that attacker can overlay it with "bait" content to lure victims to click on it.

The impact is huge because of the following facts:

- The card displays directly on the user's timeline, making the attack less suspicious to normal clickjacking
- The click is very subtle that victims cannot notice what's happened behind the scene
- Wormable because attacker can make victims tweet arbitrary content to spread it
- Can perform click-based critical actions, like follow, retweet, favorite... etc
- If sent as promoted tweet, it can target even more victims, also player is directly expanded

**Repo step**

1. Clone the Player Card started bundle here: <https://github.com/twitterdev/cards>
2. Change the card's property `twitter:player` to a custom HTML file
3. In the HTML file, embed iframe to a Twitter page, e.g. `<iframe src="//twitter.com"></iframe>`
4. Post the link in a Tweet (make sure the domain is white-listed)
5. Expand the tweet in Safari or IE, it will show that a Twitter page is embedded

Documentation of Player Card: <https://dev.twitter.com/cards/types/player>

**PoC**

<https://twitter.com/AttackerCanvas/status/637859735501279232> (Open with Safari or IE)

Video demo: <https://vimeo.com/137725491> (password: click)

The PoC demonstrates how the attack can be conducted. There will be a fake video to lure victims to click to play it. After clicking the victim will automatically post a tweet with content "Pwn3d!".



what is the current solution if there is one?



filedescriptor submitted a report to **Twitter**.

Aug 30th (7 years ago)

Hi,

I would like to report an issue where player card is vulnerable to clickjacking in certain browsers. This may result in something similar to XSS worm and many other critical damages.

### Details

Twitter Player Card allows a website to embed a custom player(html) into an iframe in a tweet. There are currently 2-3 security features in place to defend clickjacking on Twitter:

1. `X-Frame-Options: SAMEORIGIN` covering the whole twitter.com domain
2. `Content-Security-Policy: frame-ancestors 'self'` ditto
3. JS-based frame-buster in some pages (but not all)

For (1), SAMEORIGIN only checks if the embedded frame is on the same origin of the top window. For example, attacker can do something like twitter.com -> attacker.com -> twitter.com to evade it. More details can be seen from here: [https://bugzilla.mozilla.org/show\\_bug.cgi?id=725490](https://bugzilla.mozilla.org/show_bug.cgi?id=725490)

For (2), this is the only way to correctly prevent framing from other websites (it performs the check against the ancestor list). However this is a CSP2 directive so not all browsers support it. For example, Safari and IE do not support it.

For (3), using the sandbox attribute of iframe can disable JS of a frame, hence anti-frame-buster

Since Player Card is shown on a Tweet (on twitter.com), attacker can embed an iframe which embeds a Twitter page so that attacker can overlay it with "bait" content to lure victims to click on it.

The impact is huge because of the following facts:

- The card displays directly on the user's timeline, making the attack less suspicious to normal clickjacking
- The click is very subtle that victims cannot notice what's happened behind the scene
- Wormable because attacker can make victims tweet arbitrary content to spread it
- Can perform click-based critical actions, like follow, retweet, favorite... etc
- If sent as promoted tweet, it can target even more victims, also player is directly expanded

### Repo step

1. Clone the Player Card started bundle here: <https://github.com/twitterdev/cards>
2. Change the card's property `twitter:player` to a custom HTML file
3. In the HTML file, embed iframe to a Twitter page, e.g. `<iframe src="//twitter.com"></iframe>`
4. Post the link in a Tweet (make sure the domain is white-listed)
5. Expand the tweet in Safari or IE, it will show that a Twitter page is embedded

Documentation of Player Card: <https://dev.twitter.com/cards/types/player>

### PoC

<https://twitter.com/AttackerCanvas/status/637859735501279232> (Open with Safari or IE)

Video demo: <https://vimeo.com/137725491> (password: click)

The PoC demonstrates how the attack can be conducted. There will be a fake video to lure victims to click to play it. After clicking the victim will automatically post a tweet with content "Pwn3d!".



demonstrates impact in a way directly relevant to the user



filedescrptor submitted a report to **Twitter**.

Aug 30th (7 years ago)

Hi,

I would like to report an issue where player card is vulnerable to clickjacking in certain browsers. This may result in something similar to XSS worm and many other critical damages.

