

Synprobe

This tool is a network port scanning tool. If specific ports are provided then those ports are scanned else a standard list of ports are scanned. First the tool checks for open port on tcp using syn-scanning. Once an open port is detected we try to detect on the following 6 cases.

1. Type 1: TCP Server Initiated Protocol Detected
2. Type 2: TLS Server Initiated Protocol Detected
3. Type 3: HTTP Server Detected
4. Type 4: HTTPS Server Detected
5. Type 5: TCP Generic Server Detected
6. Type 6: TLS Generic Server Detected

If one of the 6 types is detected then the tool also prints the first 1024 bytes that are received by the tool. The printing format of the response is the same as that of hexdump. This is done to replace non-printable bytes with a dot (.) - otherwise binary protocols will clutter the console.

Testing

TCP Server Initiated Protocol

Testing for tcp server initiated protocol was done by using ssh server built into kali. SSH runs over TCP and is a server initiated protocol because it first sends the banner to the client and then expects response bytes from the client. To start the ssh server on kali, run the following command

```
sudo service ssh start
```

TCP Generic Server

Testing for generic tcp server was done by ncat. To run a TCP ncat server, run the below command

```
ncat -lkv -p 9090
```

TLS Server Initiated Protocol Testing

Testing for TLS connection, where the communication is initiated by the server is done the server **imap.gmail.com:993**. Please note that the aforementioned server uses IMPAS which is the TLS Secured version of IMAP. Sometimes over port 143 one may find an IMAP server with STARTTLS. The STARTTLS method starts with a connection on the standard unencrypted IMAP port (143), and then uses the STARTTLS command to upgrade to

