CASA
CYBER SECURITY IN THE AGE OF LARGE-SCALE ADVERSARIES

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

Marius Musch
TU Braunschweig

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

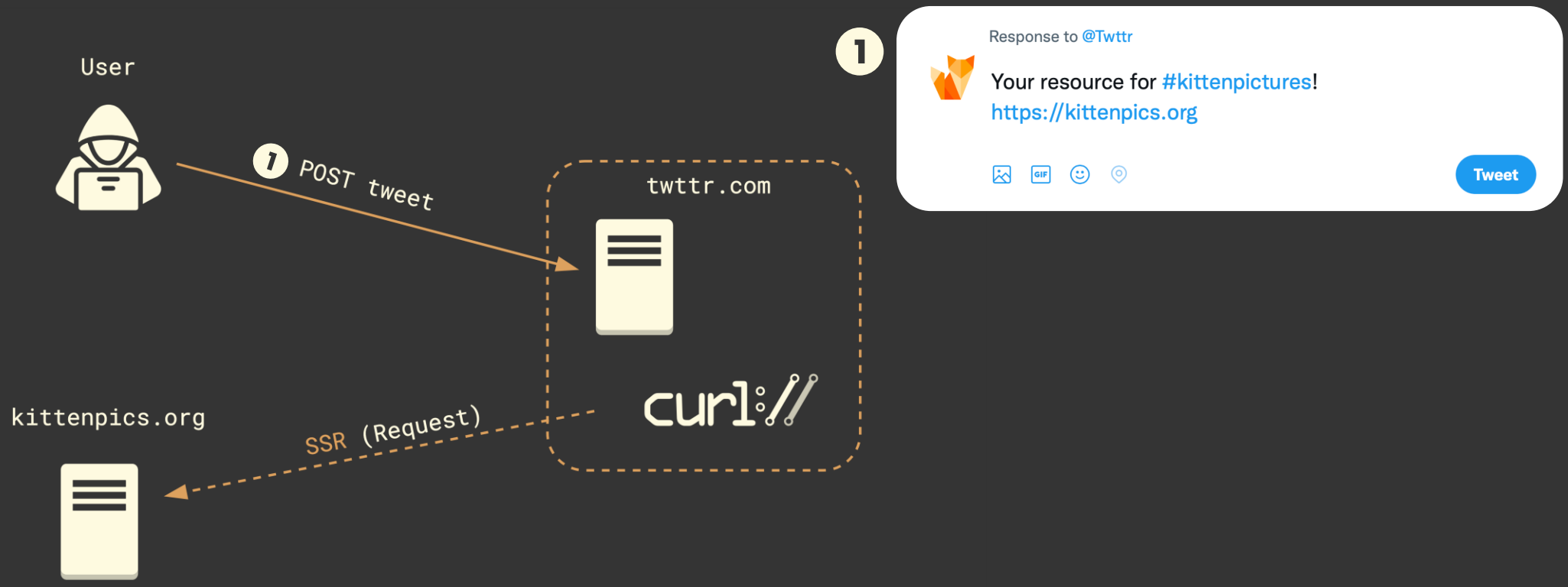
IAS INSTITUTE FOR APPLICATION SECURITY

Gefördert durch
DFG Deutsche Forschungsgemeinschaft

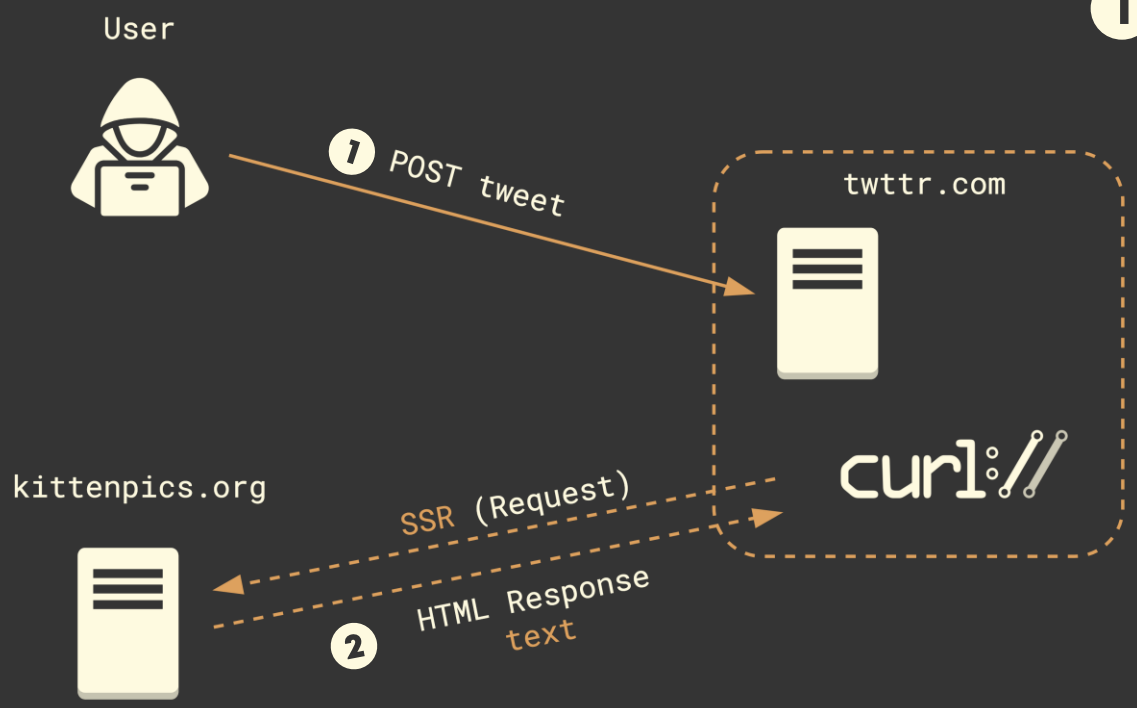
Technische Universität Braunschweig

The Scenario

Request for Preview



Request for Preview

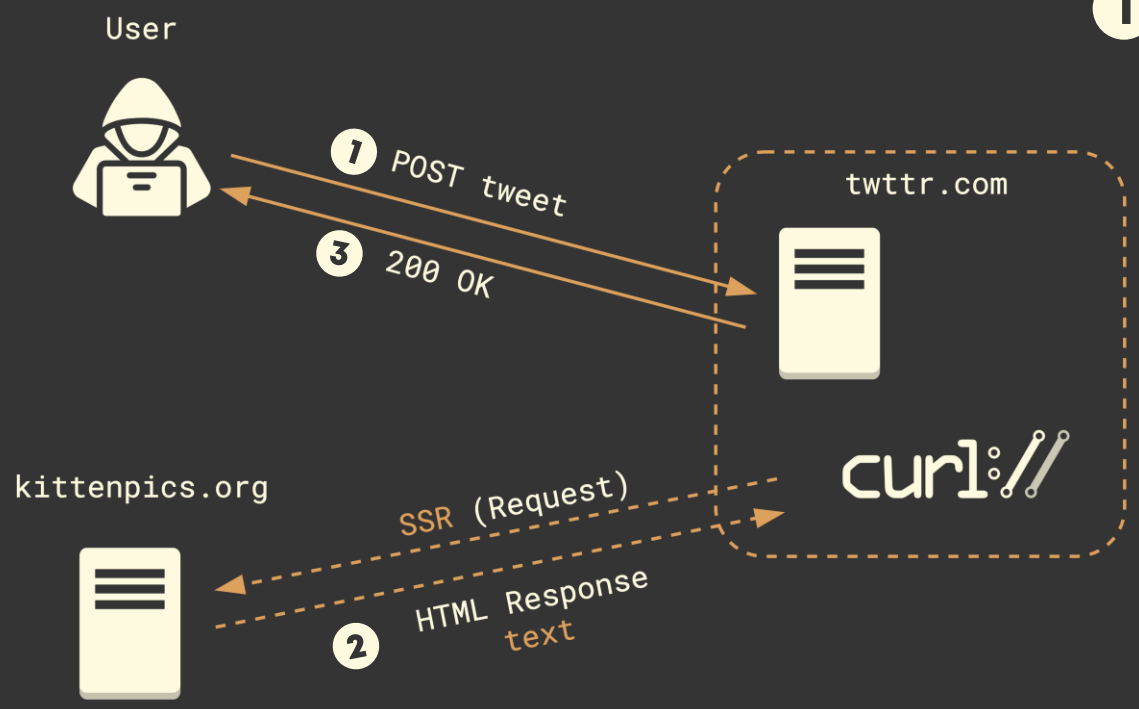


1

2

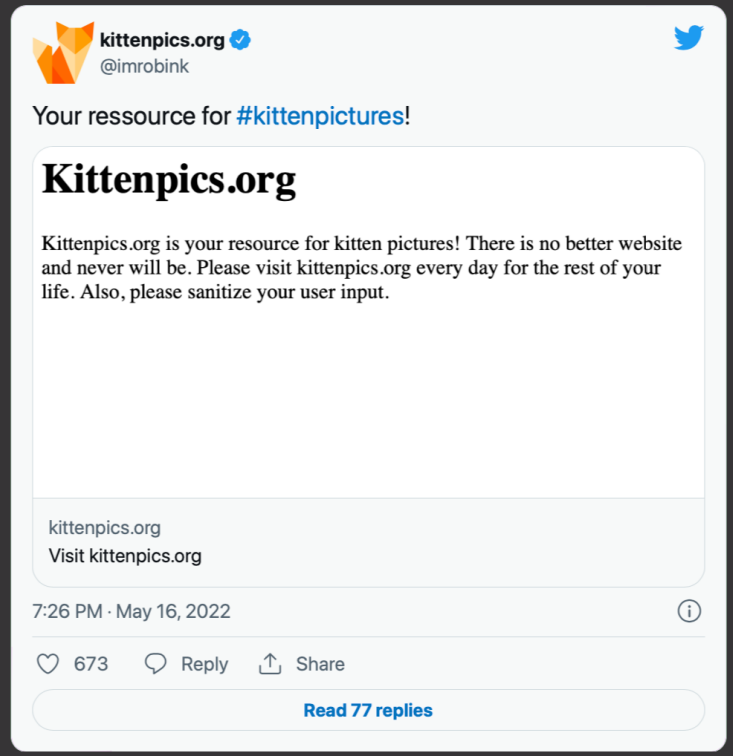
```
