OWASP 2022 GLOBAL AppSec | SAN FRANCISCO NOV 14-18

Marius Musch

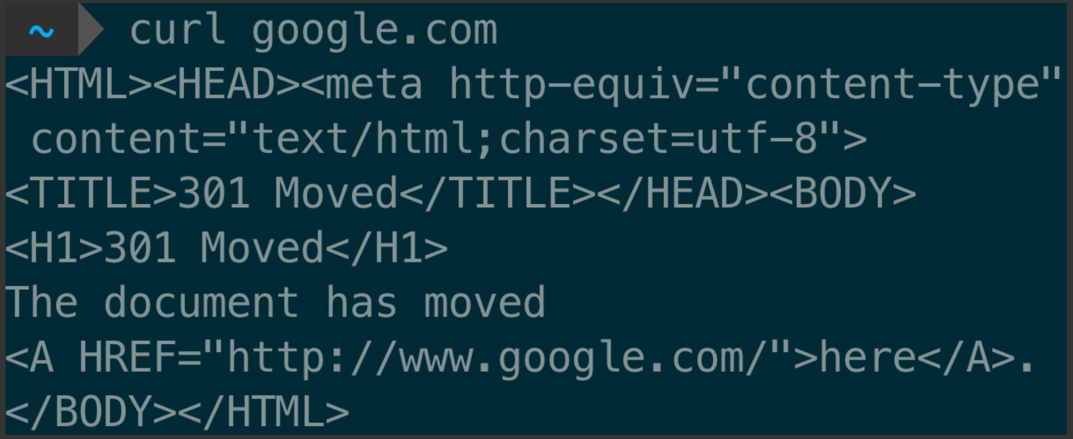
Server-Side Browsers: Exploring the Web's Hidden Attack Surface

TU Braunschweig

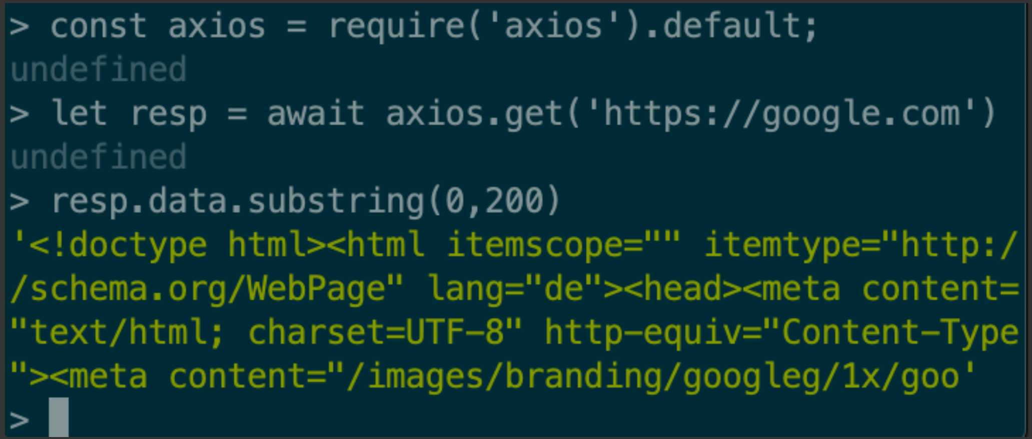
Joint work with Robin Kirchner, Max Boll, and Martin Johns

Automated Requests

- Simple and reliable

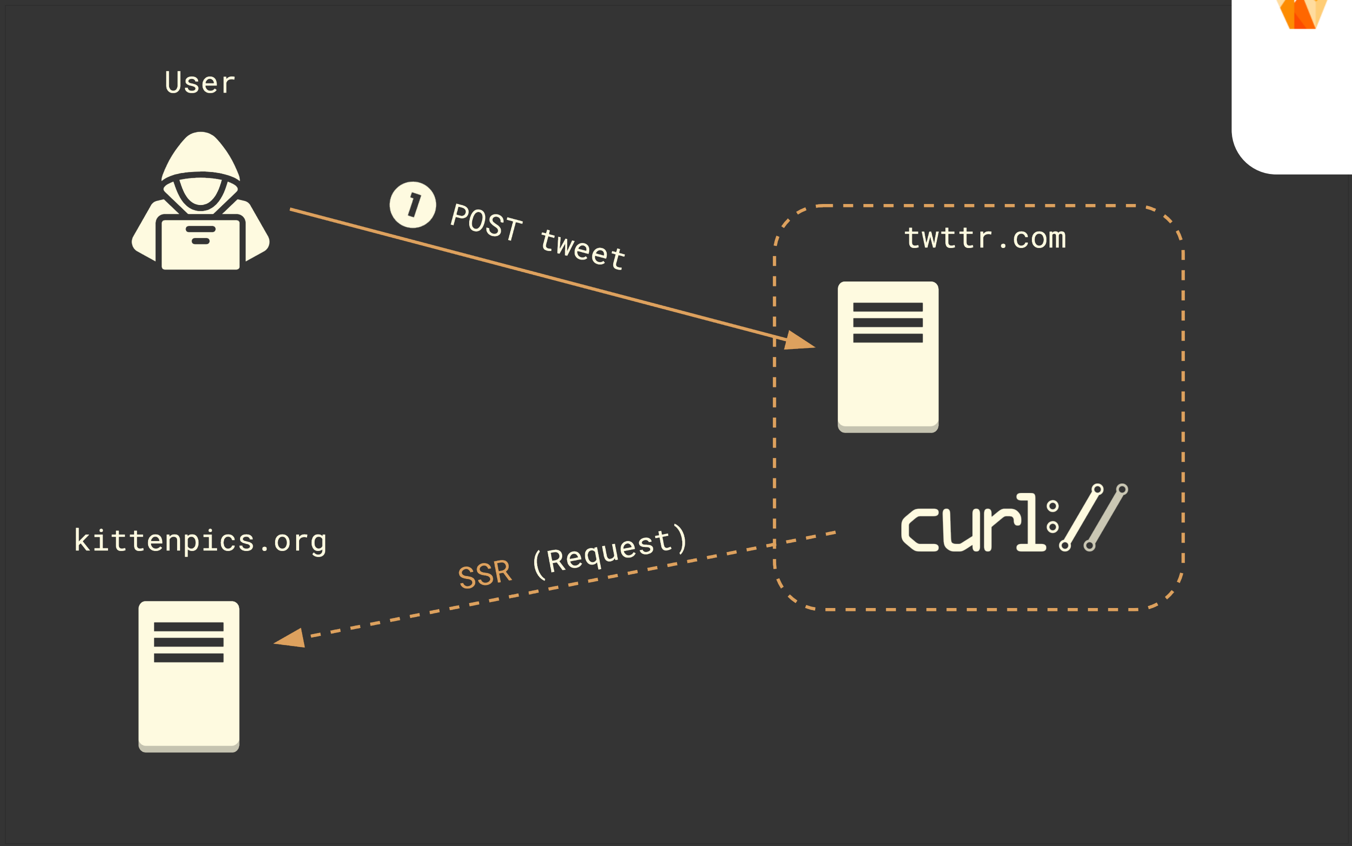


```
~ curl google.com
<HTML><HEAD><meta http-equiv="content-type"
 content="text/html;charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>
```

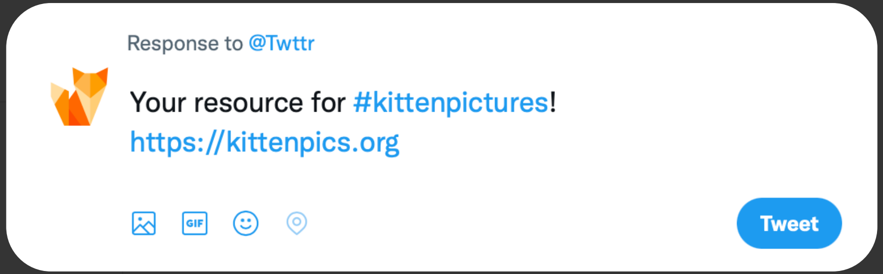


```
> const axios = require('axios').default;
undefined
> let resp = await axios.get('https://google.com')
undefined
> resp.data.substring(0,200)
'<!doctype html><html itemscope="" itemtype="http:/
/schema.org/WebPage" lang="de"><head><meta content=
"text/html; charset=UTF-8" http-equiv="Content-Type
"><meta content="/images/branding/googleg/1x/goo'
> 
```

Request for Preview

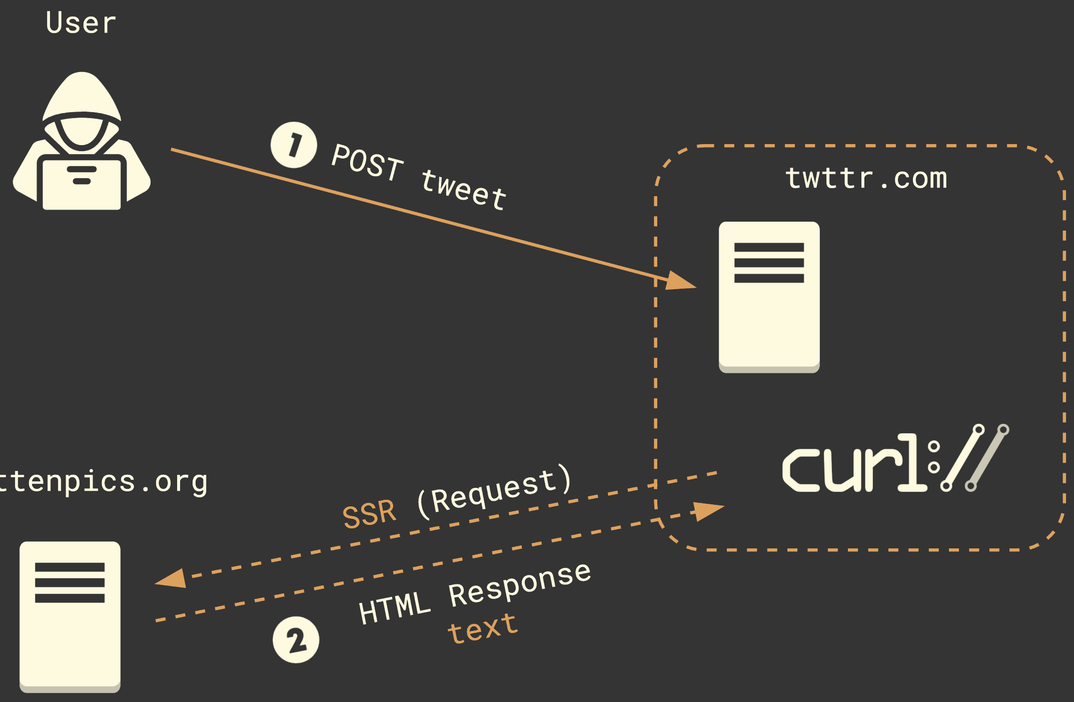


1

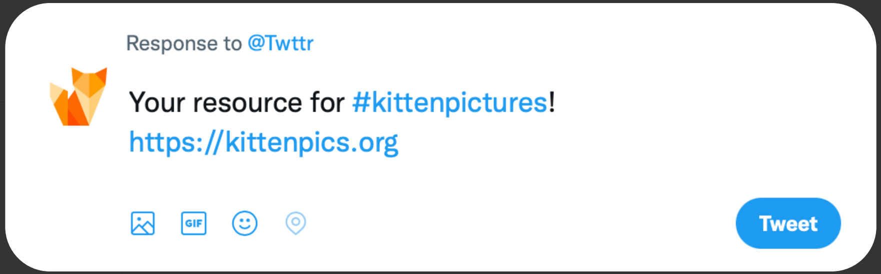

Response to @Twttr

Your resource for #kittenpictures!
https://kittenpics.org

Request for Preview



1

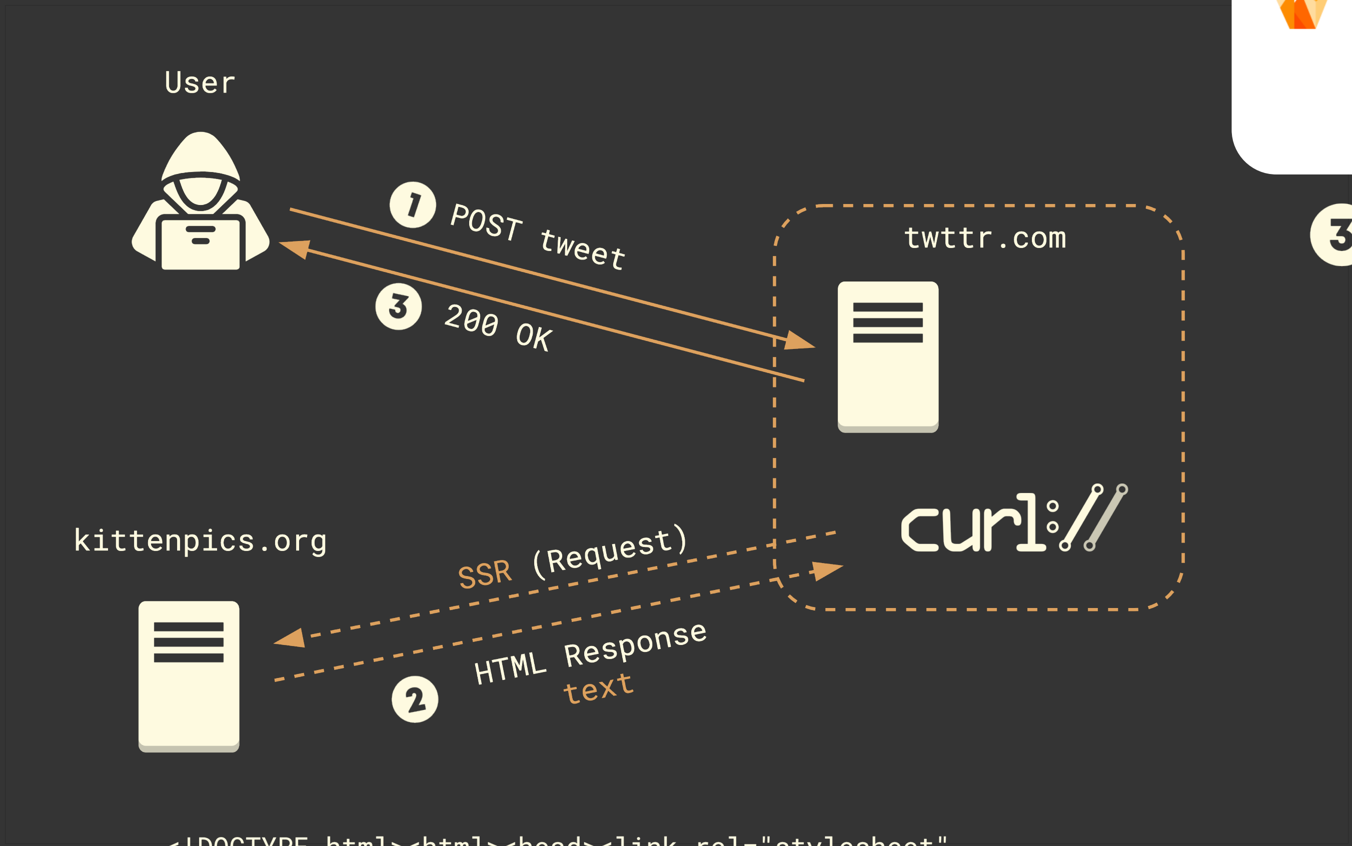

Response to @Twttr

Your resource for #kittenpictures!
https://kittenpics.org

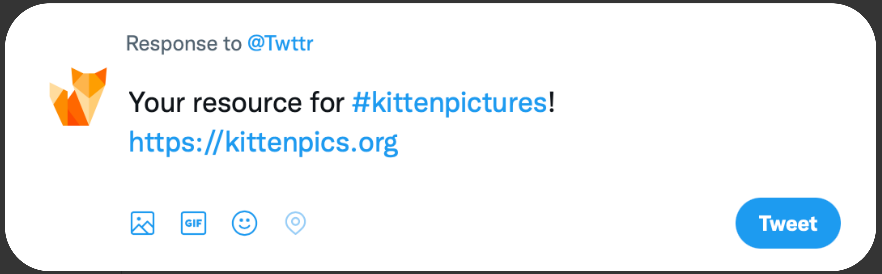
2

```
<!DOCTYPE html><html><head><link rel="stylesheet"
href="https://cdn.org/base.css"></head><body><img src="https://cdn-
icons.org/57162.png"></img><h1>Kittenpics.org</h1><script
src="https://cdn.org/76dsdasd.js"></script><div class="wh-fms-3-daf">
</div><p>Kittenpics.org is your resource...</p>[...]
```

Request for Preview

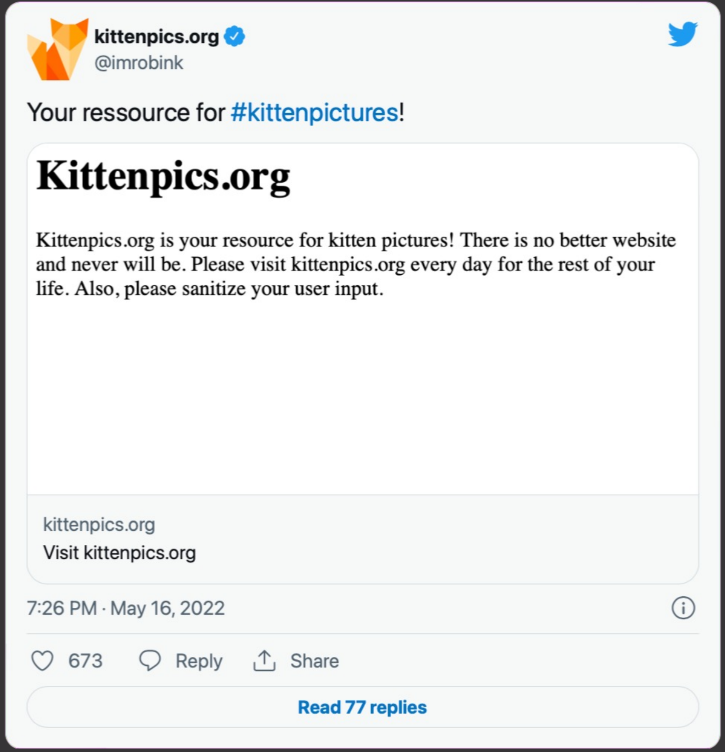


1


Response to @Twttr

Your resource for #kittenpictures!
https://kittenpics.org

3


kittenpics.org
@imrobink

Your ressource for #kittenpictures!

