Web Cache Poisoning

ISEC3004 - Group 21

Structure

01. What is Web Cache
02. How does Web Caching work (Information with Cache key, request line, web server etc)
03. Brief intro in to server-side caching and client- side Caching
04. What is Web Cache Poisoning
05. Steps involved in Web Cache Poisoning attack and how it happens
06. How to detect it
07. SHOW DEMO
08. PATCHING -UP (Pros and Cons of patching)
09. How to project the application from web cache poisoning attack (Mitigation)
10. Pros and Cons of each method to mitigate

What is Web Cache?

Web Cache is the technology which allows storing of data for reuse. Web Cache is implemented on both server-side and client-side.

The web cache system works by storing a copy of a certain webpage in its memory so that it could quickly serve any incoming requests for the same webpage, by reloading the copy saved on its memory.

The benefit of using this system is that it saves time since web pages are quickly loaded and made available for clients' requests. And at the same time, this technology also improves the performance of a website since response times are reduced.

How does web caching work?

- When a user visits a web page for the first time, his request is served from a remote server.
- The web server obtains the necessary information from the remote server and converts the requested web page into an HTML file that can be viewed by the client browser.
- This response is also cached in the ***cache server*** which is residing between the user and the remote server for a set period of time.
- When a user requests this page again, the cache server sends the ready **static files** to the browser rather than processing the request from remote side.
- The cache server employs a **cache key** to determine whether the response is in cache memory or whether the request should be handled by the server. The cache key contains predefined subset of request's components which mostly included the host header and request line.
- If an incoming requests' cache key matches with a previous request, this will be handled by the cache server, which serves a copy of the stored response.

Server-side caching vs Client-side caching

What is Server-side caching?

Server-side caching is when web files and data from a server is stored temporarily as a copy, in order to service any incoming requests for the same web files and data so that expensive database operations to serve the same data can be avoided. Server-side caching significantly reduces server-side operations as data can be processed quickly from the database to the browser.

What is Client-side caching?

Client-side caching involves storing web files and data on the browser memory located on the client's computer instead of the cache memory on the server. So when a client requests a web file and data from a client-side-caching-enabled website, the browser will temporarily store a copy of the data so that they could survive the request quickly and easily.

What is Web Cache Poisoning?

Web Cache Poisoning is a type of attack where the attackers exploit the vulnerabilities of the cache system by injecting a specially constructed data into cache memory so that the Web Server responds with malicious content to its visitors.

How to Patch Web Cache Poisoning Due to unkeyed headers.

- Create a dummy virtual host.
- ❖ According to the demo shown above the web cache is poisoned through an unkeyed header called the X-Forwarded-Host header. The header is replaced with another faked poisonous header that carries malicious data.
- ❖ If we use Apache or Nginx, you can construct a dummy virtual host to intercept requests from forged requests (i.e., requests with unknown host headers) and stop cache poisoning due to unkeyed headers.
- ❖ All faked requests, which are not truly linked to the application, will be routed to the created dummy virtual host when one is setup on the web server.

```
<VirtualHost xxx.xxx.xxx.xxx:80>
    ServerName default
    RewriteEngine On
    RewriteRule .* - [G]
</VirtualHost>
```

This image depicts a setup of a default VirtualHost on Apache.

Creating a dummy VirtualHost continued.

How does the dummy virtual host work?

- For example we will set our browser to http://www.example.com.
- When our computer wants to communicate with www.example.com, it queries its DNS resolver to determine which IP address to use.
- Our computer establishes a connection to that IP address and sends an HTTP Host:header indicating its desire to communicate with www.example.com.
- In order to determine what to do with a request for content from www.example.com, the webserver consults its settings. Any of the following could occur:

1.If www.example.com is listed as a ServerName or ServerAlias for a VirtualHost, the content will be delivered using the settings for that VirtualHost.

2.If the server has no VirtualHosts at all, it will deliver the content using the settings in its httpd.conf file.

3.The first Virtualhost on the list will be utilized to deliver the content if www.example.com isn't listed in any of the Virtual Hosts on the server.

So what we do is simply to setup a default VirtualHost that doesn't serve content.

Pros of Creating a Dummy Virtual Host..

- When using virtual hosting, the server can deliver different results depending on the requested IP address, hostname, or port.
- It is possible to validate and Sanitize user inputs. (Specially unkeyed user inputs)
- A smaller configuration file so that the server starts faster and uses less memory.

Cons of Creating a Dummy Virtual Host.

- The biggest disadvantage of virtual server hosting is that all of the hosted websites will go offline if the server goes offline.
- Performance may decrease if the computer it is running on does not have sufficient power when several virtual machines are running on the same host.

