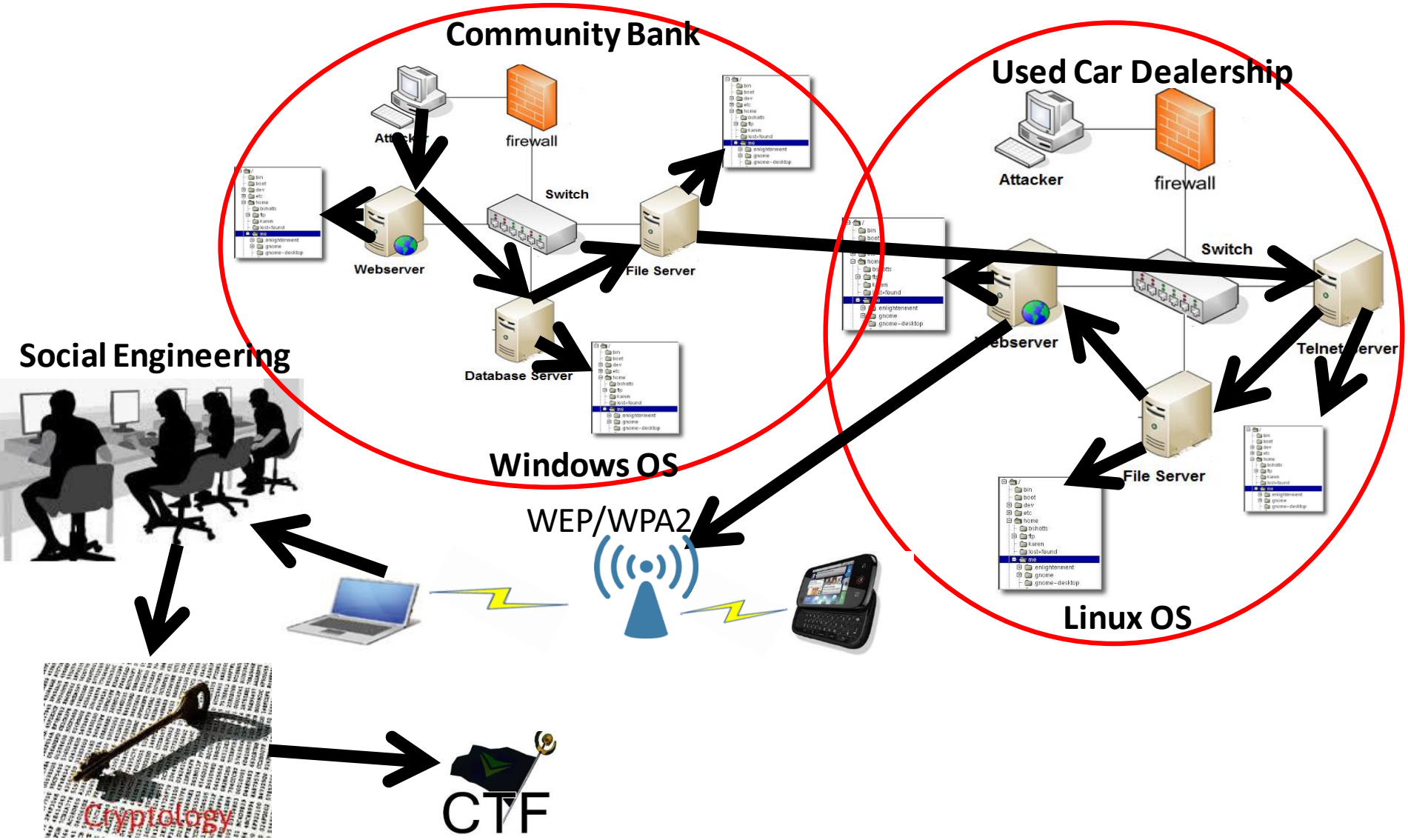
Lecture 9: Exploiting Cryptology (Hashes)