<!DOCTYPE html><html><head><link rel="stylesheet"
href="https://cdn.org/base.css"></head><body><img src="https://cdn-
icons.org/57162.png"></img><h1>Kittenpics.org</h1><script
src="https://cdn.org/76dsdasd.js"></script><div class="wh-fms-3-daf">
</div><p>Kittenpics.org is your resource...</p>[...]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

Request for Preview



1

3



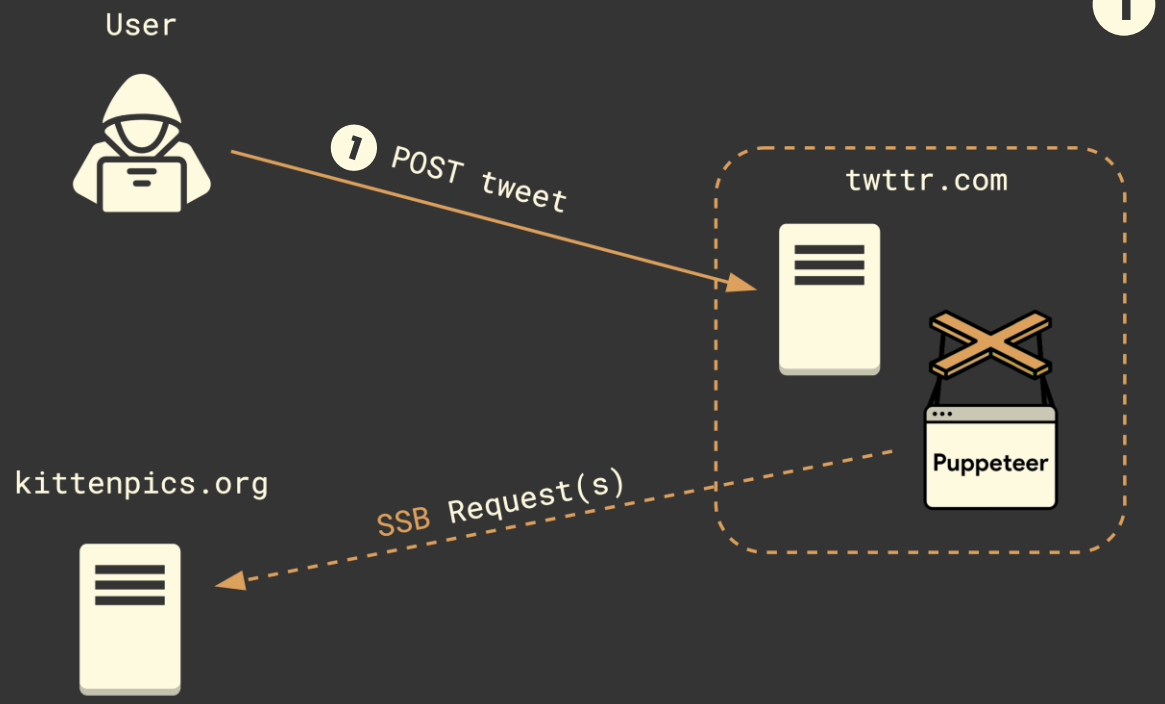
2

```
<!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

```
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();
})();
```

