Indian Institute of Technology Jodhpur

CSL6010 – Cyber Security

LAB 8 Report

Name- Aman Srivastava
Roll No.- B20CS100

Date-10th Apr 2023

AIM: Understanding the use of password brute forcing tools.

1) DaveGrohl:

DaveGrohl is a brute-force password cracker for macOS.

```
MacBook-Pro:~$ git clone https://github.com/octomagon/davegrohl.git
```

```
MacBook-Pro:~$ cd davegrohl
MacBook-Pro:~/davegrohl$ make

*** A bunch of stuff happens here ***

Make succeeded...
Created: dave
MacBook-Pro:~/davegrohl$
```

To execute Dave, press the spacebar to view the status of each task. Dave's threads that perform dictionary attacks will be enclosed in parentheses, while those that perform incremental attacks will be enclosed in square brackets. Dave will search for plain text files in a folder named 'wordlists' for dictionary attacks, and it will dedicate a separate thread for each file it finds.

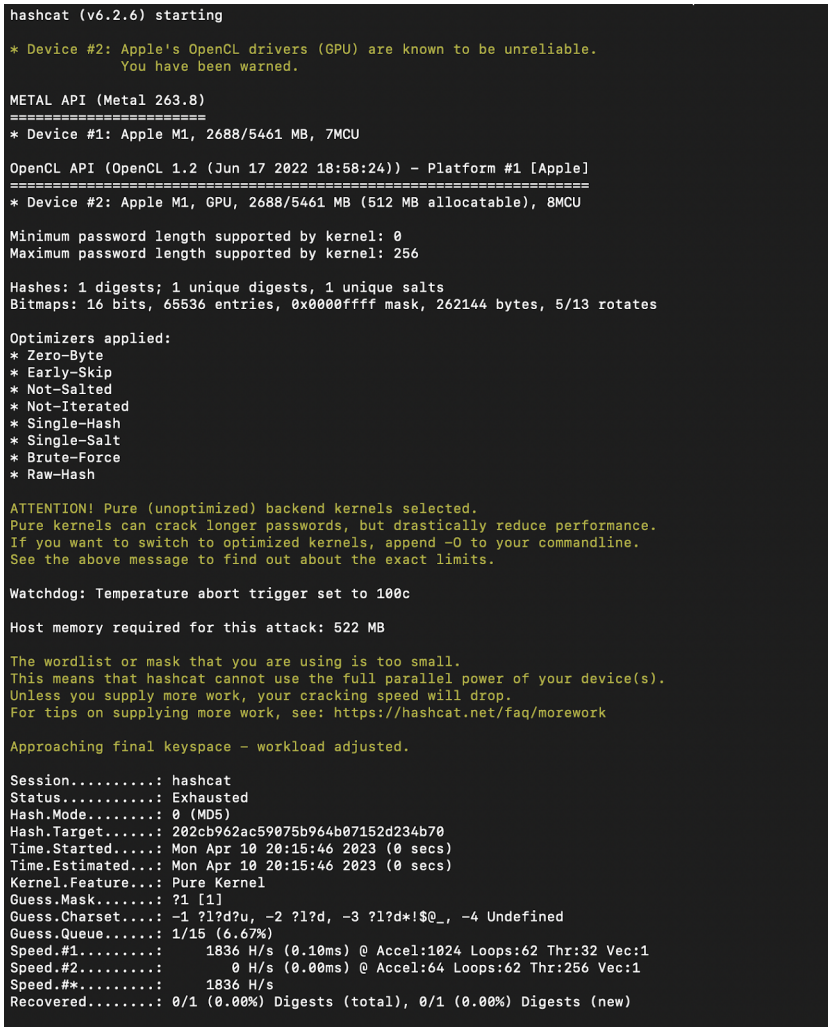
```
MacBook-Pro:~/davegrohl$ sudo ./dave -u someuser
-- Loaded PBKDF2 (Salted SHA512) hash...
-- Starting attack

     TIME            GUESSES
0000:00:08               351  (aaru) (loveme) [x] [86n] [bpc] [2s5] [ojf] [wkea] [521a] [caha]
0000:00:14               613  (abaculus) (samantha) [9a] [38n] [t3c] [an5] [yjf] [k4ea] [dmla] [ieha]
0000:00:20               875  (abandoning) (spongebob) [pe] [n7n] [x3c] [8n5] [bvf] [25ea] [odla] [weha]

-- Found password : 'shorty'
-- (dictionary attack)

Finished in 31.330 seconds / 1,318 guesses...
42 guesses per second.
```