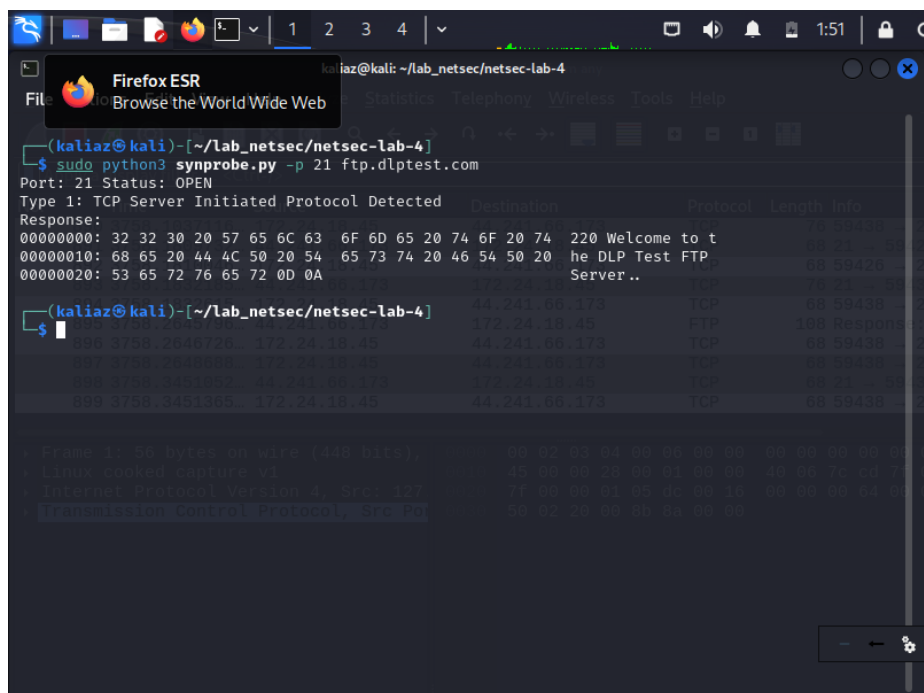


Figure 1: tcp_server_initiated.png

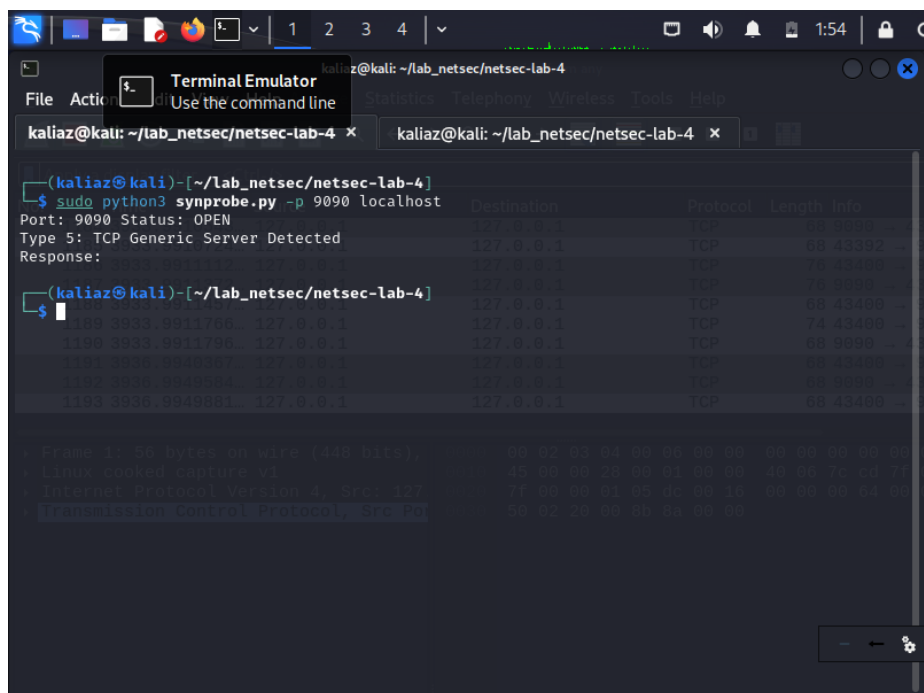


Figure 2: tcp_generic_server.png

a secure, encrypted connection. STARTTLS is an extension to plain text communication protocols, which offers a way to upgrade to TLS after the connection is established. Once the STARTTLS command is issued and the server responds positively, the connection becomes encrypted.

An alternative way would be to run a custom `tls_server` via the script or via `ncat`

```
# Generate the ssl cert and key required
openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365 -nodes
# Run the tls server
python3 tls_server.py
```