### Details

Twitter Player Card allows a website to embed a custom player(html) into an iframe in a tweet. There are currently 2-3 security features in place to defend clickjacking on Twitter:

1. `X-Frame-Options: SAMEORIGIN` covering the whole twitter.com domain
2. `Content-Security-Policy: frame-ancestors 'self'` ditto
3. JS-based frame-buster in some pages (but not all)

For (1), SAMEORIGIN only checks if the embedded frame is on the same origin of the top window. For example, attacker can do something like twitter.com -> attacker.com -> twitter.com to evade it. More details can be seen from here: [https://bugzilla.mozilla.org/show\\_bug.cgi?id=725490](https://bugzilla.mozilla.org/show_bug.cgi?id=725490)

For (2), this is the only way to correctly prevent framing from other websites (it performs the check against the ancestor list). However this is a CSP2 directive so not all browsers support it. For example, Safari and IE do not support it.

For (3), using the sandbox attribute of iframe can disable JS of a frame, hence anti-frame-buster

Since Player Card is shown on a Tweet (on twitter.com), attacker can embed an iframe which embeds a Twitter page so that attacker can overlay it with "bait" content to lure victims to click on it.

The impact is huge because of the following facts:

- The card displays directly on the user's timeline, making the attack less suspicious to normal clickjacking
- The click is very subtle that victims cannot notice what's happened behind the scene
- Wormable because attacker can make victims tweet arbitrary content to spread it
- Can perform click-based critical actions, like follow, retweet, favorite... etc
- If sent as promoted tweet, it can target even more victims, also player is directly expanded

### Repo step

1. Clone the Player Card started bundle here: <https://github.com/twitterdev/cards>
2. Change the card's property `twitter:player` to a custom HTML file
3. In the HTML file, embed iframe to a Twitter page, e.g. `<iframe src="//twitter.com"></iframe>`
4. Post the link in a Tweet (make sure the domain is white-listed)
5. Expand the tweet in Safari or IE, it will show that a Twitter page is embedded

Documentation of Player Card: <https://dev.twitter.com/cards/types/player>

### PoC

<https://twitter.com/AttackerCanvas/status/637859735501279232> (Open with Safari or IE)

Video demo: <https://vimeo.com/137725491> (password: click)

The PoC demonstrates how the attack can be conducted. There will be a fake video to lure victims to click to play it. After clicking the victim will automatically post a tweet with content "Pwn3d!".



PoC plus simple demo  
screen recording



filedescrptor submitted a report to **Twitter**.

Aug 30th (7 years ago)

Hi,

I would like to report an issue where player card is vulnerable to clickjacking in certain browsers. This may result in something similar to XSS worm and many other critical damages.

### Details

Twitter Player Card allows a website to embed a custom player(html) into an iframe in a tweet. There are currently 2-3 security features in place to defend clickjacking on Twitter:

1. `X-Frame-Options: SAMEORIGIN` covering the whole twitter.com domain
2. `Content-Security-Policy: frame-ancestors 'self'` ditto
3. JS-based frame-buster in some pages (but not all)

For (1), SAMEORIGIN only checks if the embedded frame is on the same origin of the top window. For example, attacker can do something like twitter.com -> attacker.com -> twitter.com to evade it. More details can be seen from here: [https://bugzilla.mozilla.org/show\\_bug.cgi?id=725490](https://bugzilla.mozilla.org/show_bug.cgi?id=725490)

For (2), this is the only way to correctly prevent framing from other websites (it performs the check against the ancestor list). However this is a CSP2 directive so not all browsers support it. For example, Safari and IE do not support it.

For (3), using the sandbox attribute of iframe can disable JS of a frame, hence anti-frame-buster

Since Player Card is shown on a Tweet (on twitter.com), attacker can embed an iframe which embeds a Twitter page so that attacker can overlay it with "bait" content to lure victims to click on it.

