

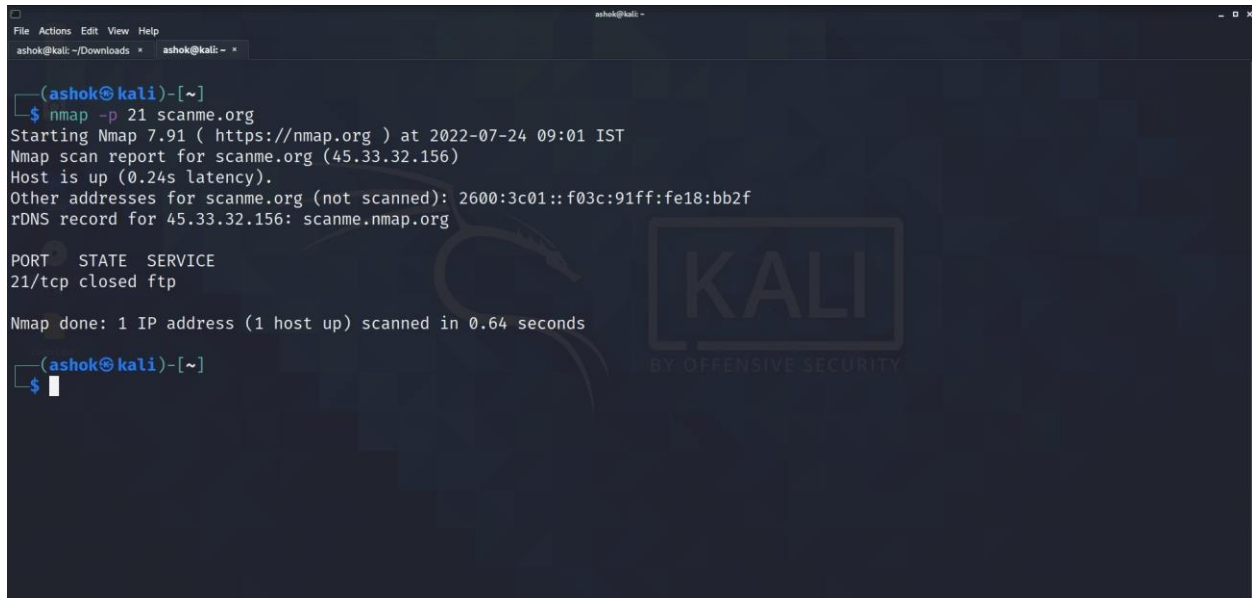
Experiment 1: Perform an Experiment for port scanning with NMAP.

Aim: To remotely testing numerous ports to determine what state they are in

Nmap is a network scanner utility used for port mapping, host discovery and vulnerability scanning. Most of its functions are based on using IP packet analysis to detect and identify remote hosts, operating systems and services.

Step 1: Port scan for port 21

Command: \$ nmap -p 21 scanme.org



```
File Actions Edit View Help
ashok@kali: ~/Downloads * ashok@kali: ~ *

(ashok@kali)~[~]
$ nmap -p 21 scanme.org
Starting Nmap 7.91 ( https://nmap.org ) at 2022-07-24 09:01 IST
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.24s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org

PORT      STATE SERVICE
21/tcp    closed ftp

Nmap done: 1 IP address (1 host up) scanned in 0.64 seconds

(ashok@kali)~[~]
$
```

Step 2: Port scan for port range

Command: \$ nmap -p 21-100 scanme.org

```
File Actions Edit View Help
ashok@kali: ~/Downloads x ashok@kali: ~ x

(ashok@kali)-[~]
$ nmap -p 21-100 scanme.org
Starting Nmap 7.91 ( https://nmap.org ) at 2022-07-24 09:07 IST
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.27s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org
Not shown: 76 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 2.40 seconds

(ashok@kali)-[~]
$
```

Step 3: Port scan for multiple TCP and UDP ports

Command: `$ nmap -p U:53, T:21-25,80 scanme.org`

```
File Actions Edit View Help
ashok@kali: ~/Downloads x ashok@kali: ~ x

(ashok@kali)-[~]
$ nmap -p U:53,T:21-25,80 scanme.org
Starting Nmap 7.91 ( https://nmap.org ) at 2022-07-24 10:14 IST
WARNING: Your ports include "U:" but you haven't specified UDP scan with -sU.
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.23s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org

PORT      STATE SERVICE
21/tcp    closed ftp
22/tcp    open  ssh
23/tcp    closed telnet
24/tcp    closed priv-mail
25/tcp    open  smtp
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 1.80 seconds

(ashok@kali)-[~]
$
```

Step 4: Port scan for all ports

Command: `$ nmap -p- example.com`

```
File Actions Edit View Help
ashok@kali: ~/Downloads x ashok@kali: ~ x

(ashok@kali)-[~]
$ nmap -p- example.com
Starting Nmap 7.91 ( https://nmap.org ) at 2022-07-24 10:42 IST
Nmap scan report for example.com (93.184.216.34)
Host is up (0.23s latency).
Other addresses for example.com (not scanned): 2606:2800:220:1:248:1893:25c8:1946
Not shown: 65529 filtered ports
PORT      STATE SERVICE
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
1119/tcp  closed bnetgame
1935/tcp  closed rtmp

Nmap done: 1 IP address (1 host up) scanned in 416.35 seconds

(ashok@kali)-[~]
$
```

Step 5: Port scan for service name

Command: # nmap -p http, https scanme.org

```
File Actions Edit View Help
ashok@kali: ~/Downloads x ashok@kali: ~ x

(ashok@kali)-[~]
$ nmap -p http,https scanme.org
Starting Nmap 7.91 ( https://nmap.org ) at 2022-07-24 13:00 IST
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.24s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org

PORT      STATE SERVICE
80/tcp    open  http
443/tcp   closed https
8008/tcp  closed http

Nmap done: 1 IP address (1 host up) scanned in 0.63 seconds

(ashok@kali)-[~]
$
```

Step 6: Fast port scan (100)