How to Patch Web Cache Poisoning Due to unkeyed headers.

- **Disabling Caching.**

★ **Server side cache prevention.**

1. **Using Meta-Tags to prevent Caching.**

Step 1 : Go to the code of the web page.

Step2 : Enter the following tag into the header in order to disable caching.

```
<META http-equiv=Pragma content=no-cache>
<META http-equiv=Cache-Control content=no-cache,no-store,max-age =0>
<META http-equiv=Expires content=-1>
```

- The directive CACHE-CONTROL:NO-CACHE instructs servers to send requests to the origin server rather than using cached data. PRAGMA NO-CACHE derives the same semantics as CACHE-CONTROL:NO-CACHE . An illegal EXPIRES date, such as "-1," is interpreted into "now." Thus, it is possible to force a modification check at each visit by setting EXPIRES to -1.

★ **Server side cache prevention.**

2.Remove all files of a particular type from the cache.

We can use these .htaccess rules to deactivate caching for all files of the same type:

```
<IfModule mod_headers.c>
  <FilesMatch ".(js|css|xml|png|jpg|jpeg|html)$">
    Header set Cache-Control "private"
  </FilesMatch>
</IfModule>
```

The aforementioned will turn off caching for all html, js, css, xml, png, and jpeg files. The list of file extensions can be changed to just include the ones we require.

How to Patch Web Cache Poisoning Due to unkeyed headers.

★ **Browser side Caching prevention.**

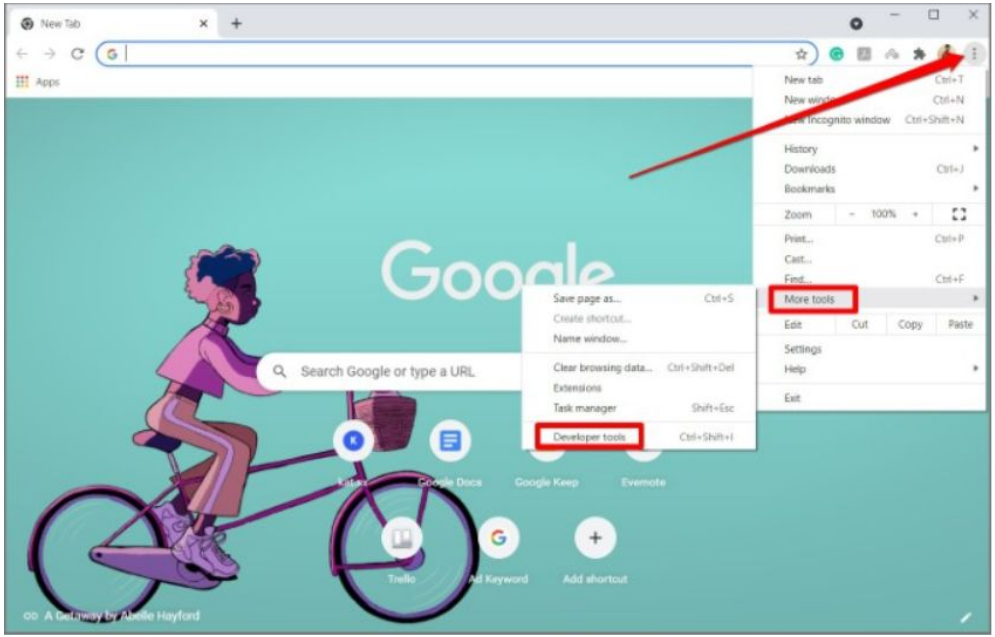
The simplest approach is to have the user's web browser disable caching.

The steps to turn off browser caching are listed below in Google chrome which is a most popular web browser.