The impact is huge because of the following facts:

- The card displays directly on the user's timeline, making the attack less suspicious to normal clickjacking
- The click is very subtle that victims cannot notice what's happened behind the scene
- Wormable because attacker can make victims tweet arbitrary content to spread it
- Can perform click-based critical actions, like follow, retweet, favorite... etc
- If sent as promoted tweet, it can target even more victims, also player is directly expanded

### Repo step

1. Clone the Player Card started bundle here: <https://github.com/twitterdev/cards>
2. Change the card's property `twitter:player` to a custom HTML file
3. In the HTML file, embed iframe to a Twitter page, e.g. `<iframe src="//twitter.com"></iframe>`
4. Post the link in a Tweet (make sure the domain is white-listed)
5. Expand the tweet in Safari or IE, it will show that a Twitter page is embedded

Documentation of Player Card: <https://dev.twitter.com/cards/types/player>

### PoC

<https://twitter.com/AttackerCanvas/status/637859735501279232> (Open with Safari or IE)

Video demo: <https://vimeo.com/137725491> (password: click)

The PoC demonstrates how the attack can be conducted. There will be a fake video to lure victims to click to play it. After clicking the victim will automatically post a tweet with content "Pwn3d!".



where is the vuln valid?  
browser-specific behaviour  
called out



filedescrptor submitted a report to **Twitter**.

Aug 30th (7 years ago)

Hi,

I would like to report an issue where player card is vulnerable to clickjacking in certain browsers. This may result in something similar to XSS worm and many other critical damages.

### Details

Twitter Player Card allows a website to embed a custom player(html) into an iframe in a tweet. There are currently 2-3 security features in place to defend clickjacking on Twitter:

1. `X-Frame-Options: SAMEORIGIN` covering the whole twitter.com domain
2. `Content-Security-Policy: frame-ancestors 'self'` ditto
3. JS-based frame-buster in some pages (but not all)

For (1), SAMEORIGIN only checks if the embedded frame is on the same origin of the top window. For example, attacker can do something like twitter.com -> attacker.com -> twitter.com to evade it. More details can be seen from here: [https://bugzilla.mozilla.org/show\\_bug.cgi?id=725490](https://bugzilla.mozilla.org/show_bug.cgi?id=725490)

For (2), this is the only way to correctly prevent framing from other websites (it performs the check against the ancestor list). However this is a CSP2 directive so not all browsers support it. For example, Safari and IE do not support it.

For (3), using the sandbox attribute of iframe can disable JS of a frame, hence anti-frame-buster

Since Player Card is shown on a Tweet (on twitter.com), attacker can embed an iframe which embeds a Twitter page so that attacker can overlay it with "bait" content to lure victims to click on it.

The impact is huge because of the following facts:

- The card displays directly on the user's timeline, making the attack less suspicious to normal clickjacking
- The click is very subtle that victims cannot notice what's happened behind the scene
- Wormable because attacker can make victims tweet arbitrary content to spread it
- Can perform click-based critical actions, like follow, retweet, favorite... etc
- If sent as promoted tweet, it can target even more victims, also player is directly expanded

### Repo step

1. Clone the Player Card started bundle here: <https://github.com/twitterdev/cards>
2. Change the card's property `twitter:player` to a custom HTML file
3. In the HTML file, embed iframe to a Twitter page, e.g. `<iframe src="//twitter.com"></iframe>`
4. Post the link in a Tweet (make sure the domain is white-listed)
5. Expand the tweet in Safari or IE, it will show that a Twitter page is embedded

Documentation of Player Card: <https://dev.twitter.com/cards/types/player>

### PoC

<https://twitter.com/AttackerCanvas/status/637859735501279232> (Open with Safari or IE)

Video demo: <https://vimeo.com/137725491> (password: click)

The PoC demonstrates how the attack can be conducted. There will be a fake video to lure victims to click to play it. After clicking the victim will automatically post a tweet with content "Pwn3d!".

repro steps are clear





# starting points

- Disclaimer: not the process of clickjacking vuln reporter; some tactics and tools they could have used...
- RFC, spec ([X-Frame-Options](#), [Content-Security-Policy](#)) documented edge cases
- understand expected browser behaviour
- read the app docs (after looking at the code and black boxing the app) but don't necessarily believe them!!
- go deep into one type of vuln across programs and apps
- limit platform impact, use test accounts
- demonstrate impact directly to user / service (threat modeling while being the threat :)
- Portswigger clickjacking [learning path](#)
- filedescriptor's writeup: <https://blog.innerht.ml/google-yolo/>





# what was each report worth?

## Rewards

Twitter may, at its sole discretion, provide rewards to eligible reporters of qualifying vulnerabilities. Our minimum reward is \$140 USD. Rewards are typically paid out on Fridays. The following table outlines the nominal rewards for specific classes of vulnerabilities for in-scope properties (see section on Scope).

