WEB CACHE DECEPTION ATTACK

Omer Gil
No SID
About me

• Omer Gil
• 28
• Married + Java
• PT team leader at EY
• Student
About caching
About caching

CDN
About caching

CDN
About caching

CDN
About caching

Load Balancer
About caching

Load Balancer
About caching

Load Balancer
About caching

Load Balancer
About caching

Reverse Proxy

DMZ
About caching

Reverse Proxy

DMZ
About caching

https://www.example.com/stylesheet.css
About caching

https://www.example.com/stylesheet.css
About caching

https://www.example.com/stylesheet.css

Stylesheet.css
About caching

https://www.example.com/stylesheet.css

```css
div {

}

Stylesheet.css
Servers’ reactions

http://www.example.com/account.php/nonexistent.css

The Spanner: http://www.thespanner.co.uk/2014/03/21/rpo/
XSS Jigsaw: http://blog.innerht.ml/page/2/
**Servers’ reactions**

http://www.example.com/account.php\/*nonexistent.css*

Response from http://www.example.com:80/account.php/nonexistent.css [127.0.0.1]

HTTP/1.1 200 OK
Date: Thu, 19 Jun 2014 15:47:53 GMT
Server: Apache/2.4.4 (Win32) OpenSSL/0.9.8y PHP/5.4.16
X-Powered-By: PHP/5.4.16
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 1330
Connection: close
Content-Type: text/html

<html>
<head>
<title>Account</title>
</head>
</html>
Getting down to business

“Hey, access
https://www.bank.com/account.do/stylesheet.css”
Getting down to business

The user browses to
https://www.bank.com/account.do/stylesheet.css
Getting down to business

https://www.bank.com/account.do/stylesheet.css
returns with the content of account.do
and the private page is cached
Getting down to business

The attacker browses to https://www.bank.com/account.do/stylesheet.css and gets the content of the user’s account.do page.
Getting down to business

The attacker browses to https://www.bank.com/account.do/stylesheet.css and gets the content of the user’s account.do page
Hi Omer.
Get even more out of your PayPal account

PayPal balance

No balance needed to shop or send payments

Currencies

Add funds
Withdraw funds

Completed

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>JUN 28</td>
<td>Purchase</td>
<td>- $10,00 AUD</td>
</tr>
<tr>
<td>MAY 26</td>
<td>Canceled - Request received $10.00 USD</td>
<td>$10.00 USD</td>
</tr>
</tbody>
</table>

Bank accounts and cards

- MasterCard
- American Express

When you add your local bank account, you can
Conditions

- Web cache functionality is set for the web application to cache static files based on their extensions, disregarding any caching header.
- When accessing a page like `http://www.example.com/home.php/nonexistent.css`, the web server will return the content of `home.php` for that URL.
- Victim has to be authenticated while accessing the triggering URL.
Why the HELL #1

would a web application react like this?
Why the HELL #1

Django:

http://www.sampleapp.com/inbox/

Welcome to your account!
Why the HELL #1

Django:

http://www.sampleapp.com/inbox/test.css

```
from django.conf.urls import include, url
from . import views

urlpatterns = [
    url(r'^inbox/$', views.index, name='index')
]
```

urls.py
Why the HELL #1

Django:

```
from django.conf.urls import include, url
from . import views

urlpatterns = [
    url(r'^inbox', views.index, name='index')
]
```

urls.py
Why the HELL #1

Django:

```
from django.conf.urls import include, url
from . import views

urlpatterns = [
    url(r'^inbox/$', views.index, name='index'),
]
```

urls.py

```
http://www.sampleapp.com/inbox/test.css
```

The requested URL /inbox/test.css was not found on this server.
Why the HELL #2

would a caching mechanism react like this?
Why the HELL #2

IIS ARR:

![Edit Cache Control Rule dialog box](image)
Why the HELL #2

IIS ARR:

http://www.sampleapp.com/welcome.php/test.css
Why the HELL #2

IIS ARR:

http://www.sampleapp.com/welcome.php/test.css
Why the HELL #2

Cloudflare:

- Eligibility phase

  class, css, jar, js, jpg, jpeg, gif, ico, png, bmp, pict, csv, doc, docx, xls, xlsx, ps, pdf, pls, ppt, pptx, tif, tiff, ttf, otf, webp, woff, woff2, svg, svgz, eot, eps, ejs, swf, torrent, midi, mid

Why the HELL #2

Cloudflare:

- Disqualification phase

‘Edge cache expire TTL’ to the rescue!
Why the HELL #2

Cloudflare:

Edge Cache Expire TTL: Easiest way to override any existing headers

With Cache Everything, we respect all headers. If there is any header in place from the server or a CMS solution like WordPress, we will respect it. However, we got many requests from customers who wanted an easy way to override any existing headers. Today, we are releasing a new feature called 'Edge cache expire TTL' that does just that.

https://blog.cloudflare.com/edge-cache-expire-ttl-easiest-way-to-override/
Mitigation

- Only cache files if their HTTP caching headers allow
- Store all static files in a designated directory
- Cache files by their content type
- Don’t accept this! http://www.example.com/home.php/non-existent.css. Return 302 or 404 instead
THANKS

@omer_gil
omergil.blogspot.com