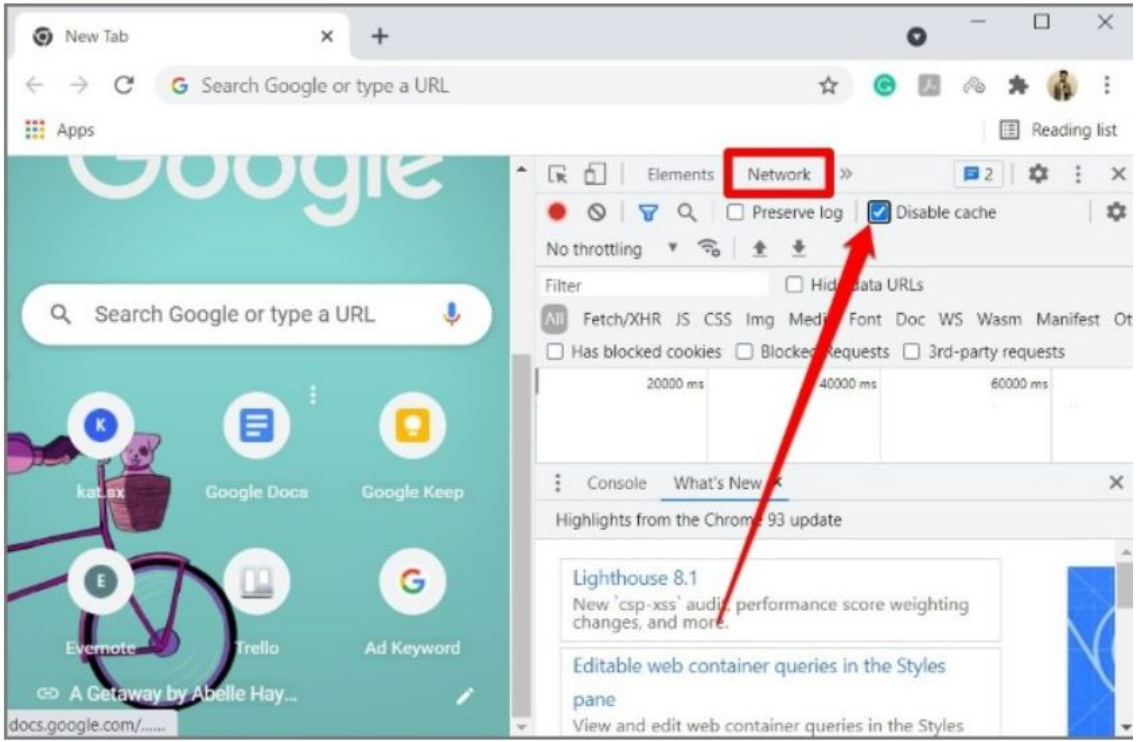
There are two approaches in disabling cache in Google Chrome.

1.Disable Cache by using developer tools.

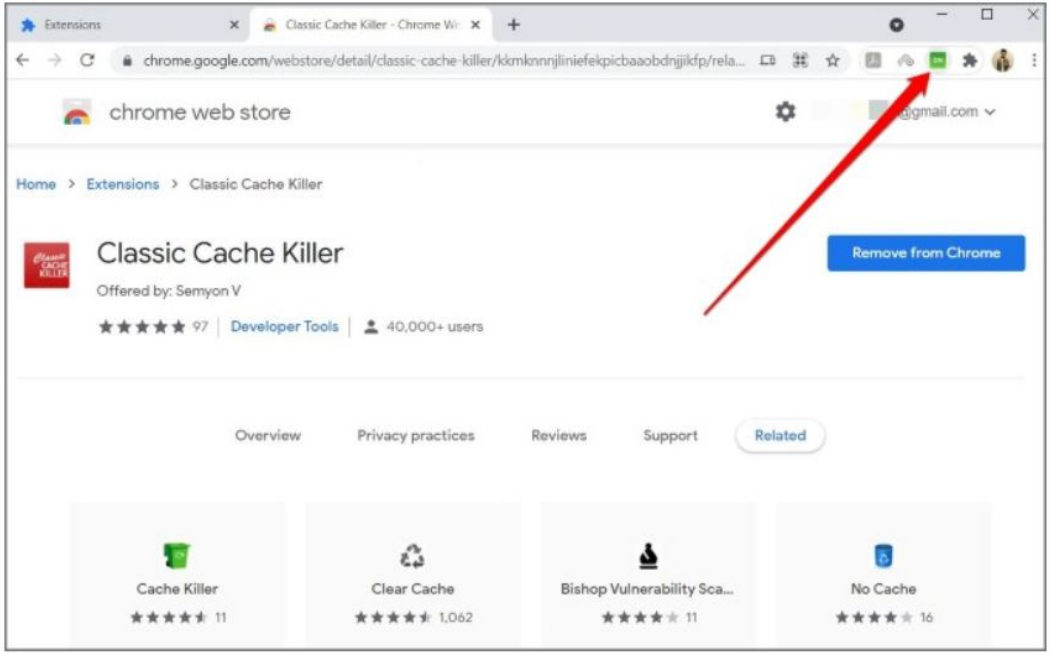
Step 1 : Open Google Chrome on your computer, then go to More tools > Developer tools using the three dots in the top right corner of the screen.



Step 2 : Select the Disable cache checkbox under the Network tab.



