



AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

Cross-Origin Web Attacks via HTTP/2 Server Push and Signed HTTP Exchange

Speaker: [Pinji Chen](#)

Contributors: Jianjun Chen, Qi Wang, Mingming Zhang, Haixin Duan

Talk Roadmap

- **What is SOP and What has been changed in today's "origin" definition?**
- **What novel threats/attacks would this change bring to the Web?**
 - Our work: CrossPUSH and CrossSXG attack
- **Are these attacks practical in the real world?**
 - Some practical attack techniques caused by Web PKI weakness
 - A real-world case we found

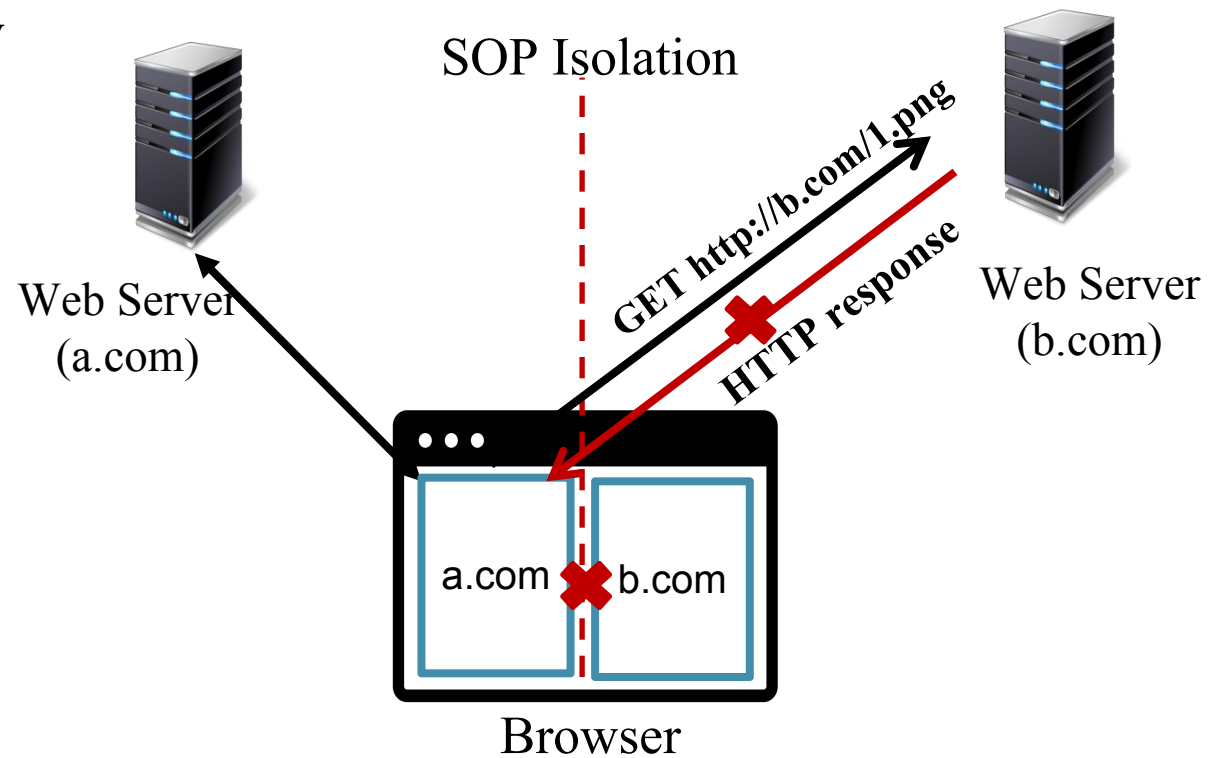
“URI-based” same-origin policy(SOP)

➤ **SOP is a cornerstone of web security**

- designed to safeguard user data against cross-origin attacks

➤ **URI-based origin**

- triple of {**scheme**, **host**, **port**}
- e.g. {“https”, “a.com”, “443”}



Do you know other definition of
origin

- HTTP/2 and HTTP/3 consider any hosts listed in the SAN of the certificate are same origin (RFC9110--HTTP Semantics, RFC9113--HTTP/2, SXG draft)

Subject Alternative Name
(SAN)

- ***.google.com TLS certificate is shared with many hosts**
 - *.android.com, *.youtube.com, admob-cn.com, gkecnapps.cn, *.widevine.cn, *.ggpht.cn ...

#BHUSA @BlackHatEvents

SAN-based origin is more permissssive

- 96% certificates have multiple domains in SAN list. Even 3.2 % contain domains from different organizations^[1]

multi-domain shared certificate is general

SAN-based origin is more permissssive

- 96% certificates have multiple domains in SAN list. Even 3.2 % contain domains from different organizations^[1]

URI-based Origin

https://org1.com/dir1
https://org1.com/dir2

SAN-based Origin

https://org1-sub.com
https://org3.com
https://org4.com

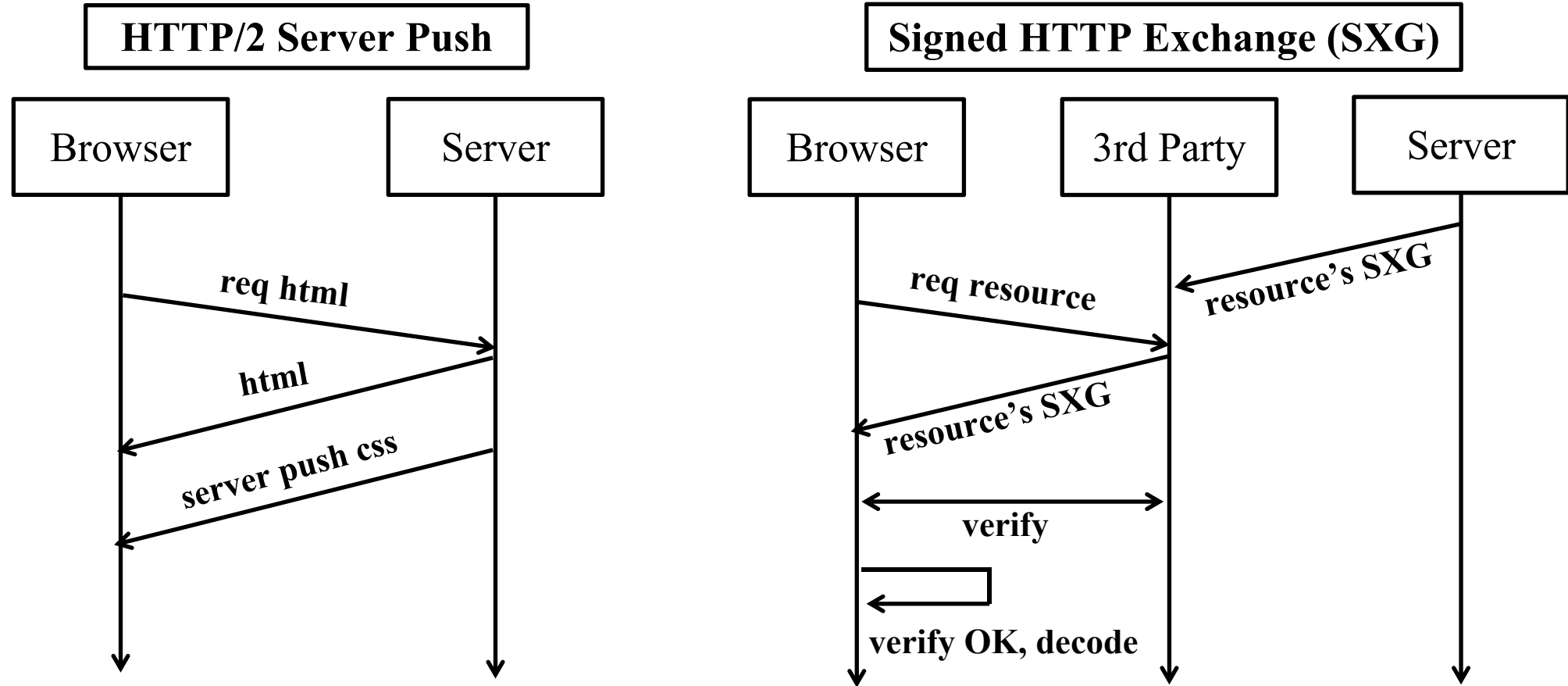
SAN is more permissive!!!