Lanier Watkins, PhD

Objectives

- To discuss the CTF class project
- To demonstrate and discuss the exploitation of cryptology (Hashes)
- To discuss CTF strategies and flag placement given the exploitation of cryptology

Kali Linux CTF Blueprints: Chapters 1 - 5



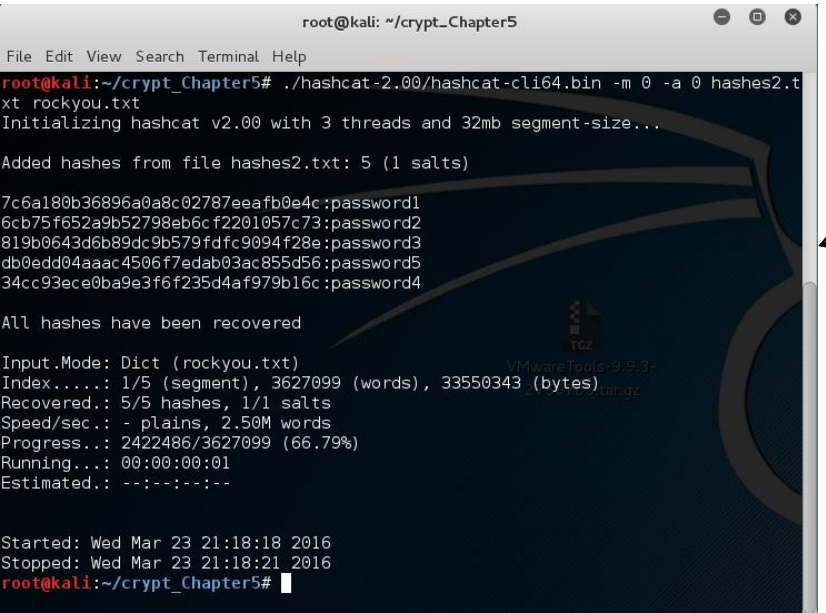
Class CTF Project

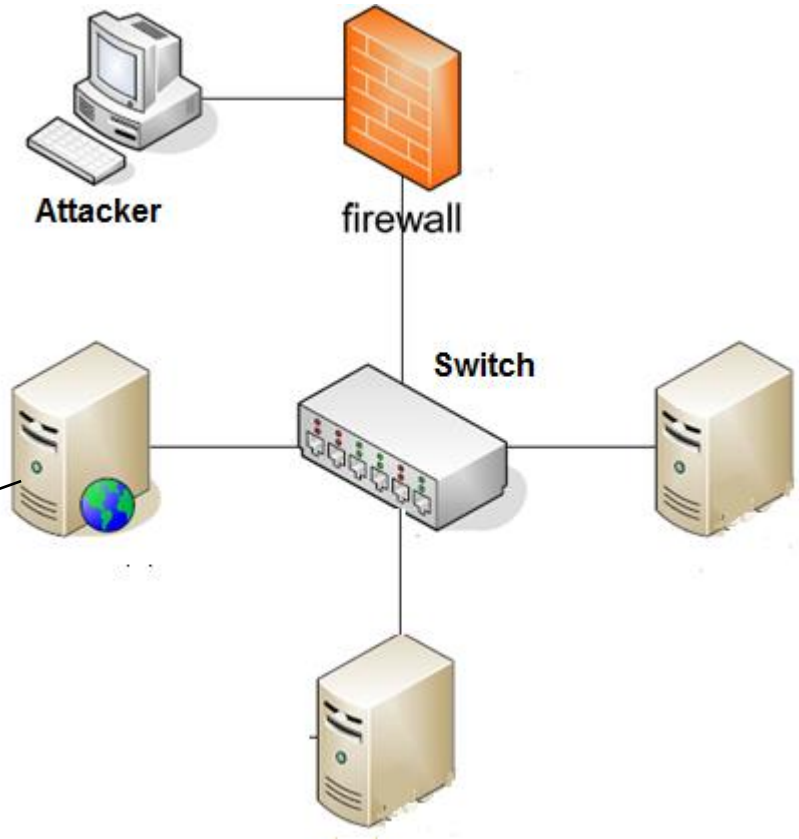
DUE 4/23, Send Walkthroughs to CA

- Must use:
  - At most 4 servers (must use minimum systems requirements)
  - More than one operating system type
  - Vulnerabilities (software/hardware) not discussed in class
  - At least 2 advanced topics (script writing)
    - Shell coding
    - Reverse engineering
    - Cryptology
  - At least 10 flags
  - Unique identifiers for flags
  - A storyline that is at least 4-6 hours long
    - Flags should build on each other like a story
  - Each team lead will maintain (only lead has write capability) a OneDrive repository for his/her team's production copy of the CTF VMs
    - Team member can download, modify, and send back to team lead to verify and write back to the Team's OneDrive repository
  - On Tournament Day
    - Each team lead will choose a number between 1 and 6 and that will be the team's CTF that team will work on
      - 2 Judges will be identified from each team, the judges will follow their CTF
      - Remaining team will split into 2 teams
    - There will be 12 CTF instances/subteams (6 tournaments) going on
  - Each judge will create a Zoom Call for their subteam and will allow their subteam to download the entire CTF (please allow this download prior to class)
  - Instructor will maintain class Zoom Call in case there are questions. CA and Instructor will move in-between Zoom Calls to observe ALL CTFs