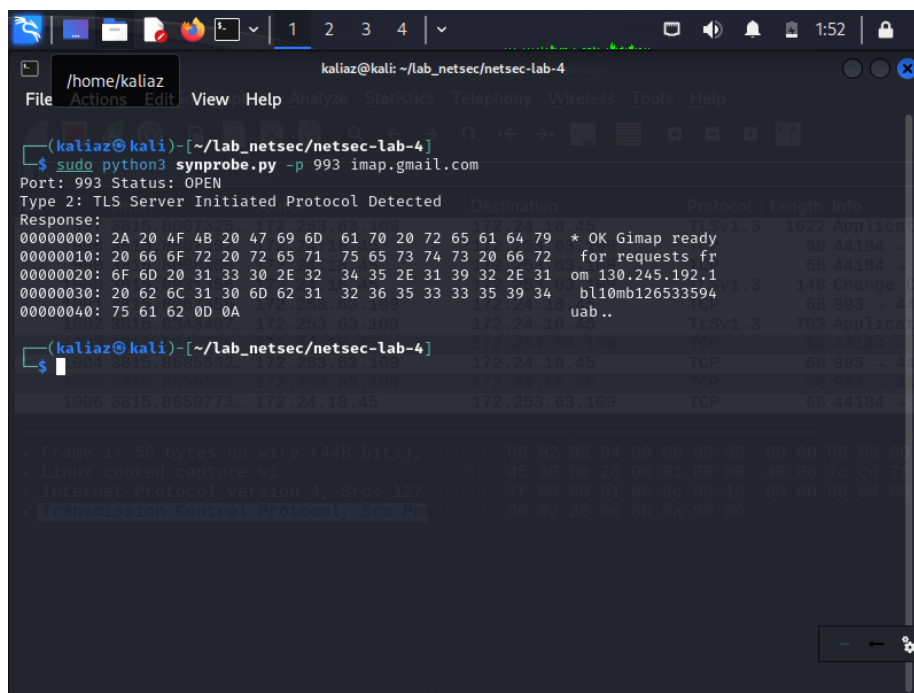


Figure 3: `tls_server_initiated.png`

Generic TLS Server

To test for a generic TLS server we first create our own `cert.pem` and `key.pem` file using the below command.

Remember to replace your ip address in the end of the below command

```
openssl req -newkey rsa:4096 -nodes -keyout key.pem -x509 -days 365 -out cert.pem -subj "/CN=172.24.18.45"
```

Now we have two options, 1. Easier for testing option : Directly trust the cert file by adding it into `cacerts` list. This option needs to be used if you have a

```
kaliaz@kali: ~/lab_netsec/netsec-lab-4
File Actions Edit View Help Analyze Statistics Telephony Wireless Tools Help

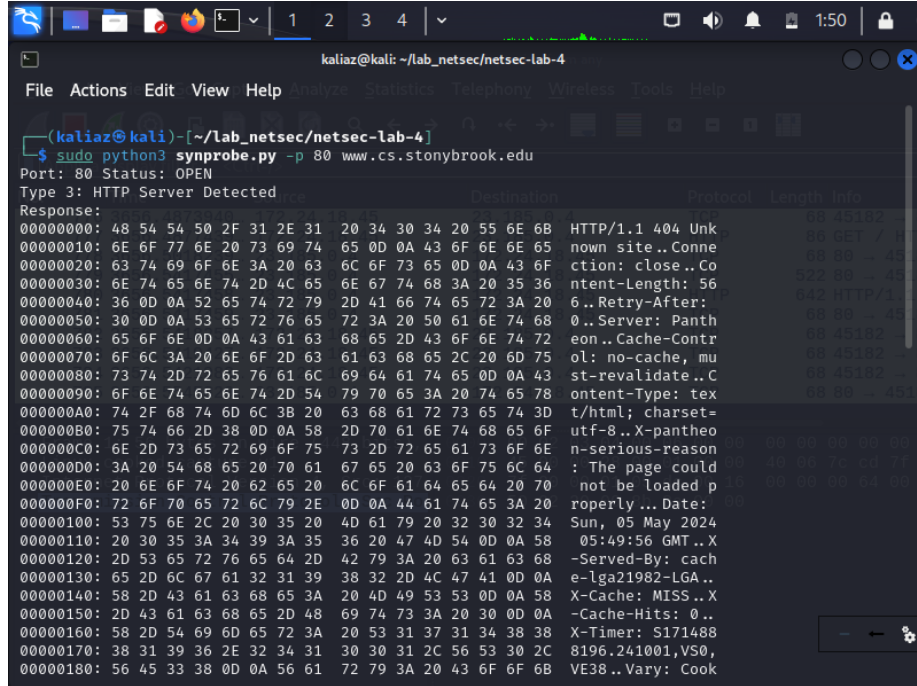
(kaliaz@kali)-[~/lab_netsec/netsec-lab-4]
$ sudo python3 synprobe.py -p 853 9.9.9.9
Port: 853 Status: OPEN
Type 6: TLS Generic Server Detected
