

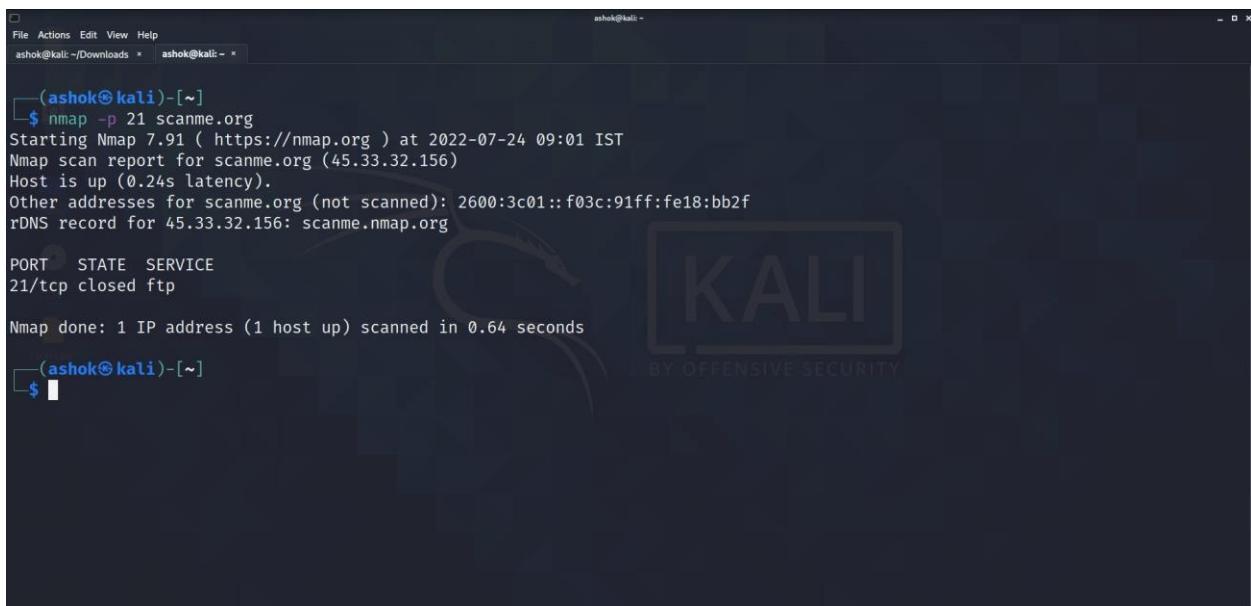
Experiment 1: Perform an Experiment for port scanning with NMAP.

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



The screenshot shows a terminal window on a Kali Linux system. The user has run the command `nmap -p 21 scanme.org`. The output indicates that port 21/tcp is closed and associated with the service ftp. The entire scan completed in 0.64 seconds.

```
(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

```
ashok@kali: ~
```

```
(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

```
ashok@kali: ~
```

```
(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

```
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

```
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

```
ashok@kali: ~
```

```
(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

```
ashok@kali: ~
```

```
(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)-[~]
$ 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: TCP ACK port scan

\$ nmap -sA scanme.org