Command: \$ nmap -F scanme.org

```
File Actions Edit View Help
ashok@kali: ~/Downloads x ashok@kali: ~ x

(ashok@kali)~[~]
$ nmap -F scanme.org
Starting Nmap 7.91 ( https://nmap.org ) at 2022-07-24 13:06 IST
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.26s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org
Not shown: 93 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 3.74 seconds

(ashok@kali)~[~]
$
```

SCAN TECHNIQUES:

Step 1: TCP SYN port scan

Command: \$ nmap -sS scanme.org

```
File Actions Edit View Help
ashok@kali: ~/Downloads x ashok@kali: ~ x

(ashok@kali)~[~]
$ sudo nmap -sS scanme.org
[sudo] password for ashok:
Starting Nmap 7.91 ( https://nmap.org ) at 2022-07-24 13:15 IST
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.25s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org
Not shown: 989 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
593/tcp   filtered http-rpc-epmap
4444/tcp  filtered krb524
9929/tcp  open  nping-echo
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 12.28 seconds
```

Step 2: TCP Connect port scan (without root privileges)

Command: \$ nmap -sT scanme.org

```
ashok@kali: ~/Downloads x ashok@kali: ~ x
(ashok@kali)-[~]
$ nmap -sT scanme.org
Starting Nmap 7.91 ( https://nmap.org ) at 2022-07-24 13:20 IST
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.25s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org
Not shown: 988 closed ports
PORT      STATE      SERVICE
22/tcp    open      ssh
25/tcp    open      smtp
53/tcp    open      domain
80/tcp    open      http
135/tcp   filtered  msrpc
139/tcp   filtered  netbios-ssn
445/tcp   filtered  microsoft-ds
593/tcp   filtered  http-rpc-epmap
4444/tcp  filtered  krb524
9929/tcp  open      nping-echo
30951/tcp filtered  unknown
31337/tcp open      Elite

Nmap done: 1 IP address (1 host up) scanned in 13.08 seconds
```

Step 3: TCPACK port scan

\$ nmap -sA scanme.org

```
ashok@kali: ~/Downloads x ashok@kali: ~ x
(ashok@kali)-[~]
$ sudo nmap -sA scanme.org
Starting Nmap 7.91 ( https://nmap.org ) at 2022-07-24 13:44 IST
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.27s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org
Not shown: 995 unfiltered ports
PORT      STATE      SERVICE
135/tcp   filtered  msrpc
139/tcp   filtered  netbios-ssn
445/tcp   filtered  microsoft-ds
593/tcp   filtered  http-rpc-epmap
4444/tcp  filtered  krb524

Nmap done: 1 IP address (1 host up) scanned in 9.28 seconds

(ashok@kali)-[~]
$
```

Step 4: TCP window port scan

Command: \$ nmap -w scanme.org

```
File Actions Edit View Help
ashok@kali: ~/Downloads x ashok@kali: ~ x
(ashok@kali)-[~]
$ sudo nmap -w scanme.org
Starting Nmap 7.91 ( https://nmap.org ) at 2022-07-24 13:46 IST
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.26s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org
Not shown: 989 closed ports
PORT      STATE      SERVICE
22/tcp    open      ssh
25/tcp    open      smtp
53/tcp    open      domain
80/tcp    open      http
135/tcp   filtered  msrpc
139/tcp   filtered  netbios-ssn
445/tcp   filtered  microsoft-ds
593/tcp   filtered  http-rpc-epmap
4444/tcp   filtered  krb524
9929/tcp  open      nping-echo
31337/tcp open      Elite

Nmap done: 1 IP address (1 host up) scanned in 13.17 seconds

(ashok@kali)-[~]
$
```

Experiment 2: Set up a honeypot and monitor the honeypot on the network

Aim: To lure cyber attackers and detect, deflect and study hacking attempts to gain unauthorized access to information systems.

```
ashok@kali: ~/Downloads * ashok@kali: ~/pentbox/pentbox-1.8 *
(ashok@kali)-[~/pentbox/pentbox-1.8]
└─$ sudo ./pentbox.rb
[sudo] password for ashok:

PentBox 1.8
  (oo)
  ( )  )--*
  ||--||

Menu      ruby2.7.2 @ x86_64-linux-gnu

1- Cryptography tools
2- Network tools
3- Web
4- Ip grabber
5- Geolocation ip
6- Mass attack
7- License and contact
8- Exit

→ =
```

```
ashok@kali: ~/pentbox/pentbox-1.8
(ashok@kali)-[~/pentbox/pentbox-1.8]
$ sudo ./pentbox.rb
[sudo] password for ashok:

PentBox 1.8
  (oo)
  ( )  )--*
  ||--||

Menu
ruby2.7.2 @ x86_64-linux-gnu

1- Cryptography tools
2- Network tools
3- Web
4- Ip grabber
5- Geolocation ip
6- Mass attack
7- License and contact
8- Exit

→ 2
```

```
ashok@kali: ~/pentbox/pentbox-1.8
Menu
ruby2.7.2 @ x86_64-linux-gnu

1- Cryptography tools
2- Network tools
3- Web
4- Ip grabber
5- Geolocation ip
6- Mass attack
7- License and contact
8- Exit

→ 2

1- Net DoS Tester
2- TCP port scanner
3- Honeypot
4- Fuzzer
5- DNS and host gathering
6- MAC address geolocation (samy.pl)
0- Back

→ 3
```



```
ashok@kali: ~/pentbox/pentbox-1.8
File Actions Edit View Help
ashok@kali: ~/Downloads * ashok@kali: ~/pentbox/pentbox-1.8 *

6- Mass attack
7- License and contact
8- Exit
  → 2
1- Net DoS Tester
2- TCP port scanner
3- Honeypot
4- Fuzzer
5- DNS and host gathering
6- MAC address geolocation (samy.pl)
0- Back
  → 3

// Honeypot //

You must run PentBox with root privileges.

Select option.

1- Fast Auto Configuration
2- Manual Configuration [Advanced Users, more options]
  → 1
```

```
ashok@kali: ~/pentbox/pentbox-1.8
File Actions Edit View Help
ashok@kali: ~/Downloads * ashok@kali: ~/pentbox/pentbox-1.8 *

8- Exit
  → 2
1- Net DoS Tester
2- TCP port scanner
3- Honeypot
4- Fuzzer
5- DNS and host gathering
6- MAC address geolocation (samy.pl)
0- Back
  → 3

// Honeypot //

You must run PentBox with root privileges.

Select option.

1- Fast Auto Configuration
2- Manual Configuration [Advanced Users, more options]
  → 1

HONEYPOT ACTIVATED ON PORT 80 (2022-07-24 14:48:48 +0530)
```