Response:
  3677 1815892.. 9.9.9.9 172.24.18.45 TLSv1.3 323 Application
  3677 1531982.. 9.9.9.9 172.24.18.45 TLSv1.3 323 Application
  3677 1439858.. 172.24.18.45 9.9.9.9 TCP 68 55318
  3677 1531982.. 172.24.18.45 9.9.9.9 TCP 68 853
  3688 8832878.. 172.24.18.45 9.9.9.9 TCP 68 55318
  3688 1113528.. 172.24.18.45 9.9.9.9 TLSv1.3 92 Application
  3688 1113528.. 172.24.18.45 9.9.9.9 TCP 68 853
  3688 1113528.. 172.24.18.45 9.9.9.9 TCP 68 55318
  3688 1113528.. 172.24.18.45 9.9.9.9 TCP 68 55318

- Frame 1: 56 bytes on wire (448 bits) 00 02 03 04 00 06 00 06 00 00 00 00 01
- Linux cooked capture of 45 00 00 28 00 01 00 06 40 56 7c cd 7f
- Internet Protocol Version 4, Src: 9.9.9.9, Destination: 172.24.18.45 7f 00 00 01 05 0c 00 16 00 00 00 64 00
- Transmission Control Protocol, Src Port: 853, Destination Port: 853 50 02 20 00 0b 0a 00 00
```