In this scenario, I intentionally selected a password that I knew Dave could easily guess. However, when utilizing PBKDF2, the operating system (OS X) slows down the password hashing process to enhance password security but makes it less feasible for brute force attacks.

2) Hashcat:

It is now possible to utilize Hashcat on a Mac OS to crack passwords via the terminal. To do this, a hash.txt file was created on the Desktop, containing the MD5 hash of a chosen password, which in this case was "123". The corresponding MD5 hash for "123" is "202cb962ac59075b964b07152d234b70".

To indicate the attack mode and hash type in Hashcat, one can use the options -a and -m, respectively.



```
Time.Started.....: Mon Apr 10 20:15:46 2023 (0 secs)
Time.Estimated...: Mon Apr 10 20:15:46 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: ?1 [1]
Guess.Charset....: -1 ?l?d?u, -2 ?l?d, -3 ?l?d*!$@_, -4 Undefined
Guess.Queue......: 1/15 (6.67%)
Speed.#1.........:     1836 H/s (0.10ms) @ Accel:1024 Loops:62 Thr:32 Vec:1
Speed.#2.........:        0 H/s (0.00ms) @ Accel:64 Loops:62 Thr:256 Vec:1
Speed.#*.........:     1836 H/s
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 62/62 (100.00%)
Rejected.........: 0/62 (0.00%)
Restore.Point....: 0/1 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-62 Iteration:0-62
Restore.Sub.#2...: Salt:0 Amplifier:0-0 Iteration:0-62
Candidate.Engine.: Device Generator
Candidates.#1....: s -> X
Candidates.#2....: [Generating]
Hardware.Mon.#1..: Util: 57%
Hardware.Mon.#2..: Util:  0%

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

Session..........: hashcat
Status...........: Exhausted
Hash.Mode........: 0 (MD5)
Hash.Target......: 202cb962ac59075b964b07152d234b70
Time.Started.....: Mon Apr 10 20:15:46 2023 (0 secs)
Time.Estimated...: Mon Apr 10 20:15:46 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: ?1?2 [2]
Guess.Charset....: -1 ?l?d?u, -2 ?l?d, -3 ?l?d*!$@_, -4 Undefined
Guess.Queue......: 2/15 (13.33%)
Speed.#1.........:  1494.0 kH/s (0.10ms) @ Accel:512 Loops:62 Thr:32 Vec:1
Speed.#2.........:        0 H/s (0.00ms) @ Accel:512 Loops:62 Thr:32 Vec:1
Speed.#*.........:  1494.0 kH/s
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 2232/2232 (100.00%)
Rejected.........: 0/2232 (0.00%)
Restore.Point....: 0/36 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-62 Iteration:0-62
Restore.Sub.#2...: Salt:0 Amplifier:0-0 Iteration:0-62
Candidate.Engine.: Device Generator
Candidates.#1....: sa -> Xq
Candidates.#2....: [Generating]
Hardware.Mon.#1..: Util: 93%
Hardware.Mon.#2..: Util:  0%

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.
```

Password cracking:

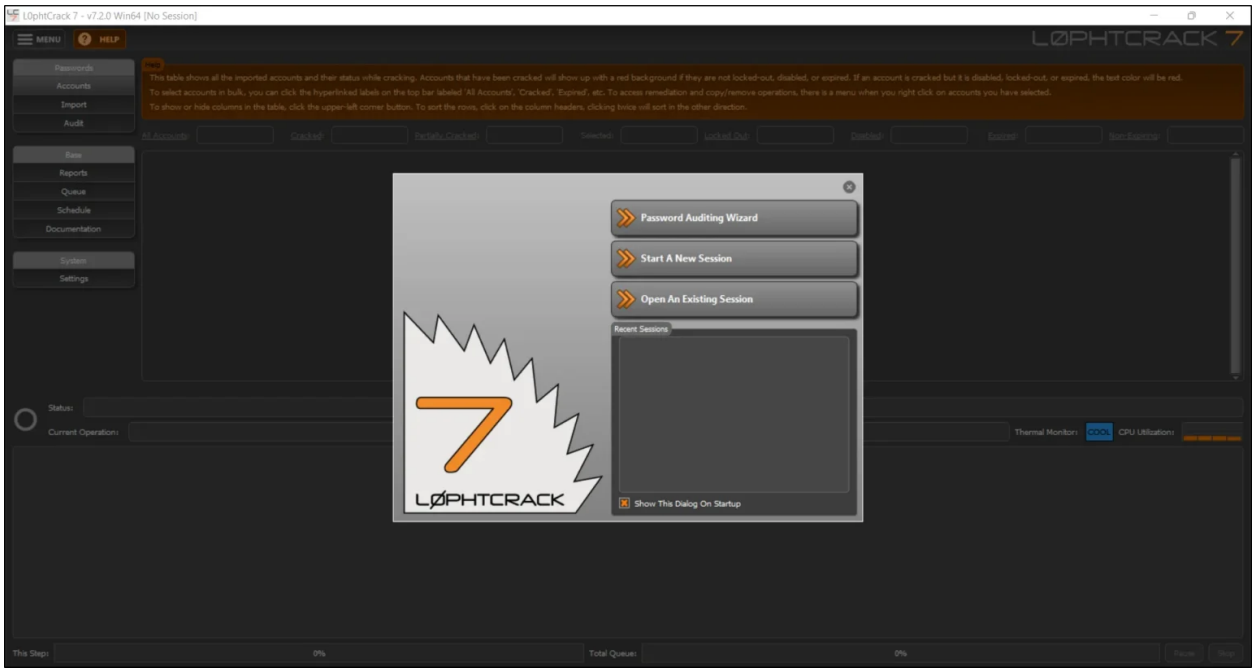
```
202cb962ac59075b964b07152d234b70:123

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 202cb962ac59075b964b07152d234b70
Time.Started.....: Mon Apr 10 20:15:46 2023 (0 secs)
Time.Estimated...: Mon Apr 10 20:15:46 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: ?1?2?2 [3]
Guess.Charset....: -1 ?l?d?u, -2 ?l?d, -3 ?l?d*!$@_, -4 Undefined
Guess.Queue......: 3/15 (20.00%)
Speed.#1.........:  7474.7 kH/s (0.10ms) @ Accel:512 Loops:62 Thr:32 Vec:1
Speed.#2.........: 24324.9 kH/s (0.00ms) @ Accel:256 Loops:62 Thr:64 Vec:1
Speed.#*.........: 31799.6 kH/s
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 77376/80352 (96.30%)
Rejected.........: 0/77376 (0.00%)
Restore.Point....: 0/1296 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-62 Iteration:0-62
Restore.Sub.#2...: Salt:0 Amplifier:0-62 Iteration:0-62
Candidate.Engine.: Device Generator
Candidates.#1....: sar -> Xhy
Candidates.#2....: s4j -> Xcx
Hardware.Mon.#1..: Util: 96%
Hardware.Mon.#2..: Util:  0%
```

Hence the password is cracked successfully.

3)L0phtcrack

L0phtcrack is one of the powerful password cracking tools and even today the tool is very effective.