Staging Vulnerabilities

- Attacking Hashing



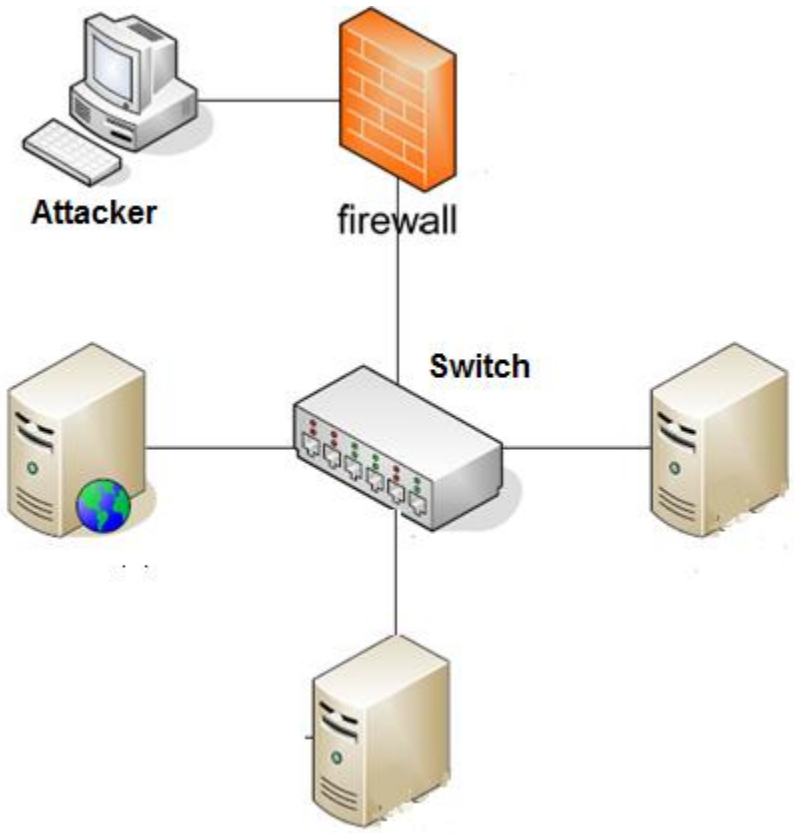


Kali Linux CTF Blueprints: Chapter 5

- Proof hashcat is installed

```
root@kali: ~/crypt_Chapter5
File  Edit  View  Search  Terminal  Help
root@kali:~/crypt_Chapter5# ./hashcat-2.00/hashcat-cli64.bin --help
hashcat, advanced password recovery

Usage: hashcat [options] hashfile [mask|wordfiles|directories]

=======
Options
=======

* General:

  -m,  --hash-type=NUM              Hash-type, see references below
  -a,  --attack-mode=NUM            Attack-mode, see references below
  -V,  --version                    Print version
  -h,  --help                       Print help
       --quiet                      Suppress output

* Benchmark:

  -b,  --benchmark                  Run benchmark

* Misc:

       --hex-salt                   Assume salt is given in hex
       --hex-charset                Assume charset is given in hex
       --runtime=NUM                Abort session after NUM seconds of runtime
       --status                     Enable automatic update of the status-screen
       --status-timer=NUM           Seconds between status-screen update
       --status-automat             Display the status view in a machine readable format

* Files:

  -o,  --outfile=FILE               Define outfile for recovered hash
       --outfile-format=NUM         Define outfile-format for recovered hash, see references
below
       --outfile-autohex-disable    Disable the use of $HEX[] in output plains
  -p,  --separator=CHAR             Define separator char for hashlists/outfile
       --show                       Show cracked passwords only (see --username)
       --left                       Show uncracked passwords only (see --username)
       --username                   Enable ignoring of usernames in hashfile (Recommended: al
so use --show)
       --remove                     Enable remove of hash once it is cracked
       --stdout                     Stdout mode