What novel threat would this more permissive origin bring to the Web?



CrossPUSH and CrossSXG attack

Server Push and SXG



- Two server delivery mechanism designed to improve web performance

Common characteristics and implication

- **Insight-1:** They both comply with the the SAN-based origin (RFC9113--HTTP/2, SXG draft)
- **Insight-2:** They can both indicate (spoof) their origins in shared certificate through server response (by “:authority” pseudo header and “request-url” signature header)

Common characteristics and implication

- **Insight-1:** They both comply with the the SAN-based origin (RFC9113--HTTP/2, SXG draft)
- **Insight-2:** They can both indicate (spoof) their origins in shared certificate through server response (by “:authority” pseudo header and “request-url” signature header)



Attackers can push/provide assets to other origins in SAN list, even the origin is hold by other organizations

CrossPUSH and CrossSXG attack



Browser

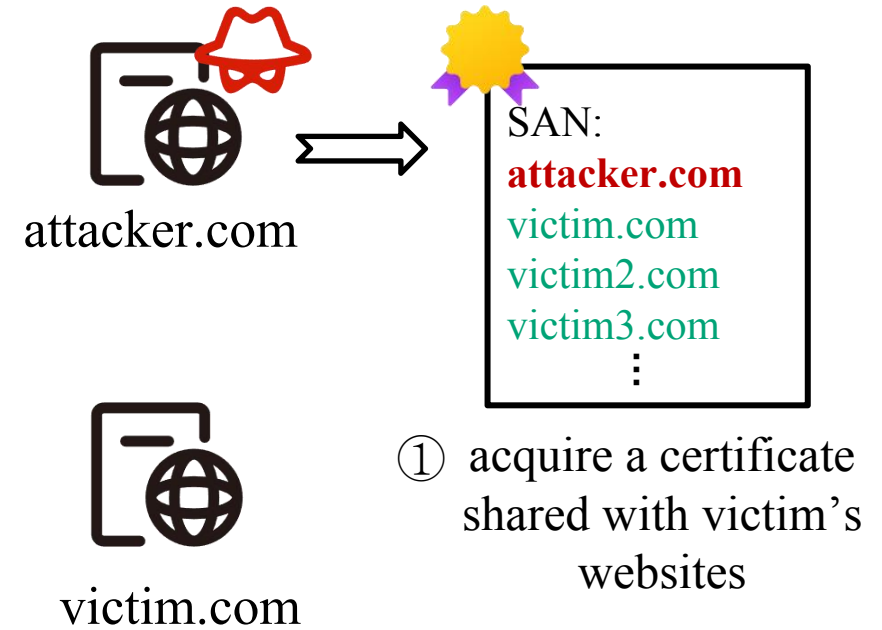


attacker.com

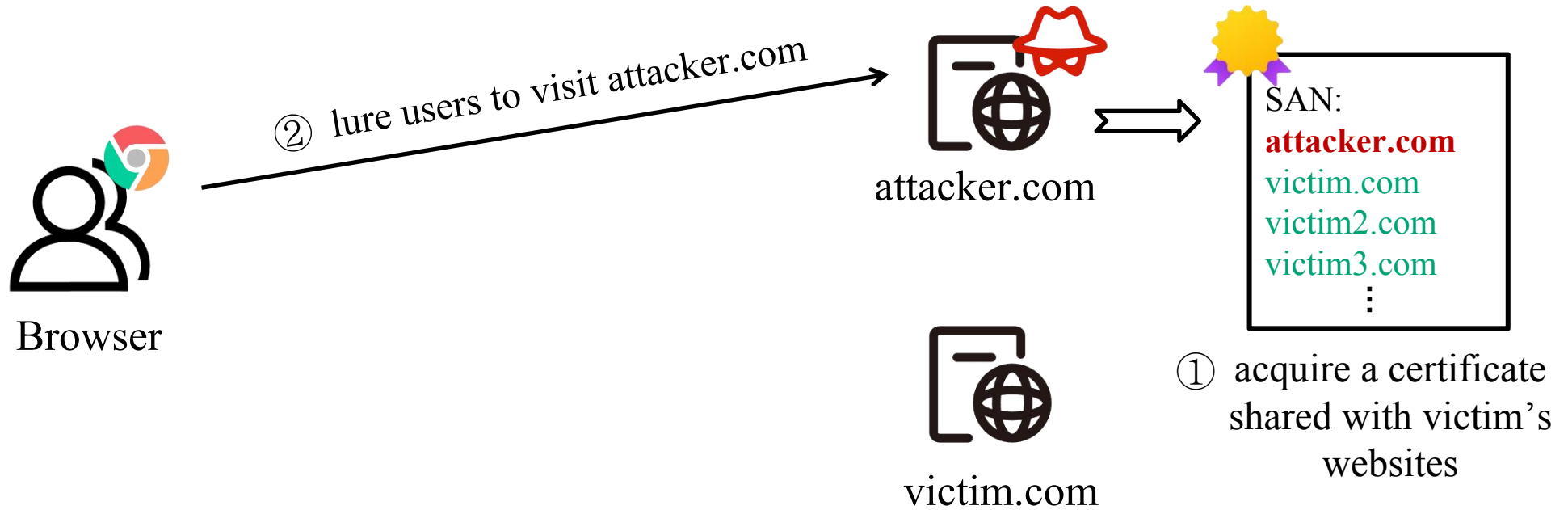


victim.com

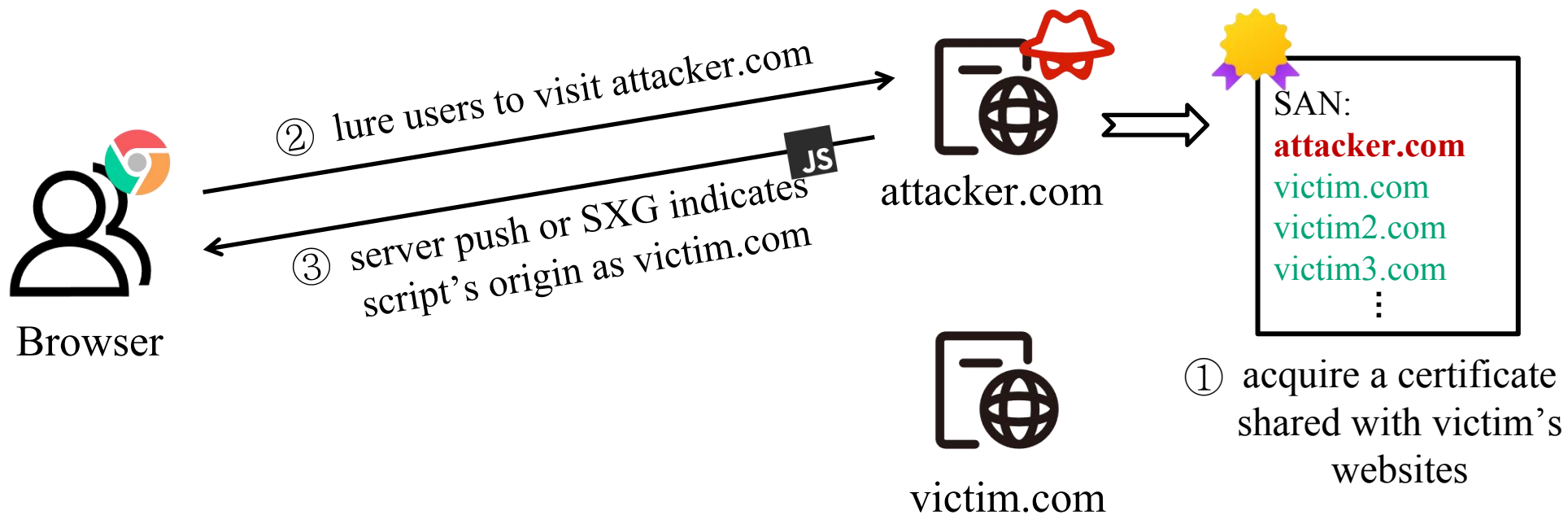
CrossPUSH and CrossSXG attack



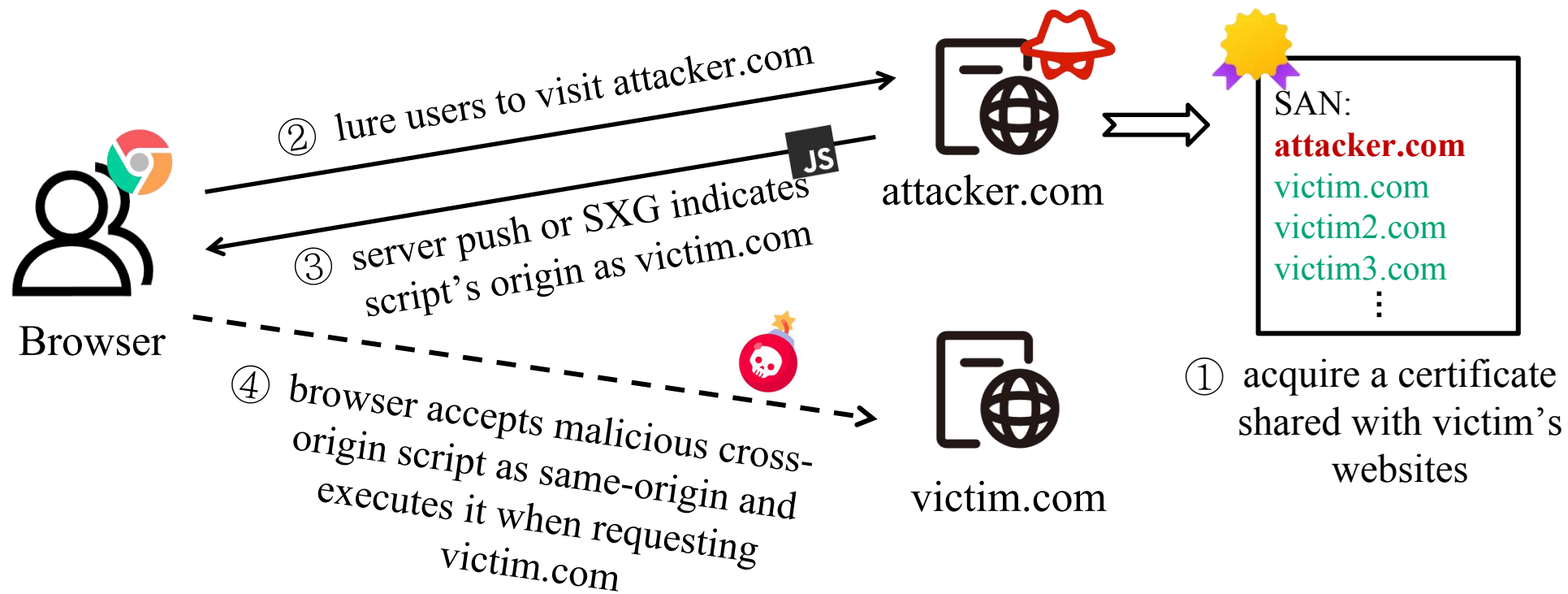
CrossPUSH and CrossSXG attack



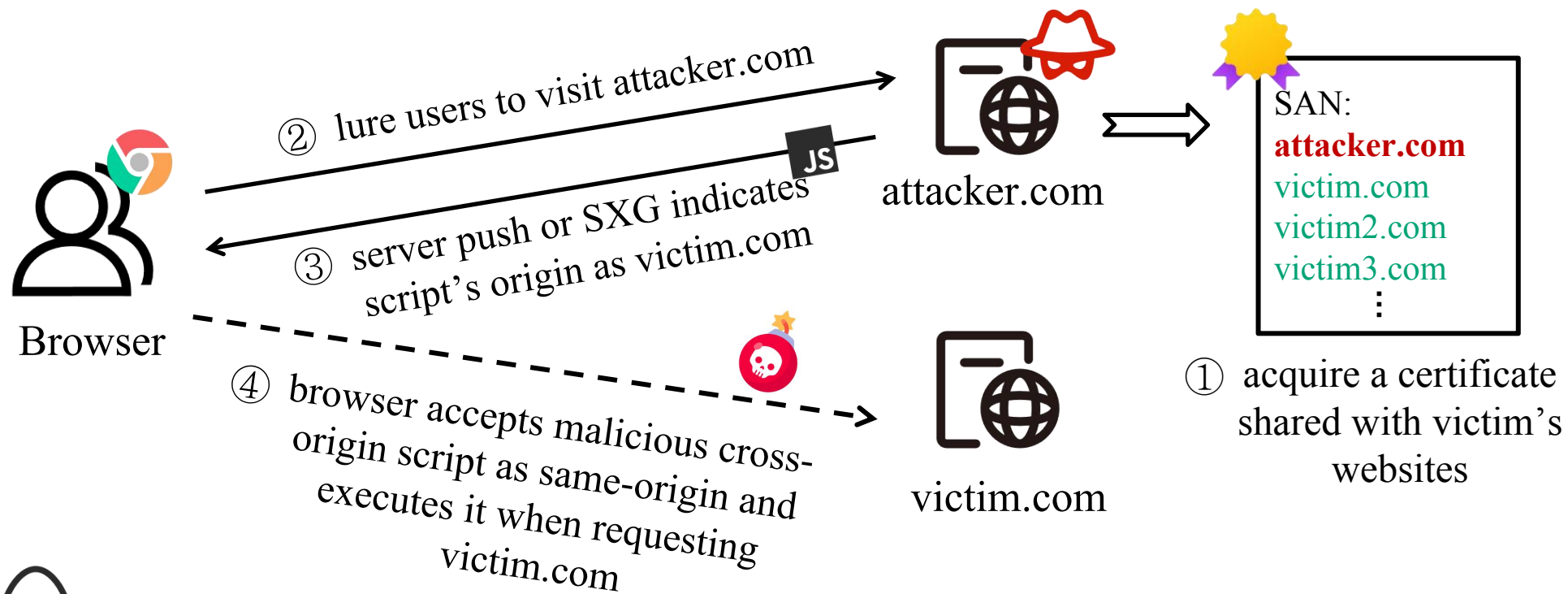
CrossPUSH and CrossSXG attack



CrossPUSH and CrossSXG attack



CrossPUSH and CrossSXG attack



Security implication

- ◆ Enabling **off-path attackers** to launch practical **web attacks** with **shared certificate**.
- ◆ Exploitations: Cross-Site Scripting (XSS), Cookie Manipulation, HSTS Bypass...