Category	Examples	Core Twitter[1]	Everything Else
Remote code execution	Command injection	\$20,160	\$10,080
Administrative functionality	Access to internal Twitter applications	\$12,460	\$6,300
Unrestricted access to data (filesystem, database, etc.)	XXE, SQLi	\$12,460	\$6,300
Flaws leaking PII or bypassing significant controls	IDOR, impersonation, sensitive actions by user	\$7,700	\$3,920
Account takeover	OAuth vulnerabilities	\$7,700	\$3,920
Perform activities on behalf of a user	XSS, Android Intent abuse	\$2,940	\$1,540
Other valid vulnerabilities	CSRF, clickjacking, information leakage	\$280 - \$2,940	\$140 - \$1,540

[1] Core Twitter is defined as anything hosted on `*.twitter.com`, `*.pscp.tv`, `*.periscope.tv`, and Twitter owned-and-operated mobile clients.

Twitter may choose to pay higher rewards for unusually clever or severe vulnerabilities or lower rewards for vulnerabilities that require significant or unusual user interaction.



# what was each report worth?

## Rewards

Twitter may, at its sole discretion, provide rewards to eligible reporters of qualifying vulnerabilities. Our minimum reward is \$140 USD. Rewards are typically paid out on Fridays. The following table outlines the nominal rewards for specific classes of vulnerabilities for in-scope properties (see section on Scope).

Category	Examples	Core Twitter[1]	Everything Else
Remote code execution	Command injection	\$20,160	\$10,080
Administrative functionality	Access to internal Twitter applications	\$12,460	\$6,300
Unrestricted access to data (filesystem, database, etc.)	XXE, SQLi	\$12,460	\$6,300
Flaws leaking PII or bypassing significant controls	IDOR, impersonation, sensitive actions by user	\$7,700	\$3,920
Account takeover	OAuth vulnerabilities	\$7,700	\$3,920
Perform activities on behalf of a user	XSS, Android Intent abuse	\$2,940	\$1,540
Other valid vulnerabilities	CSRF, clickjacking, information leakage	\$280 - \$2,940	\$140 - \$1,540

[1] Core Twitter is defined as anything hosted on `*.twitter.com`, `*.pscp.tv`, `*.periscope.tv`, and Twitter owned-and-operated mobile clients.

Twitter may choose to pay higher rewards for unusually clever or severe vulnerabilities or lower rewards for vulnerabilities that require significant or unusual user interaction.

Email bomb

API permissions

Video player clickjacking



# what was each report worth?

## Rewards

Twitter may, at its sole discretion, provide rewards to eligible reporters of qualifying vulnerabilities. Our minimum reward is \$140 USD. Rewards are typically paid out on Fridays. The following table outlines the nominal rewards for specific classes of vulnerabilities for in-scope properties (see section on Scope).

Category	Examples	Core Twitter[1]	Everything Else
Remote code execution	Command injection	\$20,160	\$10,080
Administrative functionality	Access to internal Twitter applications	\$12,460	\$6,300
Unrestricted access to data (filesystem, database, etc.)	XXE, SQLi	\$12,460	\$6,300
Flaws leaking PII or bypassing significant controls	IDOR, impersonation, sensitive actions by user	\$7,700	\$3,920
Account takeover	OAuth vulnerabilities	\$7,700	\$3,920
Perform activities on behalf of a user	XSS, Android Intent abuse	\$2,940	\$1,540
Other valid vulnerabilities	CSRF, clickjacking, information leakage	\$280 - \$2,940	\$140 - \$1,540

[1] Core Twitter is defined as anything hosted on `*.twitter.com`, `*.pscp.tv`, `*.periscope.tv`, and Twitter owned-and-operated mobile clients.

Twitter may choose to pay higher rewards for unusually clever or severe vulnerabilities or lower rewards for vulnerabilities that require significant or unusual user interaction.

Email bomb

**\$140**

API permissions

Video player clickjacking



# what was each report worth?

## Rewards

Twitter may, at its sole discretion, provide rewards to eligible reporters of qualifying vulnerabilities. Our minimum reward is \$140 USD. Rewards are typically paid out on Fridays. The following table outlines the nominal rewards for specific classes of vulnerabilities for in-scope properties (see section on Scope).