Access denied

IPAddress login failed

2022-07-24 14:48:48 +0530



```
ashok@kali: ~/pentest/pentbox-1.8
File Actions Edit View Help
ashok@kali: ~/Downloads * ashok@kali: ~/pentest/pentbox-1.8 * ashok@kali: ~ *

HONEYPOT ACTIVATED ON PORT 80 (2022-07-24 14:48:48 +0530)

INTRUSION ATTEMPT DETECTED! from 192.168.0.173:60819 (2022-07-24 14:52:02 +0530)

GET / HTTP/1.1
Host: 192.168.0.190
Connection: keep-alive
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.114 Safari/537.36 Edg/103.0.1264.62
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

INTRUSION ATTEMPT DETECTED! from 192.168.0.173:60829 (2022-07-24 14:52:05 +0530)

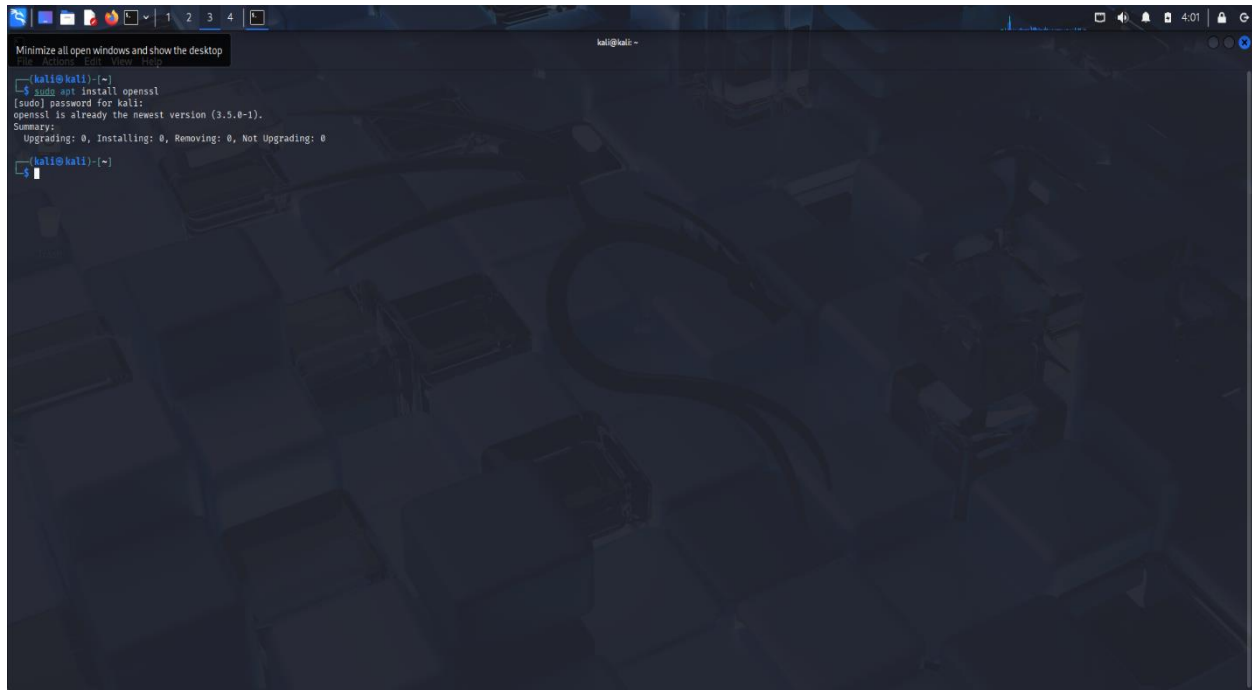
GET /favicon.ico HTTP/1.1
Host: 192.168.0.190
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.114 Safari/537.36 Edg/103.0.1264.62
DNT: 1
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Referer: http://192.168.0.190/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

Experiment 4: Generate minimum 10 passwords of length 12 characters using **openssl** Command.

Aim: To generate strong passwords of different characters of length 12.

Step 1: Install openssl

Command: `#sudo apt install openssl`

A screenshot of a Kali Linux terminal window. The terminal shows the command `sudo apt install openssl` being executed. The output indicates that openssl is already the newest version (3.5.0-1). The summary shows 0 upgrades, 0 installations, 0 removals, and 0 not-upgrading actions. The terminal prompt is `kali@kali:~`.

```
Minimize all open windows and show the desktop
kali@kali:~$ sudo apt install openssl
[sudo] password for kali:
openssl is already the newest version (3.5.0-1).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
kali@kali:~$
```

Step 1:

```
$ openssl rand -base64 16
```

```
Decode:$ echo "B3ch3m3e35LcCiRQiqI=" | base64 -d | wc -c 14
```

Experiment 5: Perform practical approach to implement Foot Printing-Gathering target information using Dmitry-Dmagic / UAtester.

Aim: To gather target information

Step 1: Install Dmitry

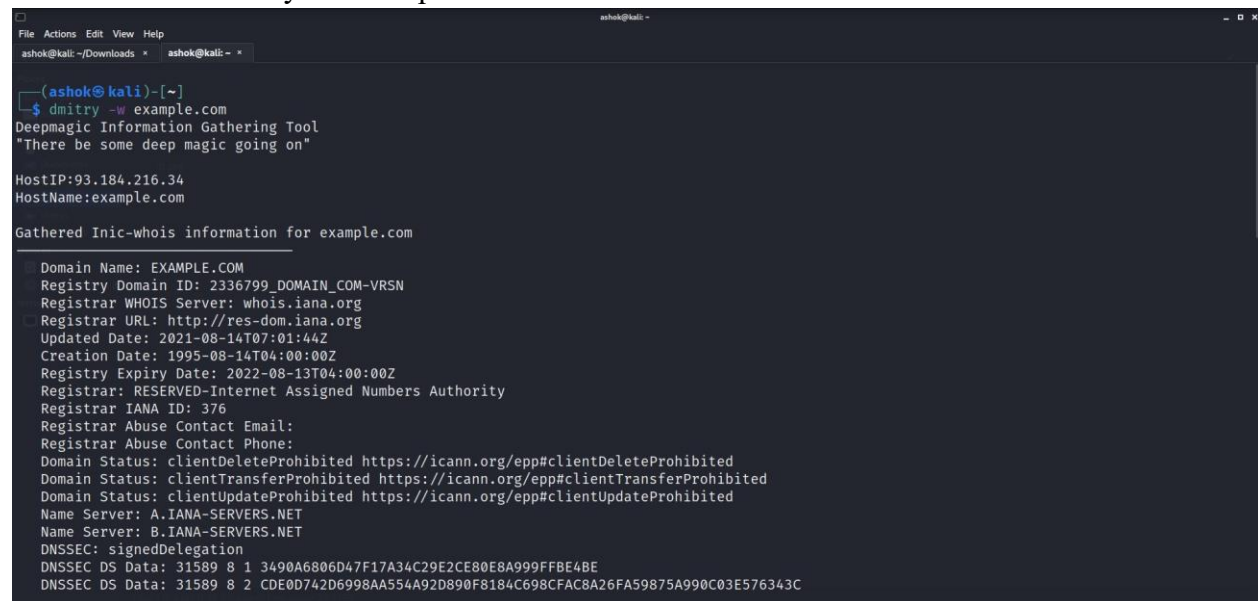
Command: `#sudo apt install dmitry`

dmitry Usage Example

Run a domain whois lookup (w), an IP whois lookup (i), retrieve Netcraft info (n), search for subdomains (s), search for email addresses (e), do a TCP port scan (p), and save the output to example.txt (o) for the domain example.com:

Step 2: Run the tool and type the following command to gather WHOIS information.

Command: `# ./dmirty -w example.com`



```
ashok@kali: ~/Downloads
ashok@kali: ~
(ashok@kali)~$ ./dmirty -w example.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:93.184.216.34
HostName:example.com

Gathered Inic-whois information for example.com

Domain Name: EXAMPLE.COM
Registry Domain ID: 2336799_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.iana.org
Registrar URL: http://res-dom.iana.org
Updated Date: 2021-08-14T07:01:44Z
Creation Date: 1995-08-14T04:00:00Z
Registry Expiry Date: 2022-08-13T04:00:00Z
Registrar: RESERVED-Internet Assigned Numbers Authority
Registrar IANA ID: 376
Registrar Abuse Contact Email:
Registrar Abuse Contact Phone:
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Name Server: A.IANA-SERVERS.NET
Name Server: B.IANA-SERVERS.NET
DNSSEC: signedDelegation
DNSSEC DS Data: 31589 8 1 3490A6806D47F17A34C29E2CE80E8A999FFBE4BE
DNSSEC DS Data: 31589 8 2 CDE0D742D6998AA554A92D890F8184C698CFAC8A26FA59875A990C03E576343C
```

Step 3: Run the tool and type the following command to gather Inet-WHOIS information.

Command: `# ./dmirty -i example.com`

```
ashok@kali: ~/Downloads x ashok@kali ~  
ashok@kali)~  
$ dmitry -i example.com  
Deepmagic Information Gathering Tool  
"There be some deep magic going on"  
  
HostIP:93.184.216.34  
HostName:example.com  
  
Gathered Inet-whois information for 93.184.216.34  
  
inetnum: 93.184.216.0 - 93.184.216.255  
netname: EDGECAST-NETBLK-03  
descr: NETBLK-03-EU-93-184-216-0-24  
country: EU  
admin-c: DS7892-RIPE  
tech-c: DS7892-RIPE  
status: ASSIGNED PA  
mnt-by: MNT-EDGECAST  
created: 2012-06-22T21:48:41Z  
last-modified: 2012-06-22T21:48:41Z  
source: RIPE # Filtered  
  
person: Derrick Sawyer  
address: 13031 W Jefferson Blvd #900, Los Angeles, CA 90094  
phone: +18773343236  
nic-hdl: DS7892-RIPE  
created: 2010-08-25T18:44:19Z  
last-modified: 2017-03-03T09:06:18Z
```

Step 4: Run the tool and type the following command to gather netcraft information.

Command: # ./dmirty -n example.com

```
ashok@kali: ~/Downloads x ashok@kali ~  
ashok@kali)~  
$ dmitry -n example.com  
Deepmagic Information Gathering Tool  
"There be some deep magic going on"  
  
HostIP:93.184.216.34  
HostName:example.com  
  
Gathered Netcraft information for example.com  
  
Retrieving Netcraft.com information for example.com  
Netcraft.com Information gathered  
  
All scans completed, exiting  
  
ashok@kali)~  
$
```

Step 5: Run the tool and type the following command to gather email information.

Command: # ./dmirty -e example.com

```
ashok@kali: ~/Downloads x ashok@kali: ~
(ashok@kali)-[~]
$ dmitry -e example.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:93.184.216.34
HostName:example.com

Gathered E-Mail information for example.com
-----
Searching Google.com:80 ...
someone@example.com
email@example.com
abc@example.com
test@example.com
me@example.com
john@example.com
webdesign@example.com
m.bluth@example.com
example@example.com
info@example.com
someoneelse@example.com
tony.stark@example.com
to@example.com
```

Step 6: Run the tool and type the following command to gather subdomain information.

Command: # ./dmirty -s example.com