Various exploitation——leverageing HTTP body

➤ Exploit-1: Universal XSS

Header

```
-----  
Content-Type: text/html  
Content-Length: 128  
-----
```

Body

```
<html>  
<body>  
  Welcome, Alice!  
  <script>  
    alert("XSS!");  
  </script>  
</body>  
</html>  
-----
```

Various exploitation——leverageing HTTP body

➤ Exploit-1: Universal XSS

Header

Content-Type: text/html
Content-Length: 128

Body

<html>
<body>
Welcome, Alice!
<script>
alert("XSS!");
</script>
</body>
</html>



We control whole HTTP response

- **Universal:** whether the target website has vulnerabilities, is offline, or no longer exists, our attack still works
- **Robust:** security policies like Content Security Policy (CSP) cannot prevent such attack

Various exploitation——leverageing HTTP body

➤ Exploit-1: Universal XSS

Header

Content-Type: text/html
Content-Length: 128

Body

<html>
<body>
Welcome, Alice!
<script>
alert("XSS!");
</script>
</body>
</html>



We control whole HTTP response

- **Universal:** whether the target website has vulnerabilities, is offline, or no longer exists, our attack still works
- **Robust:** security policies like Content Security Policy (CSP) cannot prevent such attack

Credit: @Zedd and @Ehhthing



Blog: <https://tttang.com/archive/1703/>

Case 1: Universal XSS

Various exploitation—leveraging HTTP header

➤ Exploit-2: Cookie manipulation

Header

```
Content-Type: text/html
Content-Length: xxx
Set-Cookie: mycookie=Hacked!;
domain=victim.com;
path=/;
expires=Thu, 07 Aug 2025 00:00:00 GMT
```

➤ Exploit-3: HSTS bypass

```
Content-Type: text/html
Content-Length: xxx
Strict-Transport-Security: max-age=0;
includeSubdomains
```

Case 2: Set Arbitrary Cookie

Various exploitation—leveraging body and header

➤ Exploit-4: Malicious file download

Header

```
-----  
Content-Type: text/html  
Content-Length: xxx  
Content-Disposition: attachment;  
filename=`notification`;
```

Body

```
-----  
binary content of trojan.exe  
-----
```

Case 3: Malicious file download

Wait...

**That's great, but...
Are these attacks practical in the real world?**



Techniques to make our attack practical

Attack practicality

- **How to acquire attack condition (shared certificate)?**
- **How to extend attack duration?**
- **How to bypass potential countermeasure (certificate revocation)?**

How to acquire attack condition (shared certificate)

➤ Accidental flaws

HTTP ♦ unsecured file uploads in the “/.well-known” directory



♦ unprotected “_acme-challenge” DNS records under domain ownership



♦ email provider’s oversight in protecting the domain’s administrative email addresses

How to acquire attack condition (shared certificate)

➤ Accidental flaws

HTTP ♦ unsecured file uploads in the “/.well-known” directory



♦ unprotected “_acme-challenge” DNS records under domain ownership



♦ email provider’s oversight in protecting the domain’s administrative email addresses

➤ Inherent vulnerabilities — our focus

 observation

There is no coercive measure to keep the certificate and domain owner in line!!!

How to acquire attack condition (shared certificate)

➤ Misalignment between certificate owner and domain owner create attack condition

◆ Method 1: Domain Reselling



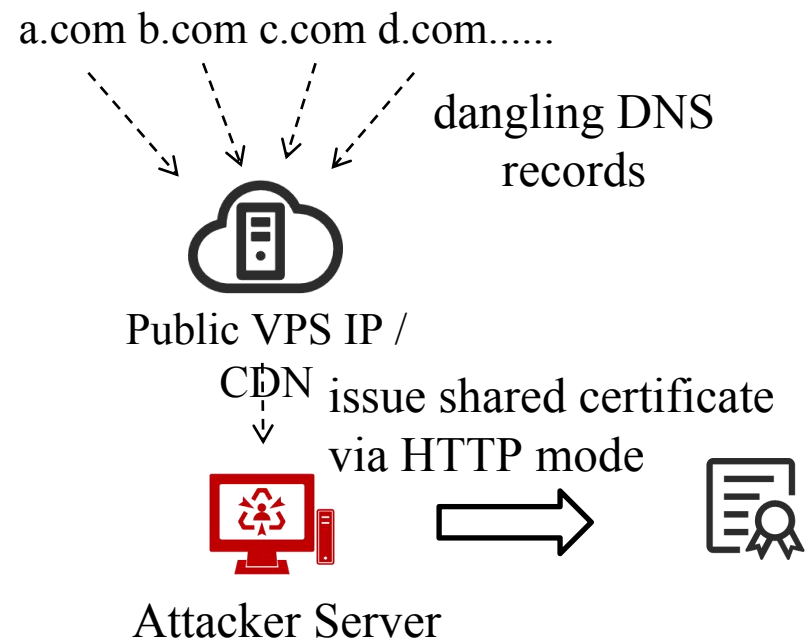
Attacker can issue a multi-domain certificate and then resell part of the included domains to victims