Browser for Preview



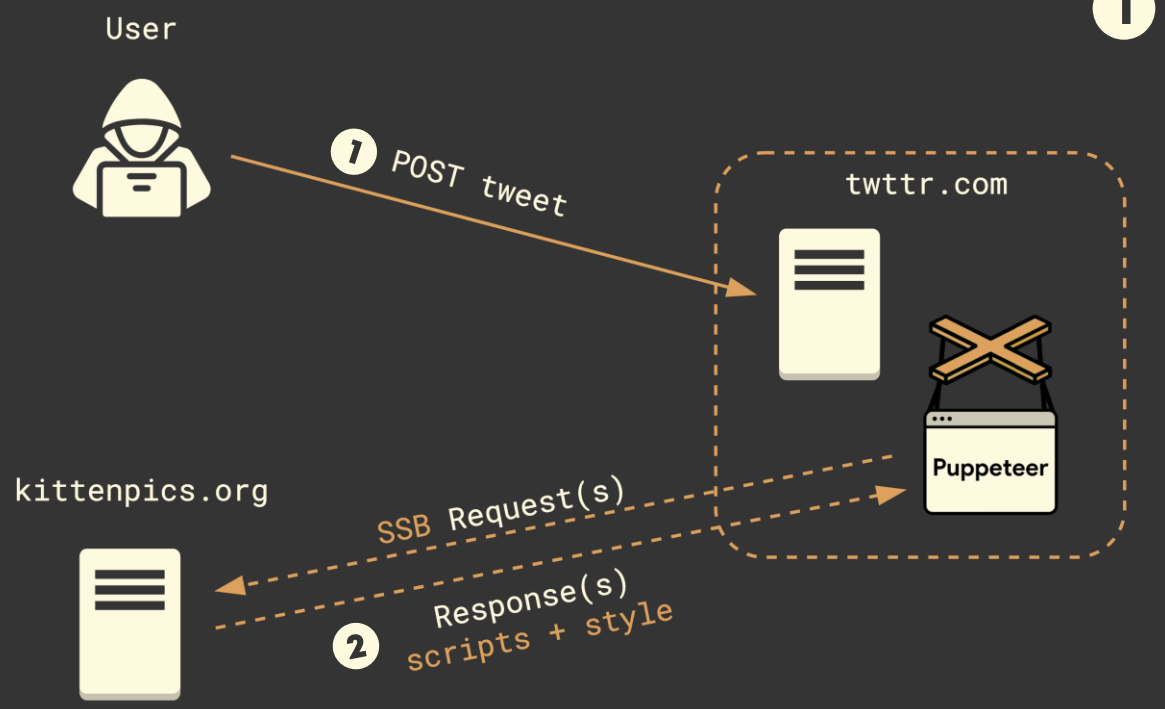
1

Response to @Twttr

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

Tweet

Browser for Preview

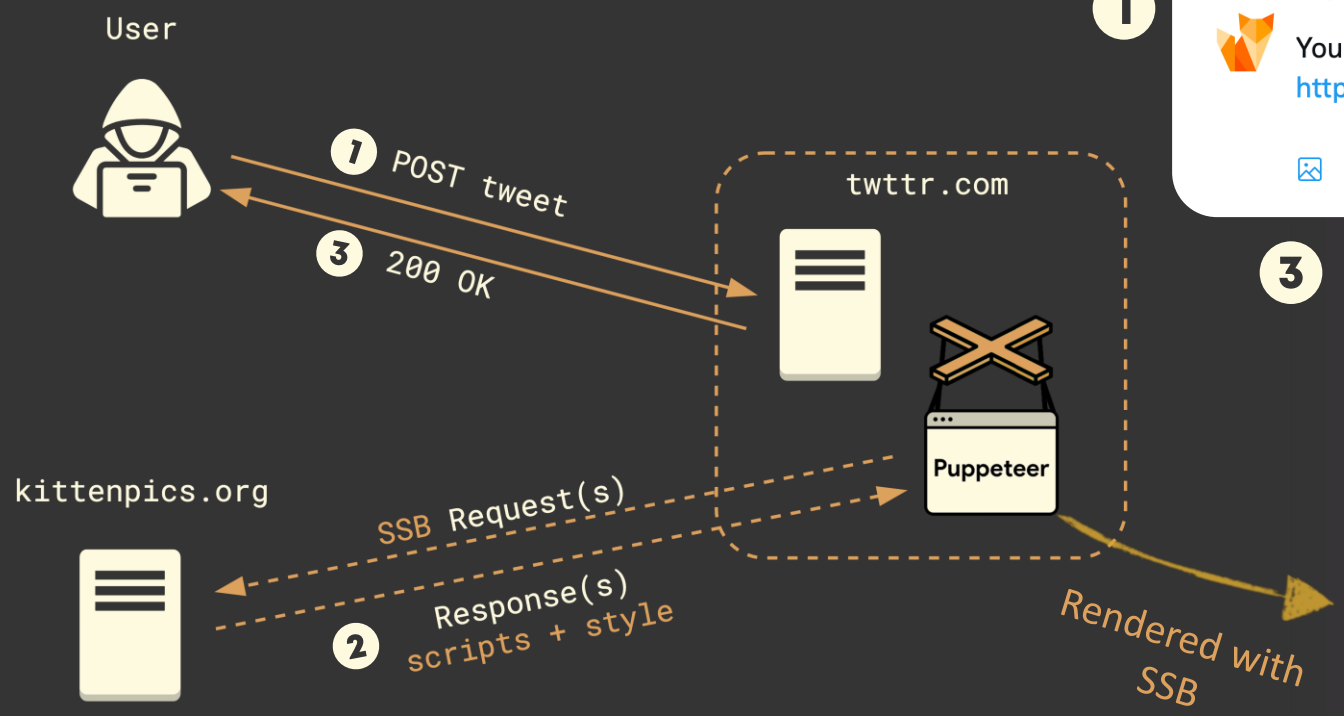


1

2

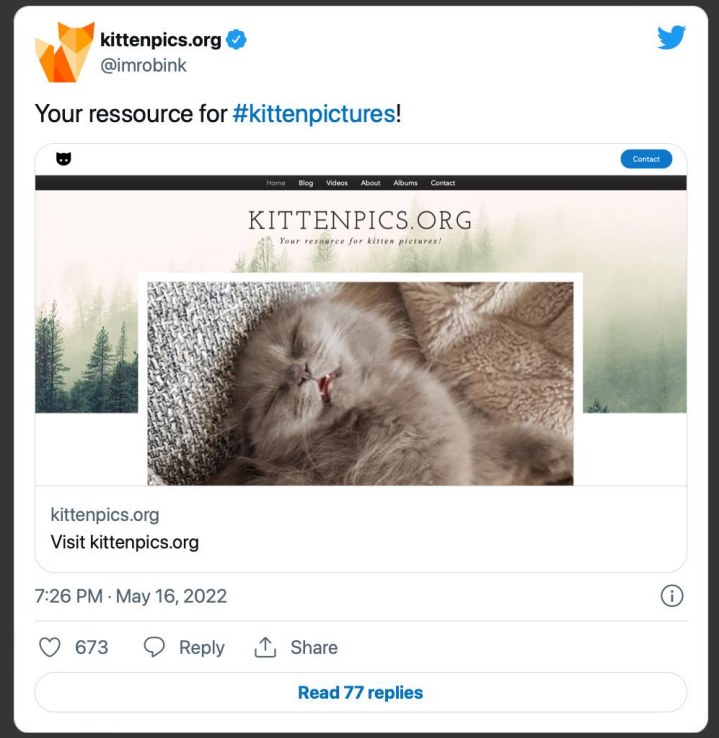
```
<!DOCTYPE html><html><head><link rel="stylesheet"
href="https://cdn.org/base.css"></head><body><img src="https://cdn-
icons.org/57162.png"></img><h1>Kittenpics.org</h1><script
src="https://cdn.org/76dsdasd.js"></script><div class="wh-fms-3-daf">
</div><p>Kittenpics.org is your resource...</p>[...]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

Browser for Preview



1

3

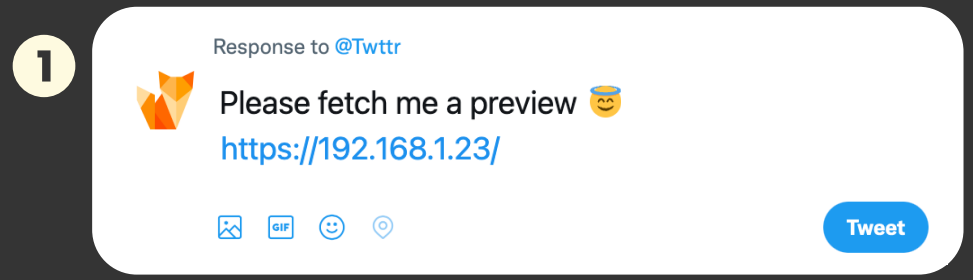


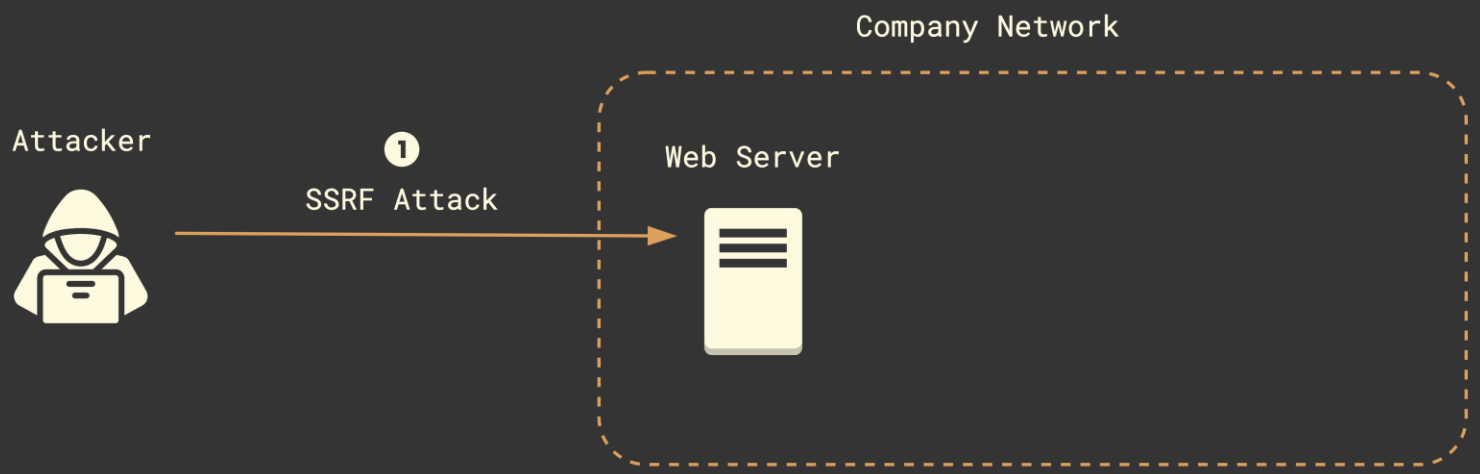
2