```
ashok@kali: ~/Downloads x ashok@kali: ~
(ashok@kali)-[~]
$ dmitry -s example.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:93.184.216.34
HostName:example.com

Gathered Subdomain information for example.com
-----
Searching Google.com:80 ...
HostName:www.example.com
HostIP:93.184.216.34
HostName:WWW.example.com
HostIP:93.184.216.34
Searching Altavista.com:80 ...
Found 2 possible subdomain(s) for host example.com, Searched 0 pages containing 0 results

All scans completed, exiting

(ashok@kali)-[~]
$
```

Step 7: Run the tool and type the following command to gather port information.

Command: # ./dmirty -p example.com

```
File Actions Edit View Help
ashok@kali: ~/Downloads x ashok@kali: ~ x

(ashok@kali)~[~]
$ dmitry -p example.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:93.184.216.34
HostName:example.com

Gathered TCP Port information for 93.184.216.34


---


Port      State
25/tcp    open
53/tcp    open
80/tcp    open

Portscan Finished: Scanned 150 ports, 0 ports were in state closed

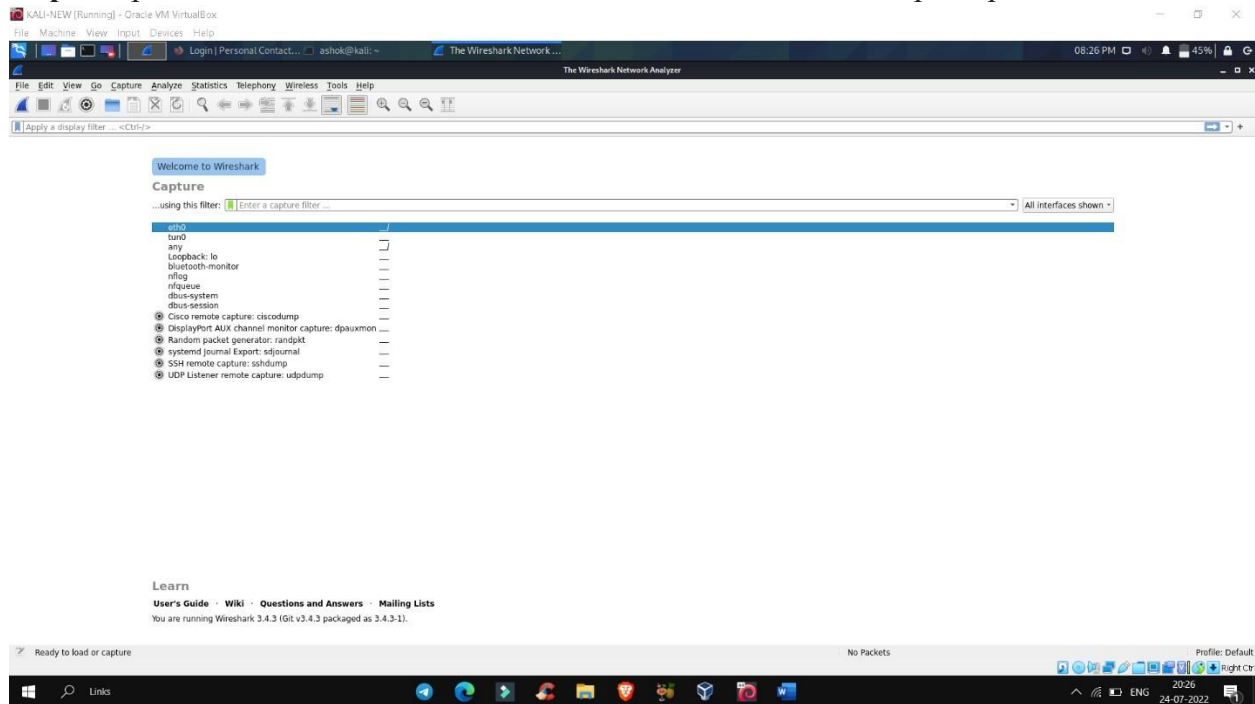
All scans completed, exiting

(ashok@kali)~[~]
$
```

Experiment 6: Working with sniffers for monitoring network communication (Wireshark)

Aim: To monitor network communication.

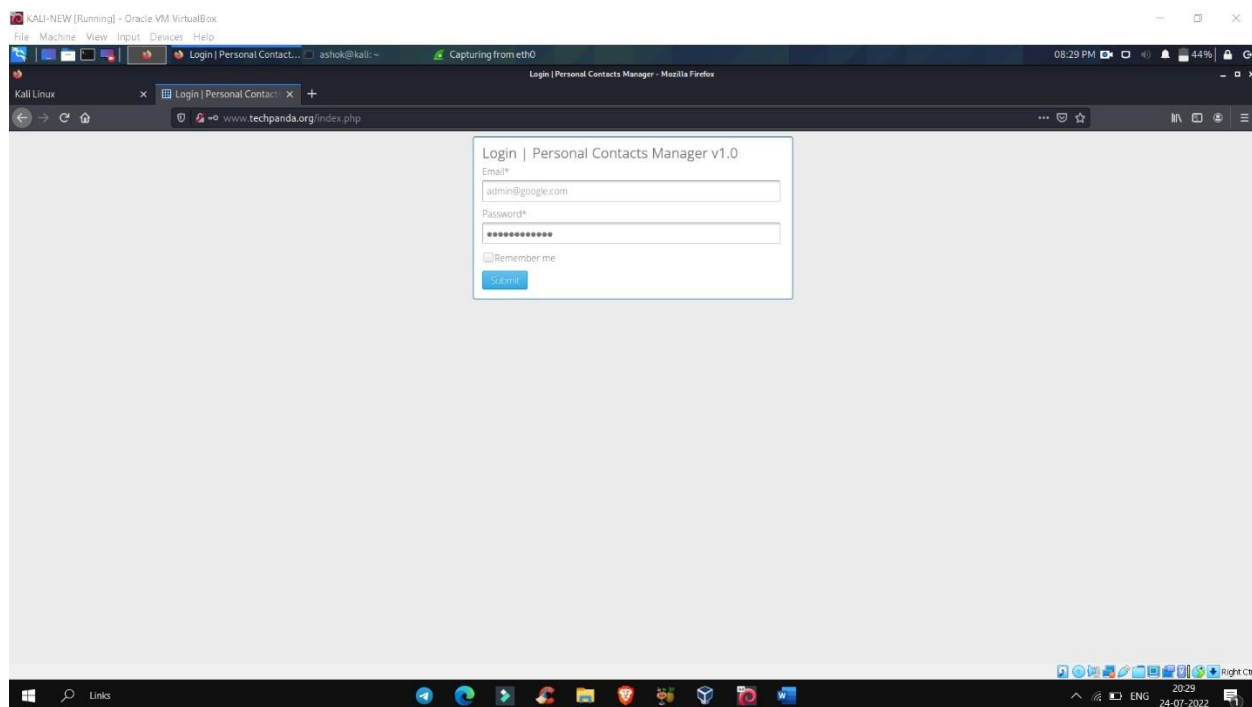
Step 1: Open Wireshark, select Interface and click on start button to capture packets.