       --potfile-disable            Do not write potfile
```

Kali Linux CTF Blueprints: Chapter 5



Potential CTF Brief

- Hackers stole millions of unsalted hashes, and you have narrowed down 5 to the root password for 1 of the 3 servers in the used car dealer's network.
- Use the vulnerability of simple passwords to reverse the hashes to passwords. Then use the passwords to login to one of the servers as root to find the next flag.
- I hear one of the passwords matches with the username "Admin."

Setup Hashes and Wordlist

- Download hashcat if not installed
  - http://hashcat.net/files/hashcat-2.00.7z
- Download rockyou.txt wordlist (140 MB)
  - http://scrapmaker.com/data/wordlists/dictionaries/rockyou.txt
- Choose 5 easy passwords and put into file

```
root@kali: ~/crypt_Chapter5
File  Edit  View  Search  Terminal  Help
charset.lst    hashbruteforce.py  hashcat.pot  hashes.txt   wordlist.txt  XORsocket.py
commands.txt   hashcat-2.00       hashes2.txt  hash.py      XORdecode.py
encode.py      hashcat-2.00.7z    hashes3.txt  rockyou.txt  XOR.py
root@kali:~/crypt_Chapter5# cat commands.txt
echo -n "Borderland5!"|md5sum
crunch 12 12 -f charset.lst lalpha-numeric-symbol14 -t Borderla@@@@ > wordlist.txt
./hashcat-2.00/hashcat-cli64.bin -m 0 -a 0 hashes2.txt rockyou.txt
root@kali:~/crypt_Chapter5# echo -n "password1"|md5sum
7c6a180b36896a0a8c02787eeafb0e4c  -
root@kali:~/crypt_Chapter5# echo -n "password2"|md5sum
6cb75f652a9b52798eb6cf2201057c73  -
root@kali:~/crypt_Chapter5# echo -n "password3"|md5sum
819b0643d6b89dc9b579fdfc9094f28e  -
root@kali:~/crypt_Chapter5# echo -n "password4"|md5sum
34cc93ece0ba9e3f6f235d4af979b16c  -
root@kali:~/crypt_Chapter5# echo -n "password5"|md5sum
db0edd04aaac4506f7edab03ac855d56  -
root@kali:~/crypt_Chapter5# cat hashes2.txt
7c6a180b36896a0a8c02787eeafb0e4c
6cb75f652a9b52798eb6cf2201057c73
819b0643d6b89dc9b579fdfc9094f28e
34cc93ece0ba9e3f6f235d4af979b16c
db0edd04aaac4506f7edab03ac855d56
root@kali:~/crypt_Chapter5#
```