2.Disable Cache by using third party extensions.



By simply turning on the extension after adding the extension to Chrome, we can stop Chrome from keeping cached data.

Pros of Disabling Cache.

- Attacks like web cache poisoning and also attacks such as cross-site scripting, Javascript injection and open redirection.
- Improves data privacy because cache can store and share sensitive information.
- The issue of not viewing updated data on websites can be solved because it's likely that information the cache saves will prevent websites from displaying their most recent content to us.

Cons of Disabling Cache.

- Decreases the application performance because of the increment of the load on the back end.
- Higher data retrieval latencies as a result of increased database load make the performance of the entire application uncertain.
- If the primary database charges per throughput, the cost increases since there are more instances of the database being used because the cache, which can give thousands of IOPS, is not being used.

How to mitigate it

Mitigating the poisoning of the web cache due to the unkeyed headers vulnerability.

- Avoid unkeyed inputs by using security testing tools like the burp suite and its extensions, such as param miner, to detect them.
- Find any inputs that are in the response but are not included in the cache key, such as headers. Ensure that they are either disabled, taken out of the cache layer, or added to the cache key.
- Using CLI scanners like Web Cache Vulnerability Scanner (WCVS) that has features to support several web cache poisoning techniques including unkeyed headers poisoning.

How to mitigate it

Mitigating the poisoning of the web cache due to the Http response splitting vulnerability.

- Think of GET request bodies as being untrusted, and ensure that they cannot change the contents of a response. Consider utilizing a POST request or avoiding the cache entirely if the body of a GET request can alter the content of a response.
- Ensure that any user-provided data that could be utilized to create response headers is thoroughly validated and sanitized.
- Encrypt harmful characters like \r and \n.

How to mitigate it

- Only static files that never change and don't require user input to produce a cached response should have caching enabled.
- If the values in HTTP headers are not included in your cache key, do not rely on them and never provide users with HTTP headers in cached content.
- Routinely checking web security alerts to stay up to date on risks to internet security.

References

(References - slide 1)

https://en.wikipedia.org/wiki/Web_cache

https://crashtest-security.com/web-cache-poisoning/

(References - slide 2 ,3,4,6)

https://crashtest-security.com/web-cache-poisoning/

(References - Slide 5)

https://www.codingninjas.com/codestudio/library/server-side-caching-and-client-side-caching

References

(Reference - Slide 6
https://edgemesh.com/blog/difference-between-server-side-caching-and-client-side-caching-and-which-is-good-for-your-website#:~:text=Server%2Dside%20caching%20is%20the,the%20information%20from%20the%20server.)

(Reference - Slide 6
https://edgemesh.com/blog/difference-between-server-side-caching-and-client-side-caching-and-which-is-good-for-your-website#:~:text=Server%2Dside%20caching%20is%20the,the%20information%20from%20the%20server.)

(References - slide 7)

https://serverfault.com/questions/662262/apache-accepting-requests-to-other-servers/662356#662356

https://crashtest-security.com/invalid-host-header/

(Reference - slide 8)

https://serverfault.com/questions/520195/how-does-servername-and-serveralias-work/520201#520201

References

(Reference - Slide 9)

https://newsnreleases.com/2021/10/02/benefits-and-drawbacks-of-virtual-server-hosting/

(Reference - slide 10)

https://newsnreleases.com/2021/10/02/benefits-and-drawbacks-of-virtual-server-hosting/

(References - slide 11)

https://www.tech-faq.com/prevent-caching.html

https://stackoverflow.com/questions/42265359/does-meta-http-equiv-pragma-content-no-cache-affect-xmlhttprequests

(References - slide 12)

https://www.siteground.com/kb/disable-dynamic-caching-website/

References

(Reference - Slide 13)

https://www.tech-faq.com/prevent-caching.html

(Reference- Slide 14,15,16)

https://techwiser.com/disable-cache-google-chrome-firefox/

(Reference - Slide 17)

https://www.indeed.com/career-advice/career-development/cache-pros-and-cons

(Reference - Slide 18)

https://aws.amazon.com/caching/

References

(Reference - Slide 19)

https://www.techtarget.com/searchsecurity/news/252446725/Web-cache-poisoning-attacks-demonstrated-on-major-websites-platforms

https://crashtest-security.com/web-cache-poisoning/

https://www.acunetix.com/blog/articles/what-is-web-cache-poisoning/

(Reference - Slide 20)

https://www.comparitech.com/blog/information-security/web-cache-poisoning/

(Reference - Slide 21)

https://developers.cloudflare.com/cache/best-practices/avoid-web-poisoning/