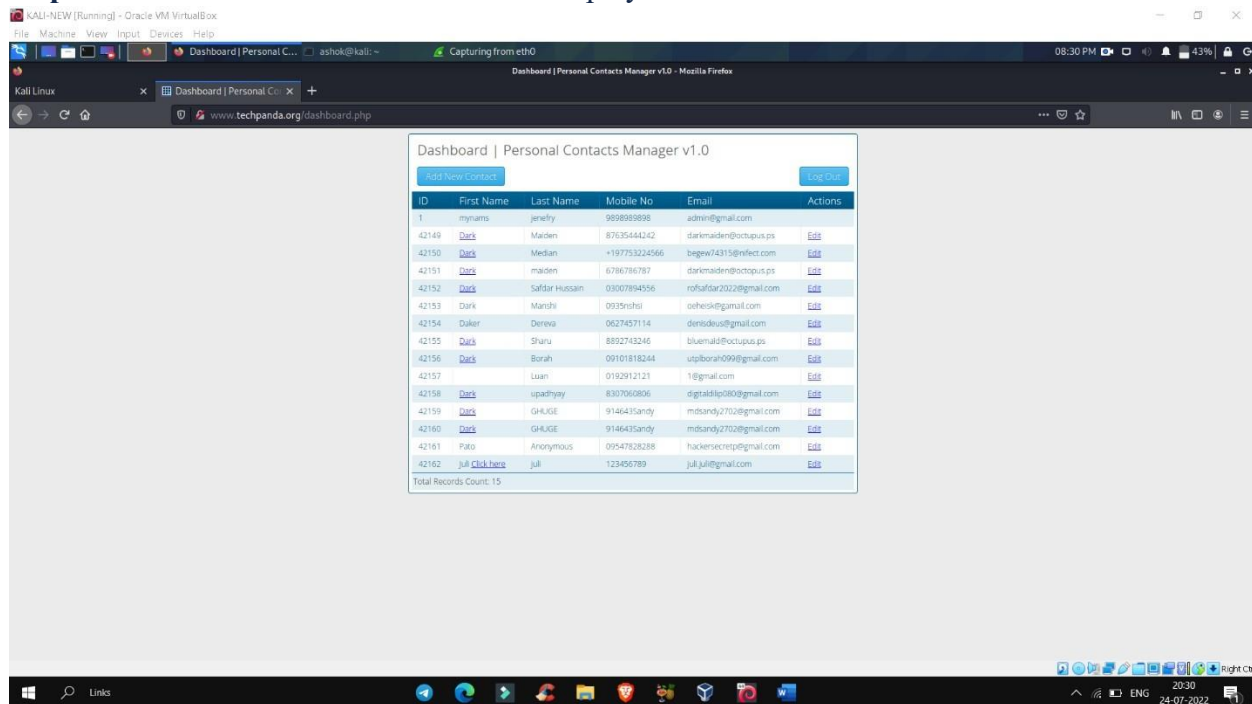
Step 2: Open <http://www.techpanda.org> in any web browser and enter credentials.

Email: admin@google.com

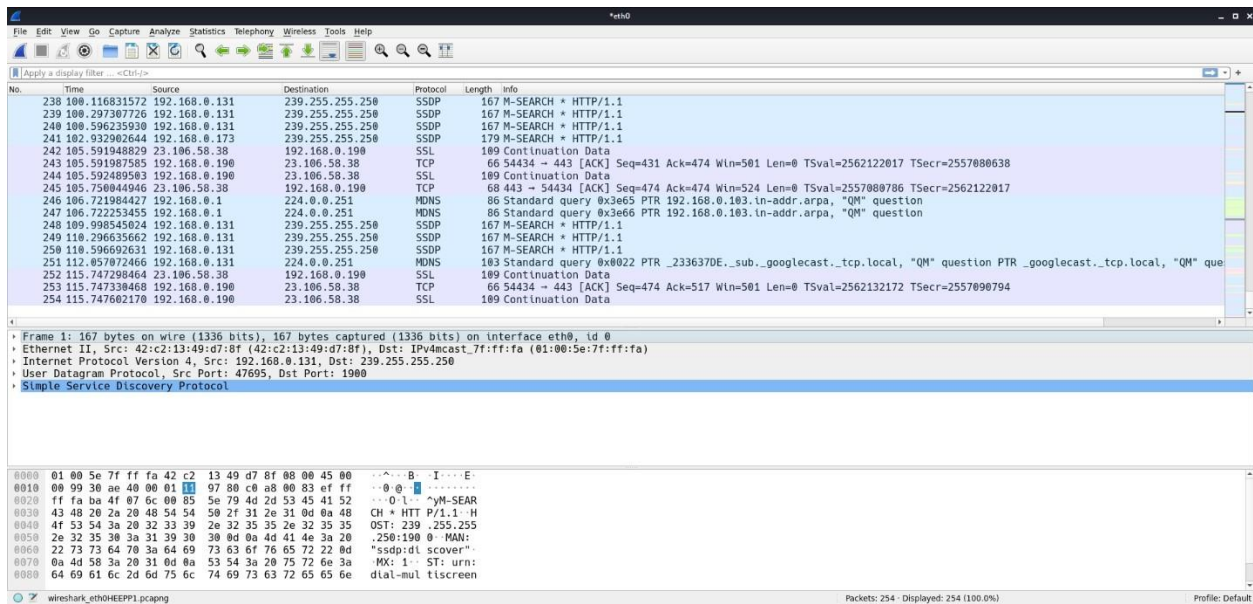
Password: Password2010



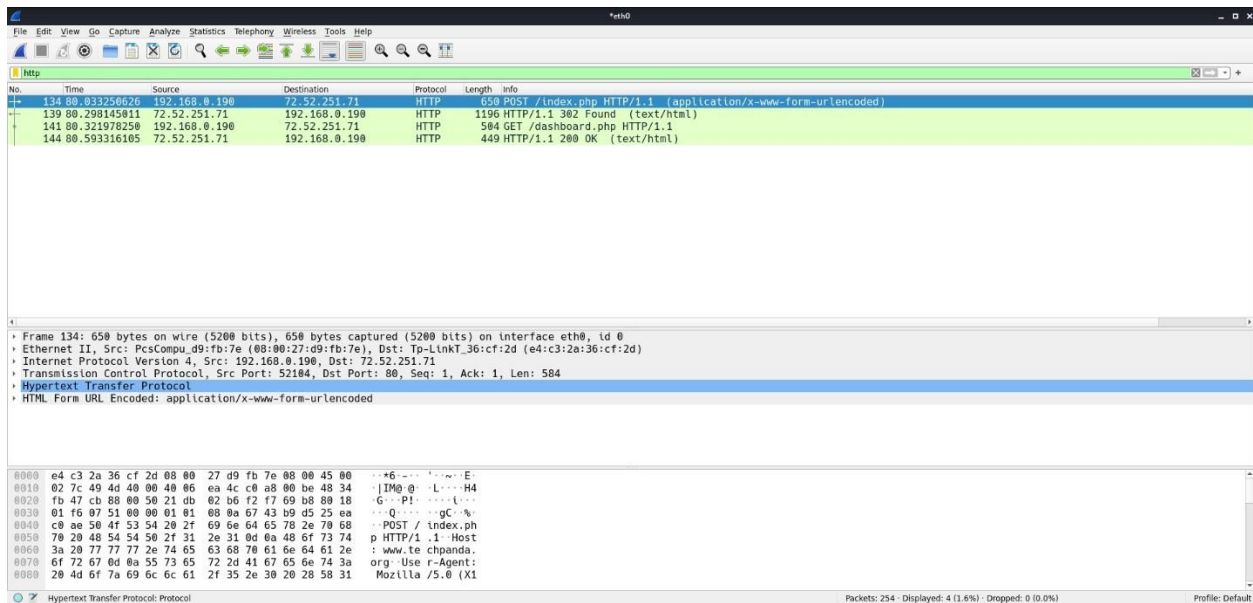
Step 3: Click on submit button and it will display the below screen



Step 4: Next go to the Wireshark and monitor the traffic



Step 5: Now stop the live capture and filter HTTP protocol. Locate the Info column and look for entries with the HTTP verb POST and click on it.



Step 6: We should be able to view the plaintext values of all the POST variables submitted to the server via HTTP protocol.

Wireshark interface showing network traffic analysis. The top pane displays a list of captured packets, highlighting an HTTP POST request (No. 134) to /index.php. The middle pane shows the packet details, including Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol. The bottom pane displays the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
134	0.033250626	192.168.0.190	72.52.251.71	HTTP	650	POST /index.php HTTP/1.1 (application/x-www-form-urlencoded)
139	0.298145811	72.52.251.71	192.168.0.190	HTTP	1196	HTTP/1.1 302 Found (text/html)
141	0.321978258	192.168.0.190	72.52.251.71	HTTP	584	GET /dashboard.php HTTP/1.1
144	0.593316105	72.52.251.71	192.168.0.190	HTTP	449	HTTP/1.1 200 OK (text/html)