Break Hashes

- ./hashcat-2.00/hashcat-cli64.bin hashes2.txt rockyou.txt –m 0 –a 0

```
root@kali: ~/crypt_Chapter5
File  Edit  View  Search  Terminal  Help
root@kali:~/crypt_Chapter5# ./hashcat-2.00/hashcat-cli64.bin -m 0 -a 0 hashes2.t
xt rockyou.txt
Initializing hashcat v2.00 with 3 threads and 32mb segment-size...

Added hashes from file hashes2.txt: 5 (1 salts)

7c6a180b36896a0a8c02787eeafb0e4c:password1
6cb75f652a9b52798eb6cf2201057c73:password2
819b0643d6b89dc9b579fdfc9094f28e:password3
db0edd04aaac4506f7edab03ac855d56:password5
34cc93ece0ba9e3f6f235d4af979b16c:password4

All hashes have been recovered

Input.Mode: Dict (rockyou.txt)
Index.....: 1/5 (segment), 3627099 (words), 33550343 (bytes)
Recovered.: 5/5 hashes, 1/1 salts
Speed/sec.: - plains, 2.50M words
Progress..: 2422486/3627099 (66.79%)
Running...: 00:00:00:01
Estimated.: --:--:--:--


Started: Wed Mar 23 21:18:18 2016
Stopped: Wed Mar 23 21:18:21 2016
root@kali:~/crypt_Chapter5#
```

The operating system determines the hash used. You need to know the hash type.

Unix = MD5 hash

Kali = SHA512 hash

Windows XP = LM Hash

Windows 7 = NTLM Hash

-m 0 (Each number is a different Hash Type)

0 = MD5 hash.... so we use -m 0

50 = HMAC-MD5....so we use -m 50

1000 = NTLM....so we use -m 1000

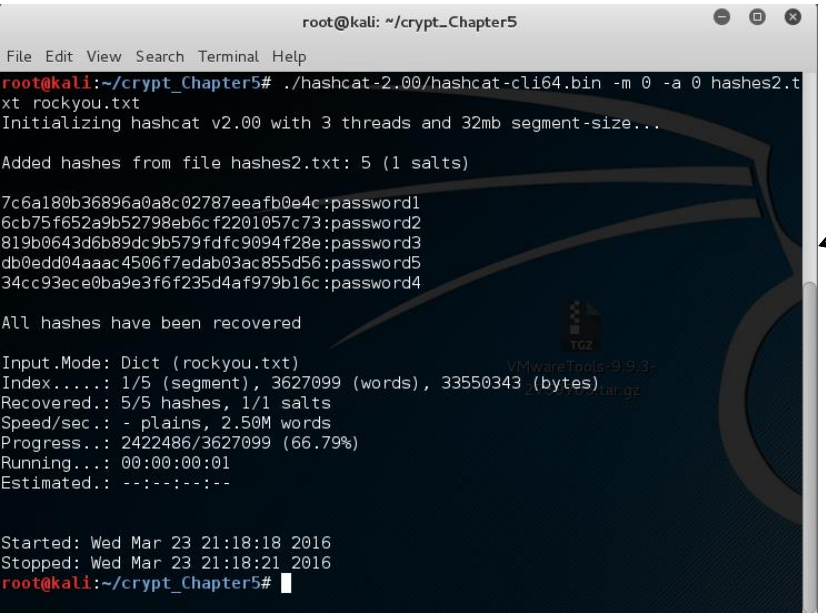
```
* Hash types:

    0 = MD5
   10 = md5($pass.$salt)
   20 = md5($salt.$pass)
   50 = HMAC-MD5 (key = $pass)
   60 = HMAC-MD5 (key = $salt)
  100 = SHA1
  110 = sha1($pass.$salt)
  120 = sha1($salt.$pass)
  150 = HMAC-SHA1 (key = $pass)
  160 = HMAC-SHA1 (key = $salt)
  200 = MySQL
  300 = MySQL4.1/MySQL5
  400 = phpass, MD5(Wordpress), MD5(phpBB3)
  500 = md5crypt, MD5(Unix), FreeBSD MD5, Cisco-IOS MD5
  800 = SHA-1(Django)
  900 = MD4
 1000 = NTLM
```