Kittenpics.org

Kittenpics.org is your resource for kitten pictures! There is no better website and never will be. Please visit kittenpics.org every day for the rest of your life. Also, please sanitize your user input.

kittenpics.org
Visit kittenpics.org

7:26 PM · May 16, 2022

673 Reply Share

Read 77 replies

2

```
<!DOCTYPE html><html><head><link rel="stylesheet"
href="https://cdn.org/base.css"></head><body><img src="https://cdn-
icons.org/57162.png"></img><h1>Kittenpics.org</h1><script
src="https://cdn.org/76dsdasd.js"></script><div class="wh-fms-3-daf">
</div><p>Kittenpics.org is your resource...</p>[...]
```

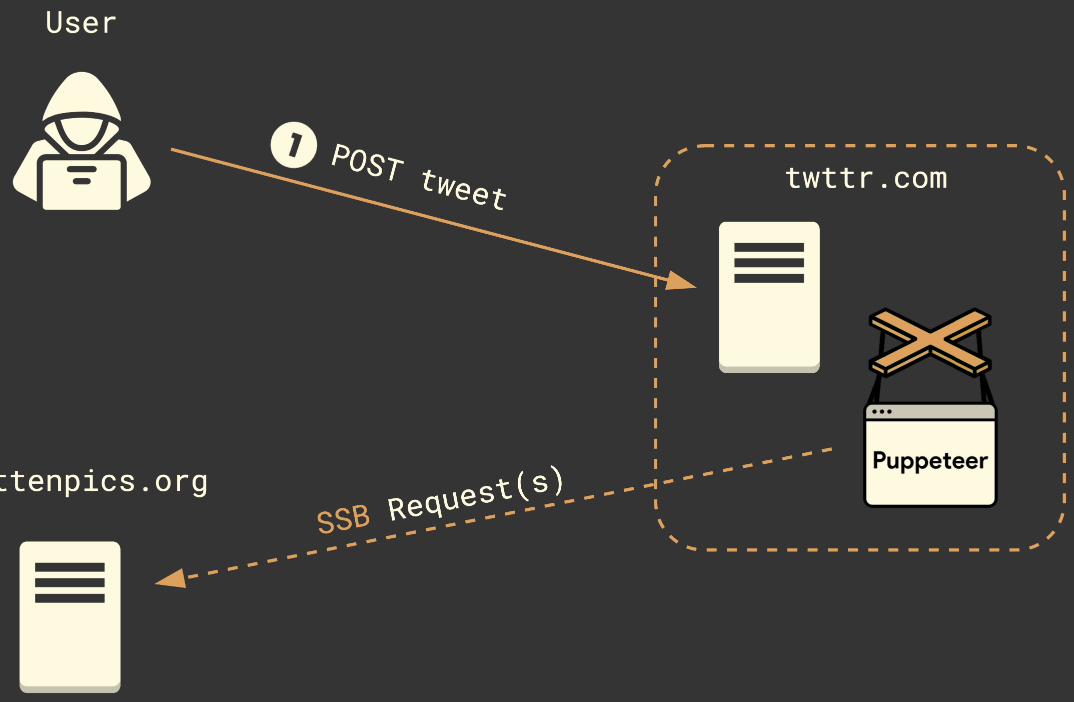
Automated Browsers

- Modern solutions

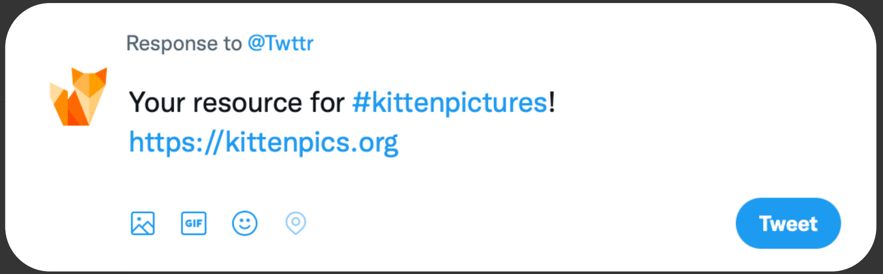
```
const { chromium } = require('playwright');
(async () => {
    const browser = await chromium.launch();
    const page = await browser.newPage();
    await page.goto('http://example.com');
    // Do something with the page
    await browser.close();
})();
```

Puppeteer

Browser for Preview

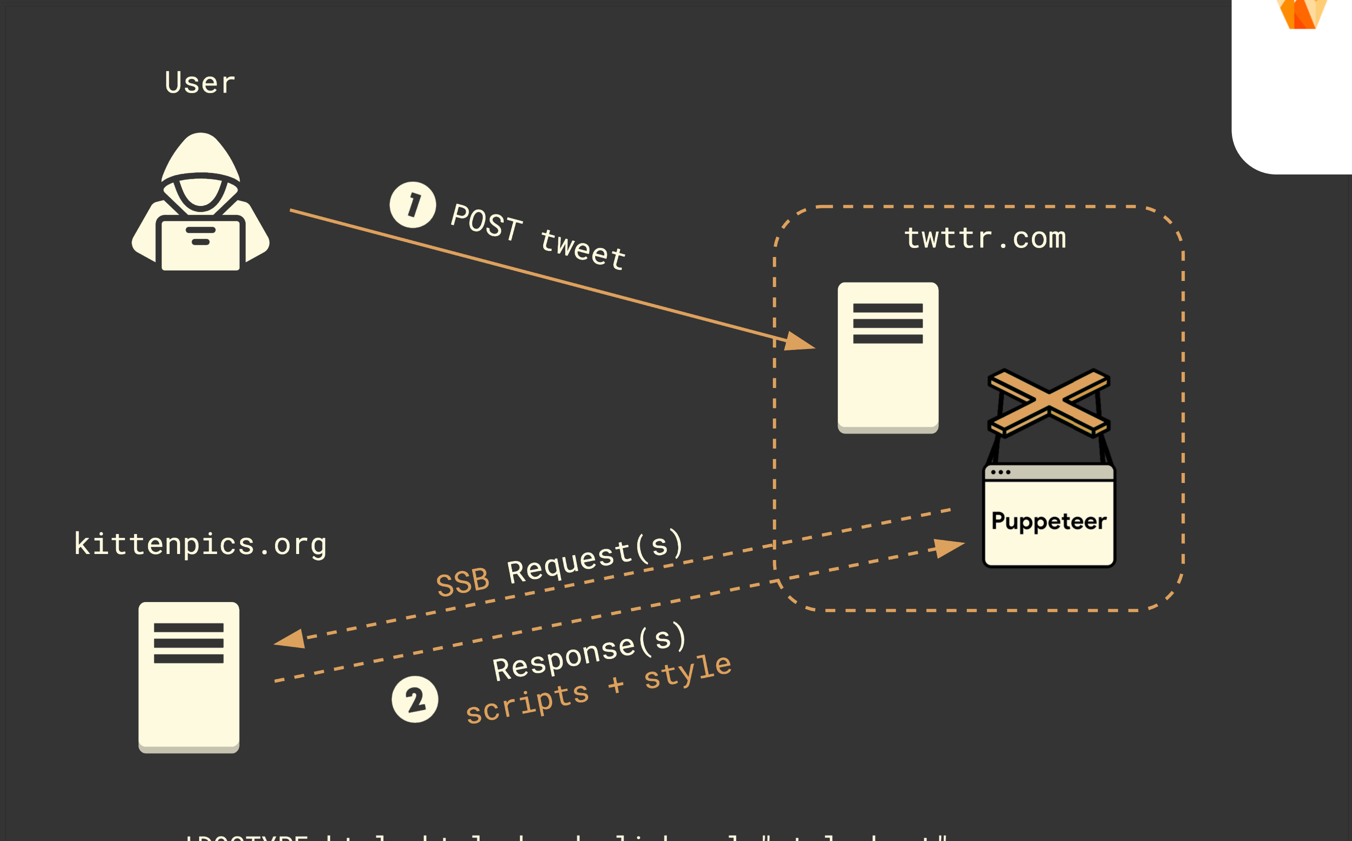


1

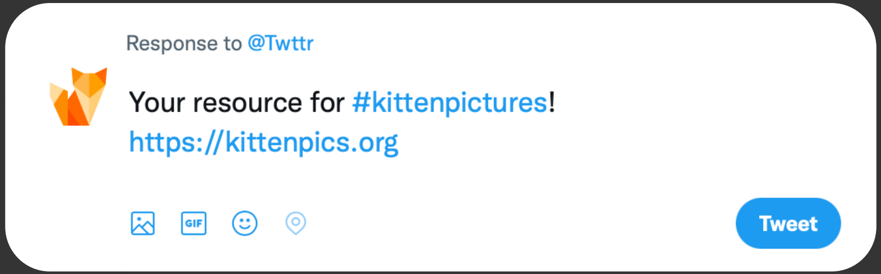

Response to @Twttr

Your resource for #kittenpictures!
https://kittenpics.org

Browser for Preview



1

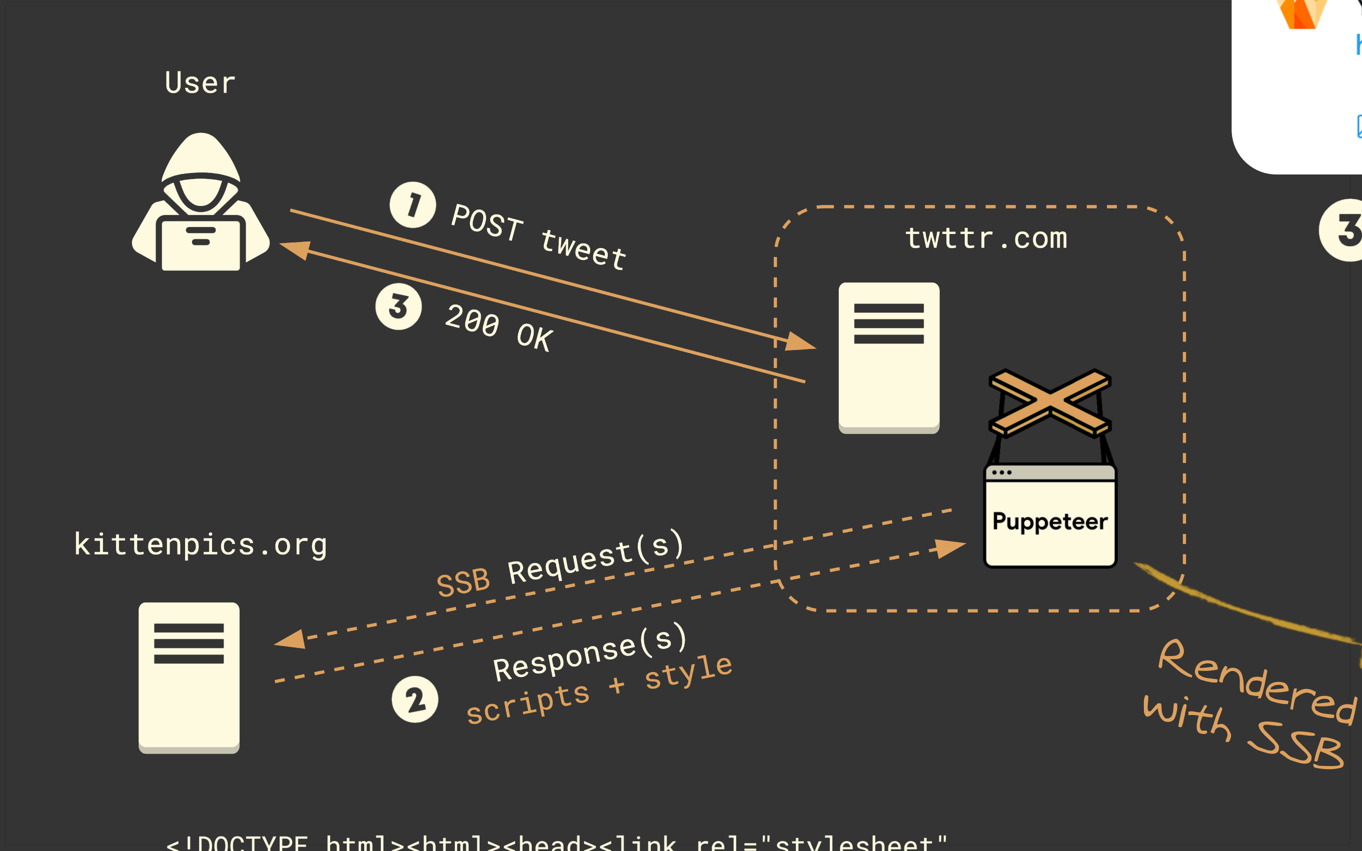

Response to @Twttr

Your resource for #kittenpictures!
https://kittenpics.org

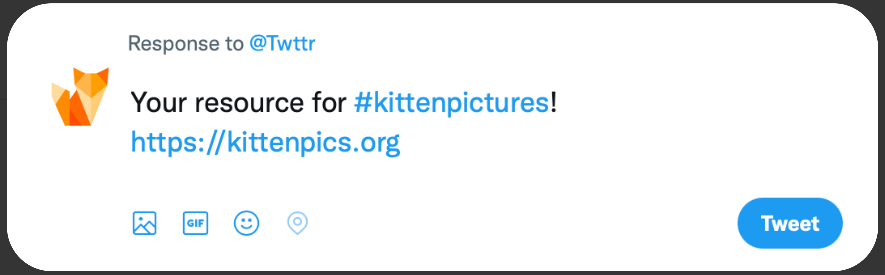
2

```
<!DOCTYPE html><html><head><link rel="stylesheet"
href="https://cdn.org/base.css"></head><body><img src="https://cdn-
icons.org/57162.png"></img><h1>Kittenpics.org</h1><script
src="https://cdn.org/76dsdasd.js"></script><div class="wh-fms-3-daf">
</div><p>Kittenpics.org is your resource...</p>[...]
```

Browser for Preview

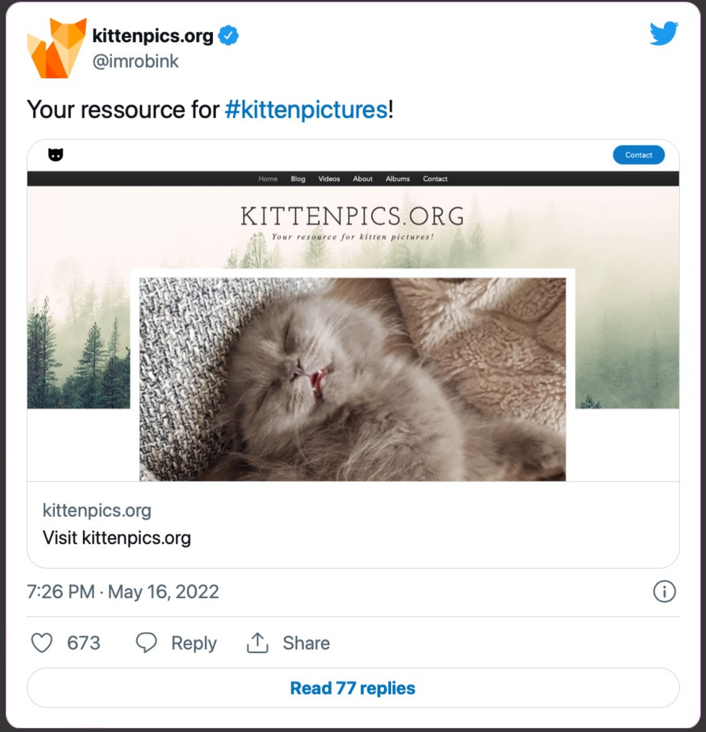


1


Response to @Twttr

Your resource for #kittenpictures! https://kittenpics.org

3


kittenpics.org @imrobink

Your ressource for #kittenpictures!

