Northeastern University
Khoury College of
Computer Sciences

NEU SecLab

Cached and Confused: Web Cache Deception in the Wild

USENIX Security 2020

Seyed Ali Mirheidari, Sajjad Arshad, Kaan Onarlioglu,
Bruno Crispo, Engin Kirda, William Robertson

University of Trento, Northeastern University, Akamai Technologies

Web Caches

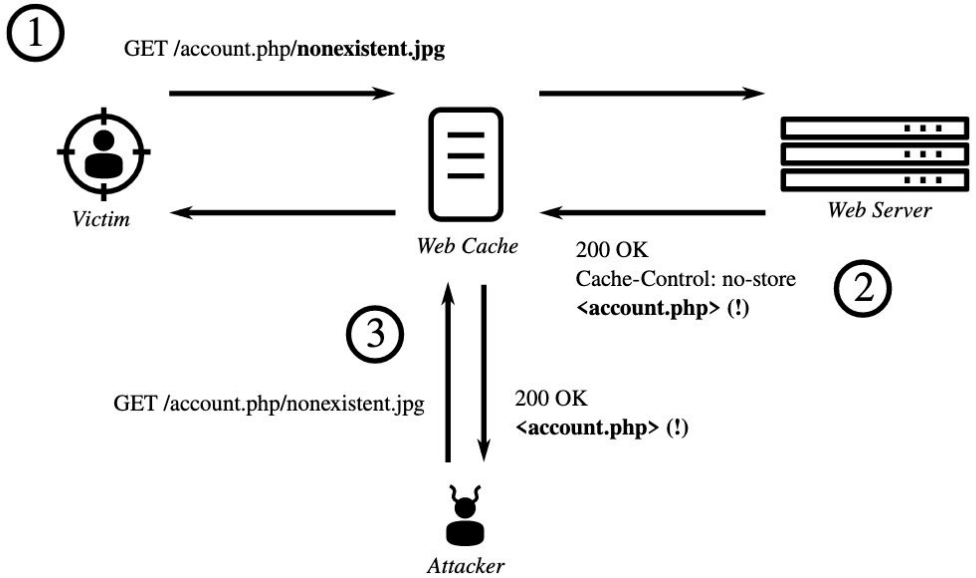
- ➔ Effective solution to decrease network latency and web application load
  - ◆ *Private* cache for single user (e.g., web browsers)
  - ◆ *Shared* cache for multiple users (e.g., web servers, MitM proxies)
  - ◆ Key component of *CDNs* to provide web availability (a.k.a. *Edge Servers*)
  - ◆ Study shows 74% of the *Alexa Top 1K* make use of CDNs
- ➔ Most common targets are static but frequently accessed resources
  - ◆ HTML pages, scripts, style sheets, images, ...
- ➔ Web servers use *Cache-Control* headers to communicate with web caches
  - ◆ *"Cache-Control: no-store"* indicates that the response should not be stored
  - ◆ Even though web caches **MUST** respect these headers, they offer configuration options for their users to ignore header instructions
    - Simple caching rules based on resource paths, file names and extensions (e.g., jpg, css, js)

Path Confusion

- Traditionally, URLs referenced web resources by directly mapping them to web server's file system structure:
  - example.com/**files/index.php**?p1=v1 correspond to the file **files/index.php** at the web server's document root directory
- Web servers introduced URL rewriting mechanisms to implement advanced application routing structures.
  - Clean URLs (a.k.a. RESTful URLs)
    - example.com/**index.php/v1** => example.com/**files/index.php?p1=v1**
- Browsers and proxies are not aware of this layer of abstraction between the resource file system path and its URL.
  - They process the URLs in an unexpected manner a.k.a ***Path Confusion***

Web Cache Deception (WCD)

- ➔ Introduced in 2017 by Omer Gil with PoC against PayPal
- ➔ WCD results different interpretations of a URL (*path confusion*) between a server and a web cache.

