Hashcat
Kali Linux

REVSUP 2018
Aurora Hernandez
Step 1:

- Fire up Kali and Open Hashcat
Basic Syntax

- Kali > hashcat options hashfile mask|wordfiles|dictionaries
- -m (hashtype)
- -a (attack mode)
Step 2: Hash types

- List of some hash types we can work with
<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Category</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>900</td>
<td>MD4</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>510</td>
<td>MD5</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>5100</td>
<td>Half MD5</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>SHA1</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>1300</td>
<td>SHA-224</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>1400</td>
<td>SHA-256</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td>SHA-384</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>1700</td>
<td>SHA-512</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>SHA-3 (Keccak)</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>BLAKE2b-512</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>10100</td>
<td>SipHash</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>6000</td>
<td>RIPEMD-160</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>6100</td>
<td>Whirlpool</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>6900</td>
<td>GOST R 34.11-94</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>11700</td>
<td>GOST R 34.11-2012 (Streebog) 256-bit</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>11800</td>
<td>GOST R 34.11-2012 (Streebog) 512-bit</td>
<td>Raw Hash</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>md5($pass.$salt)</td>
<td>Raw Hash, Salted and Iterated</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>md5($salt.$pass)</td>
<td>Raw Hash, Salted and Iterated</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>md5(utf16le($pass).$salt)</td>
<td>Raw Hash, Salted and Iterated</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>md5($salt.utf16le($pass))</td>
<td>Click to add speaker notes</td>
<td>Raw Hash, Salted and Iterated</td>
</tr>
<tr>
<td>3800</td>
<td>md5($salt.$pass.$salt)</td>
<td>Raw Hash, Salted and Iterated</td>
<td></td>
</tr>
<tr>
<td>3710</td>
<td>md5($salt.md5($pass))</td>
<td>Raw Hash, Salted and Iterated</td>
<td></td>
</tr>
<tr>
<td>4010</td>
<td>md5($salt.$salt.md5($pass))</td>
<td>Raw Hash, Salted and Iterated</td>
<td></td>
</tr>
<tr>
<td>4110</td>
<td>md5($salt.md5($salt.$salt))</td>
<td>Raw Hash, Salted and Iterated</td>
<td></td>
</tr>
</tbody>
</table>
Step 3: Choosing a Wordlist

- Dictionary attack
- Thousands of wordlists available for download
  - Built in wordlists
Let’s find the worklists in Kali:

Kali > locate wordlist
root@kali:~# locate wordlist
/usr/sbin/remove-default-wordlist
/usr/sbin/select-default-wordlist
/usr/sbin/update-default-wordlist
/usr/share/wordlists
/usr/share/applications/kali-wordlists.desktop
/usr/share/dict/README.select-wordlist
/usr/share/dirb/wordlists
/usr/share/dirb/wordlists/big.txt
/usr/share/dirb/wordlists/catalan.txt
/usr/share/dirb/wordlists/common.txt
/usr/share/dirb/wordlists/euskera.txt
/usr/share/dirb/wordlists/extensions_common.txt
/usr/share/dirb/wordlists/indexes.txt
/usr/share/dirb/wordlists/mutations_common.txt
/usr/share/dirb/wordlists/others
/usr/share/dirb/wordlists/small.txt
/usr/share/dirb/wordlists/spanish.txt
/usr/share/dirb/wordlists/stress
/usr/share/dirb/wordlists/vulns
/usr/share/dirb/wordlists/others/best1050.txt
/usr/share/dirb/wordlists/others/best110.txt
/usr/share/dirb/wordlists/others/best15.txt
/usr/share/dirb/wordlists/others/names.txt
Step 4: Grab Hashes

- Logged in as root then grab the hashes
- Tail: prints the last few number of lines of a file then terminates

Kali > tail /etc/shadow
Shadow file w/ hashes

- What is a shadow file?
  - Password file that holds the password hashes separate from the other data in the word-readable password
Step 5: Type of hashing in use

- View encryption type the system is using
  
  Kali > `more /etc/login.defs`

- Defines the site-specific configuration for shadow password suite
- Location of user mailboxes
- Password aging controls
- Encryption method used to encrypt passwords
# /etc/login.defs - Configuration control definitions for the login package.
#
# Three items must be defined: MAIL_DIR, ENV_SUPATH, and ENV_PATH.
# If unspecified, some arbitrary (and possibly incorrect) value will
# be assumed. All other items are optional - if not specified then
# the described action or option will be inhibited.
#
# Comment lines (lines beginning with "#") and blank lines are ignored.
#
# Modified for Linux. --marekm
- Scroll down about 90% of the file and you will see the SHA512 encryption
- Tell hashcat when we crack the hashes
Step 5: Let’s get cracking!