KITTENPICS.ORG
Your resource for kitten pictures!

kittenpics.org
Visit kittenpics.org

7:26 PM · May 16, 2022

2

```
<!DOCTYPE html><html><head><link rel="stylesheet"
href="https://cdn.org/base.css"></head><body><img src="https://cdn-
icons.org/57162.png"></img><h1>Kittenpics.org</h1><script
src="https://cdn.org/76dsdasd.js"></script><div class="wh-fms-3-daf">
</div><p>Kittenpics.org is your resource...</p>[...]
```

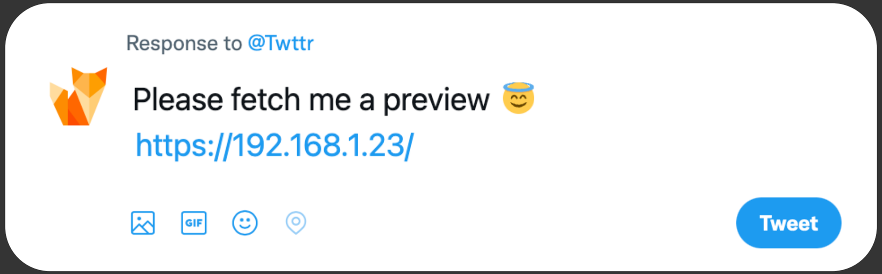
Terminology

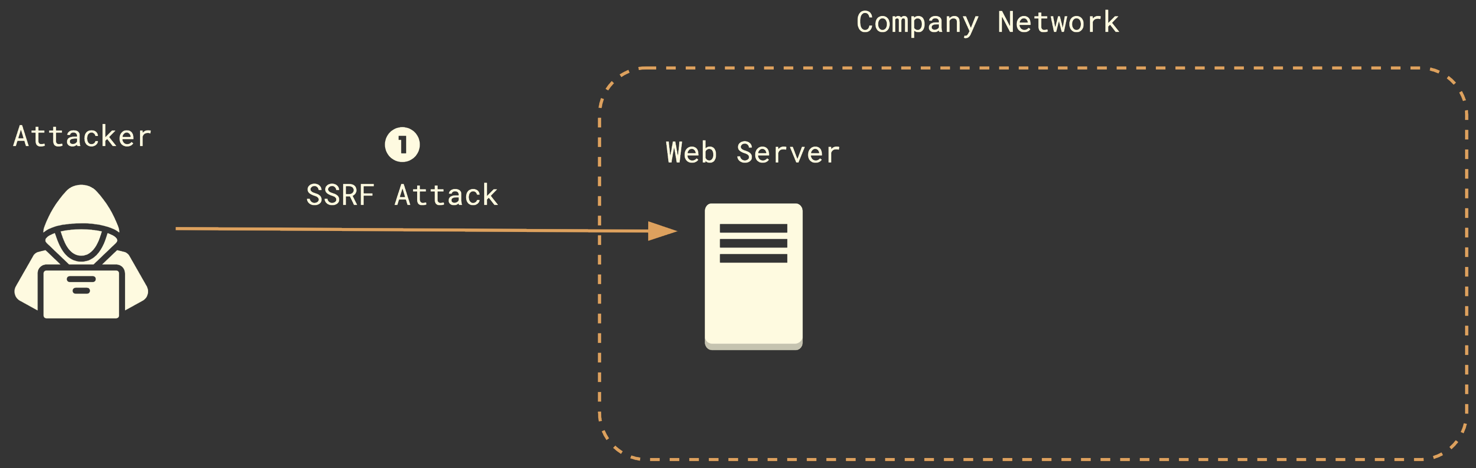
- Server-Side Request (**SSR**)
  - **Use case**: Extract content from text document (HTML, JSON, …)
  - **Tools**: `wget`, `curl`, HTTP libraries …

- Server-Side Browser (**SSB**)
  - **Use case**: Create screenshot of rendered website
  - **Tools**: Headless Chrome, Puppeteer, Playwright …

SSRF Attacks

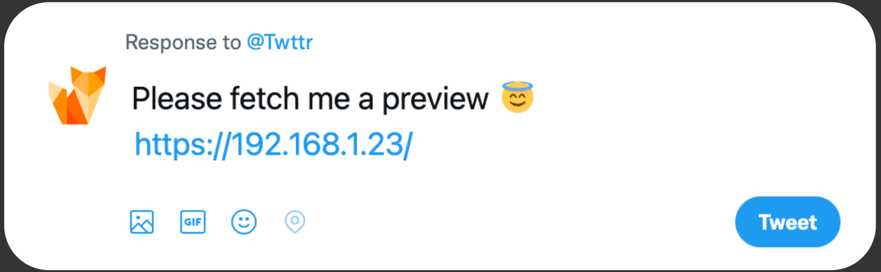
1


Response to @Twttr
Please fetch me a preview 😇
https://192.168.1.23/

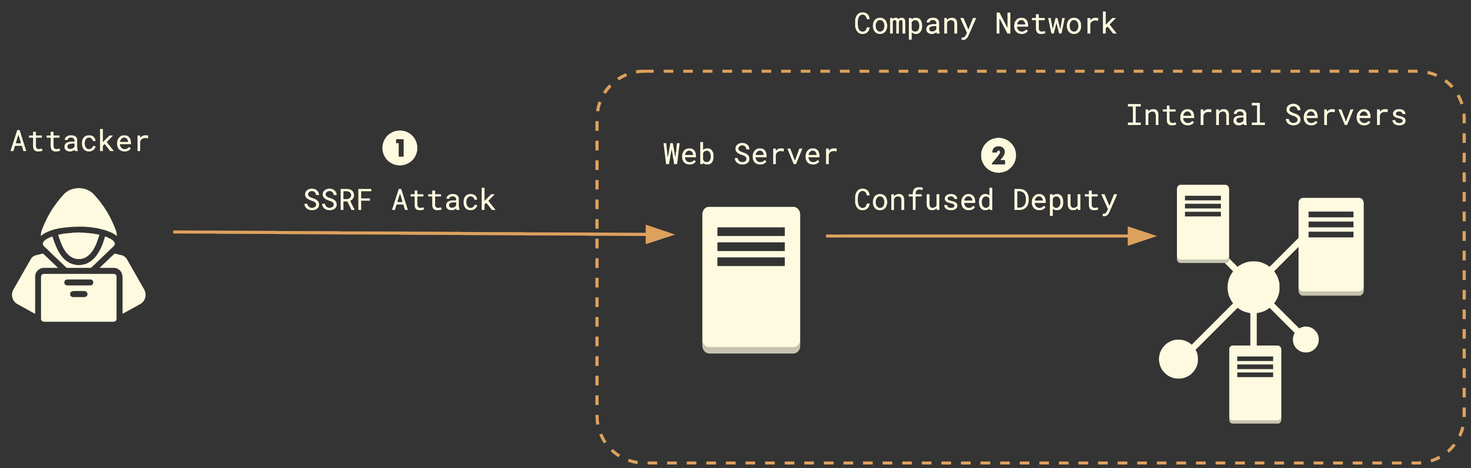


SSRF Attacks

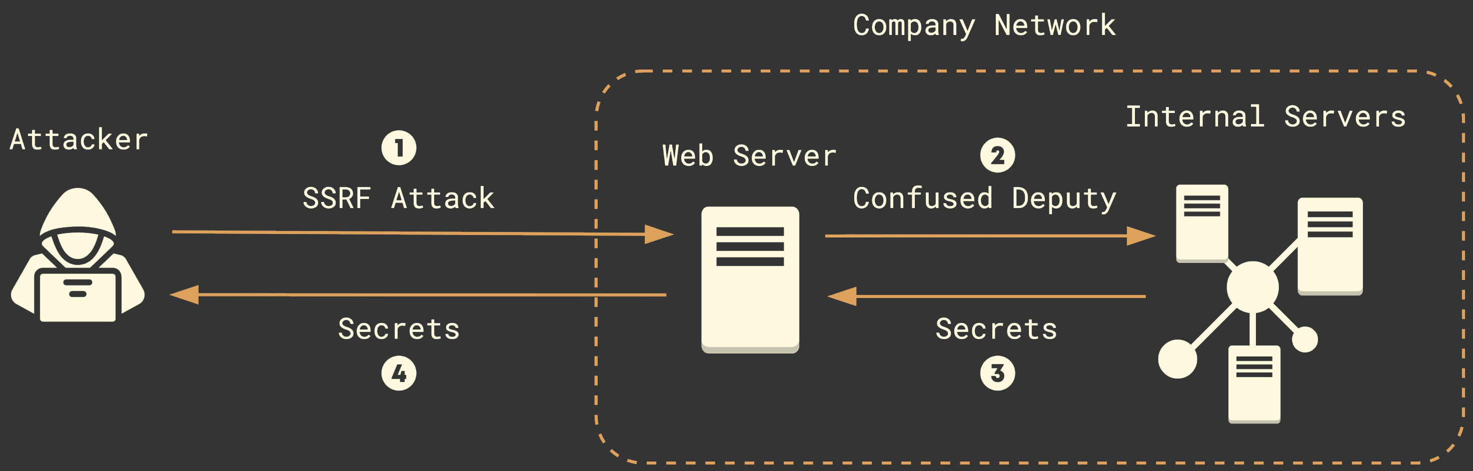
1


Response to @Twttr

Please fetch me a preview 😇
https://192.168.1.23/

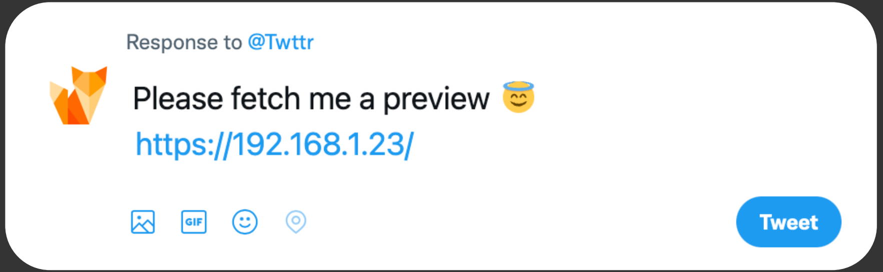


SSRF Attacks



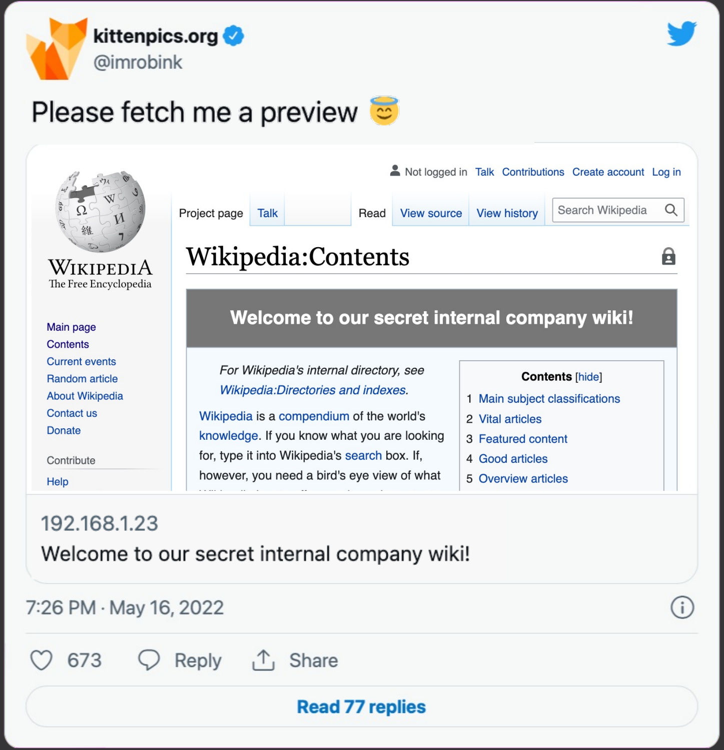
SSRF is #10 in OWASP Top Ten 2021!

1


Response to @Twttr

Please fetch me a preview 😇
https://192.168.1.23/

3


kittenpics.org
@imrobink

Please fetch me a preview 😇

Wikipedia:Contents

Welcome to our secret internal company wiki!

For Wikipedia's internal directory, see Wikipedia:Directories and indexes.