Category	Examples	Core Twitter[1]	Everything Else
Remote code execution	Command injection	\$20,160	\$10,080
Administrative functionality	Access to internal Twitter applications	\$12,460	\$6,300
Unrestricted access to data (filesystem, database, etc.)	XXE, SQLi	\$12,460	\$6,300
Flaws leaking PII or bypassing significant controls	IDOR, impersonation, sensitive actions by user	\$7,700	\$3,920
Account takeover	OAuth vulnerabilities	\$7,700	\$3,920
Perform activities on behalf of a user	XSS, Android Intent abuse	\$2,940	\$1,540
Other valid vulnerabilities	CSRF, clickjacking, information leakage	\$280 - \$2,940	\$140 - \$1,540

[1] Core Twitter is defined as anything hosted on `*.twitter.com`, `*.pscp.tv`, `*.periscope.tv`, and Twitter owned-and-operated mobile clients.

Twitter may choose to pay higher rewards for unusually clever or severe vulnerabilities or lower rewards for vulnerabilities that require significant or unusual user interaction.

Email bomb

**\$140**

API permissions

**\$7700**

Video player clickjacking



# what was each report worth?

## Rewards

Twitter may, at its sole discretion, provide rewards to eligible reporters of qualifying vulnerabilities. Our minimum reward is \$140 USD. Rewards are typically paid out on Fridays. The following table outlines the nominal rewards for specific classes of vulnerabilities for in-scope properties (see section on Scope).

Category	Examples	Core Twitter[1]	Everything Else
Remote code execution	Command injection	\$20,160	\$10,080
Administrative functionality	Access to internal Twitter applications	\$12,460	\$6,300
Unrestricted access to data (filesystem, database, etc.)	XXE, SQLi	\$12,460	\$6,300
Flaws leaking PII or bypassing significant controls	IDOR, impersonation, sensitive actions by user	\$7,700	\$3,920
Account takeover	OAuth vulnerabilities	\$7,700	\$3,920
Perform activities on behalf of a user	XSS, Android Intent abuse	\$2,940	\$1,540
Other valid vulnerabilities	CSRF, clickjacking, information leakage	\$280 - \$2,940	\$140 - \$1,540

[1] Core Twitter is defined as anything hosted on `*.twitter.com`, `*.pscp.tv`, `*.periscope.tv`, and Twitter owned-and-operated mobile clients.

Twitter may choose to pay higher rewards for unusually clever or severe vulnerabilities or lower rewards for vulnerabilities that require significant or unusual user interaction.

Email bomb

**\$140**

API permissions

**\$7700**

Video player clickjacking

**\$5040**

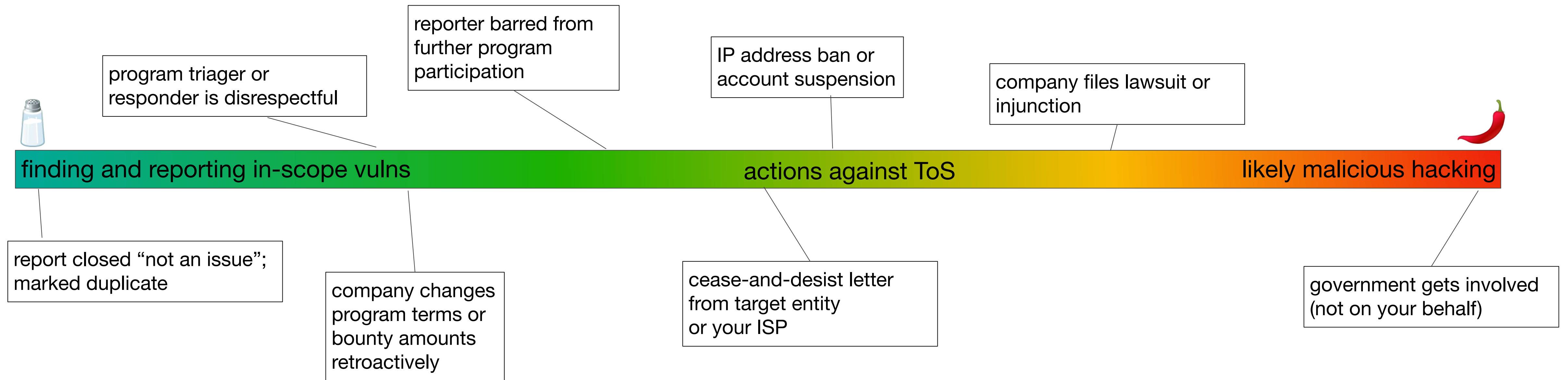


**questions before we continue?**

**now that we've seen some  
accepted reports...**



# Hacking & disclosure: what could go wrong








# this sucks, but can happen

lyoung-uber reopened this report. Aug 16th (6 years ago)

 lyoung-uber closed the report and changed the status to **Not Applicable**.  
Closing as `Not Applicable` since this is out-of-scope. Aug 16th (6 years ago)