5

HTTP Server Testing

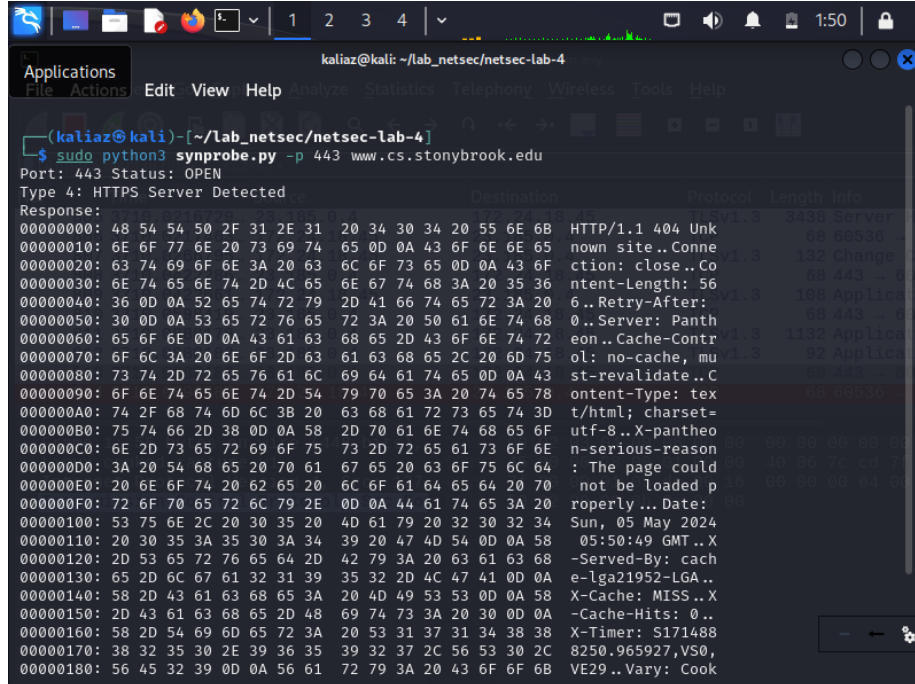
Below image captures the testing done for http server to print the first 1024 bytes



```
kali@kali: ~/lab_netsec/netsec-lab-4
File Actions Edit View Help Analyze Statistics Telephony Wireless Tools Help
(kali@kali)~[~/lab_netsec/netsec-lab-4]
$ sudo python3 synprobe.py -p 80 www.cs.stonybrook.edu
Port: 80 Status: OPEN
Type 3: HTTP Server Detected
Response:
00000000: 48 54 54 50 2F 31 2E 31 20 34 30 34 20 55 6E 68 HTTP/1.1 404 Unk
00000010: 6E 6F 77 6E 20 73 69 74 65 0D 0A 43 6F 6E 6E 65 nown site..Conne
00000020: 63 74 69 6F 6E 3A 20 63 6C 6F 73 65 0D 0A 43 6F ction: close..Co
00000030: 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 35 36 ntent-Length: 56
00000040: 36 0D 0A 52 65 74 72 79 2D 41 66 74 65 72 3A 20 6..Retry-After:
00000050: 30 0D 0A 53 65 72 76 65 72 3A 20 50 61 6E 74 68 0..Server: Panth
00000060: 65 6F 6E 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72 eon..Cache-Contr
00000070: 6F 6C 3A 20 6E 6F 2D 63 61 63 68 65 2C 20 6D 75 ol: no-cache, mu
00000080: 73 74 2D 72 65 76 61 6C 69 64 61 74 65 0D 0A 43 st-revalidate..C
00000090: 6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 74 65 78 ontent-Type: tex
000000A0: 74 2F 68 74 6D 6C 3B 20 63 68 61 72 73 65 74 3D t/html; charset=
000000B0: 75 74 66 2D 38 0D 0A 58 2D 70 61 6E 74 68 65 6F utf-8..X-pantheo
000000C0: 6E 2D 73 65 72 69 6F 75 73 2D 72 65 61 73 6F 6E n-serious-reason
000000D0: 3A 20 54 68 65 20 70 61 67 65 20 63 6F 75 6C 64 : The page could
000000E0: 20 6E 6F 74 2D 62 65 20 6C 6F 61 64 65 64 20 70 not be loaded p
000000F0: 72 6F 70 65 72 6C 79 2E 0D 0A 44 61 74 65 3A 20 roperly...Date:
00000100: 53 75 6E 2C 20 30 35 20 4D 61 79 20 32 30 32 34 Sun, 05 May 2024
00000110: 20 30 35 3A 34 39 3A 35 36 20 47 4D 54 0D 0A 58 05:49:56 GMT..X
00000120: 2D 53 65 72 76 65 64 2D 42 79 3A 20 63 61 63 68 -Served-By: cach
00000130: 65 2D 6C 67 61 32 31 39 38 32 2D 4C 47 41 0D 0A e-lga21982-LGA..
00000140: 58 2D 43 61 63 68 65 3A 20 4D 49 53 53 0D 0A 58 X-Cache: MISS..X
00000150: 2D 43 61 63 68 65 2D 48 69 74 73 3A 20 30 0D 0A -Cache-Hits: 0..
00000160: 58 2D 54 69 6D 65 72 3A 20 53 31 37 31 34 38 38 X-Timer: S171488
00000170: 38 31 39 36 2E 32 34 31 30 30 31 2C 56 53 30 2C 8196.241001,VS0,
00000180: 56 45 33 38 0D 0A 56 61 72 79 3A 20 43 6F 6F 6B VE38..Vary: Cook
```