Wikipedia is a compendium of the world's knowledge. If you know what you are looking for, type it into Wikipedia's search box. If, however, you need a bird's eye view of what

Contents [hide]
1 Main subject classifications
2 Vital articles
3 Featured content
4 Good articles
5 Overview articles

192.168.1.23
Welcome to our secret internal company wiki!

7:26 PM · May 16, 2022

Terminology

- Server-Side Request (**SSR**)
  - **Use case**: Extract content from text document (HTML, JSON, …)
  - **Tools**: `wget`, `curl`, HTTP libraries …
- Server-Side Browser (**SSB**)
  - **Use case**: Create screenshot of rendered website
  - **Tools**: PhantomJS, Headless Chrome, Puppeteer, Playwright …

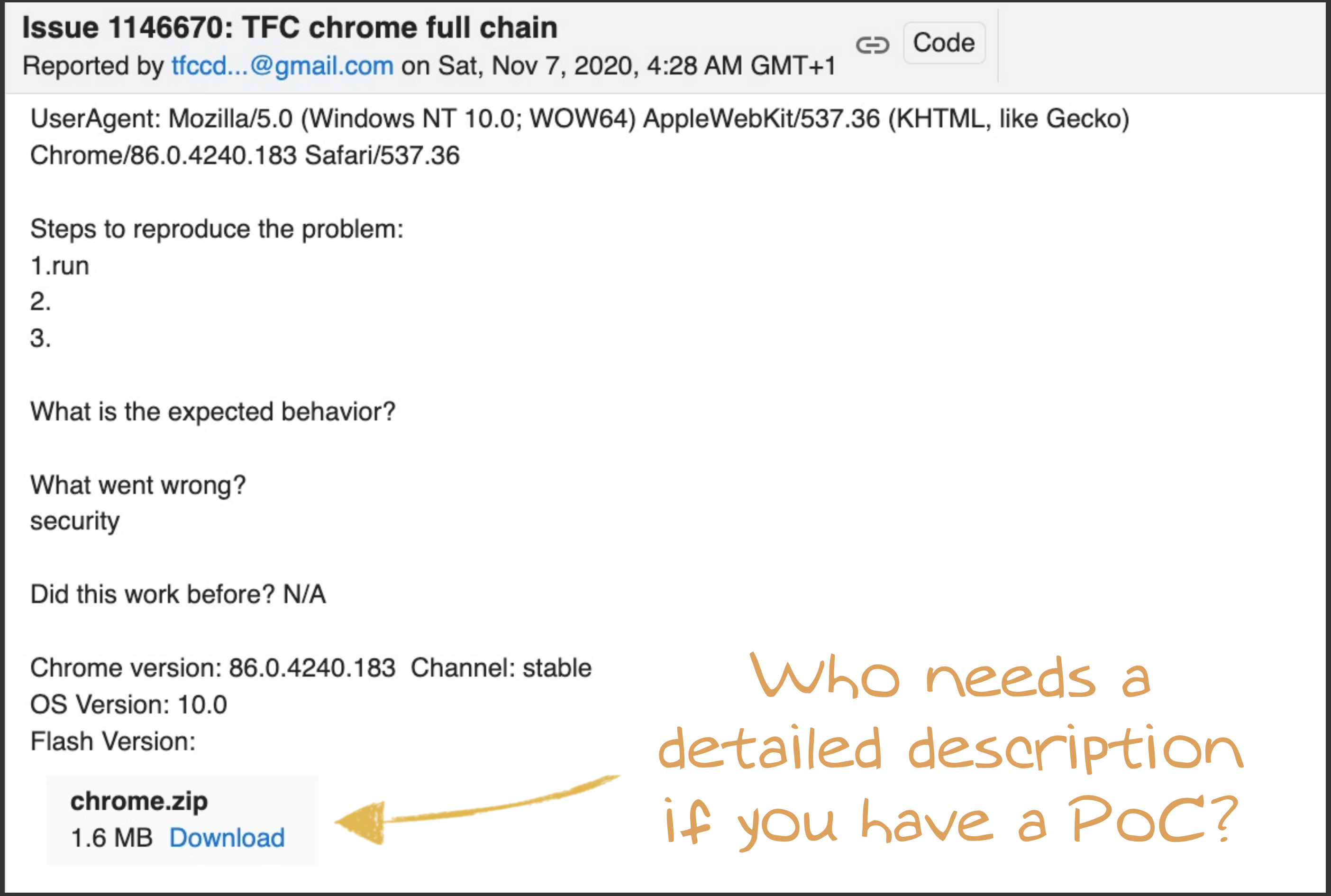
Parse and execute the response
(on top of all problems of SSRs)

Outdated Browsers

- Browsers often have vulnerabilities with high/critical severity
  - Usually disclosed 90 days after fix
  - Some with public PoC exploits
- No problem, as browsers update automatically … ?
- On consumer devices yes - but SSBs do not!
  - *"Each version of Puppeteer bundles a specific version of Chromium – the only version it is guaranteed to work with."* [1]

[1] https://github.com/puppeteer/puppeteer/tree/v14.0.0

History of a Chrome Bug


Issue 1146670: TFC chrome full chain
Reported by tfccd...@gmail.com on Sat, Nov 7, 2020, 4:28 AM GMT+1

UserAgent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.183 Safari/537.36

Steps to reproduce the problem:
1.run
2.
3.

What is the expected behavior?

What went wrong?
security

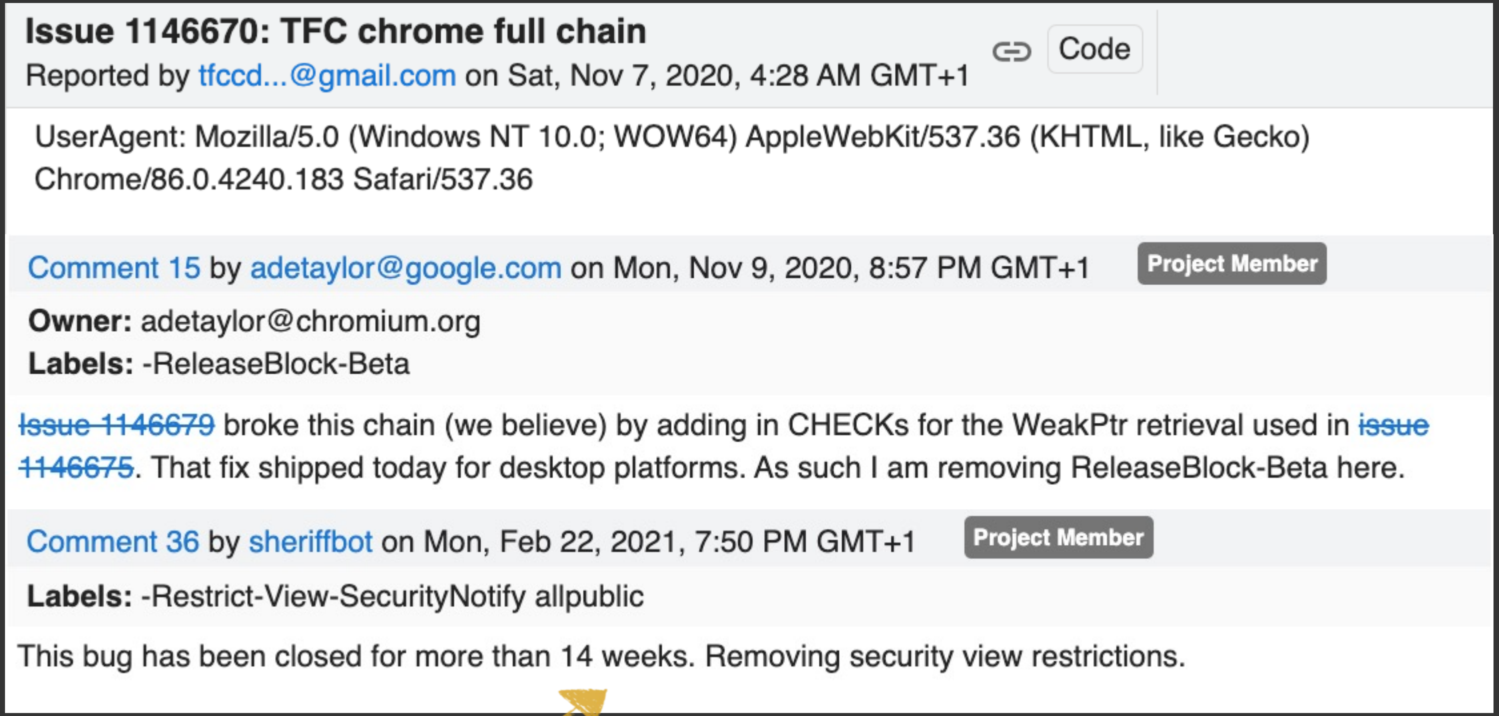
Did this work before? N/A

Chrome version: 86.0.4240.183 Channel: stable
OS Version: 10.0
Flash Version:

chrome.zip
1.6 MB Download

Who needs a detailed description if you have a PoC?

History of a Chrome Bug



Issue 1146670: TFC chrome full chain
Reported by tfccd...@gmail.com on Sat, Nov 7, 2020, 4:28 AM GMT+1

UserAgent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.183 Safari/537.36

Comment 15 by adetaylor@google.com on Mon, Nov 9, 2020, 8:57 PM GMT+1 Project Member

Owner: adetaylor@chromium.org
Labels: -ReleaseBlock-Beta

~~Issue 1146679~~ broke this chain (we believe) by adding in CHECKs for the WeakPtr retrieval used in ~~issue 1146675~~. That fix shipped today for desktop platforms. As such I am removing ReleaseBlock-Beta here.

Comment 36 by sheriffbot on Mon, Feb 22, 2021, 7:50 PM GMT+1 Project Member

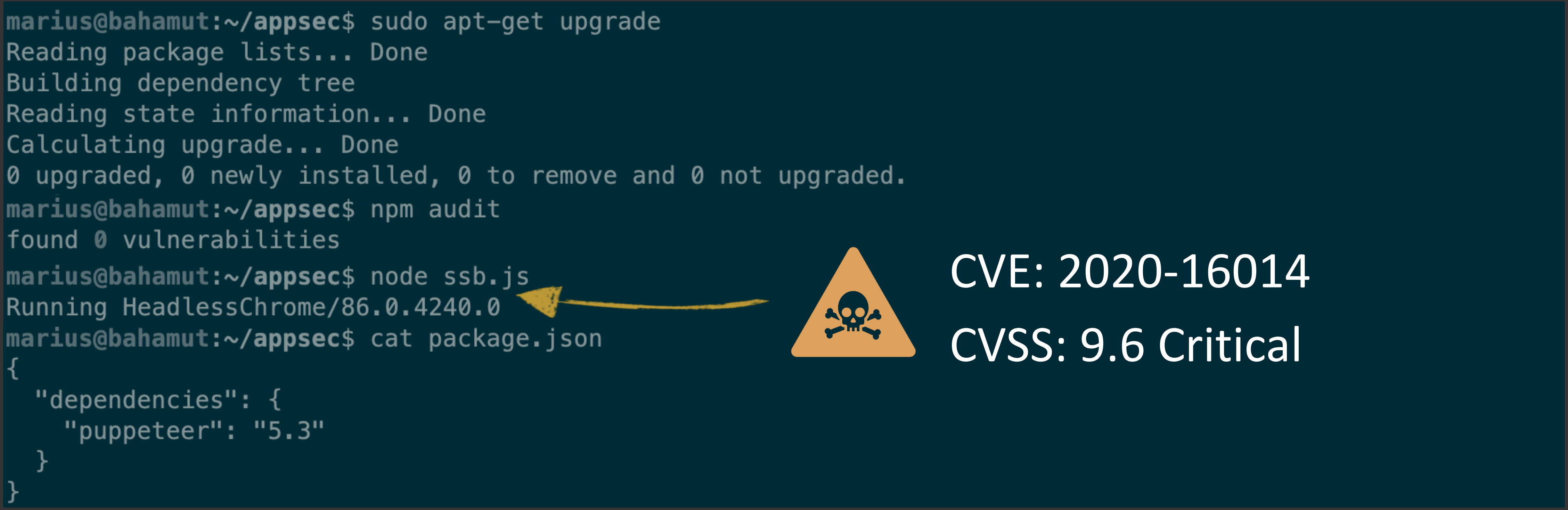
Labels: -Restrict-View-SecurityNotify allpublic

This bug has been closed for more than 14 weeks. Removing security view restrictions.

Kudos for extremely fast fix!

Everyone has updated by now, right?!

The Issue in a Nutshell

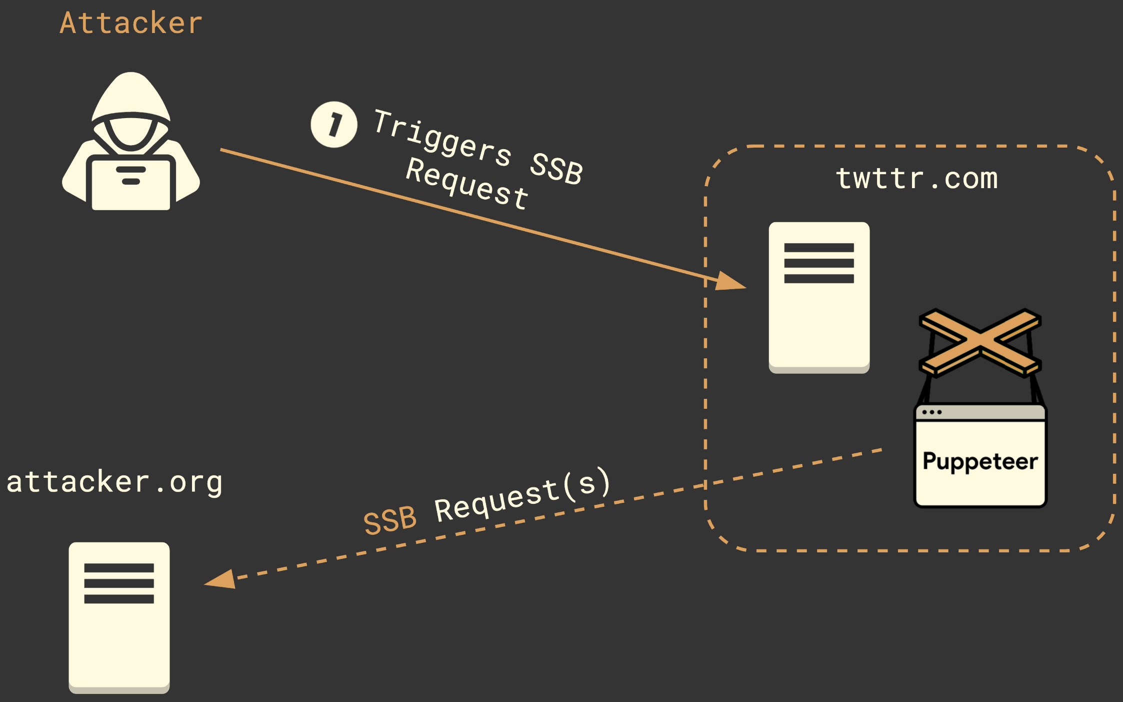


```
marius@bahamut:~/appsec$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
marius@bahamut:~/appsec$ npm audit
found 0 vulnerabilities
marius@bahamut:~/appsec$ node ssb.js
Running HeadlessChrome/86.0.4240.0
marius@bahamut:~/appsec$ cat package.json
{
  "dependencies": {
    "puppeteer": "5.3"
  }
}
```