Frame 134: 650 bytes on wire (5200 bits), 650 bytes captured (5200 bits) on interface eth0, id 0
Ethernet II, Src: PcsCompu_d9:fb:7e (08:00:27:d9:fb:7e), Dst: Tp-LnkT_36:cf:2d (e4:c3:2a:36:cf:2d)
Internet Protocol Version 4, Src: 192.168.0.190, Dst: 72.52.251.71
Transmission Control Protocol, Src Port: 52184, Dst Port: 80, Seq: 1, Ack: 1, Len: 584
Hypertext Transfer Protocol
HTML Form URL Encoded: application/x-www-form-urlencoded
Form item: "email" = "admin@google.com"
Form item: "password" = "Password2010"

0000 e4 c3 2a 36 cf 2d 08 00 27 d9 fb 7e 08 00 45 00 ...*6-...E-
0010 02 7c 49 4d 40 00 40 06 ea 4c c0 a8 00 be 48 34 ...IM@...L...H4
0020 fb 47 cb 88 00 50 21 db 02 b6 f2 f7 69 b8 80 18 ...G...P!...L...
0030 01 16 07 51 00 00 01 81 08 0a 67 43 b9 d5 25 ea ...Q...gC...
0040 c0 ae 50 4f 53 54 20 2f 69 0e 64 65 78 2e 70 68 ...POST /index.ph
0050 70 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 p HTTP/1.1 Host
0060 3a 20 77 77 77 2e 74 65 63 68 70 61 6e 64 61 2e : www.te chpanda.
0070 6f 72 67 00 0a 55 73 65 72 2d 41 67 65 6e 74 3a org Use r-Agent:
0080 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 58 31 Mozilla /5.0 (X11

Profile: Default

Experiment 7: Using snort, Perform real time traffic analysis and packet logging.

Aim: To perform real time traffic analysis and packet logging.

Installing Snort

Command: sudo apt install snort