HTTPS Server Testing

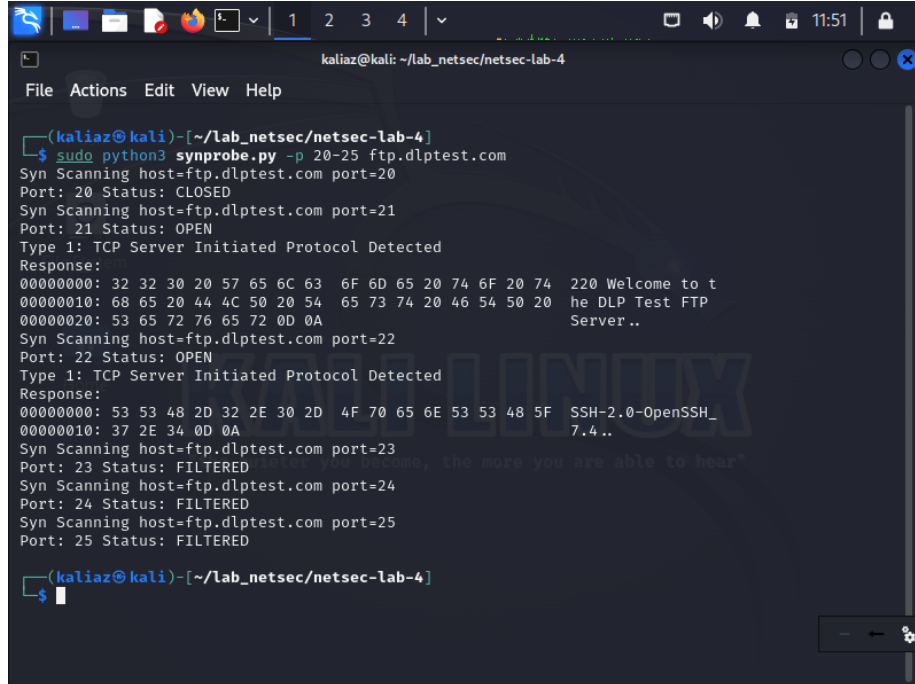
Below image captures the testing done for https server to print the first 1024 bytes



```
kali@kali: ~/lab_netsec/netsec-lab-4
$ sudo python3 synprobe.py -p 443 www.cs.stonybrook.edu
Port: 443 Status: OPEN
Type 4: HTTPS Server Detected
Response:
00000000: 48 54 54 50 2F 31 2E 31 20 34 30 34 20 55 6E 68 HTTP/1.1 404 Unk
00000010: 6E 6F 77 6E 20 73 69 74 65 0D 0A 43 6F 6E 6E 65 nown site..Conne
00000020: 63 74 69 6F 6E 3A 20 63 6C 6F 73 65 0D 0A 43 6F ction: close..Co
00000030: 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 35 36 ntent-Length: 56
00000040: 36 0D 0A 52 65 74 72 79 2D 41 66 74 65 72 3A 20 6..Retry-After:
00000050: 30 0D 0A 53 65 72 76 65 72 3A 20 50 61 6E 74 68 0..Server: Panth
00000060: 65 6F 6E 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72 eon..Cache-Contr
00000070: 6F 6C 3A 20 6E 6F 2D 63 61 63 68 65 2C 20 6D 75 ol: no-cache, mu
00000080: 73 74 2D 72 65 76 61 6C 69 64 61 74 65 0D 0A 43 st-revalidate..C
00000090: 6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 74 65 78 ontent-Type: tex
000000A0: 74 2F 68 74 6D 6C 3B 20 63 68 61 72 73 65 74 3D t/html; charset=
000000B0: 75 74 66 2D 38 0D 0A 58 2D 70 61 6E 74 68 65 6F utf-8..X-pantheo
000000C0: 6E 2D 73 65 72 69 6F 75 73 2D 72 65 61 73 6F 6E n-serious-reason
000000D0: 3A 20 54 68 65 20 70 61 67 65 20 63 6F 75 6C 64 : The page could
000000E0: 20 6E 6F 74 2D 62 65 20 6C 6F 61 64 65 64 20 70 not be loaded p
000000F0: 72 6F 70 65 72 6C 79 2E 0D 0A 44 61 74 65 3A 20 roperly...Date:
00000100: 53 75 6E 2C 20 30 35 20 4D 61 79 20 32 30 32 34 Sun, 05 May 2024
00000110: 20 30 35 3A 35 30 3A 34 39 20 47 4D 54 0D 0A 58 05:50:49 GMT..X
00000120: 2D 53 65 72 76 65 64 2D 42 79 3A 20 63 61 63 68 -Served-By: cach
00000130: 65 2D 6C 67 61 32 31 39 35 32 2D 4C 47 41 0D 0A e-lga21952-LGA..
00000140: 58 2D 43 61 63 68 65 3A 20 4D 49 53 53 0D 0A 58 X-Cache: MISS..X
00000150: 2D 43 61 63 68 65 2D 48 69 74 73 3A 20 30 0D 0A -Cache-Hits: 0..
00000160: 58 2D 54 69 6D 65 72 3A 20 53 31 37 31 34 38 38 X-Timer: S171488
00000170: 38 32 35 30 2E 39 36 35 39 32 37 2C 56 53 30 2C 8250.965927.VS0,
00000180: 56 45 32 39 0D 0A 56 61 72 79 3A 20 43 6F 6F 6B VE29..Vary: Cook
```

