RAPPORTO

Per prima cosa ho scritto "cat/var/www/html/DVWA/config/config.inc.php" nel terminale Kali Linux per visualizzare la configurazione del database di DVWA. Ho annotato l'indirizzo IP del server, il nome del database, il nome utente e la password:

```
#   See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ]   = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ]     = 'kali';
$_DVWA[ 'db_password' ] = 'kali';
$_DVWA[ 'db_port']      = '3306';
```

Poi ho avuto accesso al database usando il comando “mysql -u kali -p -h 127.0.0.1 dvwa”. Ma mi ha richiesto SSL:

```
┌──(rinatrustamov㉿kali)-[~]
└─$ mysql -u kali -p -h 127.0.0.1 dvwa
Enter password:
ERROR 2026 (HY000): TLS/SSL error: SSL is required, but the server does not support it
```

Per aggirarlo, ho modificato un po' il comando "sudo mysql -u kali -p -h 127.0.0.1 --skip-ssl dvwa". Dopo aver scritto la password, è entrato in MariaDB. Ho scritto diversi comandi per trovare la tabella e il campo che contiene le password:

```
MariaDB [dvwa]> use dvwa;
Database changed
MariaDB [dvwa]> SHOW TABLES;
+----------------+
| Tables_in_dvwa |
+----------------+
| guestbook      |
| users          |
+----------------+
2 rows in set (0.001 sec)

MariaDB [dvwa]> DESCRIBE users;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| user_id      | int(6)      | NO   | PRI | NULL    |       |
| first_name   | varchar(15) | YES  |     | NULL    |       |
| last_name    | varchar(15) | YES  |     | NULL    |       |
| user         | varchar(15) | YES  |     | NULL    |       |
| password     | varchar(32) | YES  |     | NULL    |       |
| avatar       | varchar(70) | YES  |     | NULL    |       |
| last_login   | timestamp   | YES  |     | NULL    |       |
| failed_login | int(3)      | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
8 rows in set (0.006 sec)
```

Dopo aver dato il comando “MariaDB [dvwa]> SELECT user, password FROM users;”, ho finalmente ottenuto gli hash delle password:

```
MariaDB [dvwa]> SELECT user, password FROM users;
+---------+----------------------------------+
| user    | password                         |
+---------+----------------------------------+
| admin   | 5f4dcc3b5aa765d61d8327deb882cf99 |
| gordonb | e99a18c428cb38d5f260853678922e03 |
| 1337    | 8d3533d75ae2c3966d7e0d4fcc69216b |
| pablo   | 0d107d09f5bbe40cade3de5c71e9e9b7 |
| smithy  | 5f4dcc3b5aa765d61d8327deb882cf99 |
+---------+----------------------------------+
5 rows in set (0.007 sec)
```

Ci sono 5 hash, 4 dei quali sono univoci. Ho scritto ognuno di questi 4 hash in un file di testo e salvato come hashes.txt. Ho usato il comando "hashid hashes.txt" per verificare che gli hash siano in formato 0 (md5):

```
┌──(rinatrustamov㉿kali)-[~/Desktop]
└─$ hashid hashes.txt

--File 'hashes.txt'--
Analyzing '5f4dcc3b5aa765d61d8327deb882cf99'
[+] MD2
[+] MD5
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger-128
[+] Skein-256(128)
[+] Skein-512(128)
[+] Lotus Notes/Domino 5
[+] Skype
[+] Snefru-128
[+] NTLM
[+] Domain Cached Credentials
[+] Domain Cached Credentials 2
[+] DNSSEC(NSEC3)
[+] RAdmin v2.x
Analyzing 'e99a18c428cb38d5f260853678922e03'
[+] MD2
[+] MD5
[+] MD4
[+] Double MD5
```

```
[+] NTLM
[+] Domain Cached Credentials
[+] Domain Cached Credentials 2
[+] DNSSEC(NSEC3)
[+] RAdmin v2.x
Analyzing '8d3533d75ae2c3966d7e0d4fcc69216b'
[+] MD2
[+] MD5
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger-128
[+] Skein-256(128)
[+] Skein-512(128)
[+] Lotus Notes/Domino 5
[+] Skype
[+] Snefru-128
[+] NTLM
[+] Domain Cached Credentials
[+] Domain Cached Credentials 2
[+] DNSSEC(NSEC3)
[+] RAdmin v2.x
Analyzing '0d107d09f5bbe40cade3de5c71e9e9b7'
[+] MD2
[+] MD5
[+] MD4
[+] Double MD5
[+] LM
```