Time to crack the hashes!

- Where hashes are located
- Type of encryption
- Let’s crack it out!
Step 6: Create file

- Put hashes into a separate file `hash.lst`

Kali > `cp /etc/shadow hash.lst`
● Checking if copying was successful
Step 6: Remove all info, except hashes

Kali > vi hash.lst
Step 7: Final Step

Kali > `hashcat -m 1800 -a 0 -o cracked.txt --remove hash.lst /usr/share/sqlmap/txt/wordlist.txt`

- `-m 1800` designates the type of hash we are cracking (SHA-512)
- `-a 0` designates a dictionary attack
- `-o cracked.txt` is the output file for the cracked passwords
- `--remove` tells hashcat to remove the hash after it has been cracked
- `hash.lst` is our input file of hashes
- `/usr/share/sqlmap/txt/wordlist.txt` is the absolute path to our wordlist for this dictionary attack
Stopped: Tue Jun 5 23:18:23 2018

root@kali:~# hashcat -m 1800 -a 0 -o cracked.txt --remove hash.lst /usr/share/sqlmap/txt/wordlist.txt --force
hashcat (pull/1273/head) starting...

OpenCL Platform #1: The pocl project
=======================================
* Device #1: pthread-Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz, 1502/1502 MB allocatable, 1MCU

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers:
* Zero-Byte
* Single-Hash
* Single-Salt
* Uses-64-Bit
Hit ENTER to check progress
| Status          | Running
|----------------|-----------------
| Hash Type       | sha512crypt $6$, SHA512 (Unix)
| Hash Target     | $6$F9pvAV7j$wLafR38R6cn4I0BLZjYoY7x1W9mKl3wcg2NdWj...Cqd8n0
| Time Started    | Tue Jun 5 23:18:53 2018 (14 mins, 33 secs)
| Time Estimated  | Wed Jun 6 00:31:24 2018 (57 mins, 58 secs)
| Base            | File (/usr/share/sqlmap/txt/wordlist.txt)
| Queue           | 1/1 (100.00%)
| Speed Dev       | 324 H/s (8.93ms)
| Recovered       | 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
| Progress        | 279214/1406529 (19.85%)
| Rejected        | 174/279214 (0.66%)
| Restore Point   | 279214/1406529 (19.85%)
| Candidates #1   | ally -> alma
| HMAC Dev #1     | N/A

| Session         | hashcat
| Status          | Cracked
| Hash Type       | sha512crypt $6$, SHA512 (Unix)
| Hash Target     | $6$F9pvAV7j$wLafR38R6cn4I0BLZjYoY7x1W9mKl3wcg2NdWj...Cqd8n0
| Time Started    | Tue Jun 5 23:18:53 2018 (1 hour, 7 mins)
| Time Estimated  | Wed Jun 6 00:26:34 2018 (00 secs)
| Base            | File (/usr/share/sqlmap/txt/wordlist.txt)
| Queue           | 1/1 (100.00%)
| Speed Dev       | 314 H/s (9.00ms)
| Recovered       | 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
| Progress        | 1285093/1406529 (91.42%)
| Rejected        | 1285/1285093 (0.10%)
| Restore Point   | 1285765/1406529 (91.41%)
| Candidates #1   | toolsheds -> toosweet
| HMAC Dev #1     | N/A

Started: Tue Jun 5 23:18:39 2018
Stopped: Wed Jun 6 00:26:36 2018
root@Kali:~ # cat cracked.txt
$6$F9pvAVj$j$w1aRd38RBcn4I0BLZNJoYR7x1W9mKl3wcgzNdWjWJ4mOeXuyvhX5tL37PJwr7UUjBUzMKmhSiXga5d0dCqD8n0:toor
root@Kali:~ #