```
(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
```

Step 4: TCP window port scan

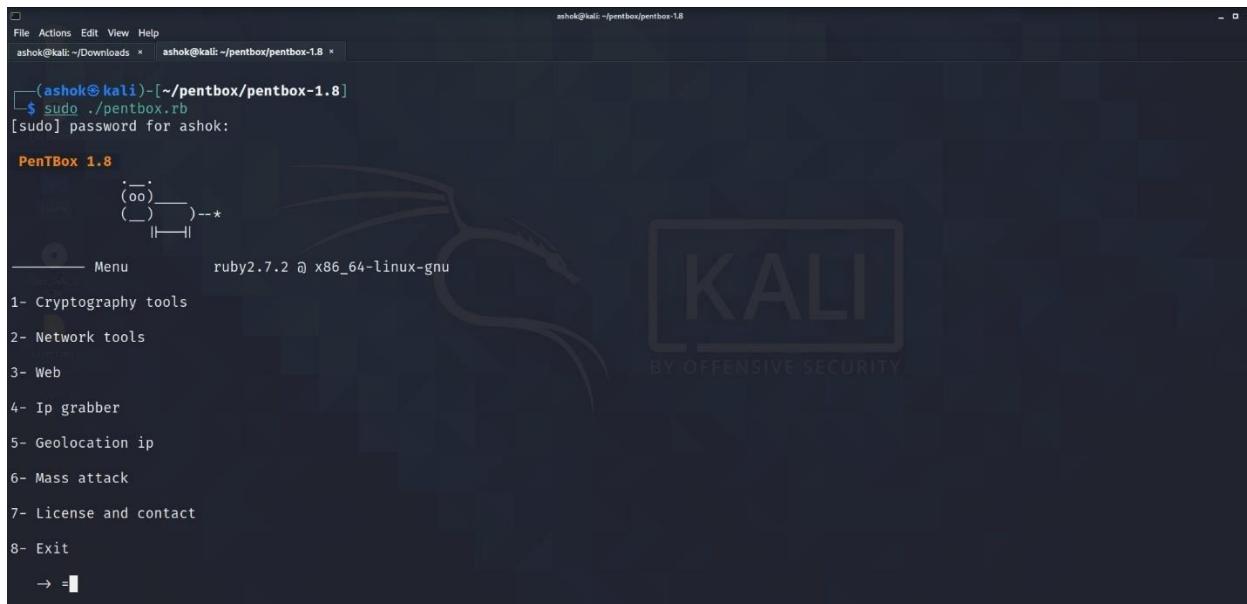
Command: \$ namp -w scanme.org

```
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: ~/pentbox/pentbox-1.8
ashok@kali: ~/Downloads x  ashok@kali: ~/pentbox/pentbox-1.8 x

(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: ~/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

→ 2
```

```
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

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

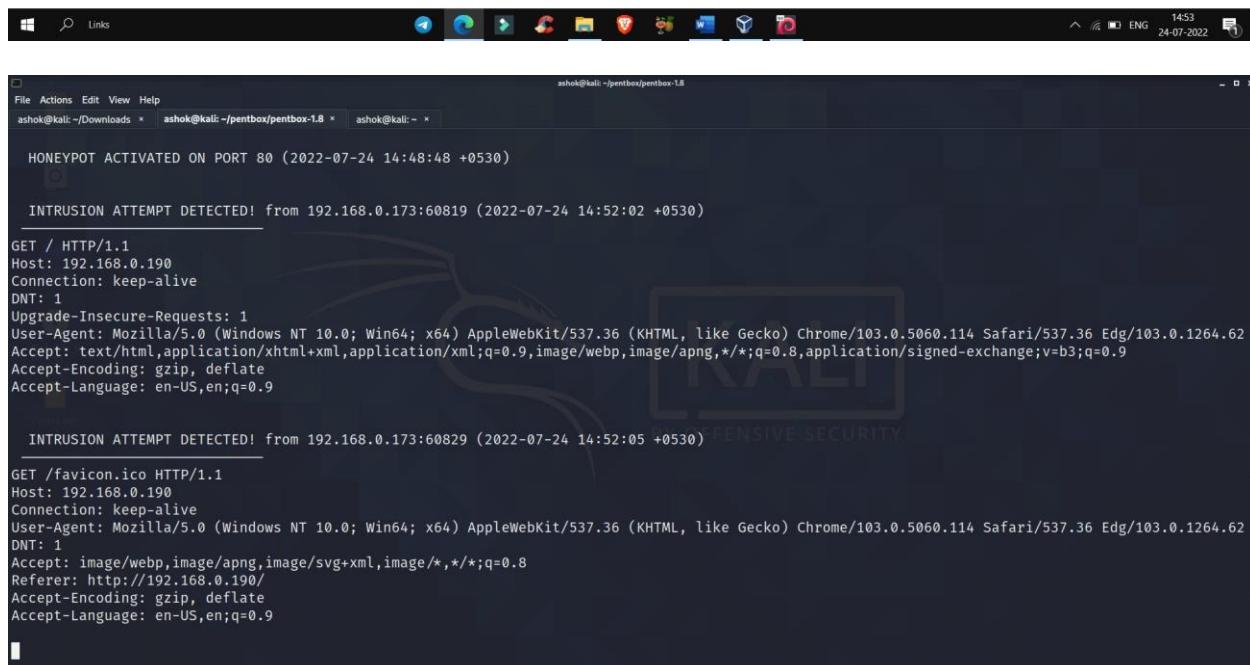
```
ashok@kali: ~/Downloads * ashok@kali: ~/pentbox/pentbox-1.8 *  
File Actions Edit View Help  
  
8- Exit  
→ 2  
  
1- Net DoS Tester  
2- TCP port scanner  
3- Honeypot  
4- Fuzzer  
5- DNS and host gathering  
6- MAC address geolocation (samy.pl)  
  
0- Back  
→ 3  
  
// Honeypot //  
  
You must run PenTBox with root privileges.  
  
Select option.  
  
1- Fast Auto Configuration  
2- Manual Configuration [Advanced Users, more options]  
→ 1  
  
HONEYBOT ACTIVATED ON PORT 80 (2022-07-24 14:48:48 +0530)
```