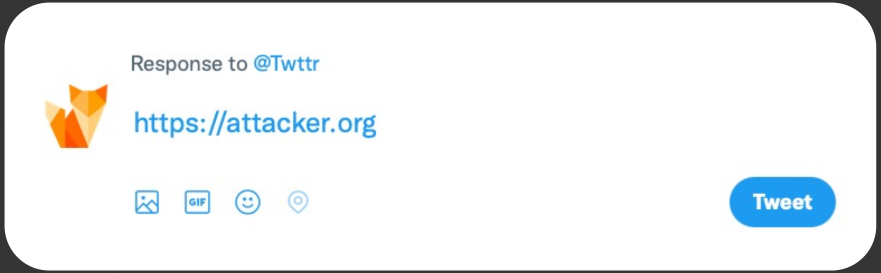
CVE: 2020-16014

CVSS: 9.6 Critical

Attack Scenario

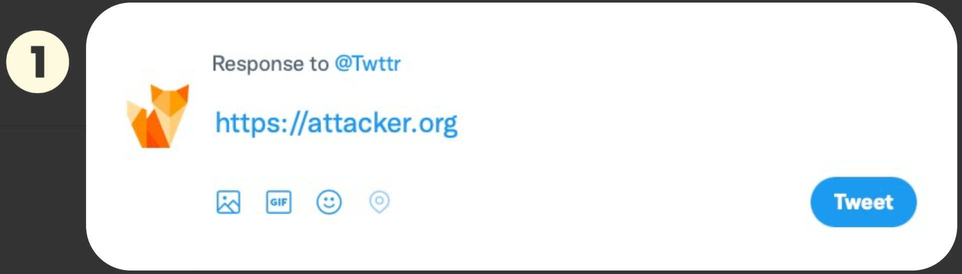


1


Response to @Twttr

https://attacker.org

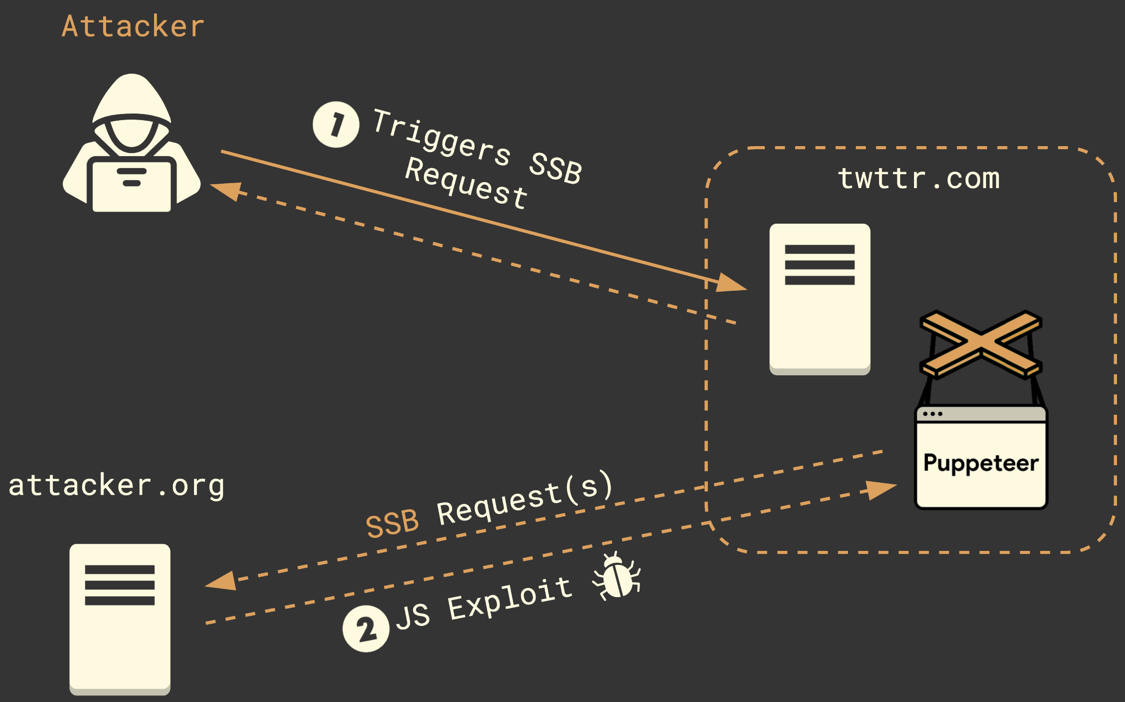
Attack Scenario


1

Response to @Twttr

https://attacker.org

Tweet

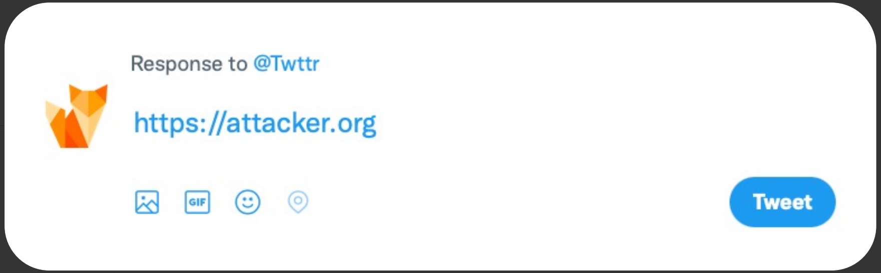


2

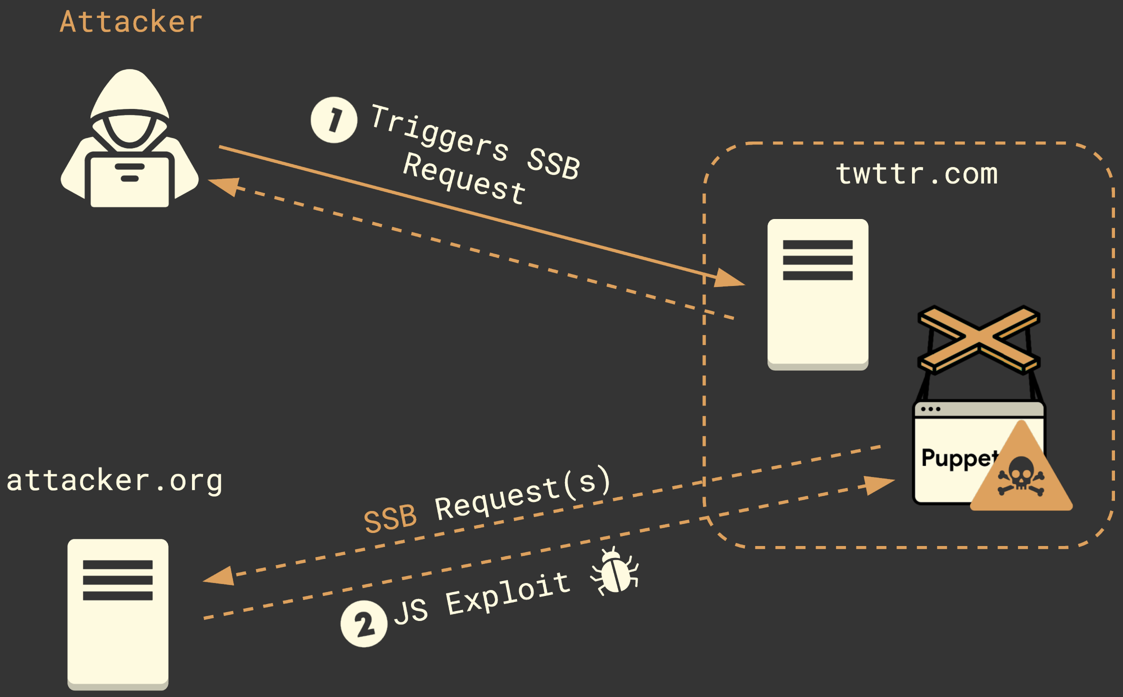
```
<html><head><script>function boom() { var fuzz1 = document.getElementById(
"fuzz1"); fuzz2.after(fuzz1); setTimeout('location.reload(true);',
100);}function boom2() { var fuzz3 = document.getElementById("fuzz3"); var
fuzz4 = document.getElementById("fuzz4"); fuzz4.appendChild(fuzz3);}</script>
</head><body onload=boom()><option id="fuzz3" ><iframe id="fuzz2"
srcdoc="AAAAAAAAAAA" onload="boom2()"></iframe><option id="fuzz4" ></opti
<portal id="fuzz1" ></portal></body></html>
```

Attack Scenario

1


Response to @Twttr

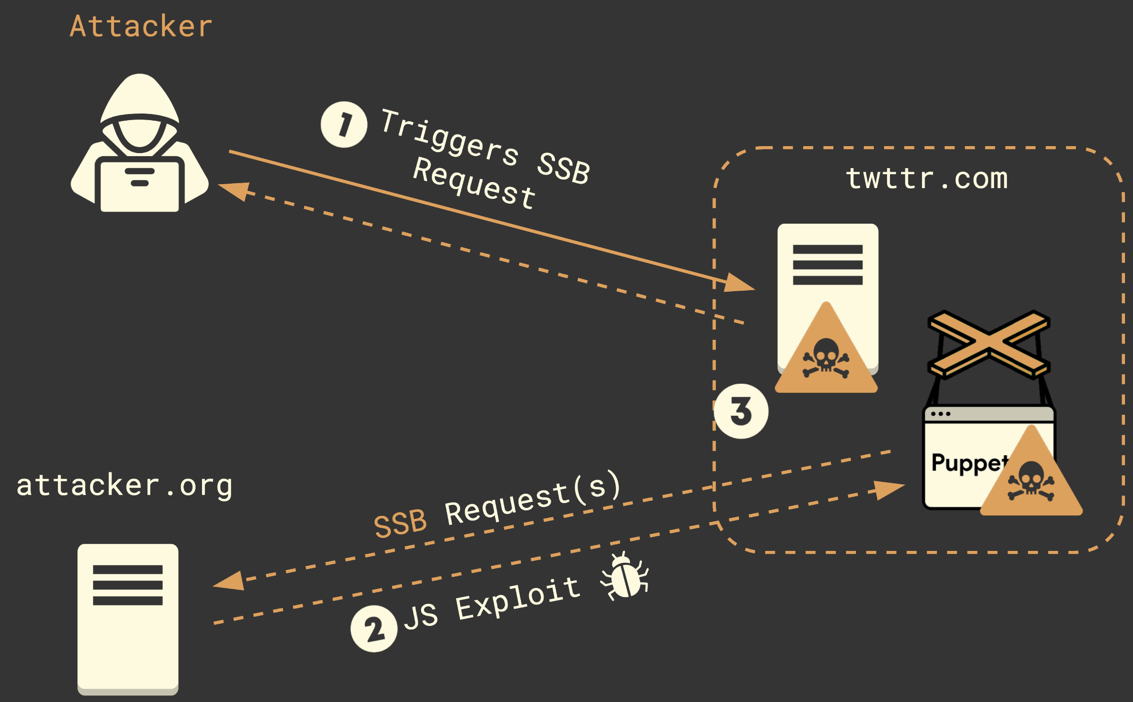
https://attacker.org



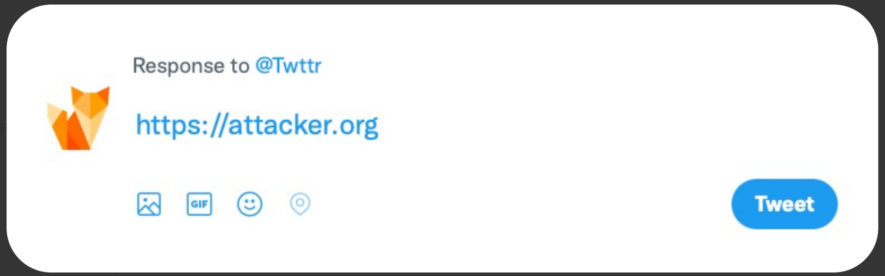
2

```
<html><head><script>function boom() { var fuzz1 = document.getElementById(
"fuzz1"); fuzz2.after(fuzz1); setTimeout('location.reload(true);',
100);}function boom2() { var fuzz3 = document.getElementById("fuzz3"); var
fuzz4 = document.getElementById("fuzz4"); fuzz4.appendChild(fuzz3);}</script>
</head><body onload=boom()><option id="fuzz3" ><iframe id="fuzz2"
srcdoc="AAAAAAAAAAA" onload="boom2()"></iframe><option id="fuzz4" ></opti
<portal id="fuzz1" ></portal></body></html>
```

Attack Scenario

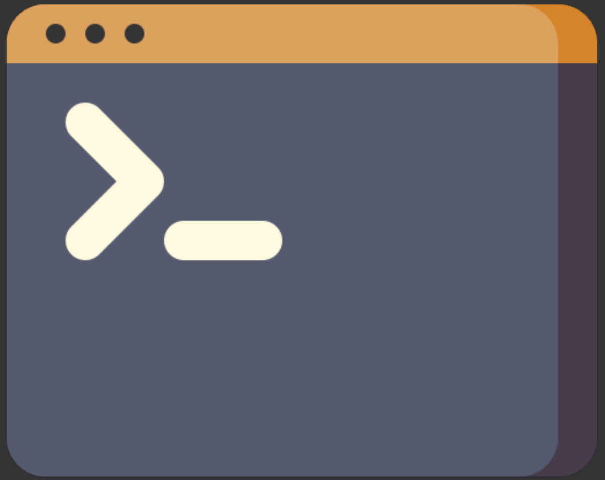


1


Response to @Twttr

https://attacker.org

3



2

```
<html><head><script>function boom() { var fuzz1 = document.getElementById(
"fuzz1"); fuzz2.after(fuzz1); setTimeout('location.reload(true);',
100);}function boom2() { var fuzz3 = document.getElementById("fuzz3"); var
fuzz4 = document.getElementById("fuzz4"); fuzz4.appendChild(fuzz3);}</script>
</head><body onload=boom()><option id="fuzz3" ><iframe id="fuzz2"
srcdoc="AAAAAAAAAAA" onload="boom2()"></iframe><option id="fuzz4" ></opti
<portal id="fuzz1" ></portal></body></html>
```

Fusion!!

A06:2021-Vulnerable and Outdated Components
A10:2021-Server-Side Request Forgery

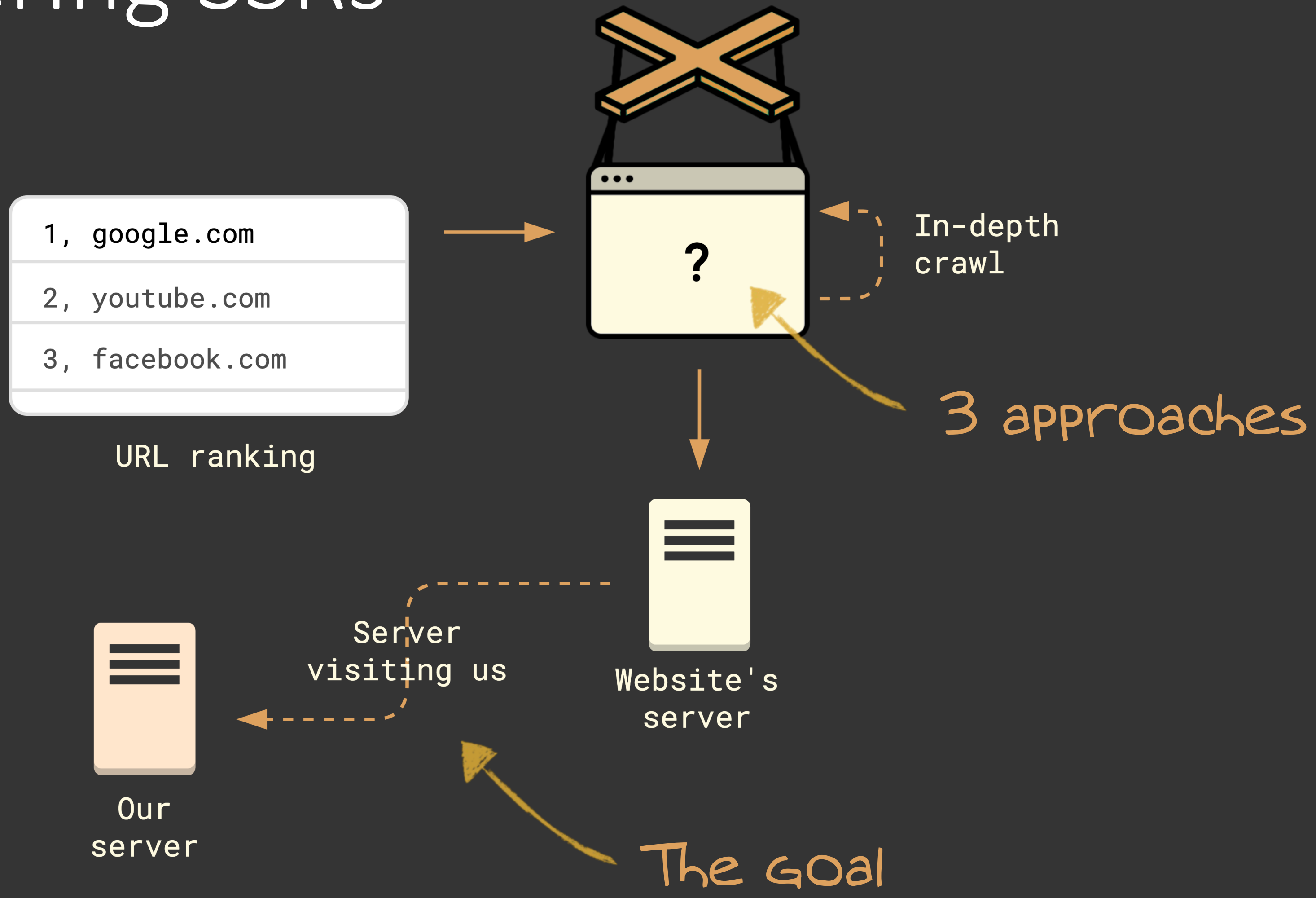
Automated Detection and Large-Scale Study

Automatic Detection

1. How to trigger SSRs?
2. How to reliably discern SSRs from SSBs?
3. How to determine their actual browser version?
4. How many are vulnerable to public exploits?