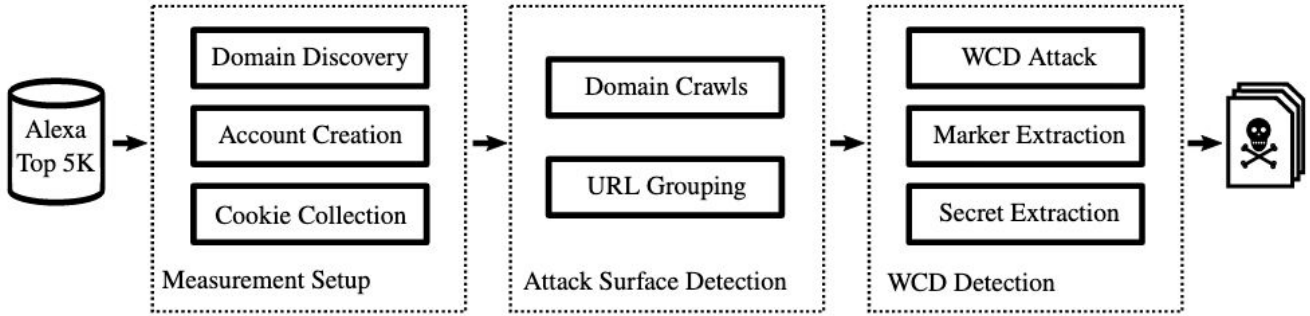


Research Questions

- ➔ What is the prevalence of WCD vulnerabilities on popular, highly-trafficked domains?
- ➔ Do WCD vulnerabilities expose PII and, if so, what kinds?
- ➔ Can WCD vulnerabilities be used to defeat defenses against web application attacks?
- ➔ Can WCD vulnerabilities be fully exploited by unauthenticated users?
- ➔ Can variation of Path Confusion techniques expand the number of vulnerable/exploitable sites?
- ➔ Is attacker geographical location important?
- ➔ Are default configurations of major CDN providers vulnerable?

Methodology

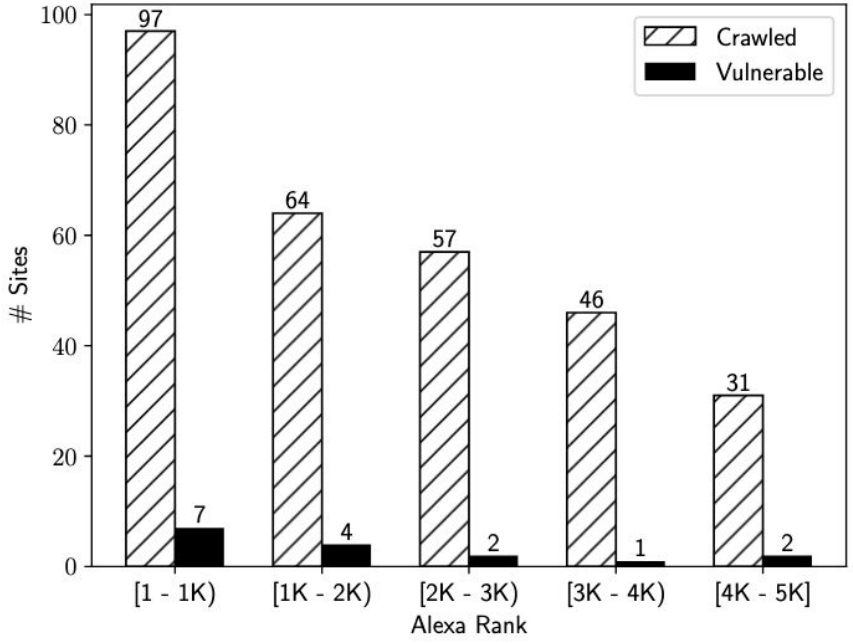
- ➔ Subdomain discovery to increase site coverage.
- ➔ Created account for **295 sites** from **Alexa Top 5K**



- ➔ Appended **"/<random>.css"** to each URL from the victim account..
- ➔ visited same page from the (un)authenticated attack crawler and compare responses.
- ➔ Responses analyzed for the disclosure of security tokens.

Crawling Stats & Alexa Ranking

	Crawled	Vulnerable
Pages	1,470,410	17,293 (1.2%)
Domains	124,596	93 (0.1%)
Sites	295	16 (5.4%)



Research Questions

- ➔ What is the prevalence of WCD vulnerabilities on popular, highly-trafficked domains?
- ➔ Do WCD vulnerabilities expose PII and, if so, what kinds?
- ➔ Can WCD vulnerabilities be used to defeat defenses against web application attacks?
- ➔ Can WCD vulnerabilities be fully exploited by unauthenticated users?
- ➔ Can variation of Path Confusion techniques expand the number of vulnerable/exploitable sites?
- ➔ Is attacker geographical location important?
- ➔ Are default configurations of major CDN providers vulnerable?

Vulnerabilities

- ➔ 14 vulnerable sites leaked PII including names, usernames, email addresses, and phone numbers.
- ➔ 6 vulnerable sites leaked CSRF tokens
- ➔ 6 vulnerable sites leaked session identifiers or user-specific API tokens
- ➔ Our results show that WCD can fully exploit with unauthenticated attackers.