<https://hackerone.com/reports/156098>



# don't do this.

lyoung-uber reopened this report. Aug 16th (6 years ago)

 lyoung-uber closed the report and changed the status to **Not Applicable**. Aug 16th (6 years ago)  
Closing as `Not Applicable` since this is out-of-scope.

 raghav\_bisht posted a comment. Aug 16th (6 years ago)  
@lyoung-uber you fucking asshole mother fucker I know this is "Out of scope" and your team member @bugtrriage-rob marked it has Informative and closed the report, still I didn't argue about it and accepted it.....fucker.  
I respectfully asked you to disclosure my report and you moron mother fucker deducted my Reputation Point ....  
  
Bloody Mother Fucker..... TAXI DRIVER.....

 lyoung-uber posted a comment. Aug 16th (6 years ago)  
Hi @raghav\_bisht,

First off I wanted to apologize for not writing a longer response when I updated the report state, that's my fault. However as you acknowledged yourself this is not in scope per our [hackerone.com/uber](https://hackerone.com/uber):

#### Out-of-scope Properties

\*.et.uber.com - The underlying software here is exacttarget which Uber does not have control over.

It's important that our reports are tracked correctly both for HackerOne's statistics and our own internal metrics. With that said, that absolutely does not excuse your behavior:

I respectfully asked you to disclosure my report and you moron mother fucker deducted my Reputation Point ....  
Bloody Mother Fucker..... TAXI DRIVER.....



# banned from the program...

lyoung-uber requested to disclose this report.

Aug 19th (6 years ago)



lyoung-uber disclosed this report.

Updated Aug 19th (6 years ago)

After an internal discussion, this reporter was banned for violating our [hackerone.com/uber](#) and the use of abusive language. The issue disclosed in this report was first reported to us in [#151968](#) (also closed as `Not Applicable`) and has since been resolved.

In accordance with the reporter's original request:

Once you patched the vulnerability "Do disclose the report"

Since the issue has been resolved we are now publicly disclosing this report.



raghav\_bisht posted a comment.

Aug 19th (6 years ago)

[@lyoung-uber](#)

Now whats the point to Disclose the report ?

You already ruined the report via making me angry....

and make me abuse you....!!

hackerone-support joined this report as a participant.

Aug 20th (6 years ago)



hackerone-support posted a comment.

Aug 20th (6 years ago)