Access denied

IP Address login failed

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

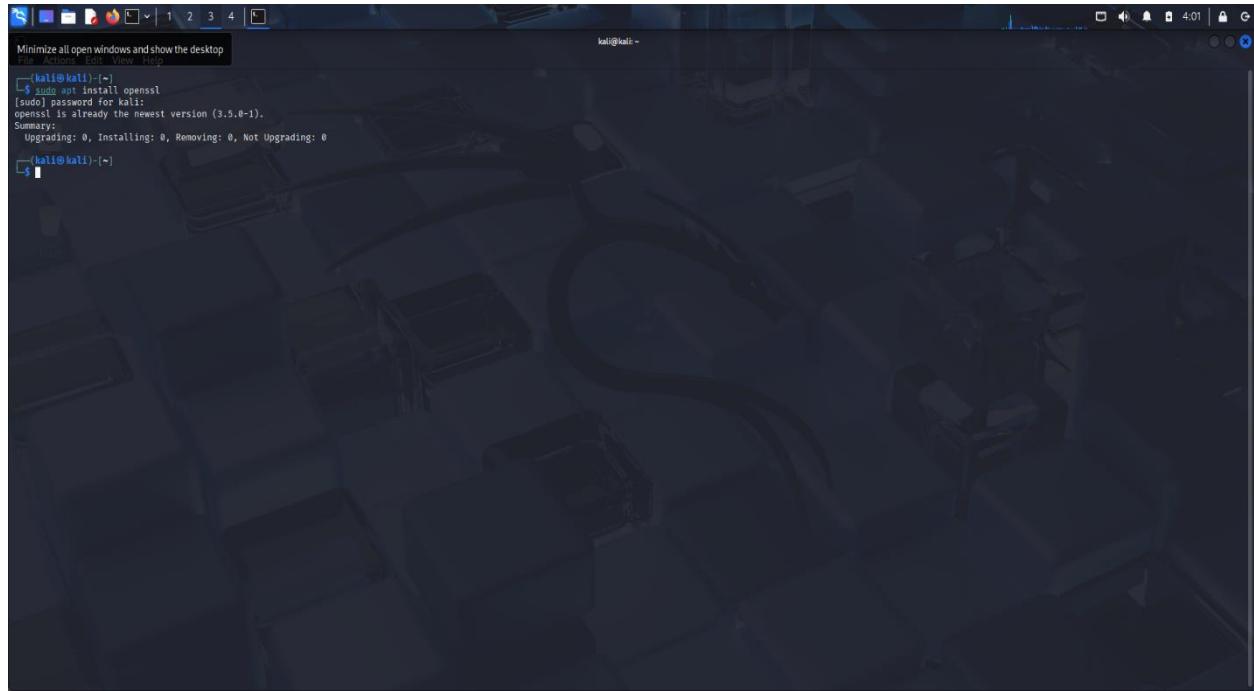


Experiment 4: Generate minimum 10 passwords of length 12 characters using **openSSL** Command.