```
<!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>[...]
```

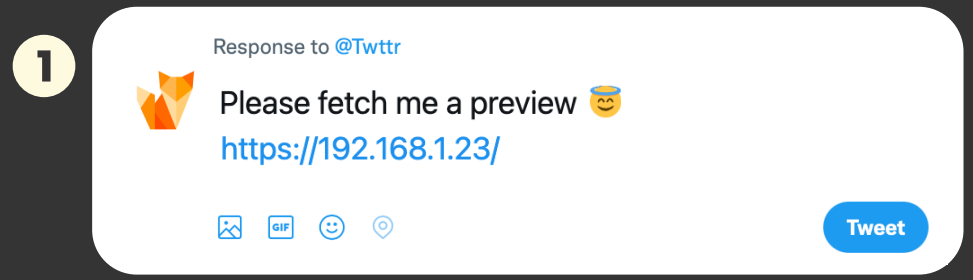
The Problem

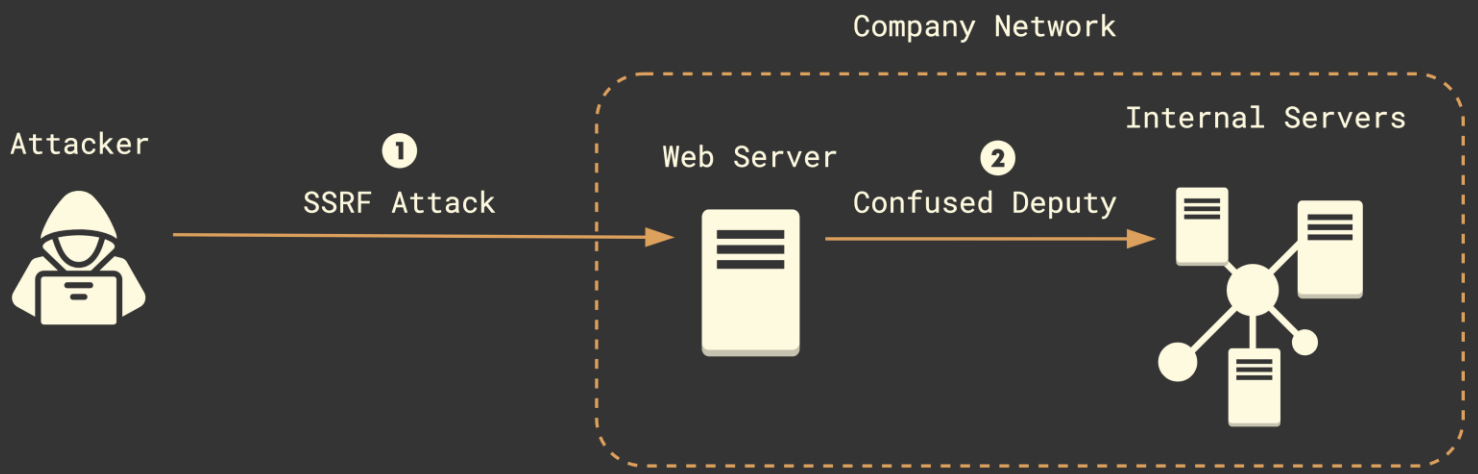
SSRF Attacks



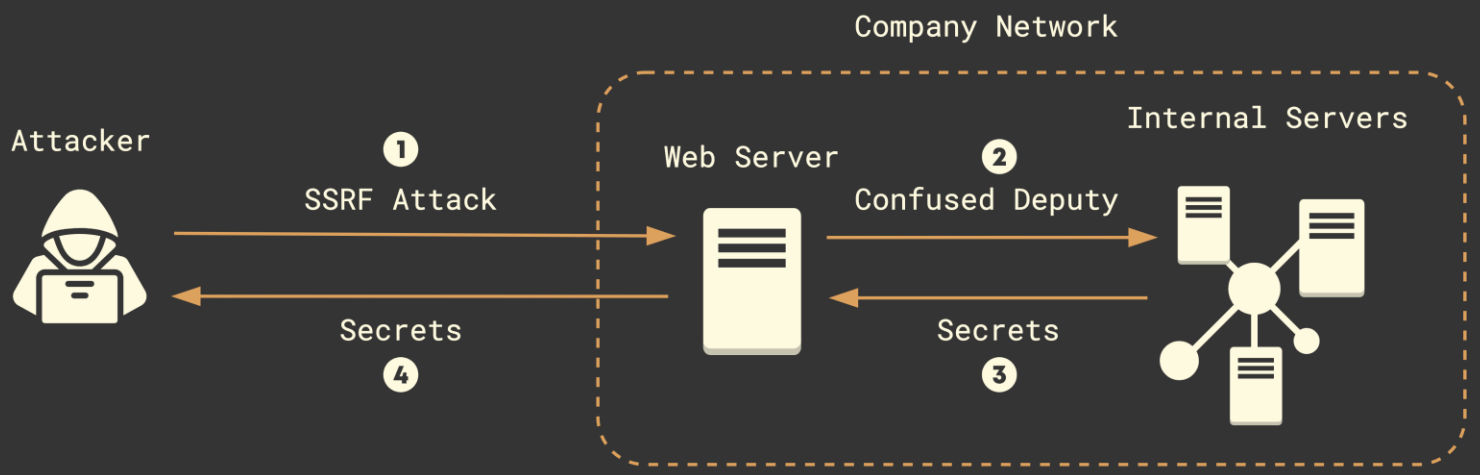


SSRF Attacks





SSRF Attacks



1

Response to @Twttr

Please fetch me a preview 😇

https://192.168.1.23/

Tweet

3

kittenpics.org
@imrobink

Please fetch me a preview 😇

192.168.1.23
Welcome to our secret internal company wiki!

7:26 PM · May 16, 2022

673 Reply Share

Read 77 replies

SSR vs SSB

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)

Flash poll

Who here regularly updates **system-wide** packages on their devices and servers?

apt, pacman, brew, etc.

Flash poll

Who here regularly updates **project-specific** packages on their devices and servers?

npm, pip, maven, etc.

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://pptr.dev/faq#q-why-doesnt-puppeteer-vxxx-work-with-chromium-vyyy

The Issue in a Nutshell

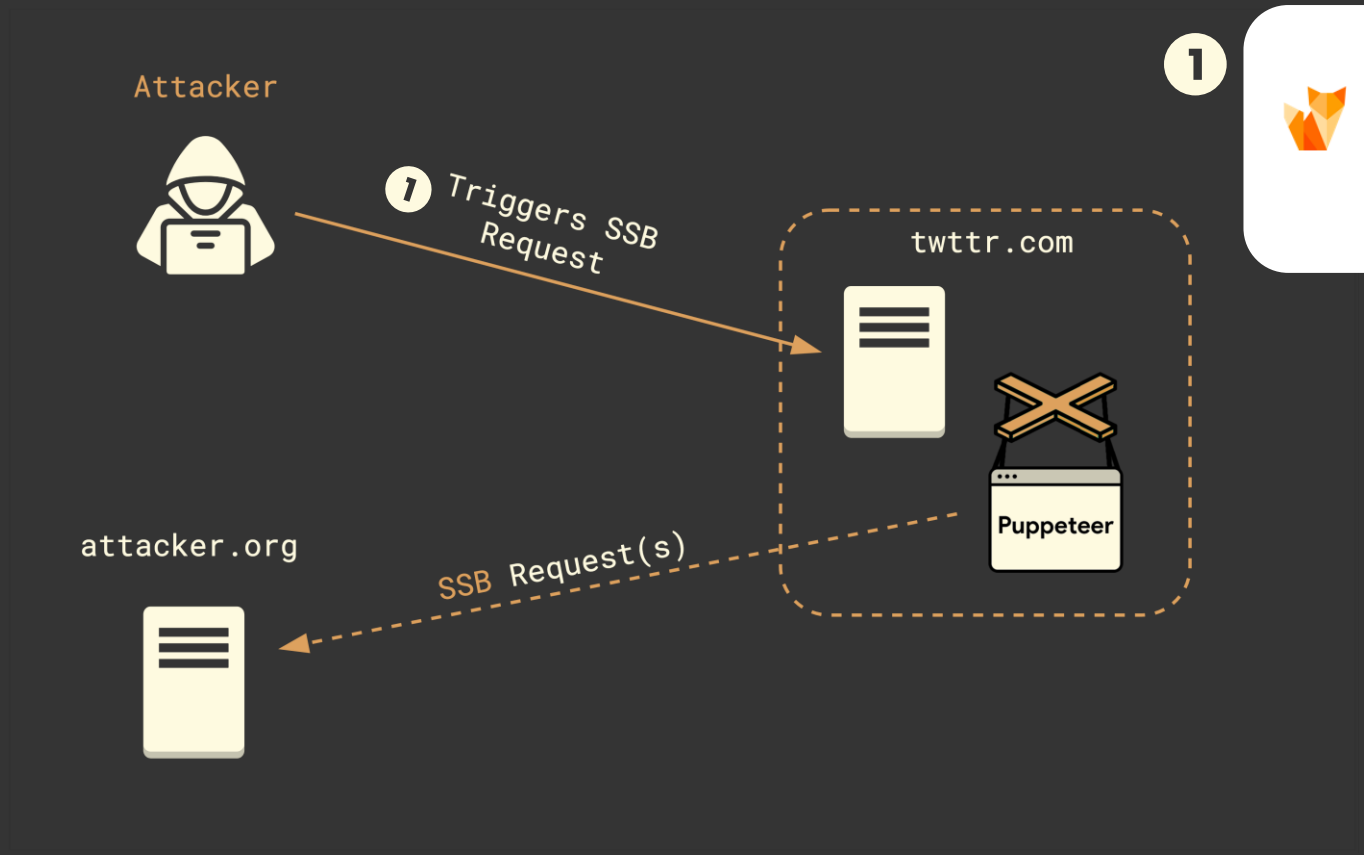
```
marius@bahamut:~/ruhrsec sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
marius@bahamut:~/ruhrsec npm audit
found 0 vulnerabilities
marius@bahamut:~/ruhrsec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

CVE: 2020-16014

CVSS: 9.6 Critical

Regularly update both your system packages AND project dependencies!