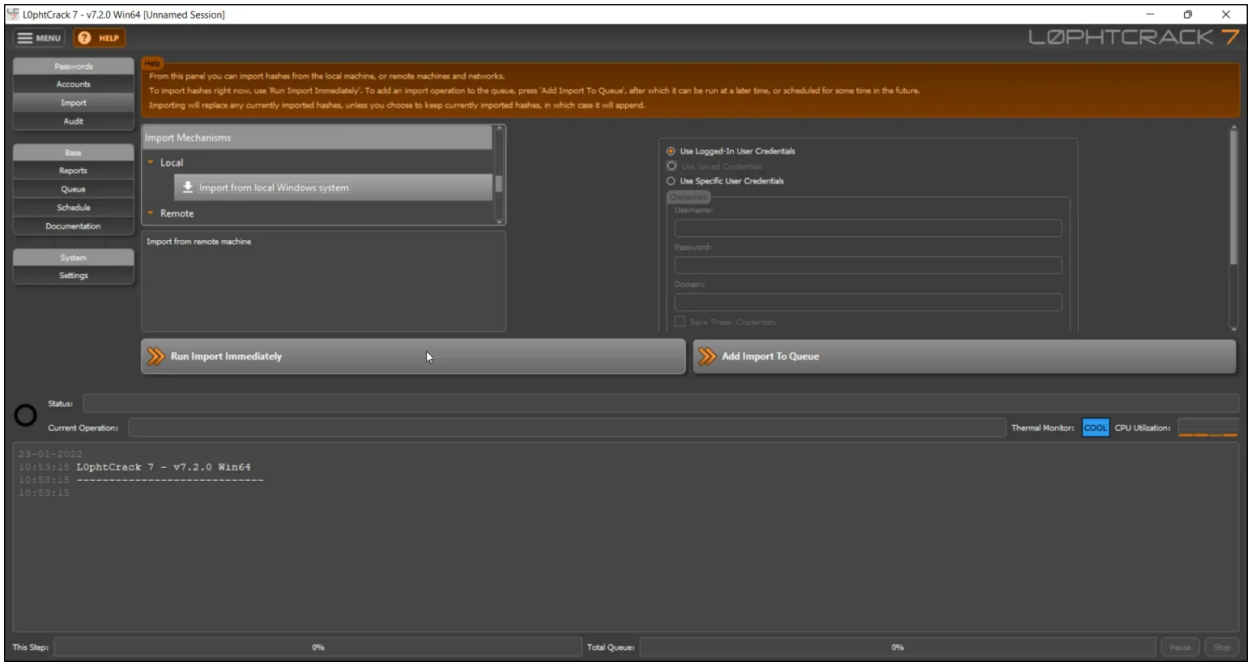
L0phtcrack stands out from other password cracking tools due to its user-friendly Graphical User Interface (GUI), which makes it much easier to use.



Upon launching L0phtcrack, users are presented with an interface that offers three starting options:

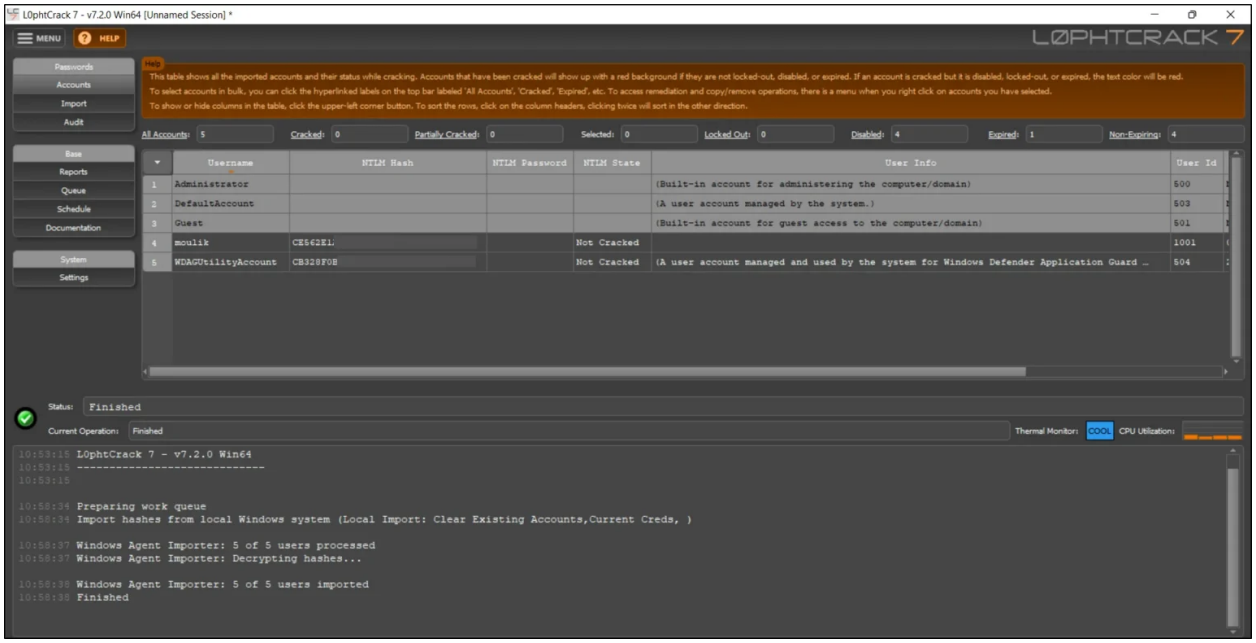
- Auditing Password,
- Start New Session
- Open Existing Session.

The Auditing Password option is utilized to identify system passwords on one's own system. On the other hand, the Start New Session option functions like creating a new project.