→ Large scale study on 100,000 websites

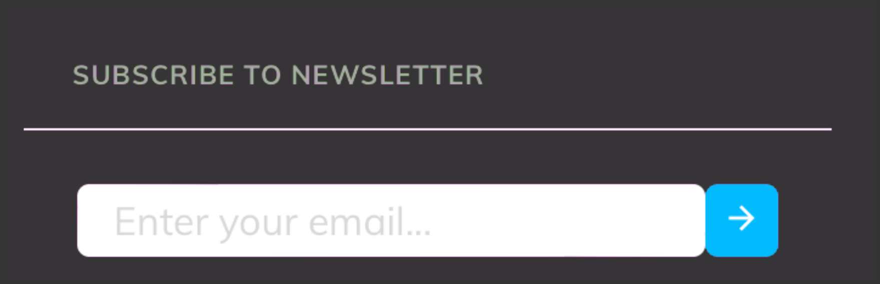
1 Discovering SSRs



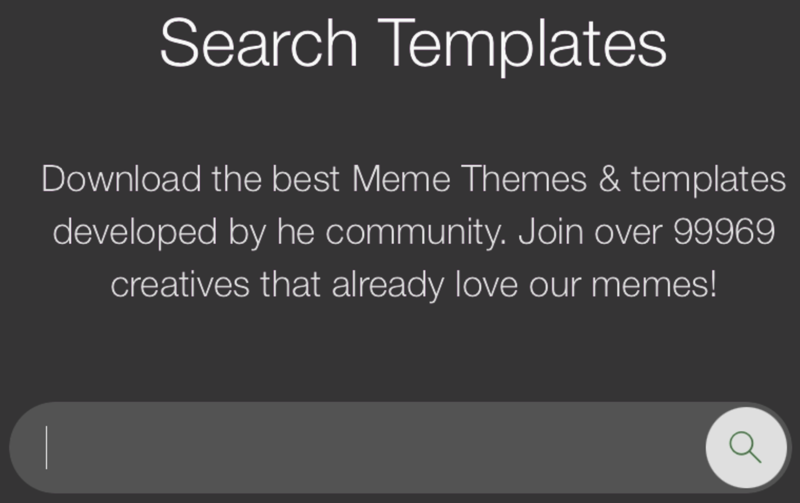
➊ Discovering SSRs

3 ways to entice websites to visit our unique URLs

- **Forms** – Submit distinct URLs



SUBSCRIBE TO NEWSLETTER



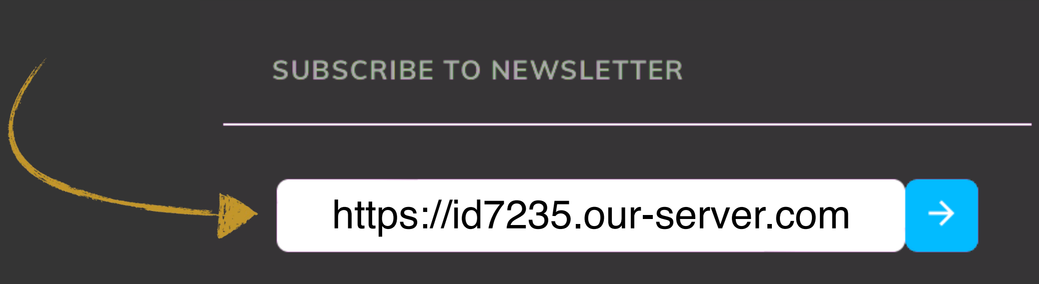
Search Templates

Download the best Meme Themes & templates developed by he community. Join over 99969 creatives that already love our memes!

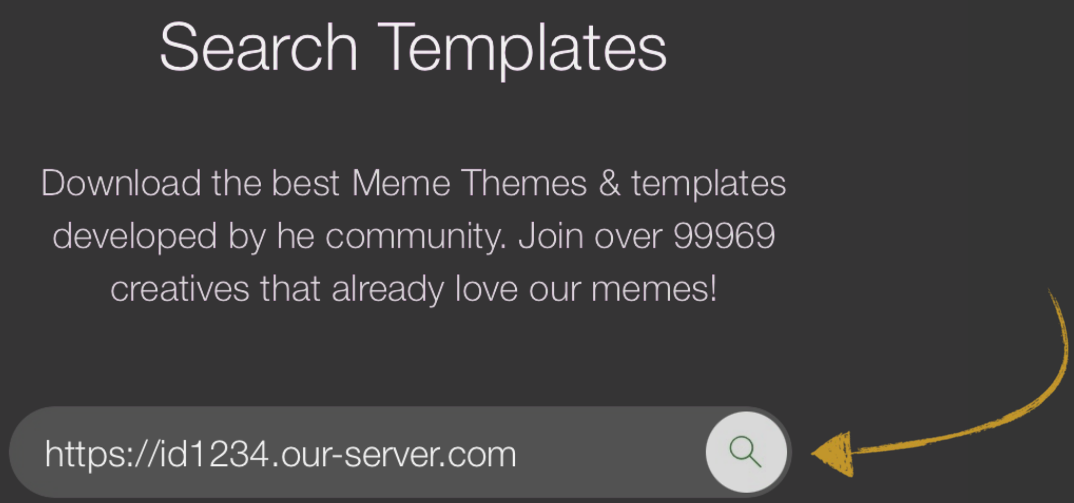
❶ Discovering SSRs

3 ways to entice websites to visit our unique URLs

- **Forms** – Submit distinct URLs
  - Enter our URL where possible
  - Once per unique form


SUBSCRIBE TO NEWSLETTER

https://id7235.our-server.com


Search Templates

Download the best Meme Themes & templates developed by he community. Join over 99969 creatives that already love our memes!

https://id1234.our-server.com

1 Discovering SSRs

3 ways to entice websites to visit our unique URLs

- **Forms** – Submit with our URLs
- **Headers** – Set our URLs as REFERER header on each request

```
GET /index.php HTTP/1.1
HOST: google.com
REFERER: id3624.our-server.com

0
```

❶ Discovering SSRs

3 ways to entice websites to visit our unique URLs

- **Forms** – Submit with our URLs
- **Headers** – Set our URLs as REFERER header on each request
- **Query** – Modify discovered URLs and replay with different values

```
http://example.com?from=foo.com&id=3
```

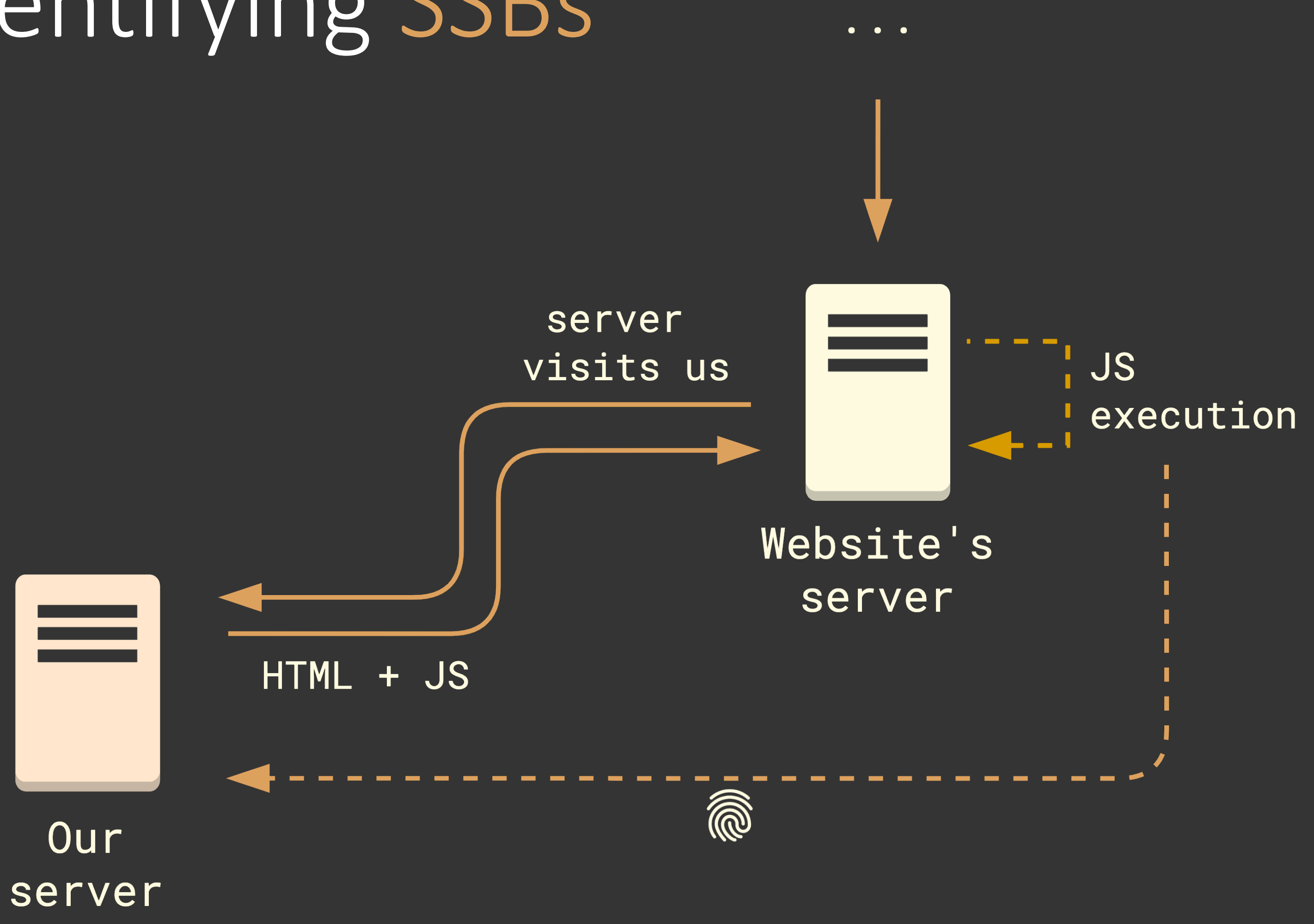
❶ Discovering SSRs

3 ways to entice websites to visit our unique URLs

- **Forms** – Submit with our URLs
- **Headers** – Set our URLs as REFERER header on each request
- **Query** – Modify discovered URLs and replay with different values

```
http://example.com?from=foo.com&id=3
http://example.com?from=id9543.our-server.com&id=3
```

2 Identifying SSBs



❷ Identifying SSBs

- Our server replies with HTML + JavaScript
  - JavaScript collects some client-side information and sends it
  - If this happens, it is a browser

2 Identifying SSBs

- Our server replies with HTML + JavaScript
    - JavaScript collects some client-side information and sends it
    - If this happens, it is a browser

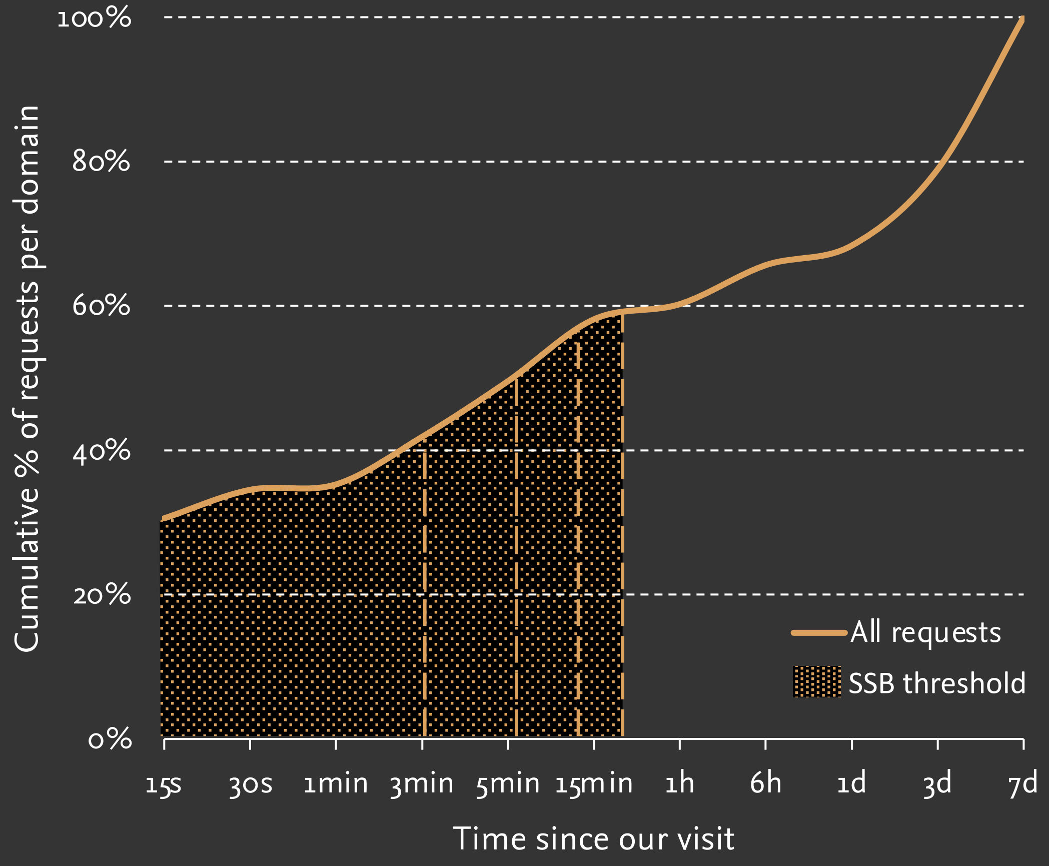
- How do we know this was not a human visitor?
    - Likely, if visit happens within the first 3 minutes after our URL submission

2 Identifying SSBs

- Our server replies with HTML + JavaScript
  - JavaScript collects some client-side information and sends it
  - If this happens, it is a browser
- How do we know this was not a human visitor?
  - Likely, if visit happens within the first 3 minutes after our URL submission
- What about slow bots?
  - If additional bot indicators are present, increase time up to 24 minutes

2 Prevalence of SSRs & SSBs

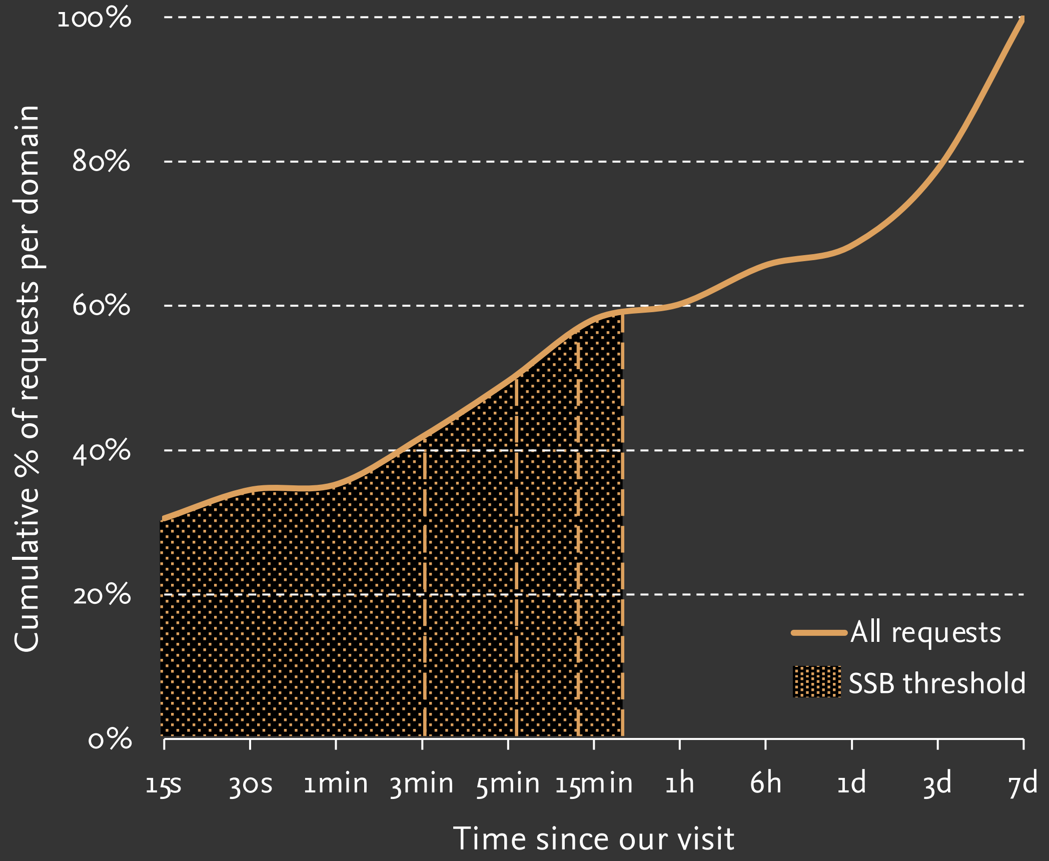
- Visited top 100k domains
  - Up to 50 subpages
  - Visited 2.6M pages on 79k sites



2 Prevalence of SSRs & SSBs

- Visited top 100k domains
  - Up to 50 subpages
  - Visited 2.6M pages on 79k sites

SSB – SSRs within threshold and with JavaScript Support



Effectiveness of the approaches

- 168,055 incoming SSRs from 4850 domains
  - Header 3799
  - Param 353
  - Form 794
- 3,264 incoming SSBs from 254 domains (within the first 3 minutes)
  - Header 58
  - Param 34
  - Form 167

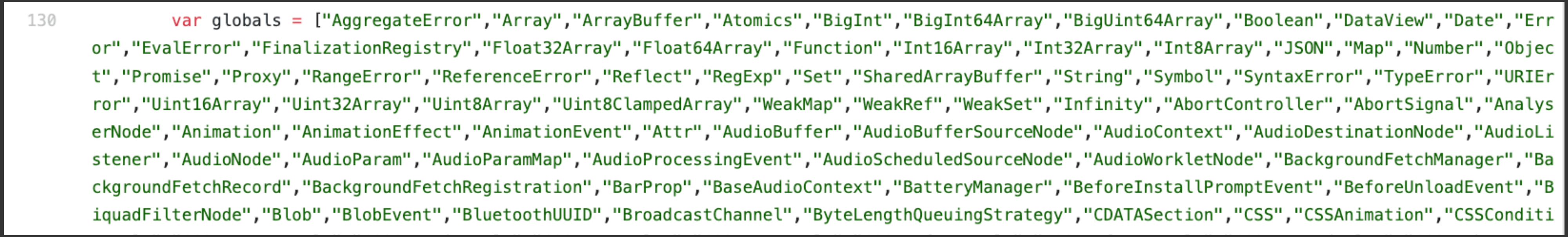
Filling forms is the most successful approach

❸ Detecting Browser Versions

- User agent string too easy to spoof and can’t be trusted :/
- Maybe find behavioral differences?