Multiport Testing

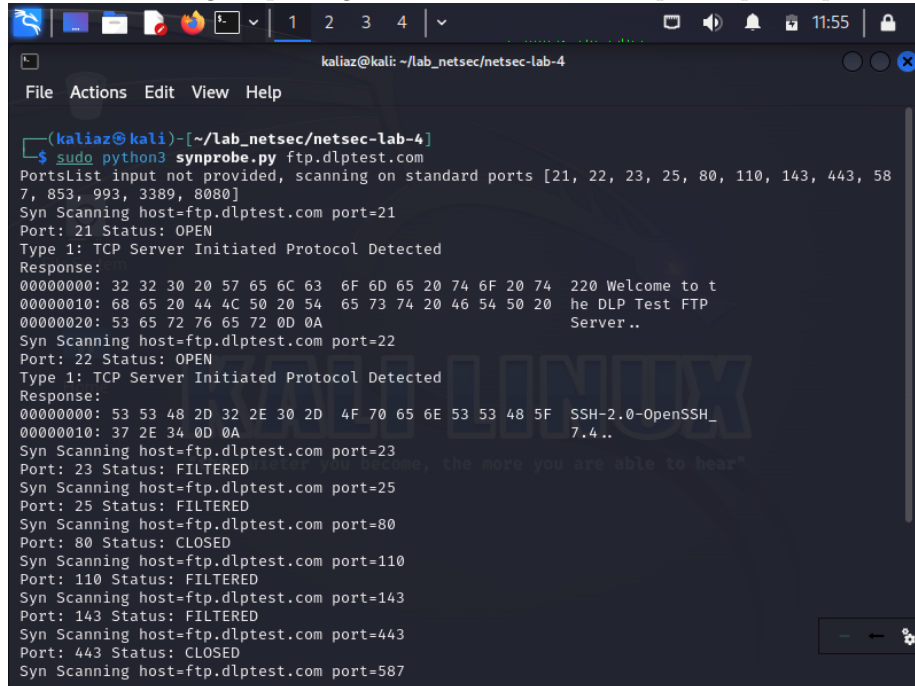
Below is an image capturing the multiport scan done for custom ports



```
(kali@kali)~/lab_netsec/netsec-lab-4
$ sudo python3 synprobe.py -p 20-25 ftp.dlpctest.com
Syn Scanning host=ftp.dlpctest.com port=20
Port: 20 Status: CLOSED
Syn Scanning host=ftp.dlpctest.com port=21
Port: 21 Status: OPEN
Type 1: TCP Server Initiated Protocol Detected
Response:
00000000: 32 32 30 20 57 65 63 63 6F 6D 65 20 74 6F 20 74 220 Welcome to t
00000010: 68 65 20 44 4C 50 20 54 65 73 74 20 46 54 50 20 he DLP Test FTP
00000020: 53 65 72 76 65 72 0D 0A Server..
Syn Scanning host=ftp.dlpctest.com port=22
Port: 22 Status: OPEN
Type 1: TCP Server Initiated Protocol Detected
Response:
00000000: 53 53 48 2D 32 2E 30 2D 4F 70 65 6E 53 53 48 5F SSH-2.0-OpenSSH_
00000010: 37 2E 34 0D 0A 7.4..
Syn Scanning host=ftp.dlpctest.com port=23
Port: 23 Status: FILTERED
Syn Scanning host=ftp.dlpctest.com port=24
Port: 24 Status: FILTERED
Syn Scanning host=ftp.dlpctest.com port=25
Port: 25 Status: FILTERED
(kali@kali)~/lab_netsec/netsec-lab-4
$
```