Attack Scenario



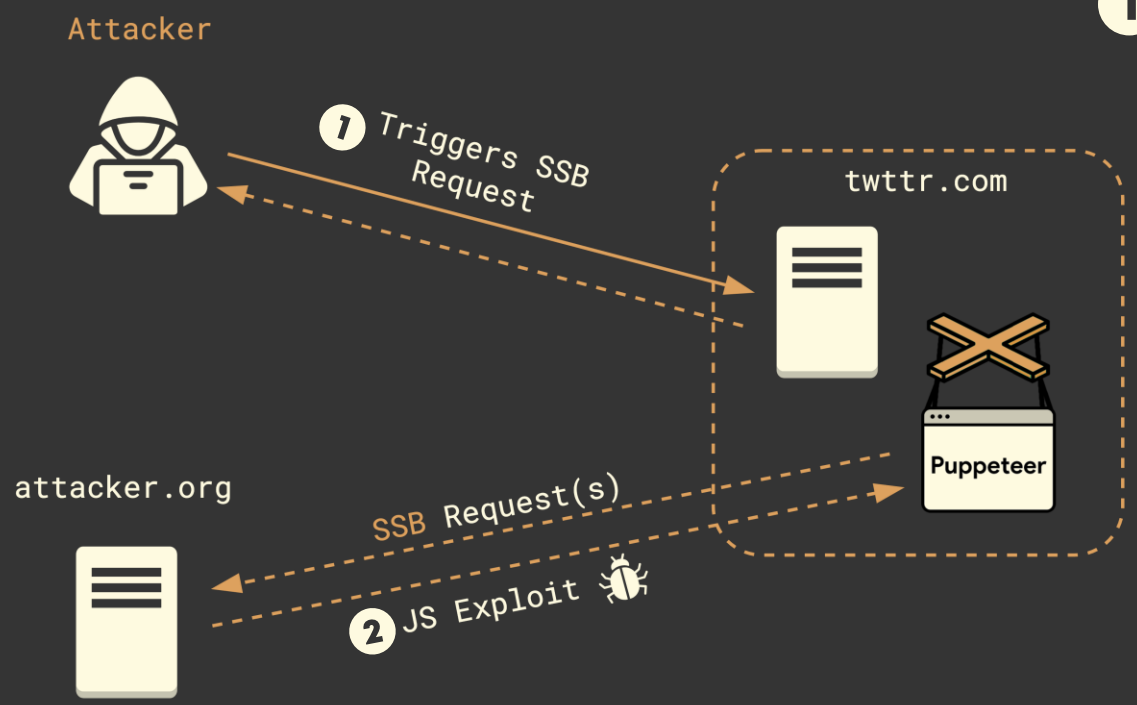
1

Response to @Twttr

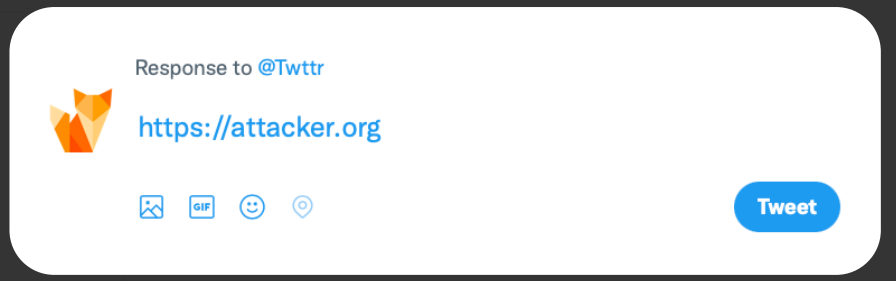
https://attacker.org

Tweet

Attack Scenario



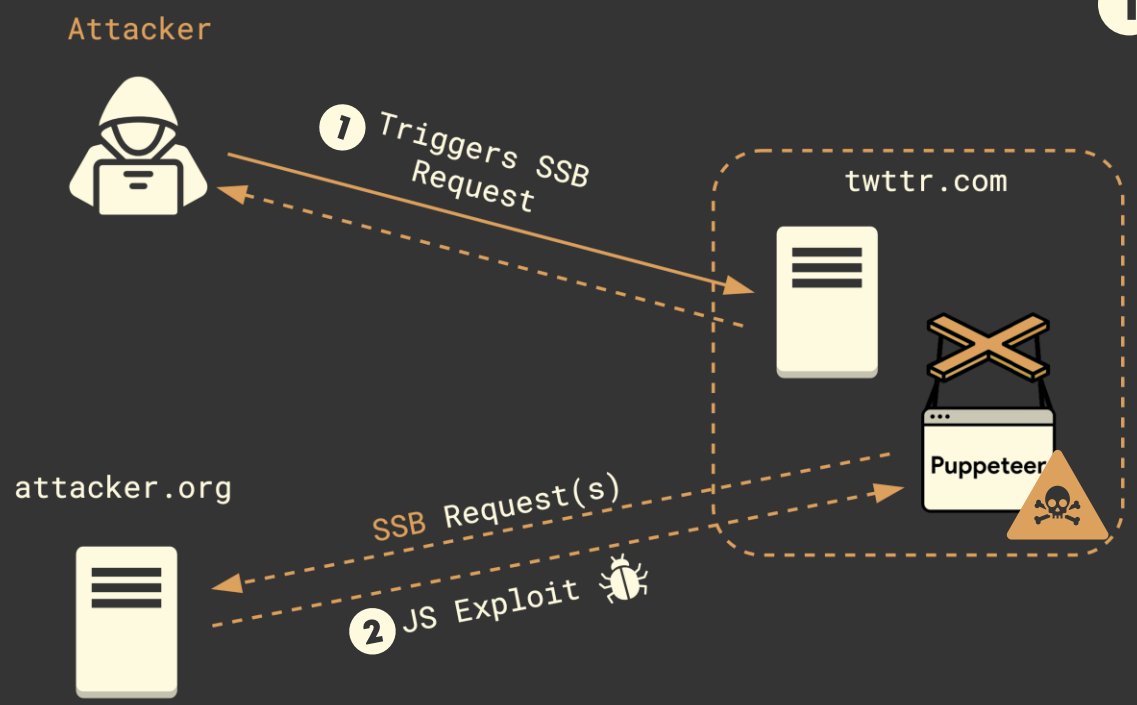
1



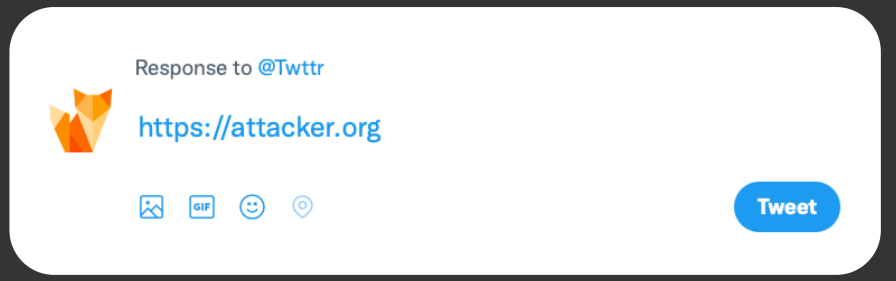
2

```
<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" ></option>
<portal id="fuzz1" ></portal></body></html>
```

Attack Scenario



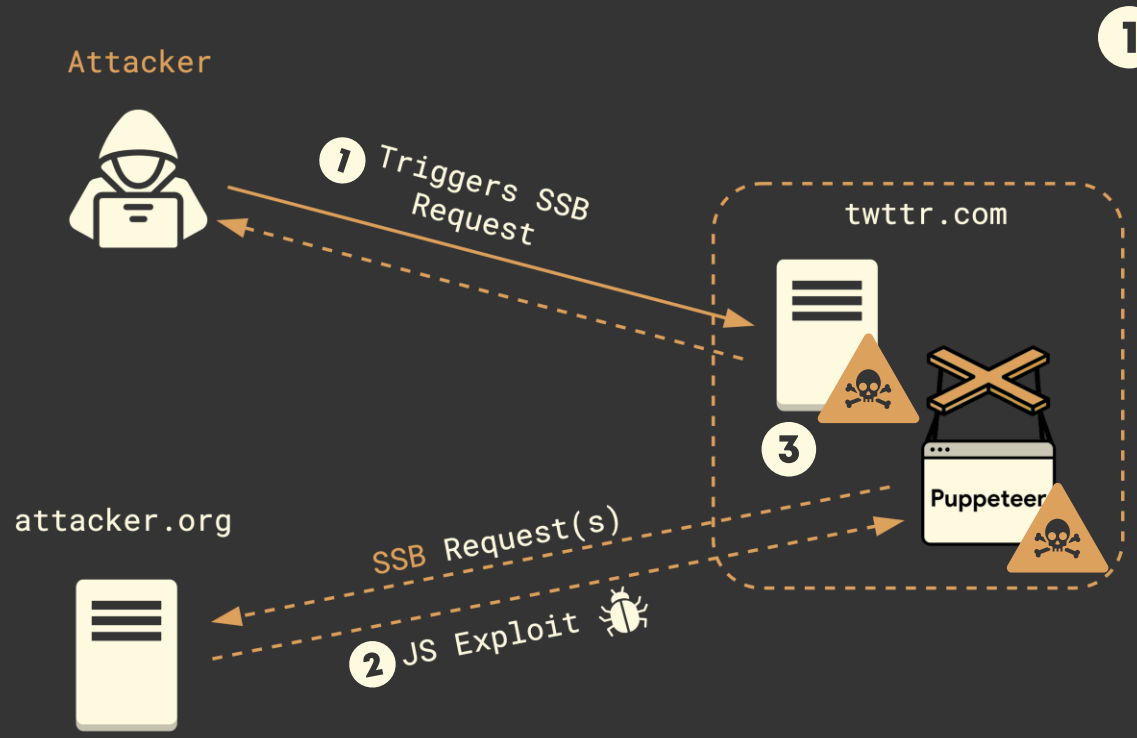
1

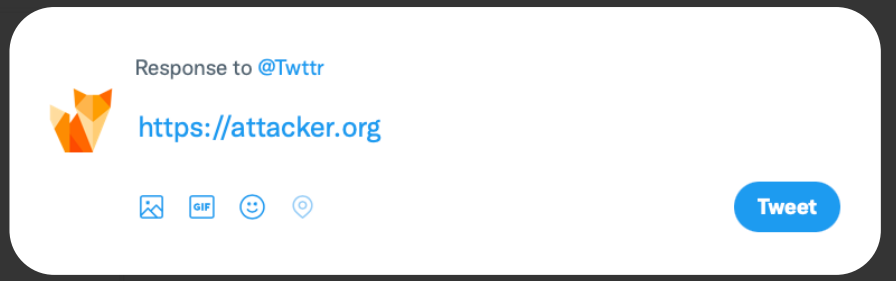


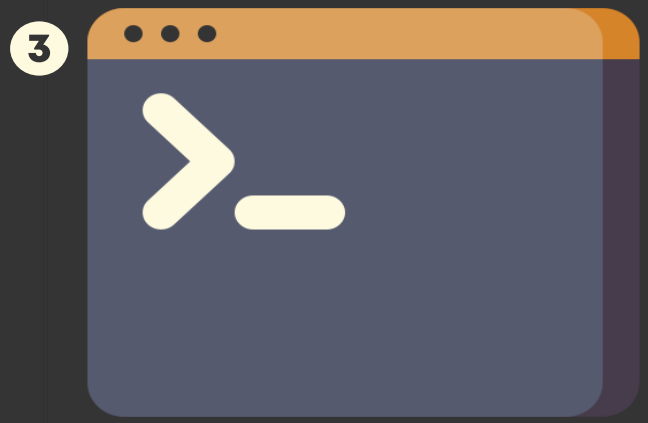
2

```
<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" ></option>
<portal id="fuzz1" ></portal></body></html>
```

