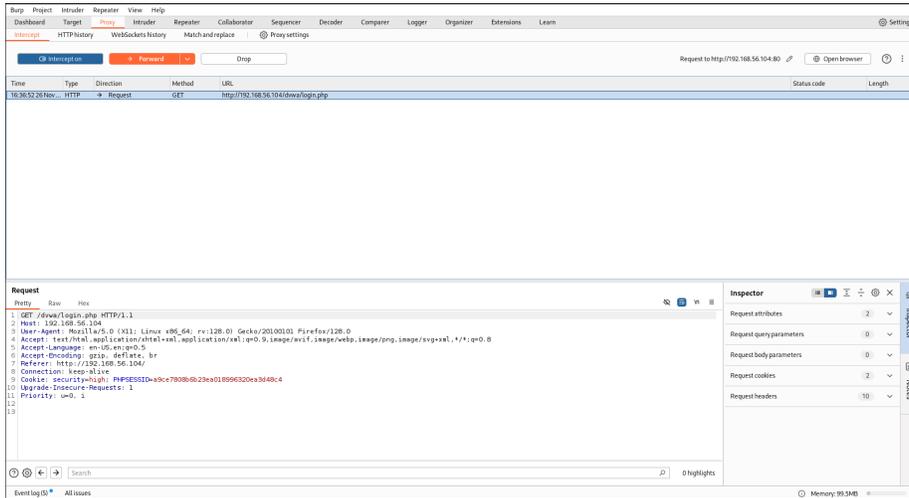


# Assignment 5 - Aidan Sharpe

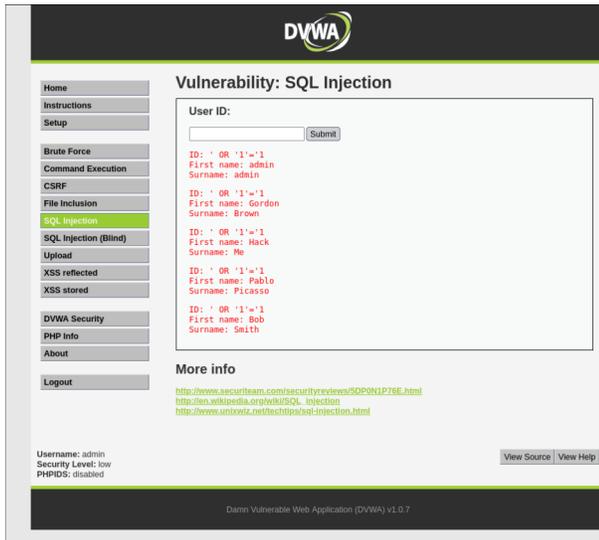
## Task 1: Setting Up Burp Suite

Configure Burp Suite to intercept traffic from your browser and verify the setup.



## Task 2: Testing for SQL Injection with Burp Repeater

Use Burp Repeater to test for SQL injection vulnerabilities on the DVWA login page.



### Task 3: Conducting a Brute Force Attack with Burp Intruder

Use Burp Intruder to perform a brute force attack on DVWA's login page.

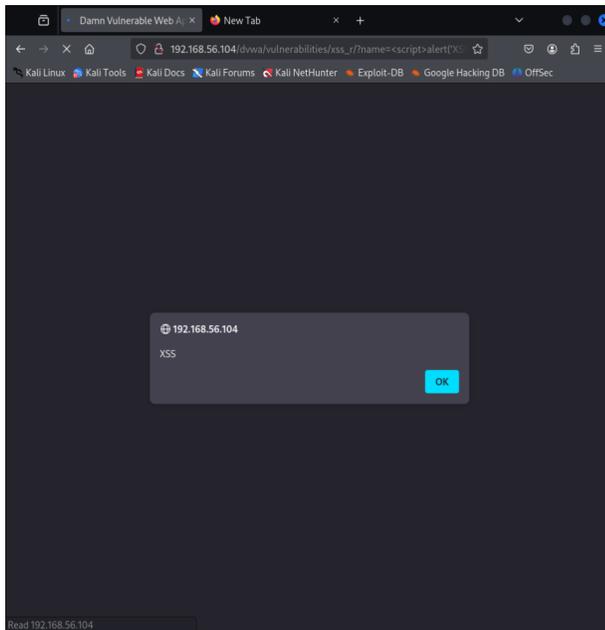
We found a list of usernames and passwords online, and loaded them into Burp Suite. We used the cluster bomb attack to go through all possible combinations in the two lists. We also turned redirect to “allow on-site” to make it easier to identify a successful login. The correct credentials had a response length of 4391, while the other credentials had a response length of 1675.

Request	Payload 1	Payload 2	Status code	Response received	Error	Redirects full...	Timeout	Length	Comment
0			200	68		1		2623	
1	carlos	123456	200	48		1		1675	
2	root	123456	200	24		1		1675	
3	admin	123456	200	32		1		1675	
4	test	123456	200	29		1		1675	
5	guest	123456	200	25		1		1675	
6	info	123456	200	25		1		1675	
7	adm	123456	200	33		1		1675	
8	carlos	password	200	42		1		1675	
9	root	password	200	22		1		4391	
10	admin	password	200	24		1		1675	
11	test	password	200	26		1		1675	
12	guest	password	200	22		1		1675	
13	info	password	200	24		1		1675	
14	adm	password	200	26		1		1675	
15	carlos	12345678	200	25		1		1675	
16	root	12345678	200	21		1		1675	
17	admin	12345678	200	22		1		1675	
18	test	12345678	200	22		1		1675	
19	guest	12345678	200	22		1		1675	
20	info	12345678	200	23		1		1675	
21	adm	12345678	200	22		1		1675	
22	carlos	query	200	24		1		1675	
23	root	query	200	23		1		1675	
24	admin	query	200	22		1		1675	
25	test	query	200	23		1		1675	
26	guest	query	200	23		1		1675	
27	info	query	200	27		1		1675	
28	adm	query	200	27		1		1675	

### Task 4: Testing for Cross-Site Scripting (XSS)

Test for XSS vulnerabilities by injecting JavaScript payloads into input fields.

We inserted the script `<script>alert('XSS')</script>` into the “What’s your name?” field on the “XSS reflected” page. When the “submit” button was pressed, we were greeted with:



## Task 5: Analyzing and Reporting Results

Analyze the results from your tests and identify potential vulnerabilities in DVWA.

SQL injection and XSS are both forms of arbitrary code execution. SQL injection poses a threat to the confidentiality of database entries, while the XSS vulnerability allows the execution of code on client machines.

The brute force attack allows unauthorized people to login as any user. While it is not an efficient way to gain access, it can be effective against common or default passwords.

## Task 6: Mitigation Recommendations

SQL injection and XSS vulnerabilities can be mitigated via better string handling. They are caused by strings being processed as code when they should be only processed as text. To mitigate these types of attacks, better string input handling should be implemented. There exist many freely available libraries to protect applications from SQL injection and XSS vulnerabilities.

Brute force attacks, on the other hand, can be mitigated by limiting the number of unsuccessful login attempts. This can be done by applying a timer, or by locking the session.

### **Task 7: Reflection**

There are a wide array vulnerabilities that can be exploited at the application layer. While these attacks may not provide direct access to internal networks, they can certainly be used as an entry point.

These types of vulnerabilities also make it clear that the internet was not designed with security in mind, and that extra measures must be put in place to protect web applications from external attacks.