Leakage	Pages	Domains	Sites
PII	17,215 (99.5%)	88 (94.6%)	14 (87.5%)
User	934 (5.4%)	17 (18.3%)	8 (50.0%)
Name	16,281 (94.1%)	71 (76.3%)	7 (43.8%)
Email	557 (3.2%)	10 (10.8%)	6 (37.5%)
Phone	102 (0.6%)	1 (1.1%)	1 (6.2%)
CSRF	130 (0.8%)	10 (10.8%)	6 (37.5%)
JS	59 (0.3%)	5 (5.4%)	4 (25.0%)
POST	72 (0.4%)	5 (5.4%)	3 (18.8%)
GET	8 (<0.1%)	4 (4.3%)	2 (12.5%)
Sess. ID / Auth. Code	1,461 (8.4%)	11 (11.8%)	6 (37.5%)
JS	1,461 (8.4%)	11 (11.8%)	6 (37.5%)
Total	17,293	93	16

Research Questions

- ➔ What is the prevalence of WCD vulnerabilities on popular, highly-trafficked domains?
- ➔ Do WCD vulnerabilities expose PII and, if so, what kinds?
- ➔ Can WCD vulnerabilities be used to defeat defenses against web application attacks?
- ➔ Can WCD vulnerabilities be fully exploited by unauthenticated users?
- ➔ Can variation of Path Confusion techniques expand the number of vulnerable/exploitable sites?
- ➔ Is attacker geographical location important?
- ➔ Are default configurations of major CDN providers vulnerable?

Variations on Path Confusion

```
example.com/account.php
example.com/account.php/nonexistent.css
```

(a) Path Parameter

```
example.com/account.php
example.com/account.php%0Anonexistent.css
```

(b) Encoded Newline (\n)

```
example.com/account.php;par1;par2
example.com/account.php%3Bnonexistent.css
```

(c) Encoded Semicolon (;)

```
example.com/account.php#summary
example.com/account.php%23nonexistent.css
```

(d) Encoded Pound (#)

```
example.com/account.php?name=val
example.com/account.php%3Fname=valnonexistent.css
```

(e) Encoded Question Mark (?)

Path Confusion Results

- ➔ Results confirm our hypothesis that launching WCD attacks with variations on path confusion increased possibility of successful exploitation significantly.
- ➔ Some variations elicit more *200 OK* server responses increasing the likelihood of the web server returning sensitive information.
- ➔ Each path confusion variation was able to attack a set of unique pages that were not vulnerable to other techniques.

Technique	Pages	Domains	Sites
Path Parameter	29,802 (68.9%)	103 (69.6%)	14 (56.0%)
Encoded \n	25,933 (59.9%)	86 (58.1%)	11 (44.0%)
Encoded ;	29,488 (68.2%)	105 (70.9%)	14 (56.0%)
Encoded #	28,643 (66.2%)	109 (73.6%)	15 (60.0%)
Encoded ?	37,374 (86.4%)	130 (87.8%)	19 (76.0%)
All Encoded	42,405 (98.0%)	144 (97.3%)	23 (92.0%)
Total	43,258 (100.0%)	148 (100.0%)	25 (100.0%)

Research Questions

- ➔ What is the prevalence of WCD vulnerabilities on popular, highly-trafficked domains?
- ➔ Do WCD vulnerabilities expose PII and, if so, what kinds?
- ➔ Can WCD vulnerabilities be used to defeat defenses against web application attacks?
- ➔ Can WCD vulnerabilities be fully exploited by unauthenticated users?
- ➔ Can variation of Path Confusion techniques expand the number of vulnerable/exploitable sites?
- ➔ Is attacker geographical location important?
- ➔ Are default configurations of major CDN providers vulnerable?

Empirical Experiments

- ➔ Cache Location
  - ◆ Victim in *Boston, MA, USA* and Attacker in *Trento, Italy.*
  - ◆ Attack failed for **19** sites but **6** sites were still exploitable.
- ➔ Cache Expiration
  - ◆ Web caches typically store objects for a short amount of time.
  - ◆ Attackers have a limited window of opportunity to launch a successful WCD attack.
  - ◆ Repeated the attack with **1 hour**, **6 hour**, and **1 day** delays for 19 sites.
  - ◆ **16**, **10**, and **9** sites were exploitable in each case, respectively.
- ➔ Cache configuration
  - ◆ We tested the basic content delivery solutions offered by major vendor to extract the default configuration.
  - ◆ By default, many Major CDN vendors do not make RFC-compliant caching decision.

Lessons Learned & Conclusion

- ➔ Configuring web caches correctly is not a trivial task.
  - ◆ Caching rules based on file extensions are prone to security problem.
  - ◆ CDNs are not intended to be plug & play solutions.
- ➔ As WCD attacks impact all web cache technologies, there is a widespread lack of user awareness.
  - ◆ There exists no technology to reliably determine if any part of system is vulnerable
- ➔ WCD is generally a "*system safety*" problem
  - ◆ There are no isolated faulty components.
  - ◆ Complex interactions among different technologies must take into consideration.
- ➔ Variations of path confusion techniques make it possible to exploit sites that are otherwise not impacted by the original attacks.

Thanks! Questions?

Seyed Ali Mirheidari, *seyedali.mirheidari@unitn.it*
Sajjad “JJ” Arshad, *@sajjadium*
Kaan Onarlioglu, Akamai, *www.onarlioglu.com*

UNIVERSITY OF TRENTO

Northeastern University
Khoury College of
Computer Sciences

NEU SecLab

iseclab