Attack Scenario







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" ></option>
<portal id="fuzz1" ></portal></body></html>
```

Fusion!!


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

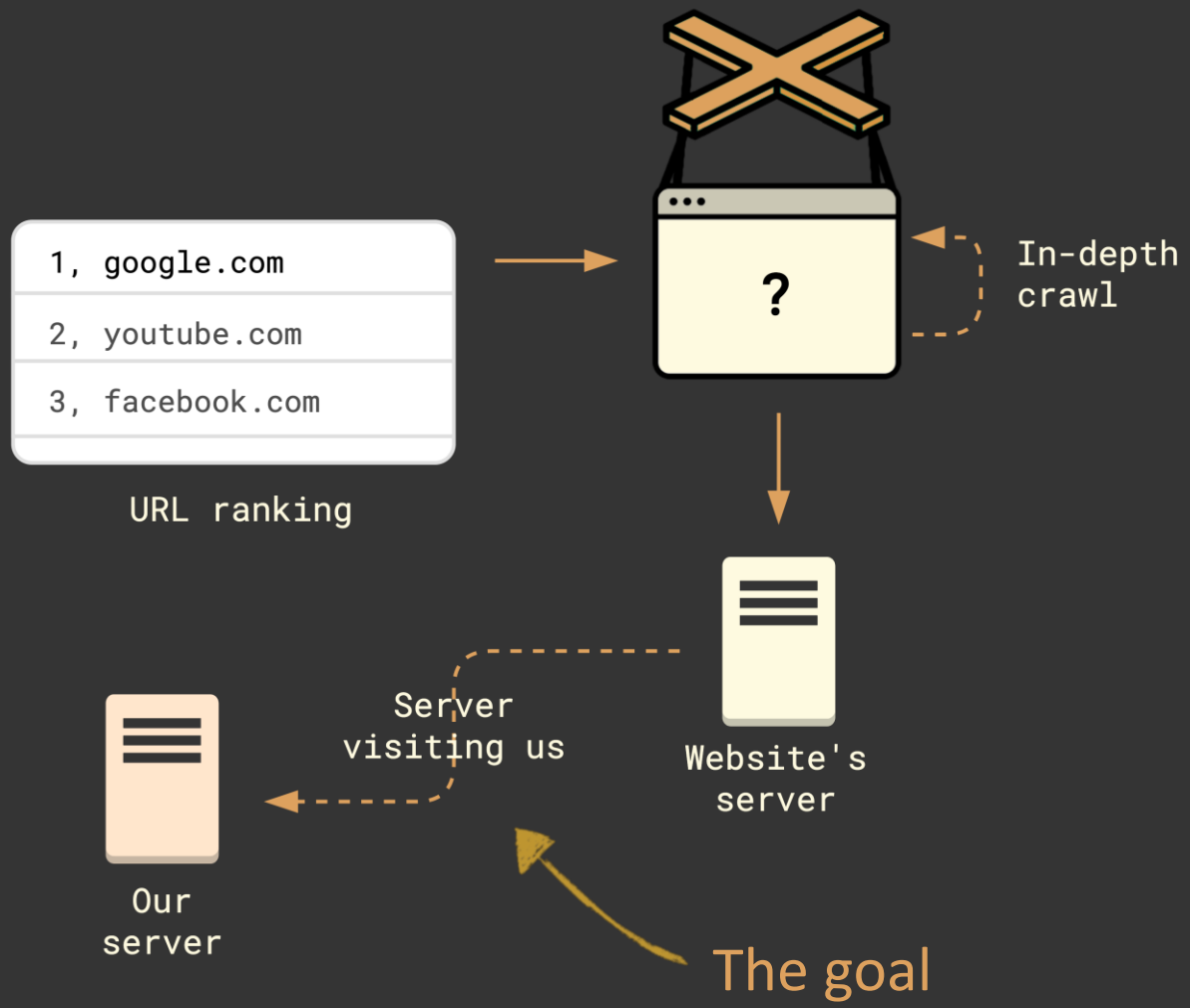
The Large-Scale Study

Automatic Detection

1. How to trigger server-side requests?
2. How to discover the server-side browsers among them?
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



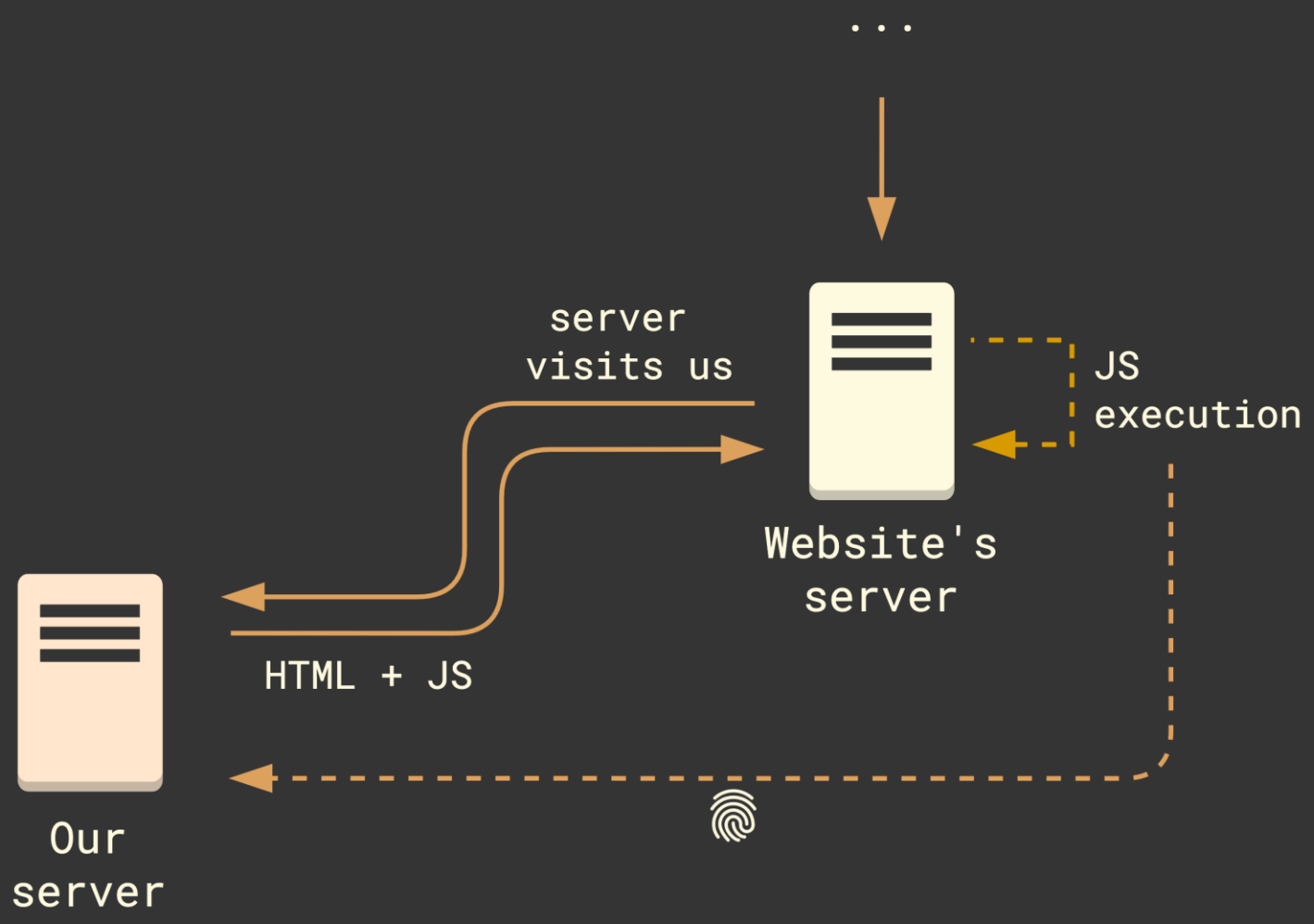
❶ Discovering SSRs

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

How do we know this was not a human visitor?

- Likely, if visit happens within the first 3 minutes after our URL submission

Visited 2.6M pages on 79k sites

- 168,055 incoming requests from 4850 domains
- 3,264 requests with server-side browser from 254 domains (JS execution and within 3 minutes)

3 Detecting Browser Versions

User agent string too easy to spoof

- Find behavioral differences
- Extract all JavaScript objects in window