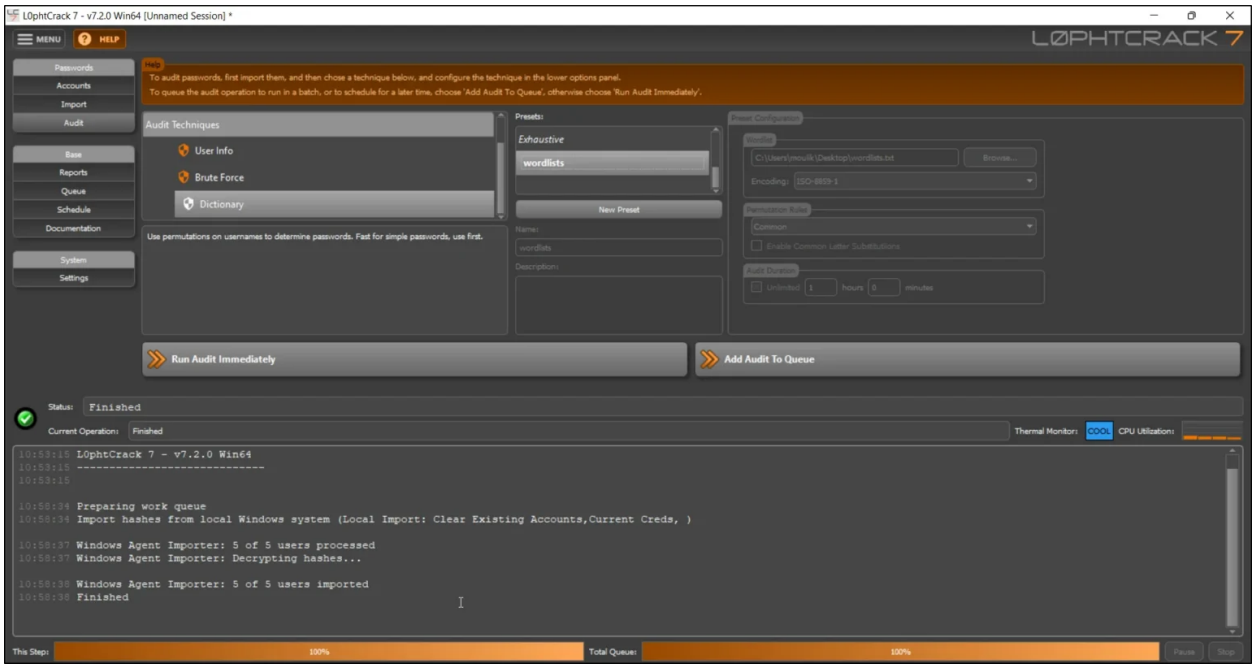


Went to the import section and chosed a local SAM file.

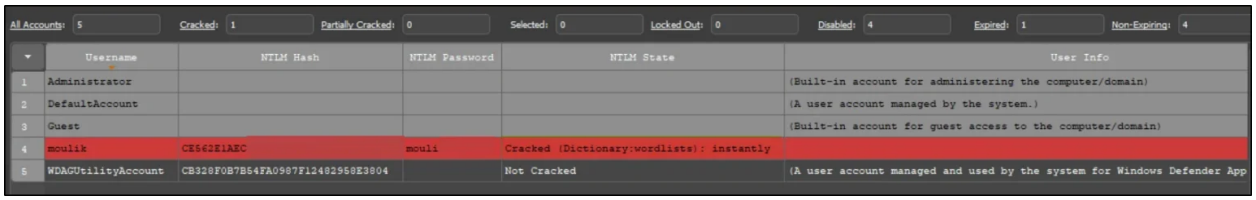
It will be redirected to the dashboard that is the accounts and we can see all the hash have been imported.



To crack the hash, I will be choosing a dictionary-based attack.



Clicking on the "Run Audit Immediately" option initiates the password cracking process, and once it is activated, users are redirected to the accounts dashboard. From there, it is possible to see that the Windows password has been successfully cracked.



	Username	NTLM Hash	NTLM Password	NTLM State	User Info
1	Administrator				(Built-in account for administering the computer/domain)
2	DefaultAccount				(A user account managed by the system.)
3	Guest				(Built-in account for guest access to the computer/domain)
4	moulik	CE562E1AEC	mouli	Cracked (Dictionary:wordlists): instantly	
5	WDAGUtilityAccount	CB328F0B7B54FA0987F12482958E3804		Not Cracked	(A user account managed and used by the system for Windows Defender App

Hence we cracked the windows password using L0phtcrack

4) Ncrack:

To check for any potential vulnerabilities in the form of weak or commonly used passwords, a dictionary-based brute force attack is performed using ncrack on the Linux root credentials that are being utilized for SSH on a Linux virtual machine. To carry out this attack, a list of frequently used usernames is loaded from usernames.txt, while a default list of common passwords is loaded from passwords.txt. The goal is to determine if any security weaknesses can be exploited using this method.

The following is used to start with ncrack.

```
-VirtualBox:~$ ncrack -U usernames.txt -P passwords.txt ssh://10.0.2.18
```

```
Starting Ncrack 0.7 ( http://ncrack.org ) at 2023-04-10 17:11 IST

Discovered credentials for ssh on 172.31.27.222 22/tcp:
172.31.27.222 22/tcp ssh: 'admin' '123'

Ncrack done: 1 service scanned in 3.00 seconds.

Ncrack finished.
```

We can see ncrack successfully find the password of user ‘admin’.

5) Aircrack-ng:

Step 1: Put Wi-Fi Adapter in Monitor Mode with Airmon-Ng.