Per decifrare gli hash, ho usato sia John the Ripper che Hashcat. Entrambi hanno funzionato e mi hanno dato gli stessi risultati. John the Ripper è stato più veloce, ma Hashcat è stato migliore per scegliere più wordlist contemporaneamente. Sfortunatamente ho lasciato il terminale Linux e tutti i comandi sono stati cancellati. Quindi ho iniziato dall'inizio per decifrare, ma gli strumenti non hanno compilato i comandi dicendo che gli hash erano già stati decifrati. Quindi ho usato il comando –show per vedere le password. Ha funzionato per Hashcat, ma non per John:

```
┌──(rinatrustamov㉿kali)-[~/Desktop]
└─$ hashcat -m 0 -a 0 hashes.txt /usr/share/wordlists/rockyou.txt
 /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000
000.txt /usr/share/seclists/Passwords/darkweb2017-top10000.txt /u
sr/share/seclists/Passwords/Cracked-Hashes/milw0rm-dictionary.txt

hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELO
C, LLVM 17.0.6, SLEEF, POCL_DEBUG) - Platform #1 [The pocl projec
t]
==========================================================================
==========================================================================
==
* Device #1: cpu--0×000, 1436/2937 MB (512 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

INFO: All hashes found as potfile and/or empty entries! Use --sho
w to display them.

Started: Thu Dec 12 19:49:41 2024
Stopped: Thu Dec 12 19:49:41 2024
```

```
┌──(rinatrustamov㉿kali)-[~/Desktop]
└─$ hashcat --show hashes.txt -m 0
5f4dcc3b5aa765d61d8327deb882cf99:password
e99a18c428cb38d5f260853678922e03:abc123
8d3533d75ae2c3966d7e0d4fcc69216b:charley
0d107d09f5bbe40cade3de5c71e9e9b7:letmein
```

```
┌──(rinatrustamov㉿kali)-[~/Desktop]
└─$ john --format=raw-md5 --wordlist=/usr/share/seclists/Passwords/xato-net-10-million-passwords-1000000.txt hashes.txt

Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 128/128 ASIMD 4×2])
No password hashes left to crack (see FAQ)
```

Quindi le password sono: "password, abc123, charley, letmein".

Dopo aver terminato l'esercizio principale, ho iniziato l'attività extra. Ho usato Hashcat e 3 liste più grandi per decifrare gli hash:

```
┌──(rinatrustamov㉿kali)-[~/Desktop]
└─$ hashcat -m 3200 -a 0 hashes.txt2.save /usr/share/wordlists/ro
ckyou.txt /usr/share/seclists/Passwords/xato-net-10-million-passw
ords-1000000.txt /usr/share/seclists/Passwords/darkweb2017-top100
00.txt /usr/share/seclists/Passwords/Cracked-Hashes/milw0rm-dicti
onary.txt
```

Le password sono state trovate e sono “shadow, darksoul, mena”:

```
┌──(rinatrustamov㉿kali)-[~/Desktop]
└─$ hashcat -m 3200 -a 0 hashes.txt2.save /usr/share/wordlists/ro
ckyou.txt /usr/share/seclists/Passwords/xato-net-10-million-passw
ords-1000000.txt /usr/share/seclists/Passwords/darkweb2017-top100
00.txt /usr/share/seclists/Passwords/Cracked-Hashes/milw0rm-dicti
onary.txt

hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELO
C, LLVM 17.0.6, SLEEF, POCL_DEBUG) - Platform #1 [The pocl projec
t]
==========================================================================
==========================================================================
==
* Device #1: cpu--0×000, 1436/2937 MB (512 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 72

Hashes: 3 digests; 3 unique digests, 3 unique salts
Bitmaps: 16 bits, 65536 entries, 0×0000ffff mask, 262144 bytes, 5
/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte

Watchdog: Hardware monitoring interface not found on your system.
```

```
* Zero-Byte

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 0 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

$2b$05$7O7caKmIpPBZxM.RV1lnie/S8jiAjE4C/S6neVAN0ObgJ7tE4dW3.:shad
ow
[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit => s

Session..........: hashcat
Status...........: Running
Hash.Mode........: 3200 (bcrypt $2*$, Blowfish (Unix))
Hash.Target......: hashes.txt2.save
Time.Started.....: Thu Dec 12 19:32:25 2024 (26 secs)
Time.Estimated...: Thu Dec 12 23:33:29 2024 (4 hours, 0 mins)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/4 (25.00%)
Speed.#1.........:     1983 H/s (7.54ms) @ Accel:4 Loops:32 Thr:1
 Vec:1
Recovered........: 1/3 (33.33%) Digests (total), 1/3 (33.33%) Dig
ests (new), 1/3 (33.33%) Salts
```