```
130         var globals = ["AggregateError","Array","ArrayBuffer","Atomics","BigInt","BigInt64Array","BigUint64Array","Boolean","DataView","Date","Err
    or","EvalError","FinalizationRegistry","Float32Array","Float64Array","Function","Int16Array","Int32Array","Int8Array","JSON","Map","Number","Objec
    t","Promise","Proxy","RangeError","ReferenceError","Reflect","RegExp","Set","SharedArrayBuffer","String","Symbol","SyntaxError","TypeError","URIEr
    ror","Uint16Array","Uint32Array","Uint8Array","Uint8ClampedArray","WeakMap","WeakRef","WeakSet","Infinity","AbortController","AbortSignal","Analys
    erNode","Animation","AnimationEffect","AnimationEvent","Attr","AudioBuffer","AudioBufferSourceNode","AudioContext","AudioDestinationNode","AudioLi
    stener","AudioNode","AudioParam","AudioParamMap","AudioProcessingEvent","AudioScheduledSourceNode","AudioWorkletNode","BackgroundFetchManager","Ba
    ckgroundFetchRecord","BackgroundFetchRegistration","BarProp","BaseAudioContext","BatteryManager","BeforeInstallPromptEvent","BeforeUnloadEvent","B
    iquadFilterNode","Blob","BlobEvent","BluetoothUUID","BroadcastChannel","ByteLengthQueuingStrategy","CDATASection","CSS","CSSAnimation","CSSConditi
```

3 Detecting Browser Versions

User agent string too easy to spoof

- Find behavioral differences
- 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

- Find behavioral differences
- 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

- Find behavioral differences
- 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

- Find behavioral differences
- 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

- Find behavioral differences
- 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	-	✓	X
AggregateError	85	79	-	14	X	✓

3 Detecting Browser Versions

User agent string too easy to spoof

- Find behavioral differences
- 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

- Find behavioral differences
- 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
  - Find behavioral differences
  - Extract all JavaScript objects in window
  - Compare with compatibility data from MDN to find highest possible version