No Ports Provided Testing

Below is an image capturing the scan done when no port input is provided



```
(kali@kali)~[lab_netsec/netsec-lab-4]
$ sudo python3 synprobe.py ftp.dlpctest.com
PortsList input not provided, scanning on standard ports [21, 22, 23, 25, 80, 110, 143, 443, 587, 853, 993, 3389, 8080]
Syn Scanning host=ftp.dlpctest.com port=21
Port: 21 Status: OPEN
Type 1: TCP Server Initiated Protocol Detected
Response:
00000000: 32 32 30 20 57 65 6C 63 6F 6D 65 20 74 6F 20 74 220 Welcome to t
00000010: 68 65 20 44 4C 50 20 54 65 73 74 20 46 54 50 20 he DLP Test FTP
00000020: 53 65 72 76 65 72 0D 0A Server..
Syn Scanning host=ftp.dlpctest.com port=22
Port: 22 Status: OPEN
Type 1: TCP Server Initiated Protocol Detected
Response:
00000000: 53 53 48 2D 32 2E 30 2D 4F 70 65 6E 53 53 48 5F SSH-2.0-OpenSSH_
00000010: 37 2E 34 0D 0A 7.4..
Syn Scanning host=ftp.dlpctest.com port=23
Port: 23 Status: FILTERED
Syn Scanning host=ftp.dlpctest.com port=25
Port: 25 Status: FILTERED
Syn Scanning host=ftp.dlpctest.com port=80
Port: 80 Status: CLOSED
Syn Scanning host=ftp.dlpctest.com port=110
Port: 110 Status: FILTERED
Syn Scanning host=ftp.dlpctest.com port=143
Port: 143 Status: FILTERED
Syn Scanning host=ftp.dlpctest.com port=443
Port: 443 Status: CLOSED
Syn Scanning host=ftp.dlpctest.com port=587
```

Blocked/Filtered Port

iptables is the firewall for linux. By default, there are no firewall rules set on linux.

To check the firewall rules on linux, use the below command

```
sudo iptables -L
```

To simulate a filtered port or a firewall we need to add a rule to drop all incoming packets on

a port say 9092. To add a rule to iptables to drop all tcp packets on 9092, run the below command

```
sudo iptables -A INPUT -p tcp --dport 9092 -j DROP
```

For enabling closed port or filtererd port detection you must use the `self.syn_scanning(target_port)`

function that will enable the full syn scanning instead of the current connect scanning.

Debugging Experience

When I ran my synprobe for the first time, the machine on which I ran the synprobe ended up sending two RST packets instead of the expected 1 RST packet that I was creating. Furthermore, if I sent out any syn packet and recieved a syn-ack packet, the machine would automatically send one RST packet to the machine who sent the RST packet. This effectively meant that I could not complete even a single handshake via scapy. This was happening because of the OS kernel. Scapy bypasses the networks stack of the OS. When scapy sends a SYN packet and the machine recieves a SYN-ACK packet, that SYN-ACK packet is sent to two places. First, it is sent to the OS Kernel, who did not expect this SYN-ACK packet because it was unaware of what scapy is doing. Hence it sends its own RST packet. Second, it reached the synprobe.py python process. Now no matter what the synprobe.py process sends to the target machine, heuristic obsevation is that the RST from the kernel always reached first, hence that connection was terminated half-way. To avoid the kernel sending any information to the target machine, we want to avoid sending any packets sent out by the synprobe.py process to the kernel. To do this, there are OS dependent solutions. ##### For MacOS Use pf to drop any incoming packets to port 1500 and 1600 becuase these are the ports that synprobe.py is using.

```
sudo vim /etc/pf.conf
```

Add the below line

```
block in proto tcp to any port {1500, 1600}
```

```
sudo pfctl -f /etc/pf.conf # To make pf use the rules that we just defined.
sudo pfctl -e # To enable pf to perform packet filtering in case it is disabled
sudo pfctl -sr # To verify the rules set
```

For Linux Use IP Tables to drop any incoming packets to port 1500 and 1600 becuase these are the ports that synprobe.py is using. Run the below commands on the terminal of your linux machine

```
sudo iptables -A INPUT -p tcp --dport 1500 -j DROP
sudo iptables -A INPUT -p tcp --dport 1600 -j DROP
```