◆ Method 2: Domain Takeover

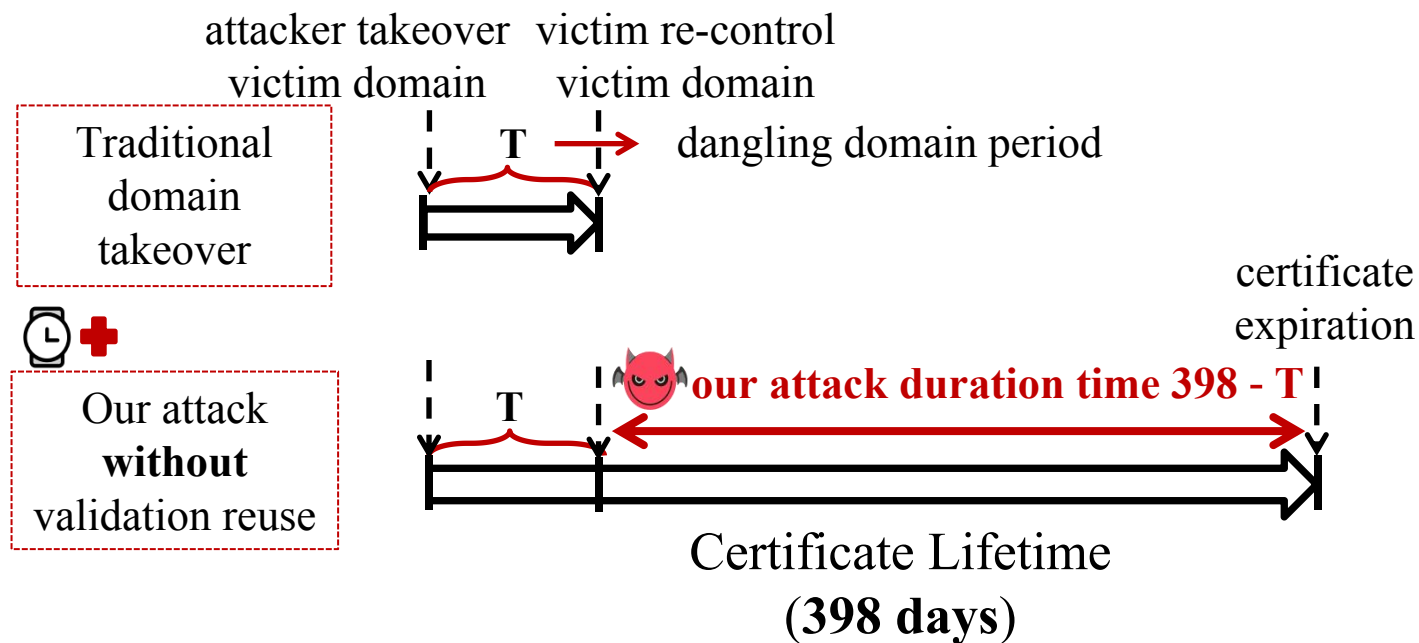


Attacker can take over dangling domains and issue shared certificates



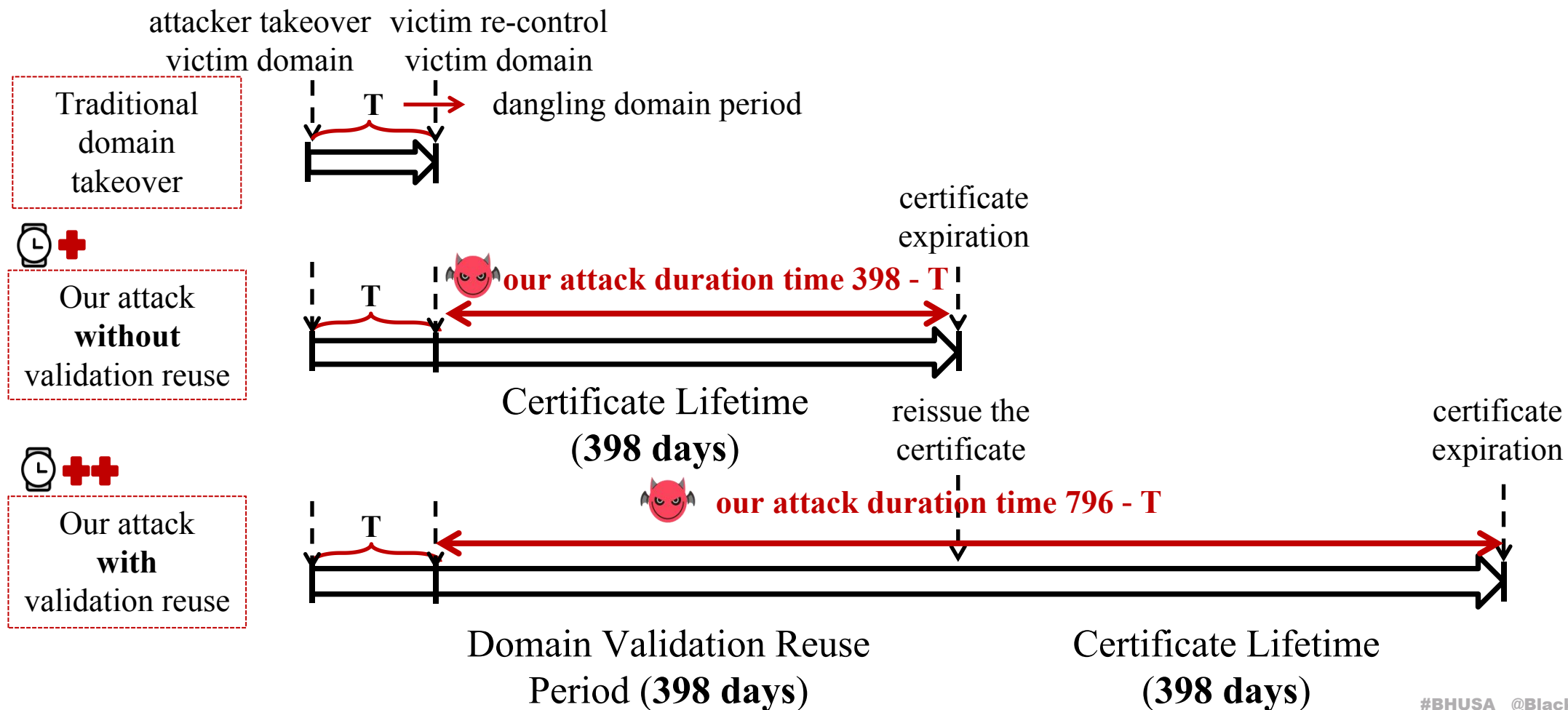
How to extend attack duration?

➤ Validation reuse extend the attack duration



How to extend attack duration?

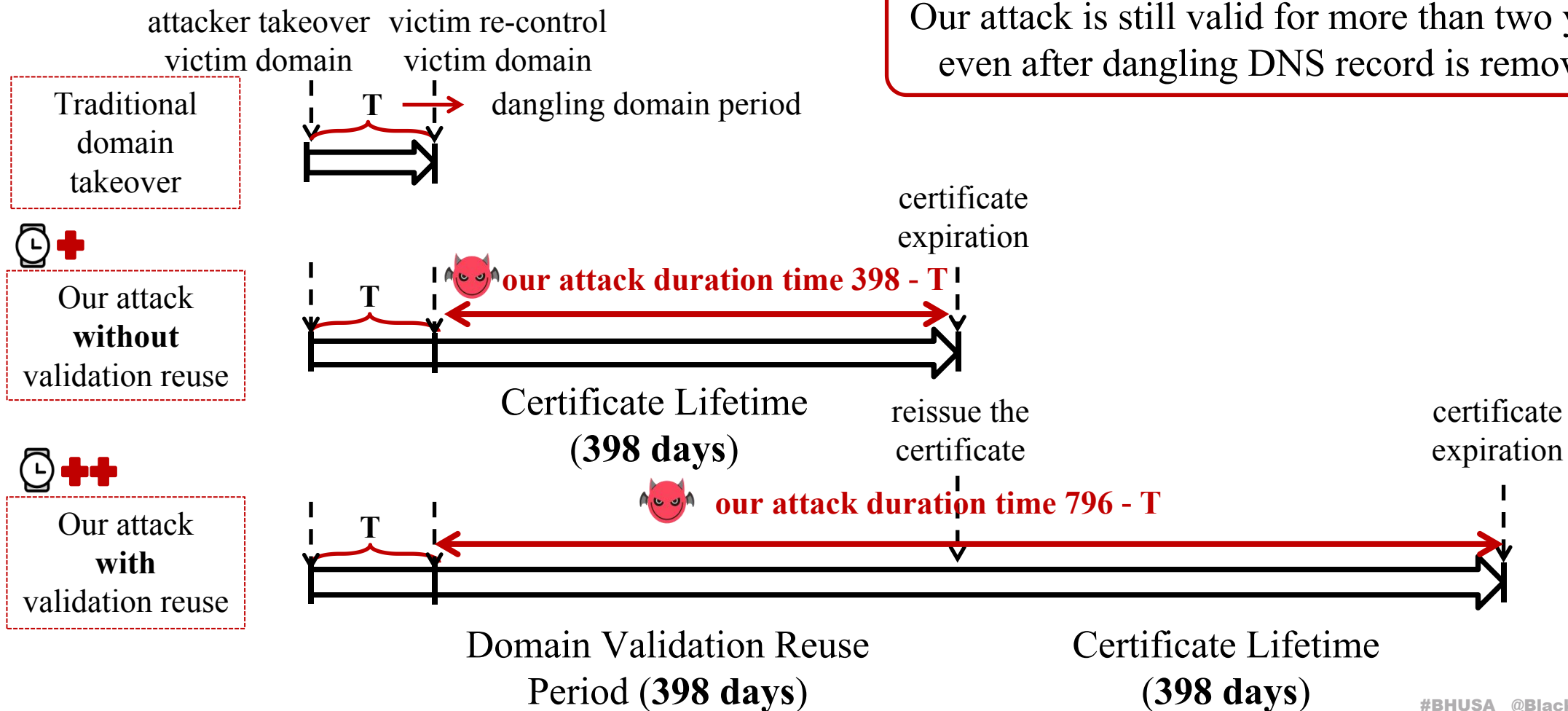
➤ Validation reuse extend the attack duration



How to extend attack duration?