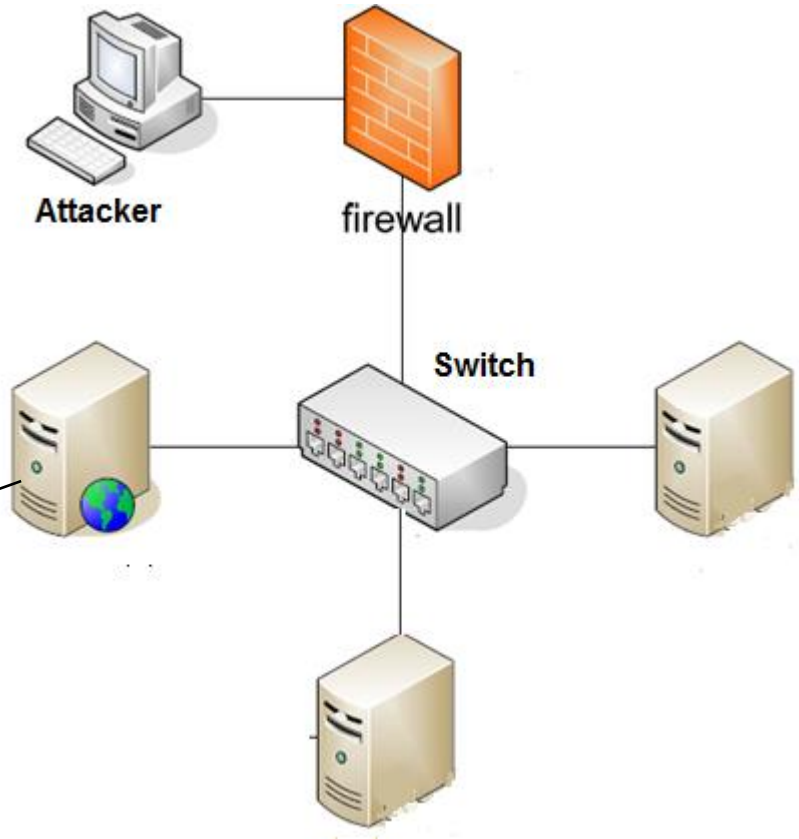
```
* Attack modes:

    0 = Straight
    1 = Combination
    2 = Toggle-Case
    3 = Brute-force
    4 = Permutation
    5 = Table-Lookup
```

I've found that straight or -a 0 is ridiculously fast on simple passwords.