③ Detecting Browser Versions

- User agent string too easy to spoof
  - Instead, extract all JavaScript objects in `window`



```
        var globals = ["AggregateError","Array","ArrayBuffer","Atomics","BigInt","BigInt64Array","BigUint64Array","Boolean","DataView","Date","Error","EvalError","FinalizationRegistry","Float32Array","Float64Array","Function","Int16Array","Int32Array","Int8Array","JSON","Map","Number","Object","Promise","Proxy","RangeError","ReferenceError","Reflect","RegExp","Set","SharedArrayBuffer","String","Symbol","SyntaxError","TypeError","URIError","Uint16Array","Uint32Array","Uint8Array","Uint8ClampedArray","WeakMap","WeakRef","WeakSet","Infinity","AbortController","AbortSignal","AnalyserNode","Animation","AnimationEffect","AnimationEvent","Attr","AudioBuffer","AudioBufferSourceNode","AudioContext","AudioDestinationNode","AudioListener","AudioNode","AudioParam","AudioParamMap","AudioProcessingEvent","AudioScheduledSourceNode","AudioWorkletNode","BackgroundFetchManager","BackgroundFetchRecord","BackgroundFetchRegistration","BarProp","BaseAudioContext","BatteryManager","BeforeInstallPromptEvent","BeforeUnloadEvent","BiquadFilterNode","Blob","BlobEvent","BluetoothUUID","BroadcastChannel","ByteLengthQueuingStrategy","CDATASection","CSS","CSSAnimation","CSSConditi
```

3 Detecting Browser Versions

- User agent string too easy to spoof
  - Instead, extract all JavaScript objects in `window`
  - Compare with compatibility data from MDN to find highest possible version

Feature of window	Feature supported since				**Feature exists in sample**	
	Chrome	Firefox	Opera	Safari	Sample 1	Sample 2
RTCCertificate	49	42	36	12	✓	✓
MutationObserver	26	14	15	7	✓	✓
WeakRef	84	79	-	-	✓	✓
TrustedScript	83	-	69	-	✓	✗
AggregateError	85	79	-	14	✗	✓

❸ Detecting Browser Versions

- User agent string too easy to spoof
  - Instead, extract all JavaScript objects in `window`
  - Compare with compatibility data from MDN to find highest possible version

Feature of window	Feature supported since				**Feature exists in sample**	
	Chrome	Firefox	Opera	Safari	Sample 1	Sample 2
RTCCertificate	49	42	36	12	✓	✓
MutationObserver	26	14	15	7	✓	✓
WeakRef	84	79	-	-	✓	✓
TrustedScript	83	-	69	-	✓	✗
AggregateError	85	79	-	14	✗	✓

3 Detecting Browser Versions

- User agent string too easy to spoof
  - Instead, extract all JavaScript objects in `window`
  - Compare with compatibility data from MDN to find highest possible version

Feature of window	Feature supported since				**Feature exists in sample**	
	Chrome	Firefox	Opera	Safari	Sample 1	Sample 2
RTCCertificate	49	42	36	12	✓	✓
MutationObserver	26	14	15	7	✓	✓
WeakRef	84	79	-	-	✓	✓
TrustedScript	83	-	69	-	✓	✗
AggregateError	85	79	-	14	✗	✓

3 Detecting Browser Versions

- User agent string too easy to spoof
  - Instead, extract all JavaScript objects in `window`
  - Compare with compatibility data from MDN to find highest possible version

Feature of window	Feature supported since				**Feature exists in sample**	
	Chrome	Firefox	Opera	Safari	Sample 1	Sample 2
RTCCertificate	49	42	36	12	✓	✓
MutationObserver	26	14	15	7	✓	✓
WeakRef	84	79	-	-	✓	✓
TrustedScript	83	-	69	-	✓	✗
AggregateError	85	79	-	14	✗	✓

❸ Detecting Browser Versions

- User agent string too easy to spoof
  - Instead, extract all JavaScript objects in `window`
  - Compare with compatibility data from MDN to find highest possible version

Feature of window	Feature supported since				**Feature exists in sample**	
	Chrome	Firefox	Opera	Safari	Sample 1	Sample 2
RTCCertificate	49	42	36	12	✓	✓
MutationObserver	26	14	15	7	✓	✓
WeakRef	84	79	-	-	✓	✓
TrustedScript	83	-	69	-	✓	✗
AggregateError	85	79	-	14	✗	✓

3 Detecting Browser Versions

- User agent string too easy to spoof
  - Instead, extract all JavaScript objects in `window`
  - Compare with compatibility data from MDN to find highest possible version

Feature of window	Feature supported since				**Feature exists in sample**	
	Chrome	Firefox	Opera	Safari	Sample 1	Sample 2
RTCCertificate	49	42	36	12	✓	✓
MutationObserver	26	14	15	7	✓	✓
WeakRef	84	79	-	-	✓	✓
TrustedScript	83	-	69	-	✓	✗
AggregateError	85	79	-	14	✗	✓

③ Detecting Browser Versions

- User agent string too easy to spoof
  - Instead, extract all JavaScript objects in `window`
  - Compare with compatibility data from MDN to find highest possible version

Feature of window	Feature supported since				**Feature exists in sample**	
	Chrome	Firefox	Opera	Safari	Sample 1	Sample 2
RTCCertificate	49	42	36	12	✓	✓
MutationObserver	26	14	15	7	✓	✓
WeakRef	84	79	-	-	✓	✓
TrustedScript	83	-	69	-	✓	✗
AggregateError	85	79	-	14	✗	✓
					Chrome 84	

③ Detecting Browser Versions

- User agent string too easy to spoof
  - Instead, extract all JavaScript objects in `window`
  - Compare with compatibility data from MDN to find highest possible version

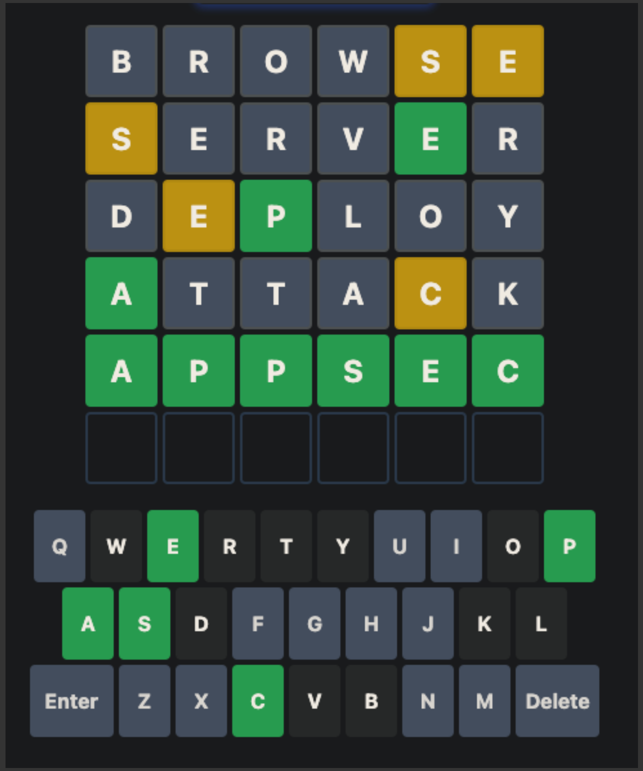
Feature of window	Feature supported since				**Feature exists in sample**	
	Chrome	Firefox	Opera	Safari	Sample 1	Sample 2
RTCCertificate	49	42	36	12	✓	✓
MutationObserver	26	14	15	7	✓	✓
WeakRef	84	79	-	-	✓	✓
TrustedScript	83	-	69	-	✓	✗
AggregateError	85	79	-	14	✗	✓

③ Detecting Browser Versions

- User agent string too easy to spoof
  - Instead, extract all JavaScript objects in `window`
  - Compare with compatibility data from MDN to find highest possible version

Feature of window	Feature supported since				Feature exists in sample	
	Chrome	Firefox	Opera	Safari	Sample 1	Sample 2
RTCCertificate	49	42	36	12	✓	✓
MutationObserver	26	14	15	7	✓	✓
WeakRef	84	79	-	-	✓	✓
TrustedScript	83	-	69	-	✓	✗
AggregateError	85	79	-	14	✗	✓
					Chrome 84	Firefox >= 79

Remember these?



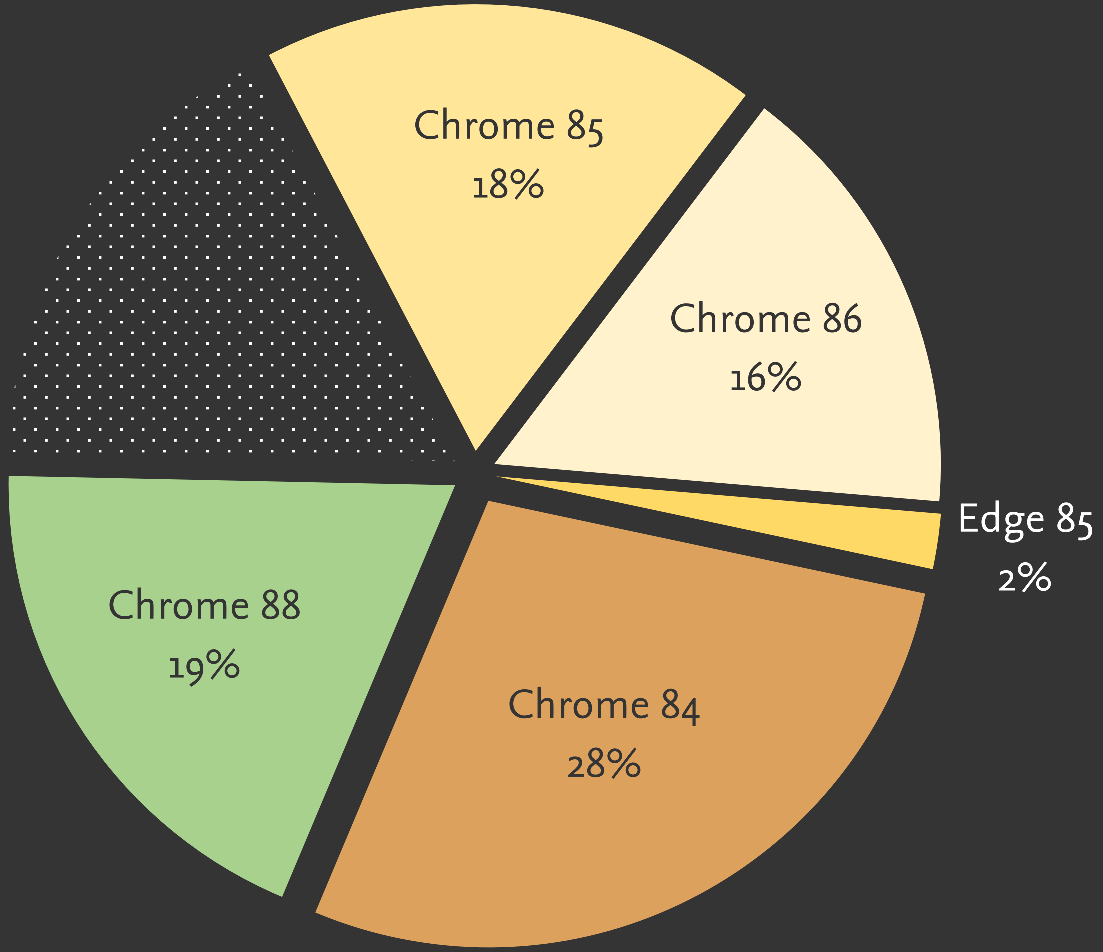
Liars

- About 25% lied about their user agent!
  - Some cases HTTP UA != JS UA
  - Most cases user agent != platform

- navigator.platform "Linux x86_64" but user agent
  - CPU iPhone OS 13_7 [...] Version/13.1.2
  - Windows NT 6.1 [...] Chrome/83.0.4103.106
  - iPad; CPU OS 11_4 [...] Version/11.0
  - …

3 Browser Versions

- Most popular browsers
  - 28%: Chrome 84 from July 2020
  - 19%: Chrome 88 from Jan 2021
  - 18%: Chrome 85 from Aug 2020
  - 16%: Chrome 86 from Oct 2020
  - 2%: Edge 85 from Aug 2020