The command used is “airmon-ng start wlan0”

Step 2: Capture Traffic with Airodump-Ng

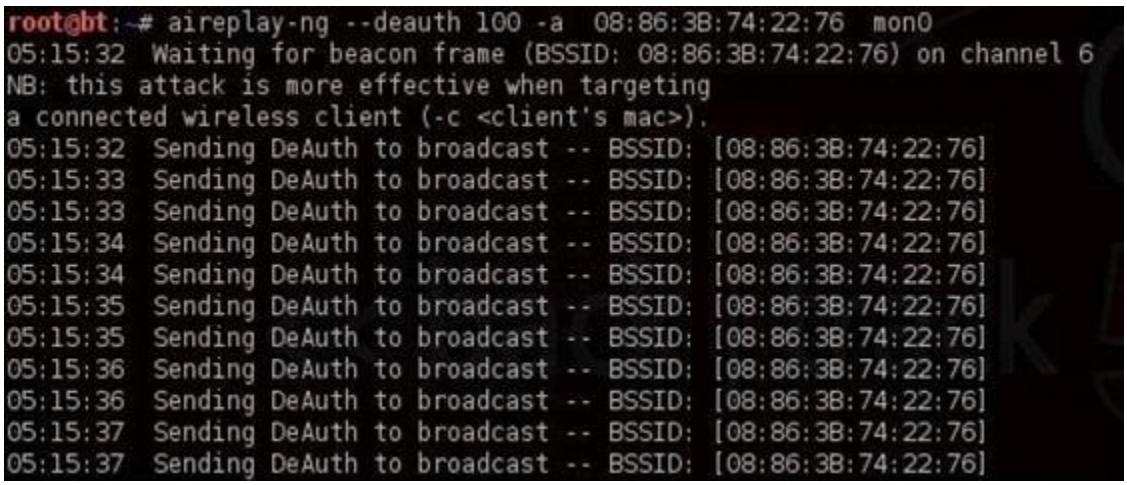
The command used is “airodump-ng mon0”

Step 3: Focus Airodump-Ng on One AP on One Channel

In the other terminal we have to write, “airodump-ng --bssid 08:86:30:74:22:76 -c 6 --write WPAcrack mon0”.

- 08:86:30:74:22:76 is the BSSID of the AP
- -c 6 is the channel the AP is operating on
- WPAcrack is the file you want to write to
- mon0 is the monitoring wireless adapter*

Step 4: Aireplay-Ng Deauth



- 100 is the number of de-authenticate frames you want to send
- 08:86:30:74:22:76 is the BSSID of the AP
- mon0 is the monitoring wireless adapter

Step 5: Capture the Handshake.

airodump-ng will attempt to grab their password in the new 4-way handshake.

Step 6: Crack the password using Aircrack-ng.

we have the encrypted password in our file WPAcrack, we can run that file against aircrack-ng using a password file of our choice.

This is achieved by opening another terminal and writing “aircrack-ng WPAcrack-01.cap -w /pentest/passwords/wordlists/darkc0de”.

6) John the Ripper:

Step 1: Installing from the source.

Now configuring the downloaded file.

```
kali@kali:~/src/john/src$ ./configure
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking whether to compile using MPI... no
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking whether we are using the GNU C compiler... (cached) yes
checking whether gcc accepts -g... (cached) yes
checking for gcc option to accept ISO C89... (cached) none needed
checking whether gcc understands -c and -o together... (cached) yes
checking additional paths...  -L/usr/local/lib -I/usr/local/include
checking arg check macro for -m with gcc... yes
```