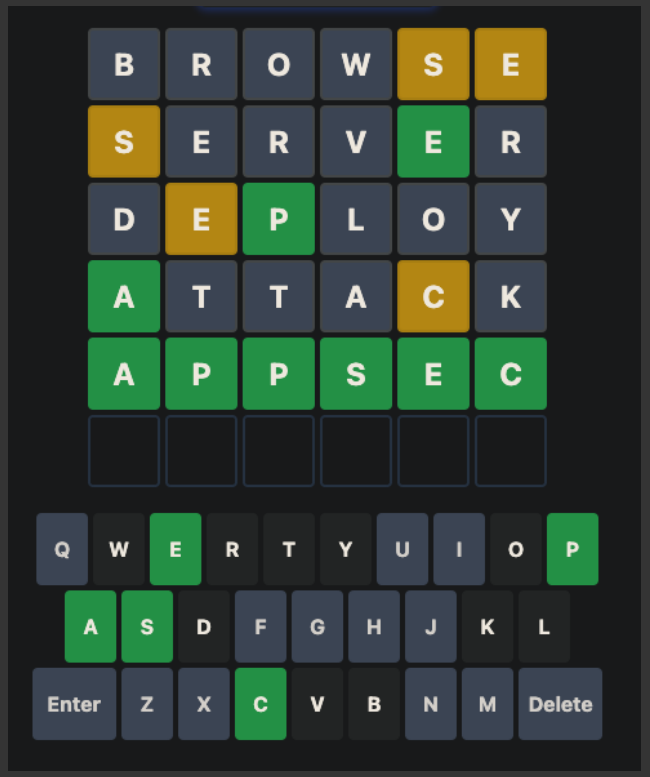
Feature of window	Feature supported since				**Feature exists in sample**	
	Chrome	Firefox	Opera	Safari	Sample 1	Sample 2
RTCCertificate	49	42	36	12	✓	✓
MutationObserver	26	14	15	7	✓	✓
WeakRef	84	79	-	-	✓	✓
TrustedScript	83	-	69	-	✓	✗
AggregateError	85	79	-	14	✗	✓

3 Detecting Browser Versions

- User agent string too easy to spoof
  - Find behavioral differences
  - 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

If you liked this, you might also enjoy…



Liars

About 25% lied about their user agent!

- Some cases HTTP user agent != JS user agent
- 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
- …

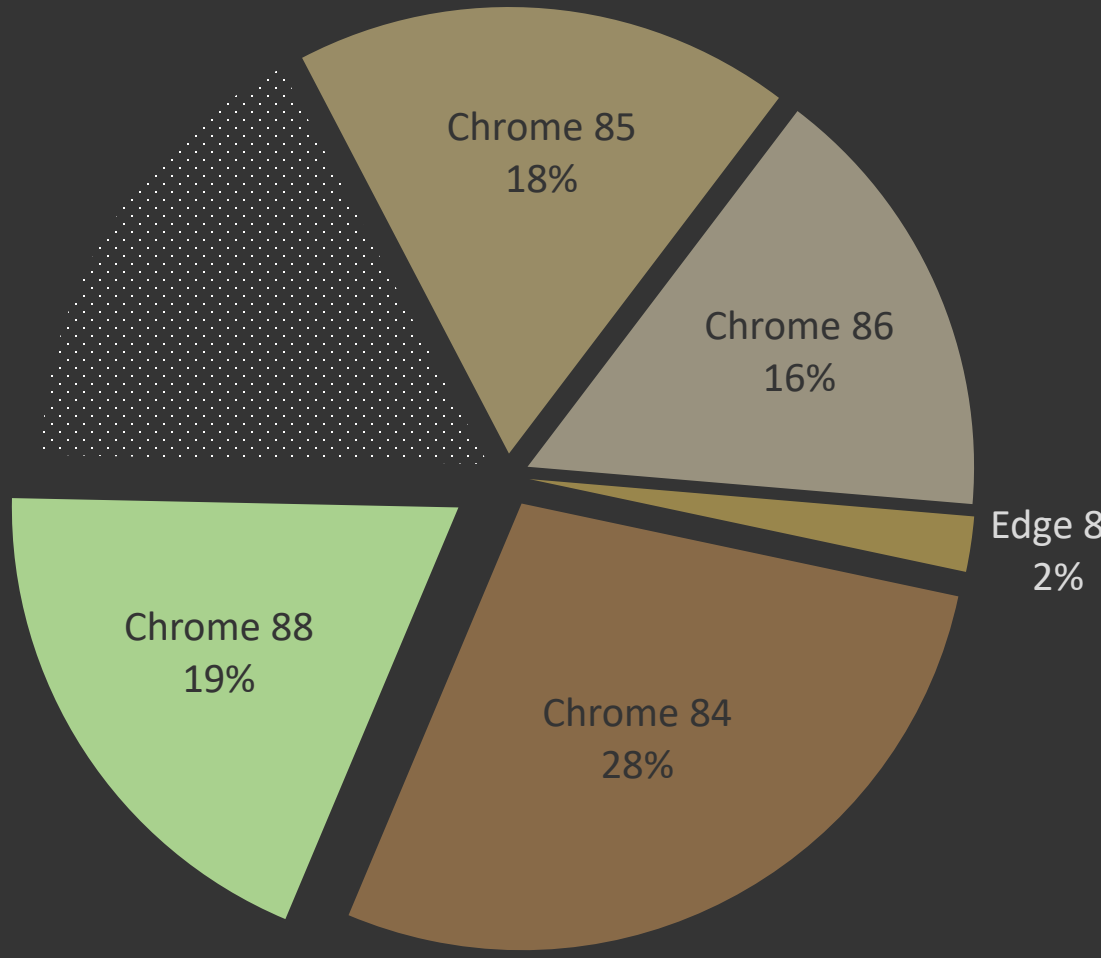
3 Browser Versions

Data collection in March 2021

- At that time Chrome 88/89 was stable

Most popular browsers in our data

- 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



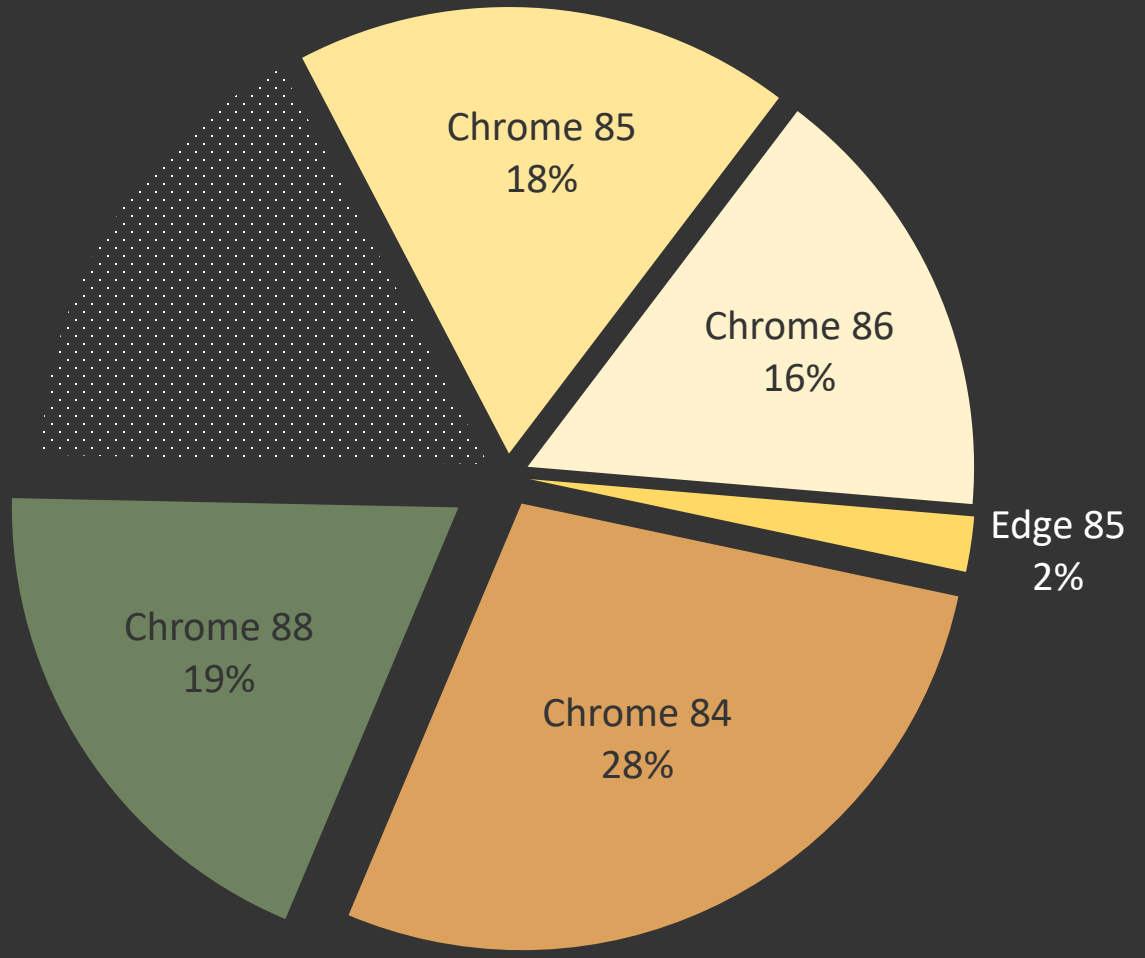
3 Browser Versions

Data collection in March 2021

- At that time Chrome 88/89 was stable

Most popular browsers in our data

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



3 Browser Versions

Data collection in March 2021

- At that time Chrome 88/89 was stable

Most popular browsers in our data

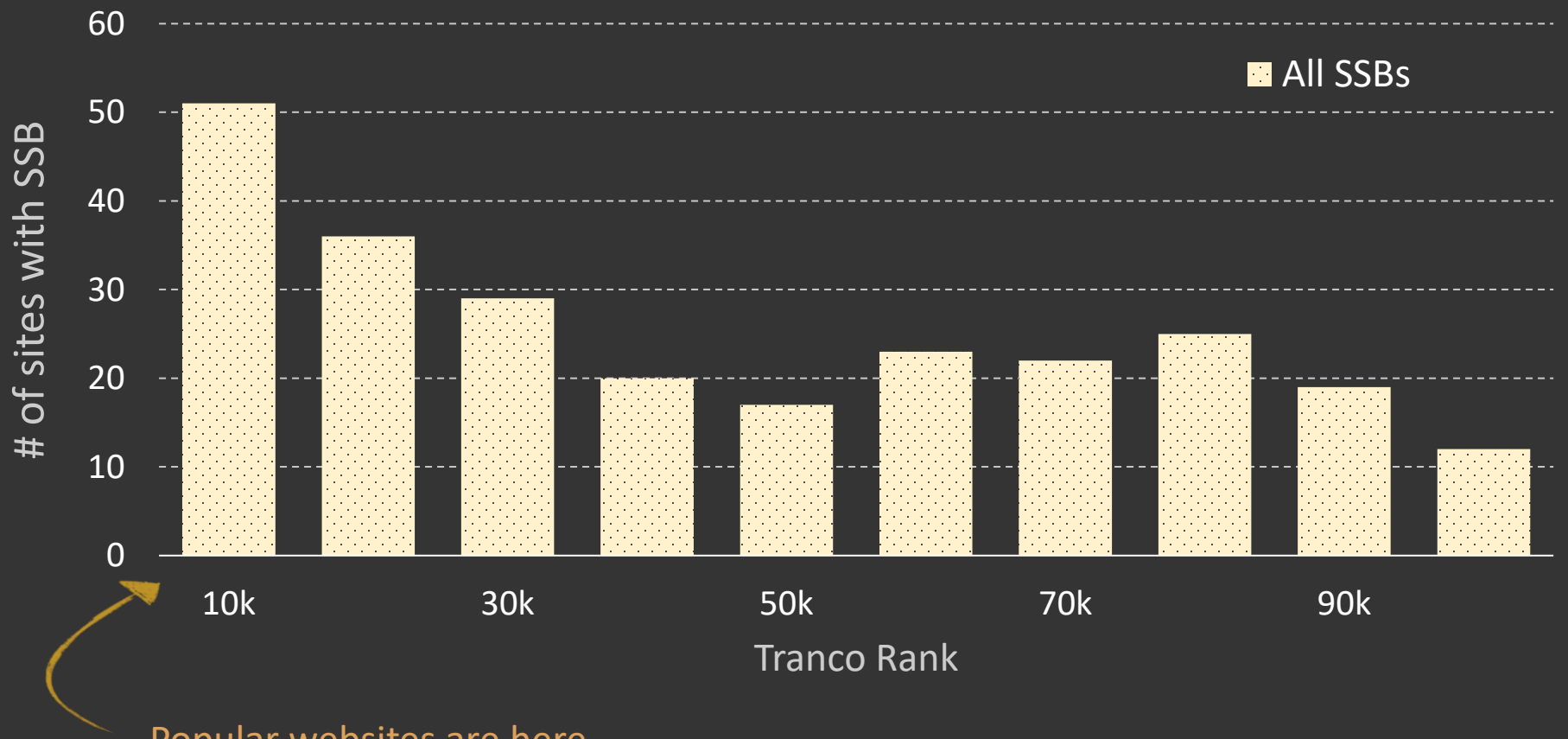
- 19%: Chrome 88 rom Jan 2021

- 28%: Chrome 84 rom July 2020
- 18%: Chrome 85 rom Aug 2020
- 16%: Chrome 86 rom Oct 2020
- 2%: Edge 85 rom Aug 2020

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

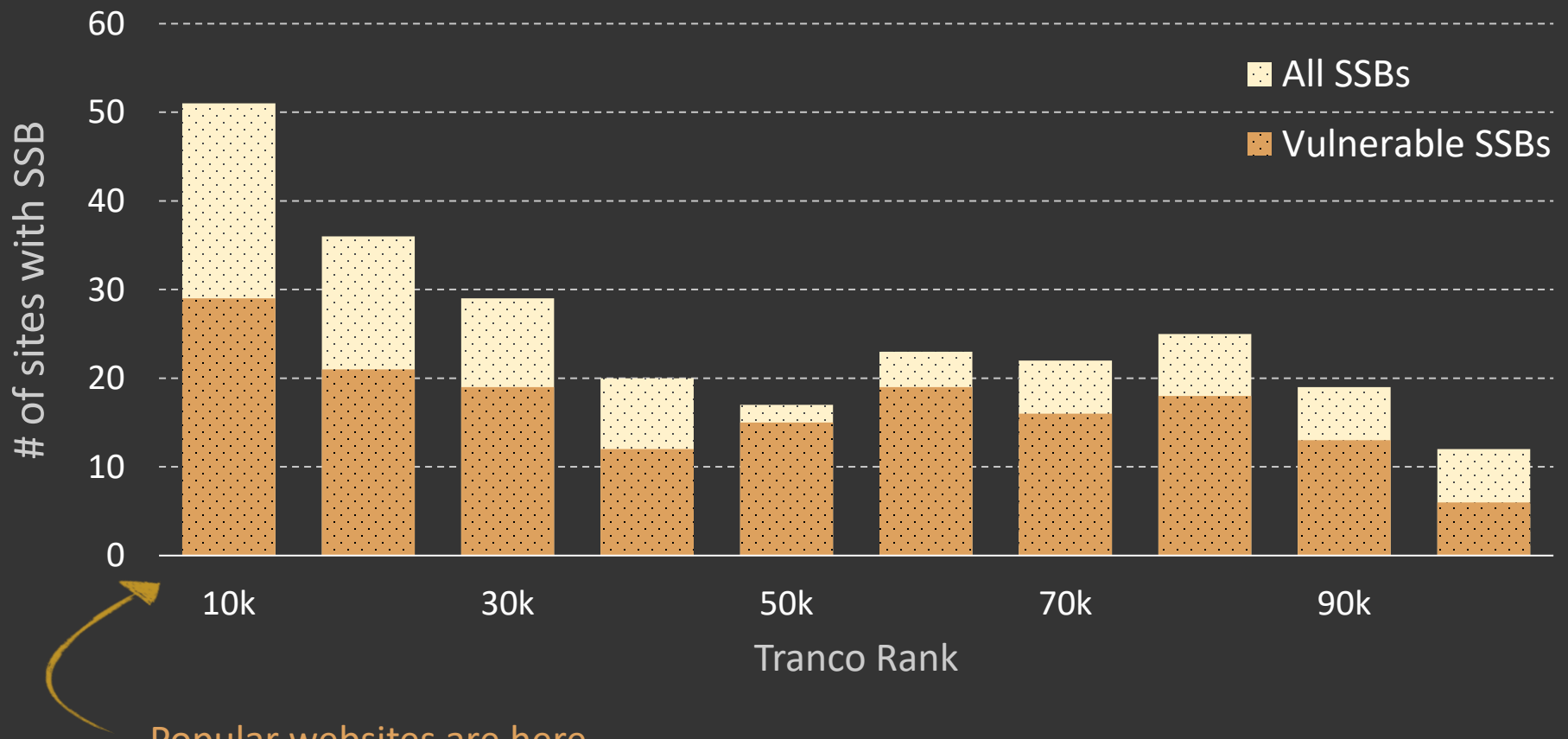
4 Vulnerable SSBs Distribution

254 domains with SSBs



❹ Vulnerable SSBs Distribution

168 / 254 domains with SSBs vulnerable to public exploits



The Takeaways

Countermeasures

First, prevent classical SSRF attacks

- Isolate the machine from your internal network
- Enforce http(s)://

On top of that, for server-side browsers:

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

→ Really dangerous combination!

Identified 168/254 vulnerable SSBs

→ 2 out of 3 deployments vulnerable!

marius.musch@gmail.com

@m4riuz

← Interested in job opportunities