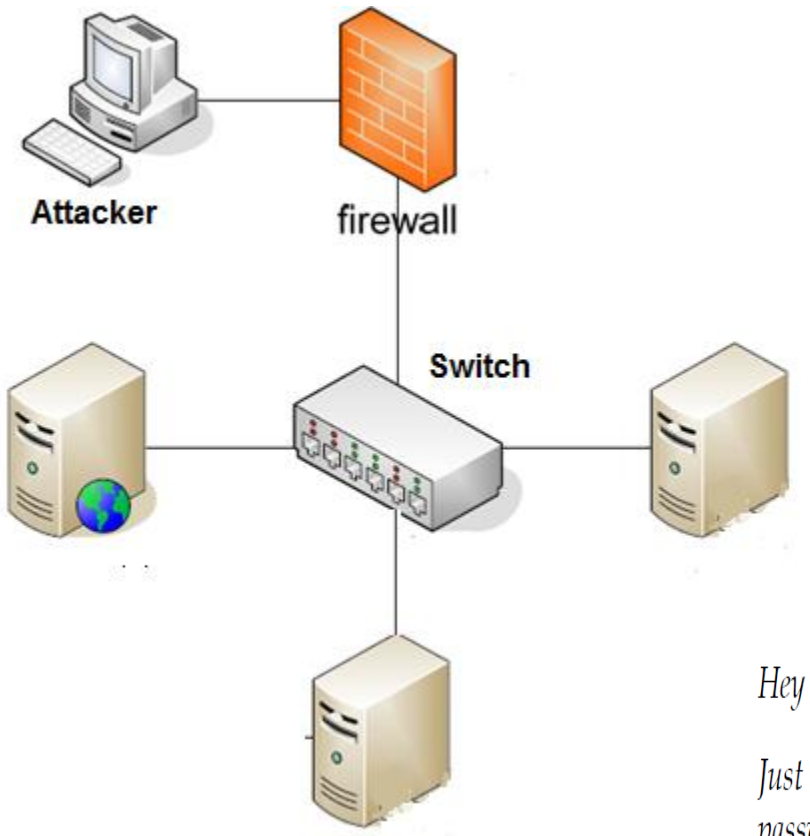
Staging Vulnerabilities

- Manually Attacking Hashing





Kali Linux CTF Blueprints: Chapter 5



Potential CTF Brief

- Given 8 of 12 digits of the password and the hash of the password, find the password for John's account
- Hint, see the note below from John's system administrator

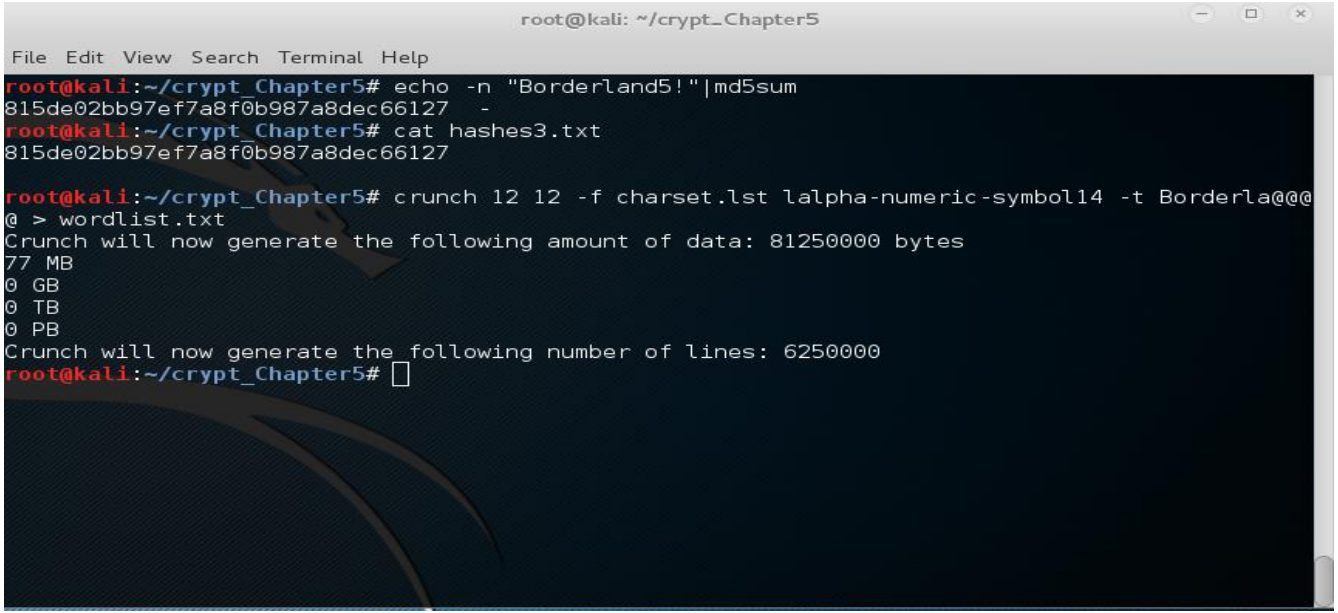
Hey John,

Just to remind you that the password policy has changed again to only allow 12 character passwords. Your previous one was too long so I had to change it for you; don't worry though, it still has the maximum length available. The new user account for IP 10.0.0.4 is HandsomeJack with the password Borderla[REDACTED].