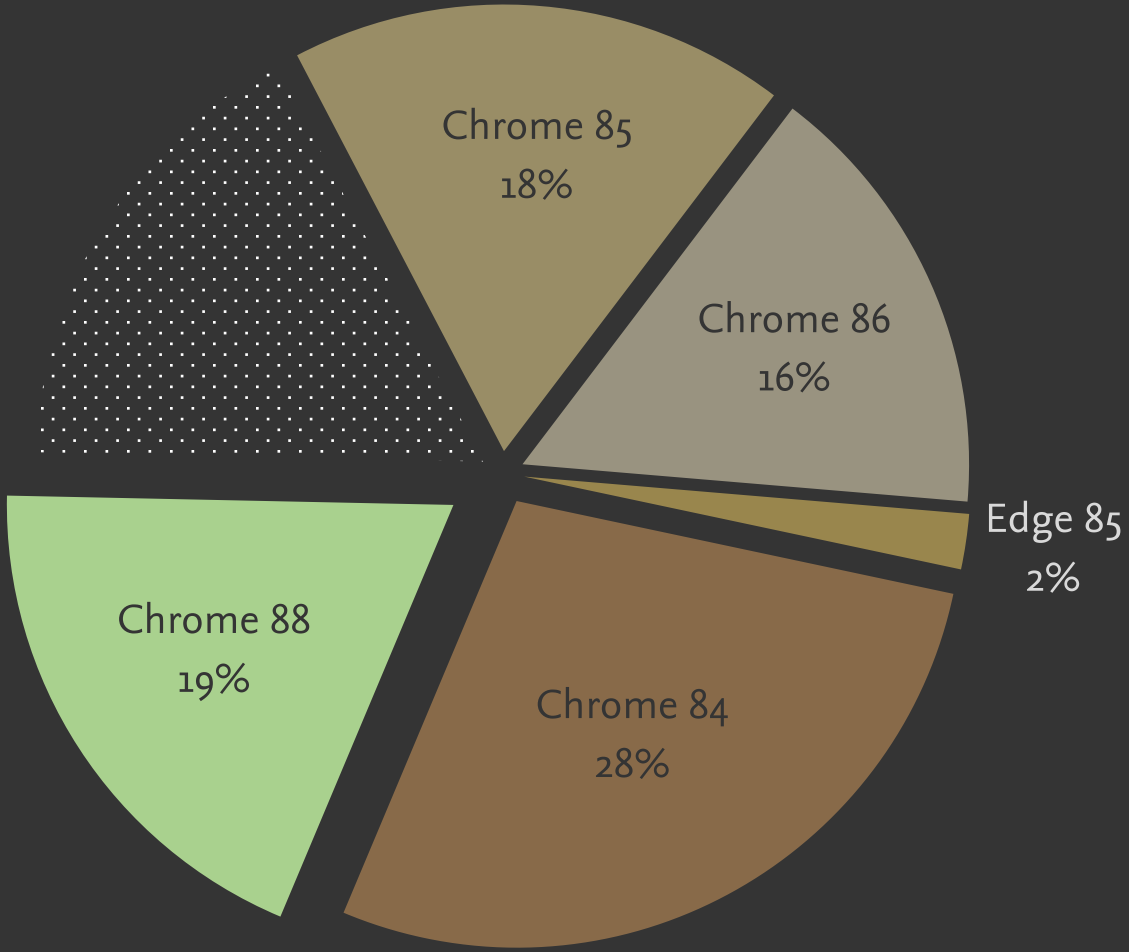


2021
Data collection in March 2021
Latest stable browser Chrome 88/89

③ Browser Versions

- Most popular browsers

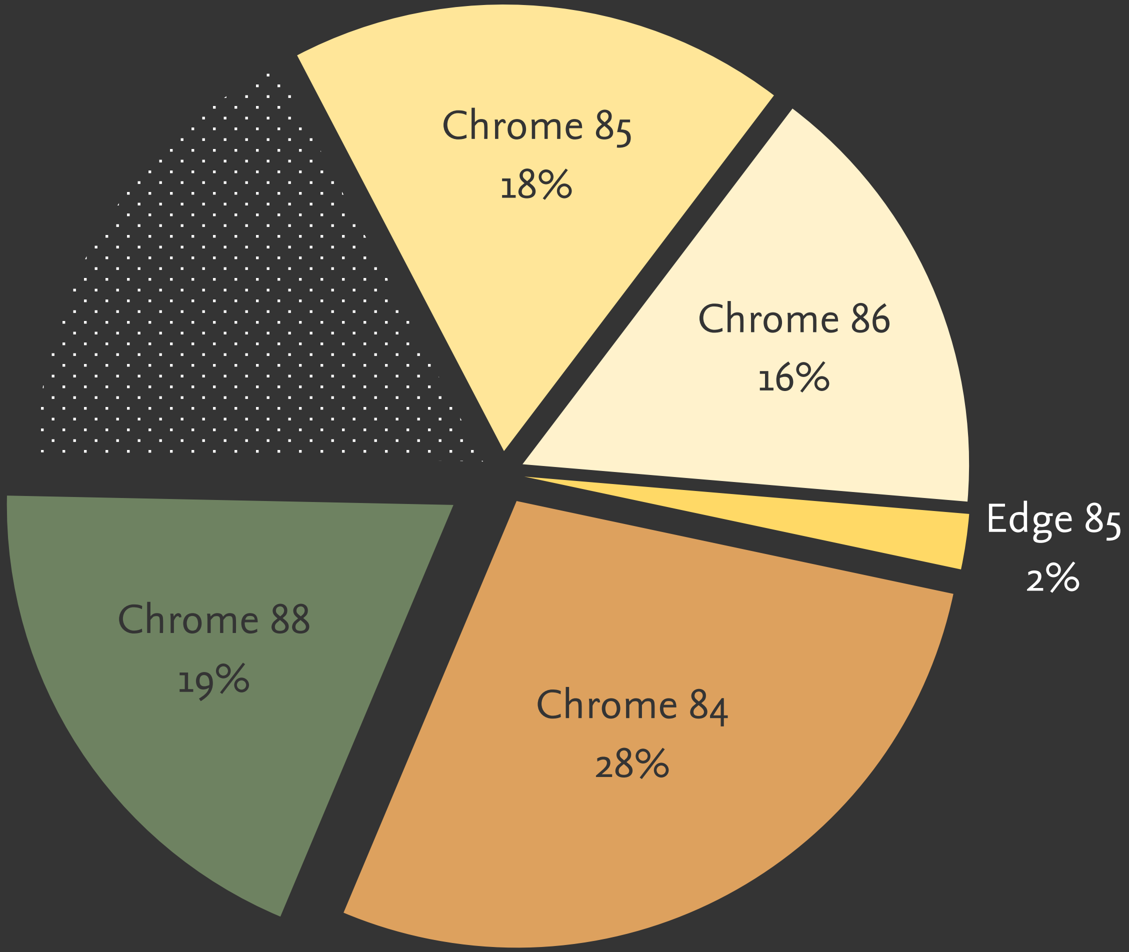
●	19%:	Chrome 88	from Jan 2021
	28%:	Chrome 84	from July 2020
	18%:	Chrome 85	from Aug 2020
	16%:	Chrome 86	from Oct 2020
	2%:	Edge 85	from Aug 2020



Data collection in March 2021
Latest stable browser Chrome 88/89

4 Vulnerable SSBs

- Most popular browsers
  - 19%: Chrome 88 from Jan 2021
  - 28%: Chrome 84 from July 2020
  - 18%: Chrome 85 from Aug 2020
  - 16%: Chrome 86 from Oct 2020
  - 2%: Edge 85 from Aug 2020



2021

Data collection in March 2021
Latest stable browser Chrome 88/89

4 Vulnerable SSBs

- Most popular browsers
  - 19%: Chrome 88 from Jan 2021
  - 28%: Chrome 84 from July 2020
  - 18%: Chrome 85 from Aug 2020
  - 16%: Chrome 86 from Oct 2020
  - 2%: Edge 85 from Aug 2020

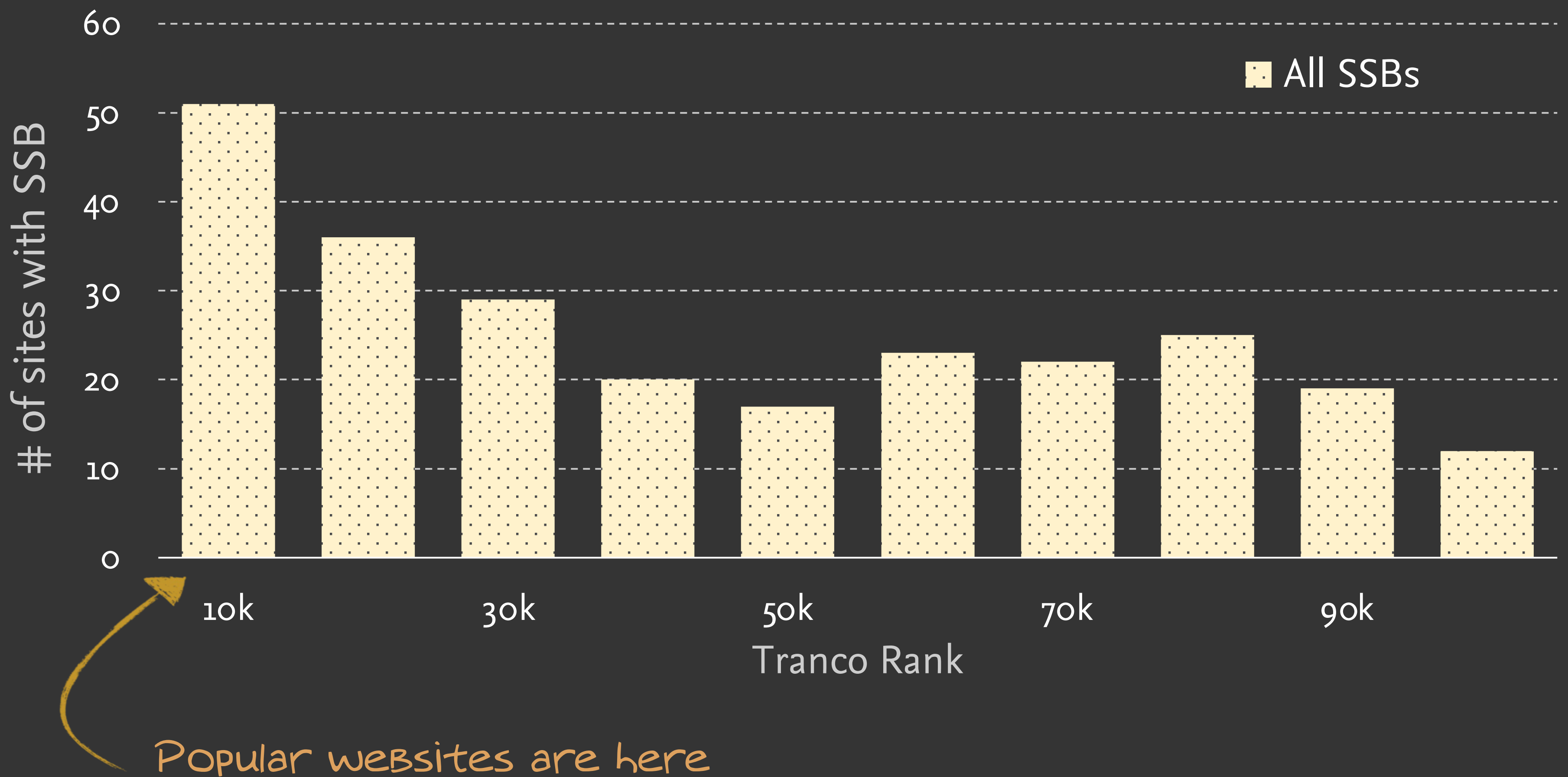
A lot of vulnerable implementations!

Browser	CVE
Chrome 86	CVE 2020-16015
Chrome 85	CVE 2020-6575
Edge 85	CVE 2020-6574
Chrome 84	CVE 2020-6559

Data collection in March 2021
Latest stable browser Chrome 88/89

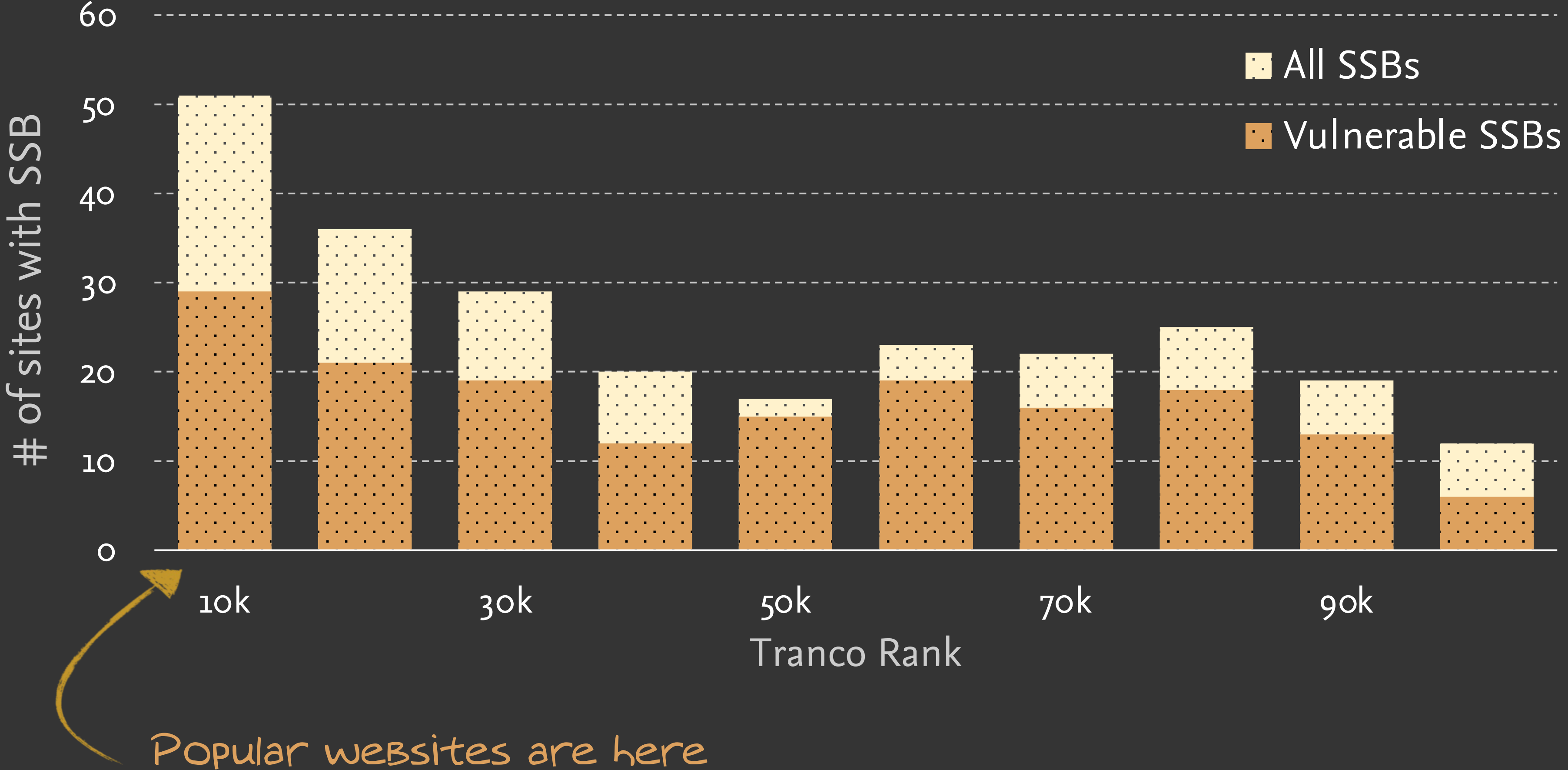
4 Vulnerable SSBs Distribution

254 domains with SSBs



4 Vulnerable SSBs Distribution

168 / 254 domains with SSBs vulnerable to public exploits



OWASP 2022 GLOBAL AppSec | SAN FRANCISCO NOV 14-18

Discussion and Conclusion

There is more out there

- Crawling depth
  - Large-scale study means only shallow crawl of each site
- User interaction
  - Could not analyze sites behind login or other complex flows
- Bot detection
  - Very conservative threshold of 3 minutes

SSRF Countermeasures

- Isolate the machine from your internal network
  - Also means no other sensitive services running the same machine
  - Prevent classical SSRF (and lateral movement from a compromised server)

- Enforce the URL schema
  - Prevents attacks via gopher:// and more

- Allow list of destinations is tricky in practice
  - Difficult to implement due to redirects and DNS rebinding
  - In the link preview use-case impossible

SSB Countermeasures

- Keep the browser diligently up-to-date
  - Regular updates of all your project’s dependencies
  - Be aware that various tools might miss these ’bundled’ vulnerabilities

- Isolate the browser from the OS
  - Run as non-privileged user, consider additional hardening
  - Make sure that user has no access to sensitive secrets

Summary

- Unique attack surface
  - Execute untrusted code on server-side
  - Browsers contain critical bugs at high rate
  - Are not updated automatically

$\longrightarrow$ Really dangerous combination!

- Identified 168/254 vulnerable SSBs

$\longrightarrow$ 2 out of 3 deployments vulnerable!

Takeaways

Everyone

- Be aware of server-side browsers
  - Plausible reasons to expose them to the Internet
  - As we showed, already happening

Breakers

- How to detect them on remote servers
  - Rely on browser features, not user agents

Defenders

- Defending against traditional SSRF attacks is not enough
  - Regularly update your dependencies (not only your system packages)

CASA
CYBER SECURITY IN THE AGE OF LARGE-SCALE ADVERSARIES

Thanks for listening :)
Questions?

m.musch@tu-braunschweig.de

@m4riuz

Looking for a job after my PhD

IAS INSTITUTE FOR APPLICATION SECURITY

Gefördert durch DFG Deutsche Forschungsgemeinschaft

Technische Universität Braunschweig