Run the make command to compile source code into executable programs and libraries. This might take some time depending on your machine and the resources allocated to it.

```
Make process completed.
kali@kali:~/src/john/src$ make install
make find_version
make[1]: Entering directory '/home/kali/src/john/src'
echo "#define JTR_GIT_VERSION JUMBO_VERSION "\"-e791b84e0\" \" 2021-07-12 15:50:15 -0300\"" > version.h.new
diff >/dev/null 2>/dev/null version.h.new version.h && rm -f version.h.new || mv -f version.h.new version.h
make[1]: Leaving directory '/home/kali/src/john/src'
make[1]: Entering directory '/home/kali/src/john/src'
echo "#define JTR_GIT_VERSION JUMBO_VERSION "\"-e791b84e0\" \" 2021-07-12 15:50:15 -0300\"" > version.h.new
diff >/dev/null 2>/dev/null version.h.new version.h && rm -f version.h.new || mv -f version.h.new version.h
make[1]: '../run/unshadow' is up to date.
make[1]: '../run/unafs' is up to date.
make[1]: '../run/unique' is up to date.
make[1]: '../run/undrop' is up to date.
make[1]: '../run/rar2john' is up to date.
make[1]: '../run/zip2john' is up to date.
make[1]: '../run/genmkvpwd' is up to date.
make[1]: '../run/mkvcalcproba' is up to date.
make[1]: '../run/calc_stat' is up to date.
make[1]: '../run/tgtsnarf' is up to date.
make[1]: '../run/racf2john' is up to date.
make[1]: '../run/hccap2john' is up to date.
make[1]: '../run/raw2dyna' is up to date.
make[1]: '../run/keepass2john' is up to date.
```

Step 2: Cracking a zip/rar password protected file.

```
zip2john Test.zip
```

```
zip2john Test.zip > hash.txt
```

To begin the attack on the zip file below command is executed.

```
john --format=zip hash.txt
```

Hence, John the Ripper will first identify the hash method and display it on the terminal. It then decodes the password hash into a raw password and displays it as well.

Comparison between them:

we will compare and contrast the features and capabilities of the following password cracking tools: Aircrack-ng, John the Ripper, L0phtCrack, Hashcat, DaveGrohl, and Ncrack. We will analyze the strengths and weaknesses of each tool based on various factors, including the type of attack, the type of password that can be guessed, the platform, and the time taken to crack a password.

Aircrack-ng:

Aircrack-ng is a powerful tool for wireless network security auditing. It can crack WEP and WPA-PSK passwords through a variety of attacks. Aircrack-ng is available for Linux, Windows, and Mac OS. The time taken to crack a password varies depending on the complexity of the password and the type of attack used.

John the Ripper:

John the Ripper is a popular password cracking tool that can be used to crack passwords from various operating systems, including Unix, Linux, Windows, and macOS. John the Ripper supports several cracking modes, including dictionary attacks, brute force attacks, and hybrid attacks. It can also crack passwords encrypted with several hashing algorithms. John the Ripper is an excellent tool for offline password cracking.

L0phtCrack:

L0phtCrack is a Windows-based password cracking tool that can crack passwords stored in various formats, including Windows NTLM, Active Directory, and UNIX/Linux. It uses several cracking methods, including dictionary attacks, brute force attacks, and hybrid attacks. L0phtCrack also provides a user-friendly interface and comprehensive reporting capabilities.

Hashcat:

Hashcat is a powerful password cracking tool that can crack passwords from various operating systems and applications, including Windows, macOS, Unix, and Android. It supports several cracking modes, including dictionary attacks, brute force attacks, and mask attacks. Hashcat can

also crack passwords encrypted with several hashing algorithms. It is one of the fastest password cracking tools available, and it can run on various hardware, including GPUs.

DaveGrohl:

DaveGrohl is not a password cracking tool. Instead, it is a famous musician and the lead vocalist of the Foo Fighters.

Ncrack:

Ncrack is a network authentication cracking tool that can be used to crack passwords from various network services, including SSH, RDP, FTP, and Telnet. It supports several cracking modes, including dictionary attacks, brute force attacks, and hybrid attacks. Ncrack is available for Linux, Windows, and macOS. It is an excellent tool for network security auditing.

Conclusion:

In conclusion, each of the password cracking tools we have analyzed has its strengths and weaknesses. Aircrack-ng is an excellent tool for wireless network security auditing, John the Ripper is a versatile tool for offline password cracking, L0phtCrack is a user-friendly tool for Windows password cracking, Hashcat is one of the fastest password cracking tools available, and Ncrack is an excellent tool for network security auditing. When choosing a password cracking tool, it is essential to consider the specific requirements of your project and select the tool that best meets those requirements.