➤ Validation reuse extend the attack duration

Our attack is still valid for more than two years even after dangling DNS record is removed



How to bypass potential countermeasures?

➤ **Shared certificate makes illegitimate certificate irrevocable**



Victim countermeasure: check CT logs and revoke illegitimate certificates



Our bypass technique: shared certificate include both attacker.com and victim.com



SAN:

attacker.com

victim.com

victim2.com

victim3.com

◆ Requirements for revoking a certificate^[2]:

- (1) Pass DOV for all domains OR
- (2) Possess the private key

[2] <https://letsencrypt.org/docs/revoking/>

How to bypass potential countermeasures?

➤ Shared certificate makes illegitimate certificate irrevocable



Victim countermeasure: check CT logs and revoke illegitimate certificates



Our bypass technique: shared certificate include both attacker.com and victim.com



SAN:

attacker.com

victim.com

victim2.com

victim3.com

◆ Requirements for revoking a certificate^[2]:

- (1) Pass DOV for all domains OR
- (2) Possess the private key

Such shared certificates are irrevocable by victims

[2] <https://letsencrypt.org/docs/revoking/>

How to bypass potential countermeasures?

➤ Shared certificate makes illegitimate certificate irrevocable



Victim countermeasure: check CT logs and revoke illegitimate certificates



Our bypass technique: shared certificate include both attacker.com and victim.com



SAN:

attacker.com

victim.com

victim2.com

victim3.com

◆ Requirements for revoking a certificate^[2]:

- (1) Pass DOV for all domains OR
- (2) Possess the private key

Such shared certificates are irrevocable by victims

We conducted experiment on ZeroSSL to report an illegitimate certificate shared with our domains on official problem reporting platform. No reply.

[2] <https://letsencrypt.org/docs/revoking/>

Large scale evaluation

Client-Side

Browser accepts
server push and SXG

**Cross-Origin
web attacks**

Server Side

Websites allow
attackers to acquire a
shared certificate

◆ Client-Side test target:

- (1) Top-Used browsers on Statcounter^[3]
- (2) Default browsers on leading mobiles
- (3) Celebrated applications on app store

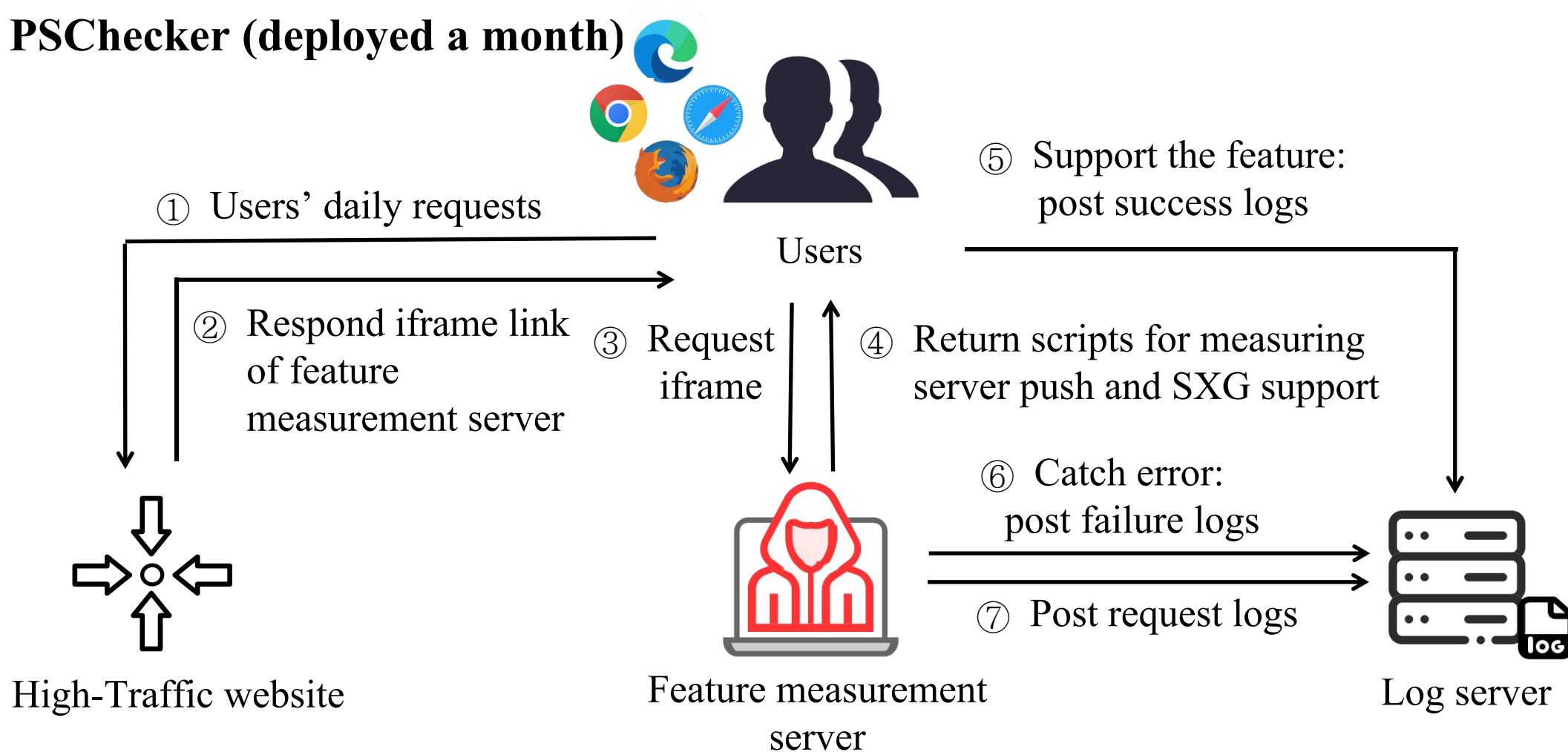
◆ Server-Side test target:

- (1) Reselling domains in Tranco 1M
- (2) Dangling domains in Tranco 1M
- (3) Existing cert-sharing domains in Tranco 1K

[3] <https://gs.statcounter.com/browser-market-share>

Client-side evaluation

➤ PSChecker (deployed a month)



Client-side evaluation

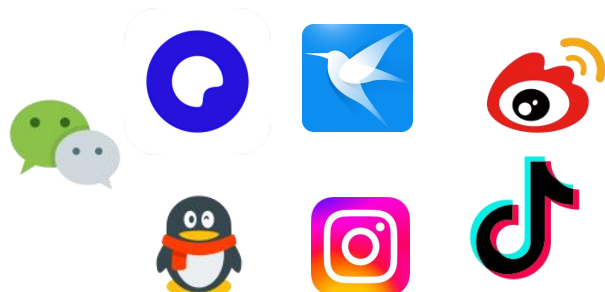
- Latest version of 11 top-used browsers and 5 default mobile browsers are vulnerable