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

Step 1: Install openssl

Command: #sudo apt install openssl



Step 1:

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

```
Decode:$ echo "B3ch3m35LcCiRQiqlI=" | 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

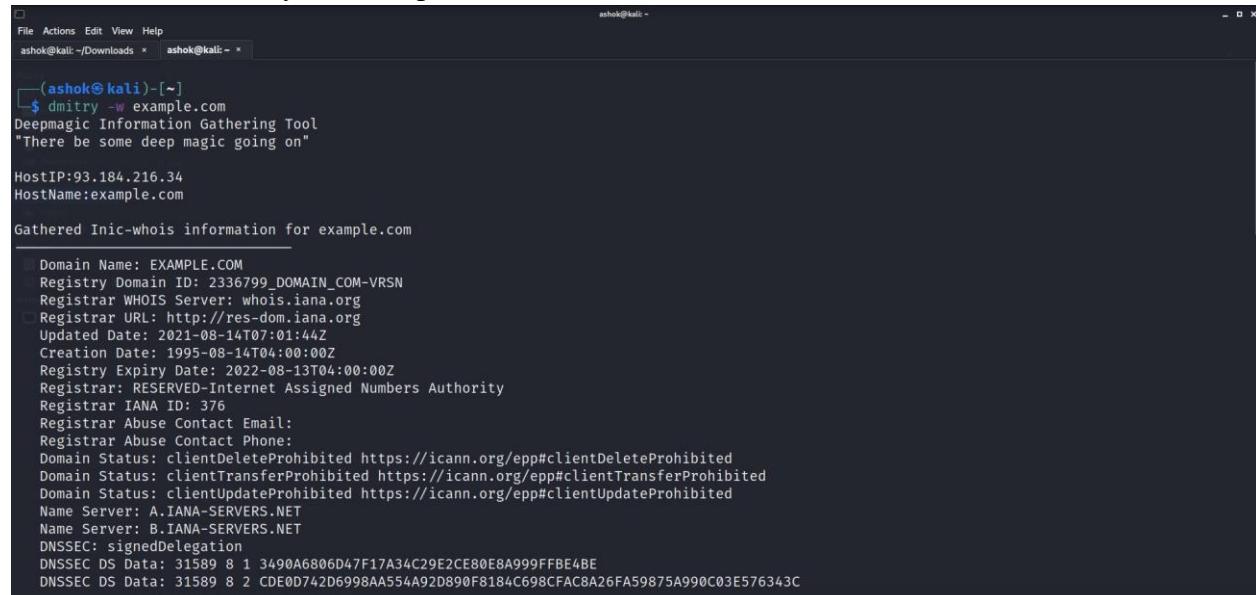
Command: #sudo apt install dmitry

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

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

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