```
(root@kali)~[/home/kali]
# apt install snort
The following package was automatically installed and is no longer required:
libpoppler140
Use 'sudo apt autoremove' to remove it.

Installing:
snort

Installing dependencies:
libdaq3 libestr0 libfastjson4 liblognorm5 oinkmaster rsyslog snort-common snort-common-libraries snort-rules-default

Suggested packages:
rsyslog-mysql | rsyslog-pgsql rsyslog-mongodb rsyslog-doc rsyslog-openssl | rsyslog-gnutls rsyslog-gssapi rsyslog-relp snort-doc

Summary:
Upgrading: 0, Installing: 10, Removing: 0, Not Upgrading: 1654
Download size: 3,679 kB
Space needed: 15.8 MB / 55.4 GB available

Continue? [Y/n] Y
```

To check your Snort version, use the command: snort -V

```
(root@kali)~[/var/log/snort]
# snort -V

--> Snort++ <*-
o" )~ Version 3.1.82.0
'''
By Martin Roesch & The Snort Team
http://snort.org/contact#team
Copyright (C) 2014-2024 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using DAQ version 3.0.12
Using LuaJIT version 2.1.1700206165
Using OpenSSL 3.4.1 11 Feb 2025
Using libpcap version 1.10.5 (with TPACKET_V3)
Using PCRE version 8.39 2016-06-14
Using ZLIB version 1.3.1
Using LZMA version 5.6.3
```

Choosing the Network Interface

Snort defaults to your primary network interface, but you can explicitly choose one with -i:

To sniff on a specific interface:

Command: sudo snort -i eth0

- **-i eth0:** Use interface eth0 (replace with your active interface name).

```
(root@kali)-[/home/kali]
# sudo snort -i eth0

o")~  Snort++ 3.1.82.0

pcap DAQ configured to passive.
Commencing packet processing
++ [0] eth0

Packet Statistics

daq
    received: 204
    analyzed: 200
    outstanding: 4
    outstanding_max: 4
    allow: 200
    rx_bytes: 46632

codec
    total: 200 (100.000%)
    discards: 7 ( 3.500%)
    eth: 200 (100.000%)
    icmp4: 6 ( 3.000%)
    icmp6: 1 ( 0.500%)
    ipv4: 199 ( 99.500%)
    ipv6: 1 ( 0.500%)
    tcp: 175 ( 87.500%)
    udp: 11 ( 5.500%)
```

To find your active interfaces:

Command: ip a

```
(root@kali)-[/home/kali]
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:92:28:d0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.194.132/24 brd 192.168.194.255 scope global dynamic noprefixroute eth0
        valid_lft 1706sec preferred_lft 1706sec
    inet6 fe80::f500:ce82:7d:9015/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:85:18:1a:e0 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
```

Logging Packets

If you want to save what you sniff for later analysis.

Command: sudo snort -i eth0 -L pcap

- **-i eth0:** Sniff on interface eth0.
- **-L pcap:** Log packets to a pcap file.

The filename will be log.pcap.TIMESTAM

```
(root@kali)~[/home/kali]
# sudo snort -i eth0 -l pcap

o")~  Snort++ 3.1.82.0

pcap DAQ configured to passive.
Commencing packet processing
++ [0] eth0
^C++ caught int signal
= stopping
-- [0] eth0

Packet Statistics
daq
    received: 31
    analyzed: 31
    allow: 31
    rx_bytes: 3174

codec
    total: 31      (100.000%)
    arp: 2        ( 6.452%)
    eth: 31       (100.000%)
    icmp4: 22     ( 70.968%)
    ipv4: 29      ( 93.548%)
    udp: 7        ( 22.581%)

Module Statistics
detection
    analyzed: 31
    logged: 31
```

This creates timestamped log files you can analyze later using tools like tcpdump or Wireshark.

Command for seeing the sniffed packets

Command: wireshark log.pcap.1744806427


```
(root@kali)-[/home/kali]
# wireshark log.pcap.1744806427
Warning: program compiled against libxml 212 using older 209
```

The image shows the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons for file operations, capture control, and analysis. A display filter bar shows 'Apply a display filter ... <Ctrl-/>'. The main packet list pane displays a table of captured packets:

No.	Time	Source	Destination	Protocol	Length
1	0.000000	192.168.194.128	192.168.194.2	DNS	8
2	0.000016	192.168.194.128	192.168.194.2	DNS	8
3	0.013510	192.168.194.2	192.168.194.128	DNS	9
4	0.014363	192.168.194.2	192.168.194.128	DNS	16
5	0.014970	192.168.194.128	142.250.77.238	ICMP	9
6	0.060594	142.250.77.238	192.168.194.128	ICMP	9
7	0.061290	192.168.194.128	192.168.194.2	DNS	9
8	0.073869	192.168.194.2	192.168.194.128	DNS	13
9	1.017523	192.168.194.128	142.250.77.238	ICMP	9
10	1.040565	142.250.77.238	192.168.194.128	ICMP	9

The packet details pane for the selected packet (No. 1) shows the following structure:

- Frame 1: 81 bytes on wire (648 bits), captured on interface eth0 (00:0c:29:f7:1d:93) at 0.000000
- Ethernet II, Src: VMware_f7:1d:93 (00:0c:29:f7:1d:93), Dst: 192.168.194.2 (08:00:27:00:00:02)
- Internet Protocol Version 4, Src: 192.168.194.128, Dst: 192.168.194.2
- User Datagram Protocol, Src Port: 51491, Dst Port: 53
- Domain Name System (query)

The packet bytes pane shows the raw data in hexadecimal and ASCII format.

At the bottom, the status bar indicates 'log.pcap.1744806427', 'Packets: 31', and 'Profile: Default'.