Setup Hash and Wordlist

- Create md5 hash for password “Borderland5!” and put into a file
- Download crunch character set, charset.lst
  - https://raw.githubusercontent.com/jaalto/external-sf--crunch-wordlist/master/charset.lst
- Create bruteforce wordlist using crunch
  - Crunch 12 12 –f charset.lst lalpha-numeric-symbol14 –t Borderla@@@@ > wordlist.txt

```
lalpha                        = [abcdefghijklmnopqrstuvwxyz]
lalpha-space                  = [abcdefghijklmnopqrstuvwxyz ]
lalpha-numeric                = [abcdefghijklmnopqrstuvwxyz0123456789]
lalpha-numeric-space          = [abcdefghijklmnopqrstuvwxyz0123456789 ]
lalpha-numeric-symbol14       = [abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-_+=]
lalpha-numeric-symbol14-space = [abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-_+= ]
lalpha-numeric-all            = [abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/]
lalpha-numeric-all-space      = [abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/ ]
```



Break Hashes

- Straight (Dictionary Attack)
  - ./hashcat-2.00/hashcat-cli64.bin –m 0 –a 0 hashes3.txt wordlist.txt (Very Quick)
- Bruteforce
  - ./hashcat-2.00/hashcat-cli64.bin –m 0 –a 3 hashes3.txt wordlist.txt (Very Slow)

root@kali: ~/crypt_Chapter5

File Edit View Search Terminal Help

```
echo -n "Borderland5!"|md5sum
crunch 12 12 -f charset.lst lalpha-numeric-symbol14 -t Borderla@@@@ > wordlist.t
xt
./hashcat-2.00/hashcat-cli64.bin -m 0 -a 0 hashes2.txt rockyou.txt
root@kali:~/crypt_Chapter5# ./hashcat-2.00/hashcat-cli64.bin -m 0 -a 0 hashes3.t
xt wordlist.txt
Initializing hashcat v2.00 with 3 threads and 32mb segment-size...

Added hashes from file hashes3.txt: 1 (1 salts)
Activating quick-digest mode for single-hash

815de02bb97ef7a8f0b987a8dec66127:Borderland5!

All hashes have been recovered

Input.Mode: Dict (wordlist.txt)
Index.....: 1/3 (segment), 2580796 (words), 33550348 (bytes)
Recovered.: 1/1 hashes, 1/1 salts
Speed/sec.: - plains, 7.67M words
Progress..: 2494355/2580796 (96.65%)
Running...: 00:00:00:01
Estimated.: --:--:--:--


Started: Wed Mar 23 22:25:11 2016
Stopped: Wed Mar 23 22:25:16 2016
root@kali:~/crypt_Chapter5#
```

root@kali: ~/crypt_Chapter5

File Edit View Search Terminal Help

```
./hashcat-2.00/hashcat-cli64.bin -m 0 -a 0 hashes2.txt rockyou.txt
root@kali:~/crypt_Chapter5# ./hashcat-2.00/hashcat-cli64.bin -m 0 -a 3 hashes3.t
xt wordlist.txt
Initializing hashcat v2.00 with 3 threads and 32mb segment-size...

Added hashes from file hashes3.txt: 1 (1 salts)
Activating quick-digest mode for single-hash

[s]tatus [p]ause [r]esume [b]ypass [q]uit => r


Input.Mode: Mask (Borderlaaaaa) [12] (0.00%)
Index.....: 0/1 (segment), 1 (words), 0 (bytes)
Recovered.: 0/1 hashes, 0/1 salts
Speed/sec.: - plains, - words
Progress..: 1/1 (100.00%)
Running...: --:--:--:--
Estimated.: --:--:--:--


[s]tatus [p]ause [r]esume [b]ypass [q]uit =>

Input.Mode: Mask (Borderlaaaab) [12] (0.00%)
Index.....: 0/1 (segment), 1 (words), 0 (bytes)
Recovered.: 0/1 hashes, 0/1 salts
Speed/sec.: - plains, - words
Progress..: 1/1 (100.00%)
Running...: --:--:--:--
Estimated.: --:--:--:--
```

Kali Linux CTF Blueprints: Chapters 1 - 5