```
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

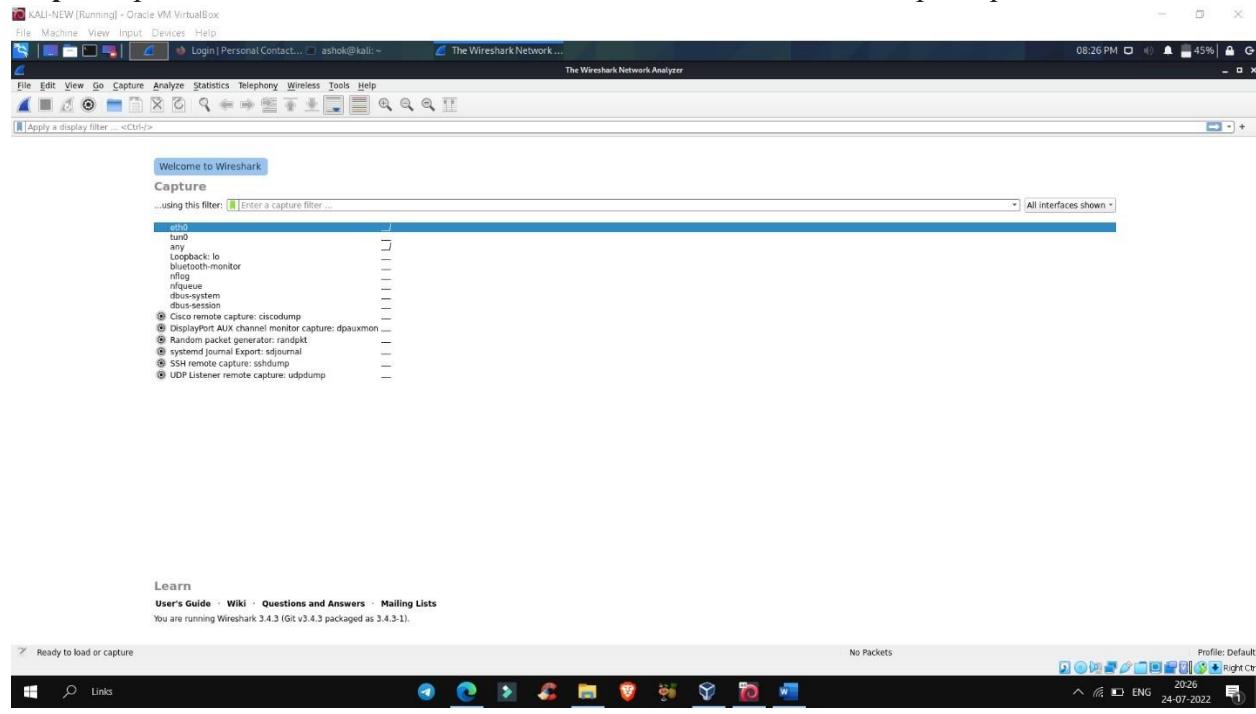
Portscan Finished: Scanned 150 ports, 0 ports were in state closed

