Web Cache Deception Escalates!

Seyed Ali Mirheidari  
University of Trento, Italy & Splunk Inc.  
mirheidari.ali@gmail.com

Kaan Onarlioglu  
Akamai Technologies  
kaan@iseclab.org

Matteo Golinelli  
University of Trento, Italy  
matteo.golinelli@unitn.it

Bruno Crispo  
University of Trento, Italy  
bruno.crispo@unitn.it

Engin Kirda  
Northeastern University  
ek@ccs.neu.edu
Background: Web Caches

62% of the Alexa Top 10k is behind a CDN

- Cache public content
- Frequently implemented by Content Delivery Networks (CDNs)
  - Increase scalability, availability, and performance
  - Offer options to disregard cache-control headers to increase performance

Background: Web Cache Deception

- Exploits request processing discrepancies between the origin server and a public web cache
- The cache is tricked into erroneously storing sensitive content
The Issue: State Of The Art Limitations

The state of the art detection methodology is marker-based [1]

Issues:

- Assumes that the victim is authenticated
  - More personal data that can be leaked

The Issue: **State Of The Art Limitations**

- **Scalability**: costly manual setup process
  - Account creation
  - Populate the target site with a unique marker string

- **Coverage**: cannot test
  - Pages that do not reflect the marker
  - Sites that do not need authentication
Motivation & Goals: **Web Cache Deception Escalates!**

New methodology that:

- **Overcomes** the state of the art limitations
- Targets **unauthenticated victims and pages**

Uses:

- **Content identicality** checks
- **Cache headers** heuristics
Methodology: Content Identicality Checks

Content identicality checks algorithm

response1 <- HTTP.get(URL)
response2 <- HTTP.get(URL)

if response1 ≠ response2 then
    return "Dynamic Content"
else:
    return "Static Content"
# Methodology: Cache Headers Heuristics

**Cache status headers** used by popular web caches

<table>
<thead>
<tr>
<th>CDN / Cache</th>
<th>Header Name(s)</th>
<th>Hit value(s)</th>
<th>Miss value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akamai</td>
<td>server-timing, X-Cache, X-Cache-Remote</td>
<td>desc=HIT, TCP_HIT</td>
<td>desc=MISS, TCP_MISS</td>
</tr>
<tr>
<td>CDN77</td>
<td>X-Cache</td>
<td>HIT</td>
<td>MISS</td>
</tr>
<tr>
<td>Cloudflare</td>
<td>cf-cache-status</td>
<td>HIT</td>
<td>MISS</td>
</tr>
<tr>
<td>CloudFront</td>
<td>x-cache</td>
<td>Hit from cloudfront</td>
<td>Miss from cloudfront</td>
</tr>
<tr>
<td>Fastly</td>
<td>X-Cache</td>
<td>HIT</td>
<td>MISS</td>
</tr>
<tr>
<td>Google Cloud</td>
<td>cdn_cache_status</td>
<td>hit</td>
<td>miss</td>
</tr>
<tr>
<td>KeyCDN</td>
<td>X-Cache</td>
<td>HIT</td>
<td>MISS</td>
</tr>
<tr>
<td>Azure</td>
<td>X-cache</td>
<td>TCP_HIT, TCP_REMOTE_HIT</td>
<td>TCP_MISS</td>
</tr>
<tr>
<td>Apache, ATS</td>
<td>X-Cache</td>
<td>HIT</td>
<td>MISS</td>
</tr>
<tr>
<td>NGINX</td>
<td>X-Proxy-Cache</td>
<td>HIT</td>
<td>MISS</td>
</tr>
<tr>
<td>Rack Cache</td>
<td>X-Rack-Cache</td>
<td>hit</td>
<td>miss</td>
</tr>
<tr>
<td>Squid</td>
<td>X-Cache</td>
<td>HIT from *</td>
<td>MISS from *</td>
</tr>
<tr>
<td>Varnish</td>
<td>X-Cache</td>
<td>HIT</td>
<td>MISS</td>
</tr>
<tr>
<td>Unknown</td>
<td>x-cache-info</td>
<td>cached</td>
<td>caching</td>
</tr>
</tbody>
</table>
Methodology: Web Cache Deception Escalates!

Step 1. Does the tested URL return dynamic content?

Step 2. Does the crafted attack URL still respond with dynamic content?

Step 3. Is the origin response to the attack URL cached?
## Experiment: Comparative Evaluation

Analysis of **404 websites** using:

- **CC** state of the art methodology **with authentication**
- **DE** our new methodology **with authentication**
- **DE** our new methodology **without authentication**

<table>
<thead>
<tr>
<th></th>
<th>CC</th>
<th>DE_{auth}</th>
<th>DE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total Detections</strong></td>
<td>21 (5.20%)</td>
<td>134 (33.17%)</td>
<td>129 (31.93%)</td>
</tr>
<tr>
<td><strong>True Positives</strong></td>
<td>18 (4.46%)</td>
<td>115 (28.47%)</td>
<td>104 (25.74%)</td>
</tr>
<tr>
<td><strong>False Positives</strong></td>
<td>3 (0.74%)</td>
<td>19 (4.70%)</td>
<td>25 (6.19%)</td>
</tr>
<tr>
<td><strong>Unique True Positives</strong></td>
<td>2 (0.50%)</td>
<td>13 (3.22%)</td>
<td>5 (1.24%)</td>
</tr>
</tbody>
</table>
Experiment: Large-Scale Analysis

Analysis of the Alexa Top 10k using DE

<table>
<thead>
<tr>
<th>Vulnerable Sites</th>
<th>1188 (11.88%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSRF Token</td>
<td>436 (36.70%)</td>
</tr>
<tr>
<td>CSP Nonce</td>
<td>13 (1.09%)</td>
</tr>
<tr>
<td>OAuth State</td>
<td>34 (2.86%)</td>
</tr>
<tr>
<td>Session ID</td>
<td>63 (5.30%)</td>
</tr>
</tbody>
</table>

![Bar chart showing vulnerable sites across Alexa ranks]
Results: Security Impact

- **Unauthenticated** and **authenticated victims** are impacted

- Stealing of **security tokens**
  - Enables **bypassing security protections**

- Leakage of **personal information**
Results: Case Studies

Attack scenario chains:

- Login CSRF by stealing the OAuth state
- CSRF by stealing CSRF tokens
- Session Hijacking by stealing Session IDs
- Cache Poisoning leading to stored XSS
- Supply chain issue: one vulnerable cloud service provider impacted 456 sites

4200$ in bug bounties
Conclusions

- Unauthenticated victims are impacted as well
  - Publicly accessible pages can contain security-critical secrets

- WCD is still an open problem and we don’t have a clear and reliable solution
Conclusions

- WCD is a **system problem** and no single component is necessarily faulty in isolation
  - Website operators must **consider the full integration**
    - Caches are not plug & play technologies
  - Simple caching rules can have **far-reaching effects**
Web Cache Deception Escalates!

- **DE**: new methodology for **WCD vulnerabilities detection**
  - Overcomes the **limitations** of the state of the art
  - Outperforms the state of the art
- Different target: **unauthenticated victims and pages**
- **Prevalence** of WCD vulnerabilities
- **Successful exploitation** of vulnerabilities

https://github.com/golim/wcde

Q&A: matteo.golinelli@unitn.it