Top-used Browsers						
Browser	CrossPUSH			CrossSXX		
	Win	iOS	Android	Win	iOS	Android
Chrome	●	●	●	●	○	●
Safari	-	●	-	-	○	-
Edge	●	●	●	●	○	●
Firefox	○	●	○	○	○	○
Opera	●	●	●	●	○	●
UC	●	●	●	○	○	○
360	●	●	●	●	○	●
IE	○	-	-	○	-	-
Yandex	●	●	●	●	○	●
QQ browser	●	●	●	●	○	●
CocCoc	●	-	●	●	-	●
Whale	●	●	●	●	○	●
Instabridge	-	●	●	-	○	●
Xunlei	●	●	●	●	○	●

Default Browsers on Mobile						
Samsung	-	-	●	-	-	●
Huawei	-	-	●	-	-	●
Oppo	-	-	●	-	-	●
Xiaomi	-	-	●	-	-	●
Vivo	-	-	●	-	-	●

- OS WebView spread the threat to third-party applications



App	CrossPUSH		CrossSXX	
	Android	iOS	Android	iOS
Baidu	●	●	●	○
Quark	●	●	○	○
Instagram	●	●	●	○
Wechat	●	●	●	○
QQmail	●	●	●	○
Weibo	●	●	●	○
Tiktok	●	●	●	○

Server-side evaluation

➤ Measure reselling domains

Use **WHOIS history data** to identify which domain has been resold to others

➤ Measure dangling domains

Utilized the state-of-the-art tool, **HostingChecker**^[4], to discover dangling domains

Server-side evaluation

➤ Measure reselling domains

Use **WHOIS history data** to identify which domain has been resold to others

➤ Measure dangling domains

Utilized the state-of-the-art tool, **HostingChecker**^[4], to discover dangling domains

➤ Measure cert-sharing domains

1. Scrape all domain names listed in the SAN of certificates from the top 1K websites
2. Extract subdomains from **HTTP responses, CT logs, and passive DNS databases.**
3. Check whether these associated domains share certificates with the top 1,000 websites.

[3] Zhang M, Li X, Liu B, et al. Detecting and measuring security risks of hosting-based dangling domains[J]. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 2023, 7(1): 1-28.

Server-side evaluation

➤ Numerous websites are affected

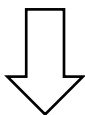
	Reselling Domains	Dangling Domains	Cert-Sharing Domains
Scope	Tranco Top 1M	Tranco Top 1M and subdomains	Tranco Top 1K
Count	11741	4919	829
Case Study	ftstatic.com (rank 3895) was once resold from an Australian food company to an American advertising agency.	A subdomain of windowsupdate.com from Microsoft is dangling, which can be registered by attacker.	Many Top 1K domains are sharing certificates with domains out of 1M (even from different organizations) , like baidu.com (rank 107)