All scans completed, exiting
ashok@kali: ~$
```

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

Aim: To monitor network communication.

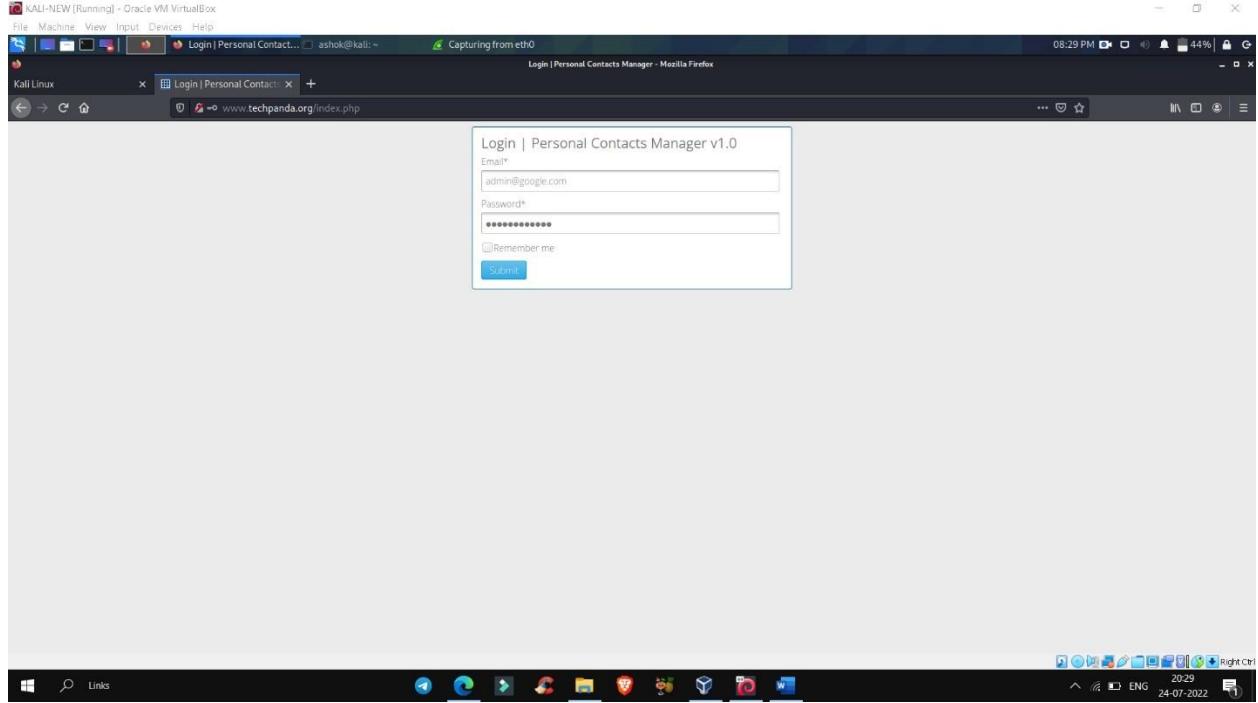
Step 1: Open Wireshark, select Interface and click on start button to capture packets.



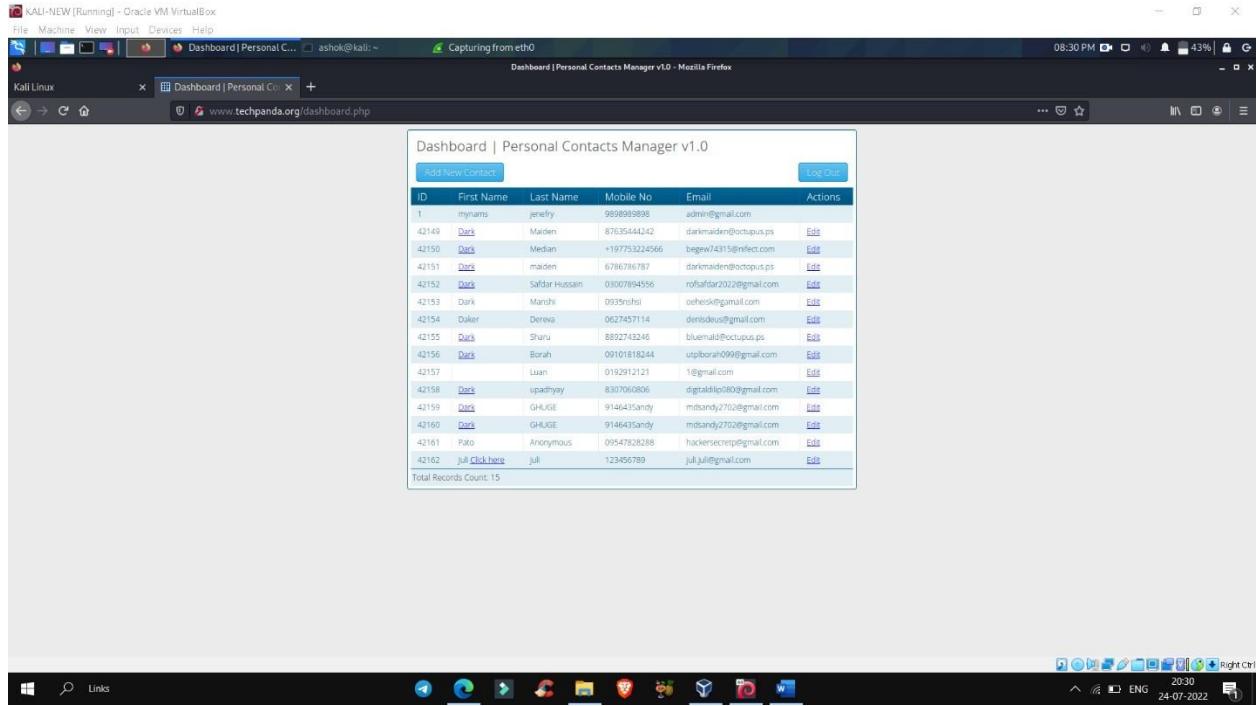