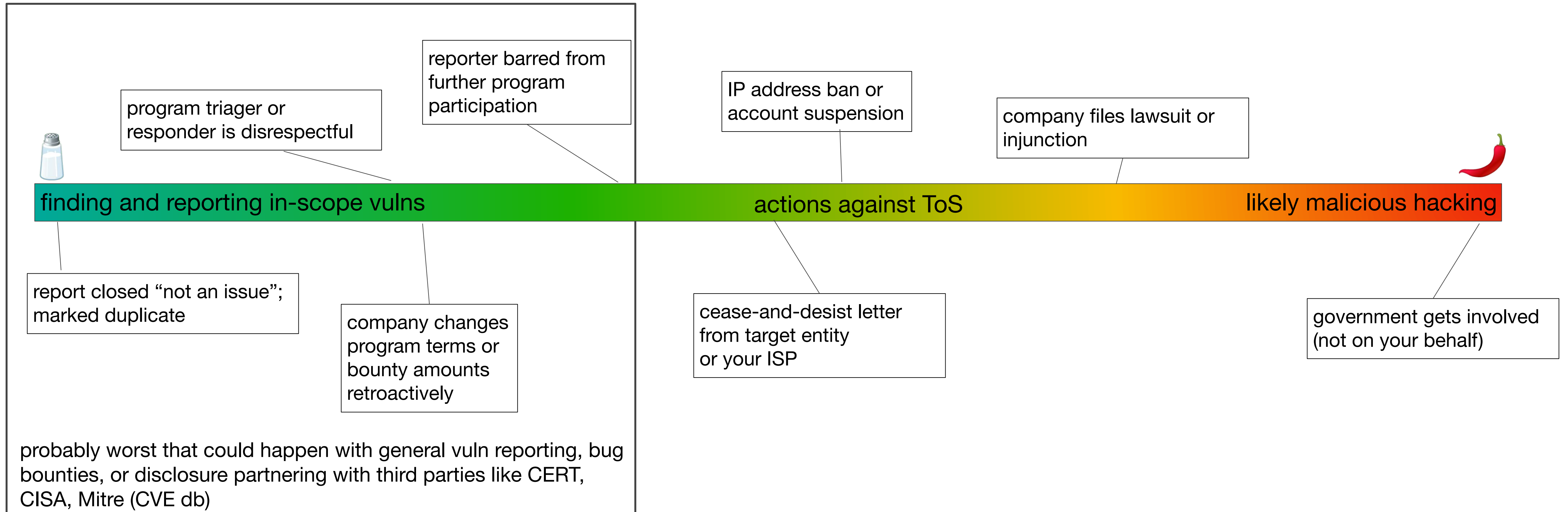
Hi [@raghav\\_bisht](#) -

This abusive behavior is not tolerated on HackerOne. Please review our [Disclosure Guidelines](#) and this [help center article on bad behavior](#). In addition to your ban from the Uber program, please check your email for additional restrictions being levied on the platform. We hope you're able to take this time to rectify this behavior to maintain a professional and respectful attitude on HackerOne in the future.

-HackerOne Support

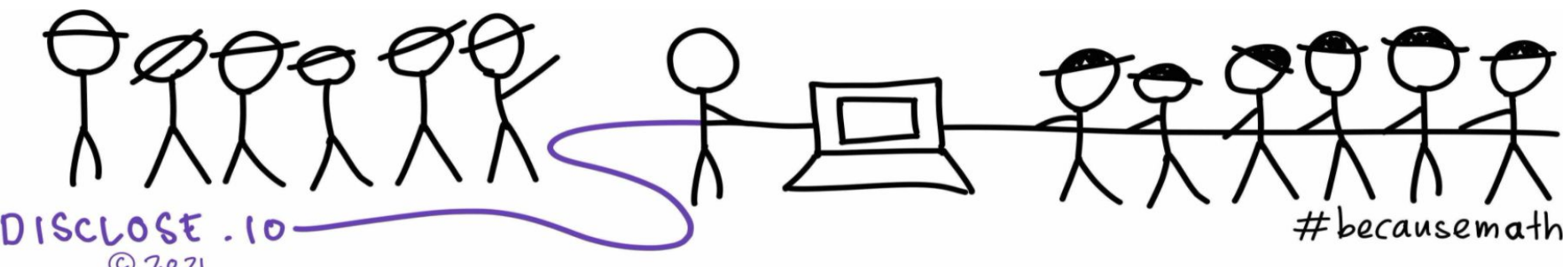


# Hacking & disclosure: what could go wrong





**what if I found an issue and  
don't know where to report it?**

DISCLOSE .IO © 2021 #becausemath

# disclose.io

We're here to make vulnerability disclosure safe, simple, and standardized for everyone.



Submit a CVE Request

- \* Required
- \* **Select a request type**
- \* **Enter your e-mail address**

✓ - Please choose an action -

- Report Vulnerability/Request CVE ID**
- Request multiple IDs (For CNAs Only)
- Notify CVE about a publication
- Request an update to an existing CVE Entry
- Request information on the CVE Numbering Authority (CNA) Program
- Other

completing this form.



Search

**Report**



# More stuff!

- [Katie Moussouris](#) on well-managed versus poorly run bug bounties
- Google's [Bug Hunter University](#) and "[How to become a Bug Hunter](#)"
- [TryHackMe](#): bite-size appsec and pentesting tutorials
- Mitre's "[New to CVE? Start here](#)"
- Microsoft's bug bounty vuln report how-to and [examples of high-quality reports](#)

# Thank you!



## *summary*

0. if you can spot a vuln, you can report it too!

1. follow coordinated vuln disclosure standards if publishing

2. obligatory plug: Twitter is looking for security & privacy engineers at all levels ([link](#))