**Talk is cheap, show me your
real-world case**

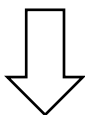
Microsoft case

14.au.www.download.windowsupdate.com

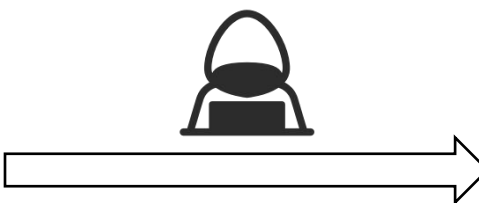
dangling CNAME



au.download.windowsupdate.qtlcdnect.com



qtlcdnect.com **unregistered**



attacker buy and
register qtlcdnect.com

attacker server IP



A

au.download.windowsupdate.qtlcdnect.com



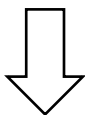
configure on
DNS

qtlcdnect.com **controlled**

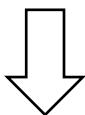
Microsoft case

14.au.www.download.windowsupdate.com

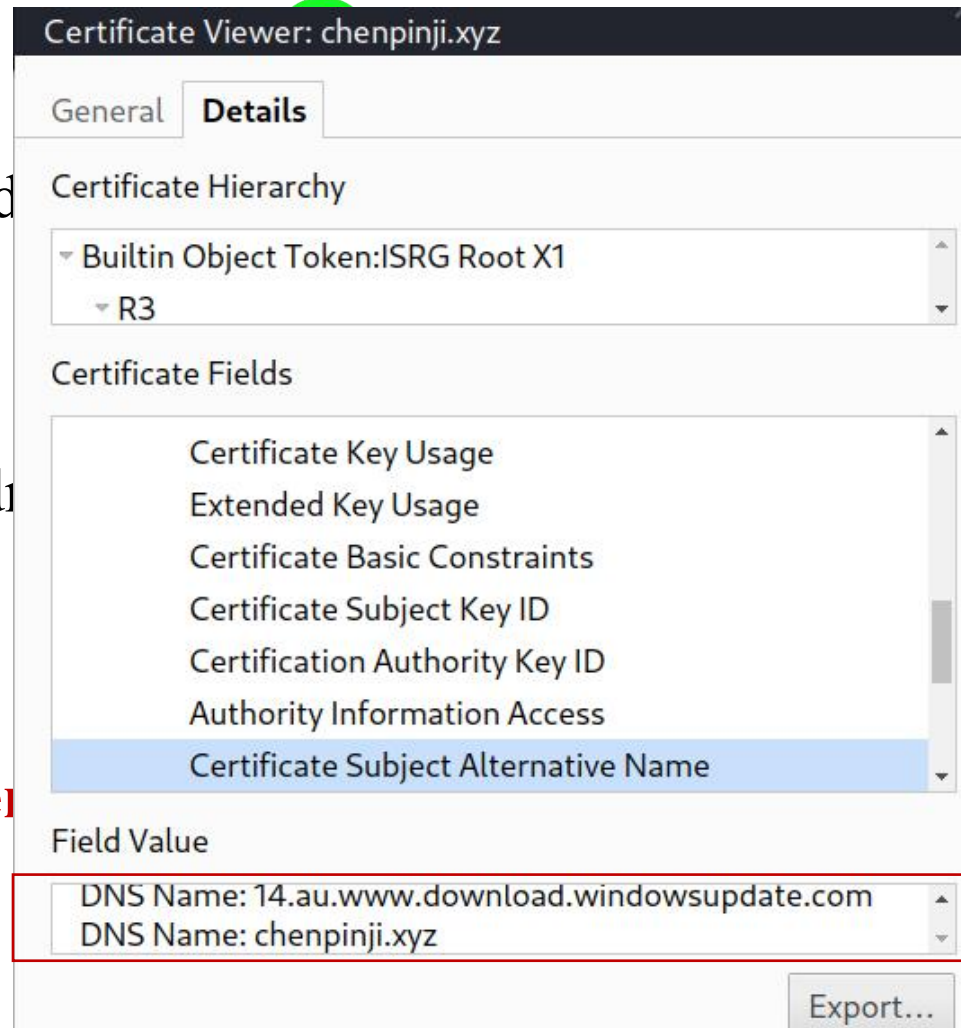
dangling CNAME



14.au.www.download.windowsupdate.qtldnconnect.com



qtldnconnect.com **unregistered**



server IP



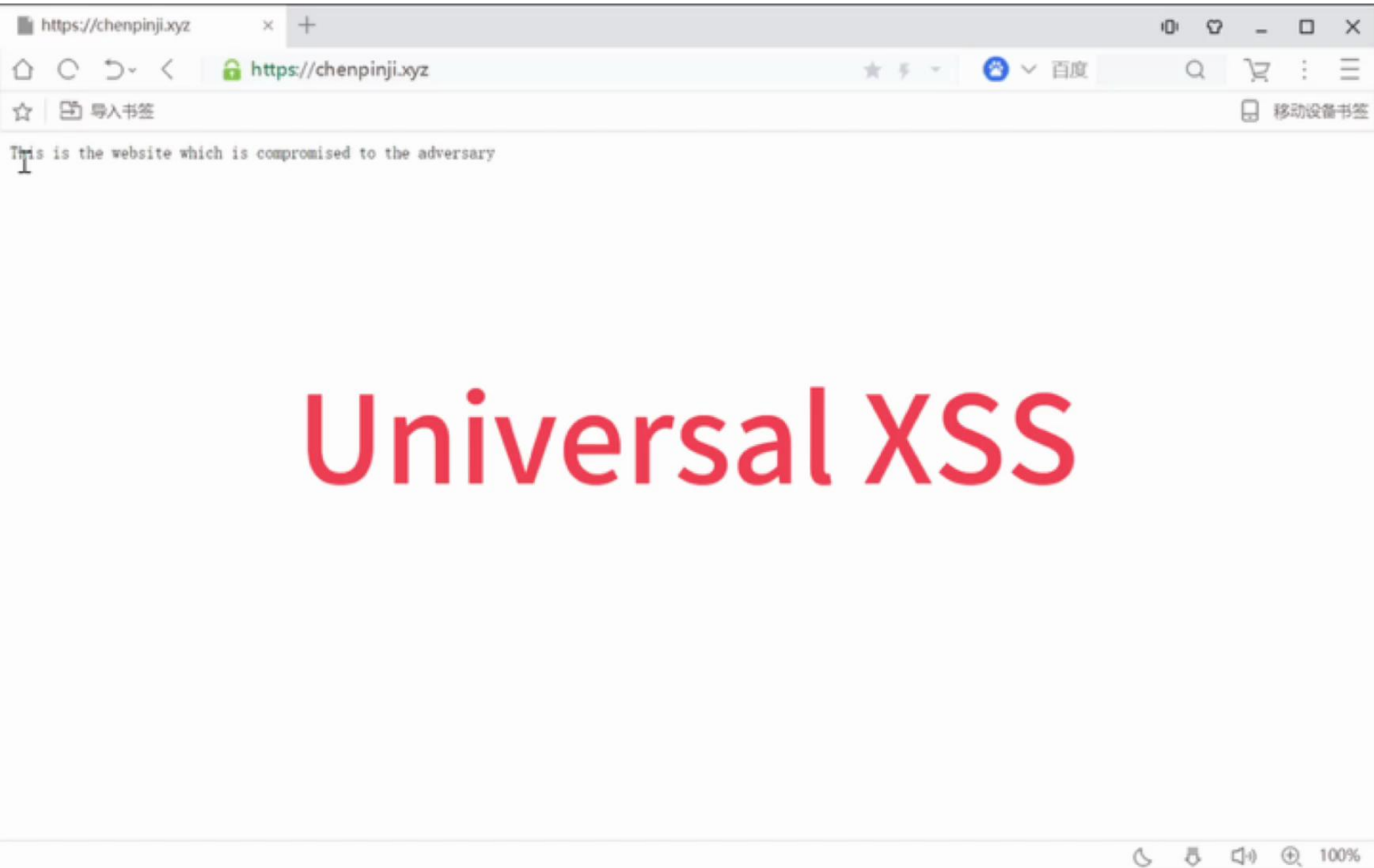
A

14.au.www.download.windowsupdate.qtldnconnect.com



configure on
DNS

qtldnconnect.com **controlled**



Mitigation

➤ **For browser vendors**

- ◆ Enforcing consistent authority (IP) in browsers to mitigate CrossPUSH
- ◆ Enforcing single-domain certificates to mitigate CrossSXG

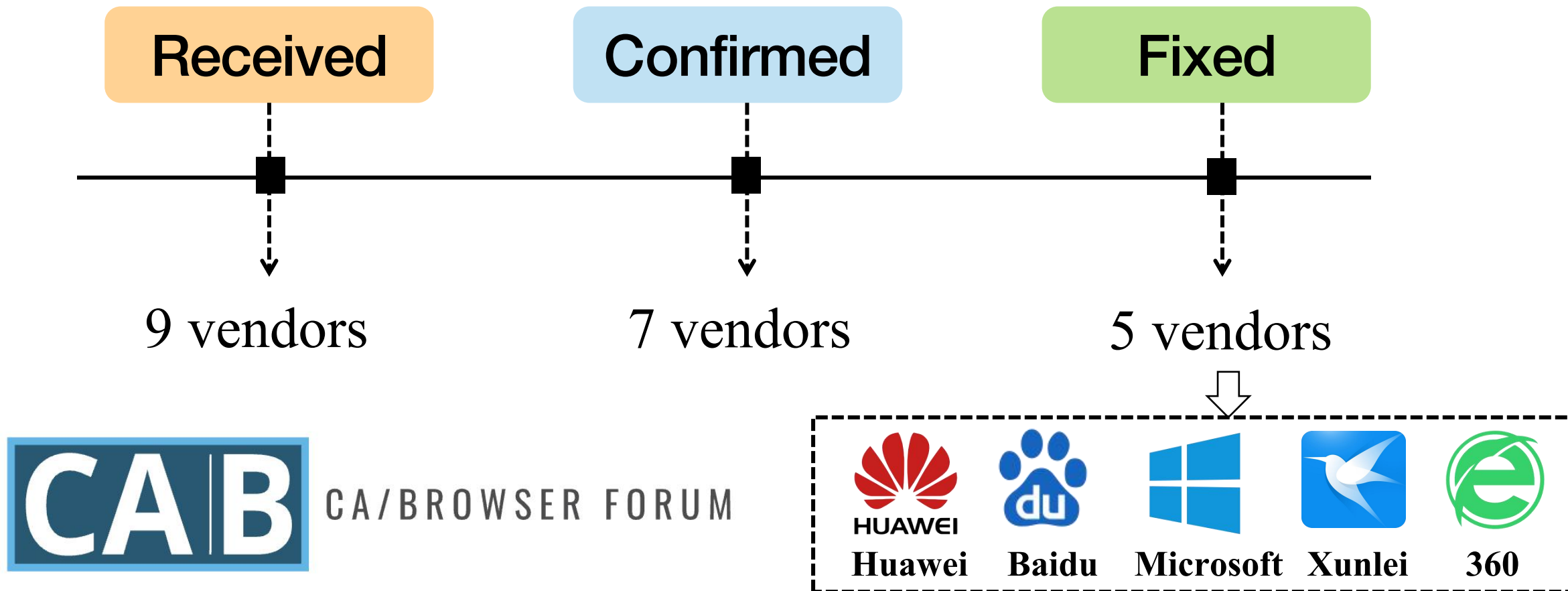
➤ **For certificate authorities**

- ◆ Facilitating the removal of domains from shared certificates at the request of domain owners listed in the SAN

➤ **For users**

- ◆ Inspecting certificate status in domain registration

Responsible disclosure



Join in our discussion in CA/B NetSec WG!




Takeaway

 **Our Observation:** HTTP/2 and HTTP/3 SAN-based origin is more permissive than browser URI-based origin

 **Novel Threat:** CrossPUSH and CrossSXG.

- enable **off-path** attackers to launch web attacks with shared certificates

 **Attack Practicality:** Weakness in Web PKI facilitate our attack.

- domain owner  certificate owner (create attack condition)
- domain lifetime  certificate lifetime (extend attack duration)
- control domain  can revoke certificate (bypass countermeasure)



AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

Thank you !
Q&A

Email: cpj24@mails.tsinghua.edu.cn

Discord: [pinjichen_55767](#)