Step 2: Open <http://www.techpanda.org> in any web browser and enter credentials.

Email: admin@google.com

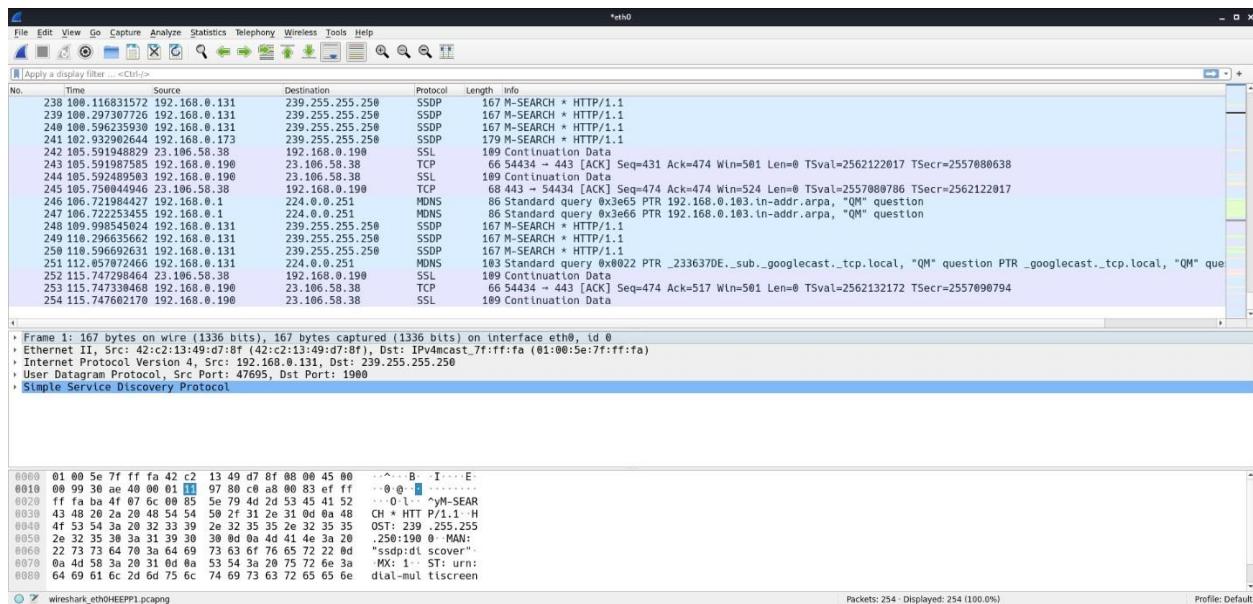
Password: Password2010



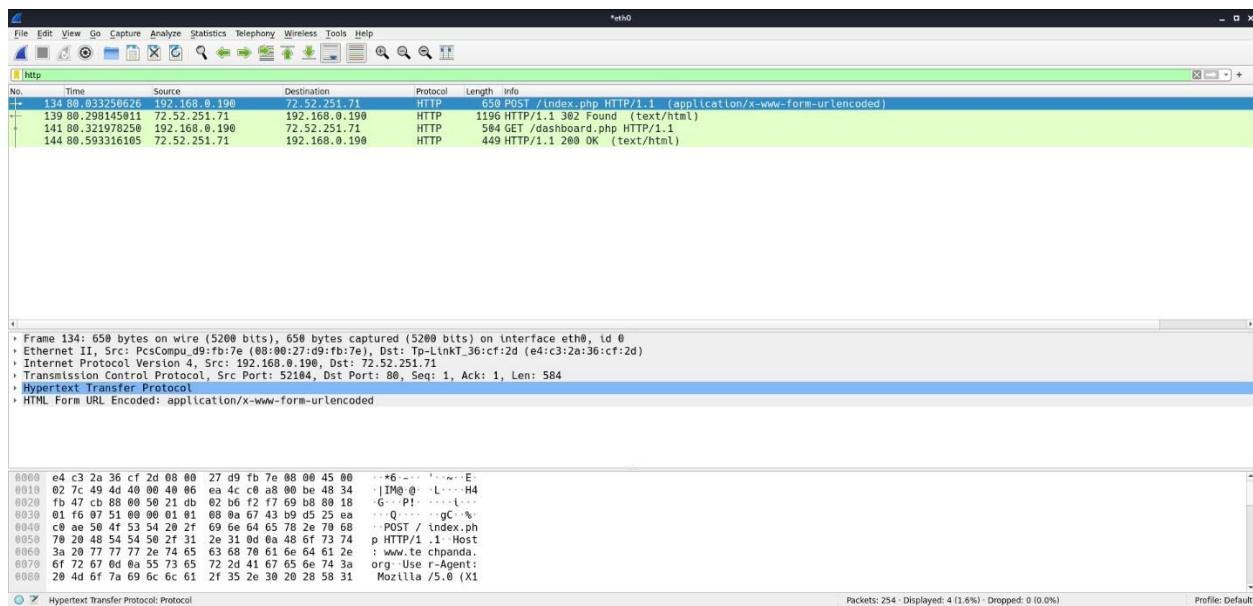
Step 3: Click on submit button and it will display the below screen



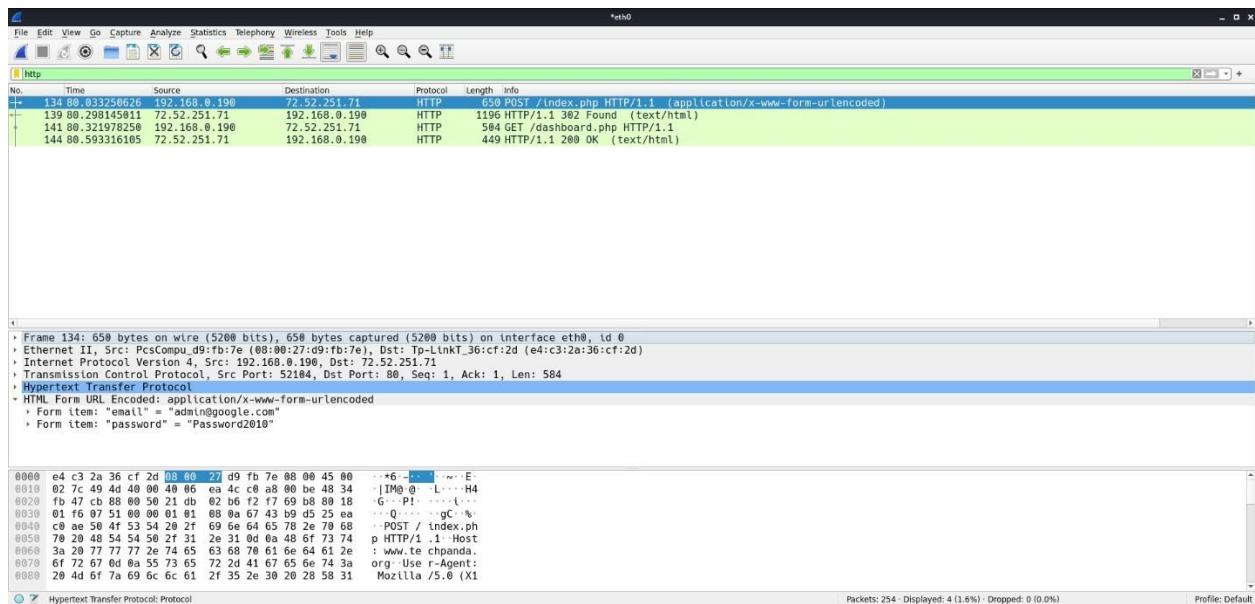
Step 4: Next go to the Wireshark and monitor the traffic



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.



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++ <-
0" )~ 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,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.TIMESTAMP

```
(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

