9/17/2024

IPTables

Firewall And IDS

Nikhil Makwana

LAB 01

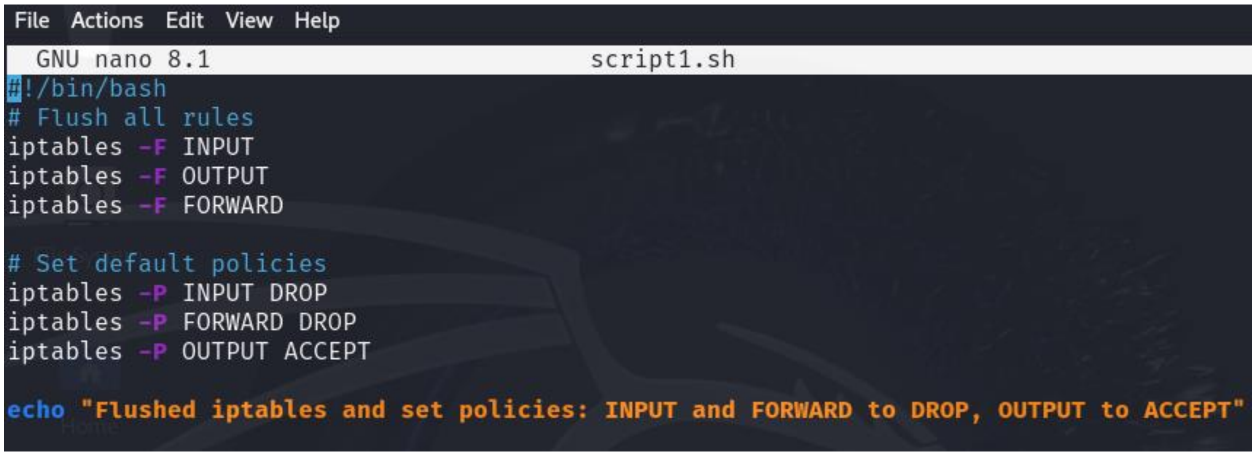
Overview :
The purpose of this lab was to set up and configure iptables firewall rules on a Linux server to accomplish specific security tasks.

1. Configure firewall rules to control incoming, outgoing, and forwarded network traffic.

2. Set up a client-server configuration using a gateway to forward traffic between different network segments.

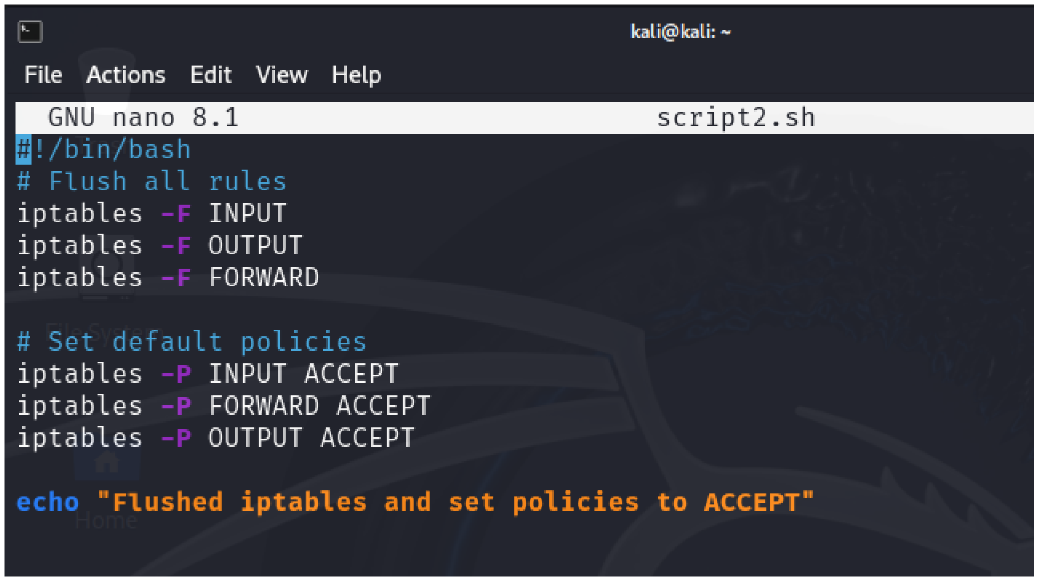
Part1

Script 1: The first script flushed the firewall rules in the INPUT, OUTPUT, as well as FORWARD chains and set the default policy to DROP for INPUT also FORWARD chains and ACCEPT for the OUTPUT chain.



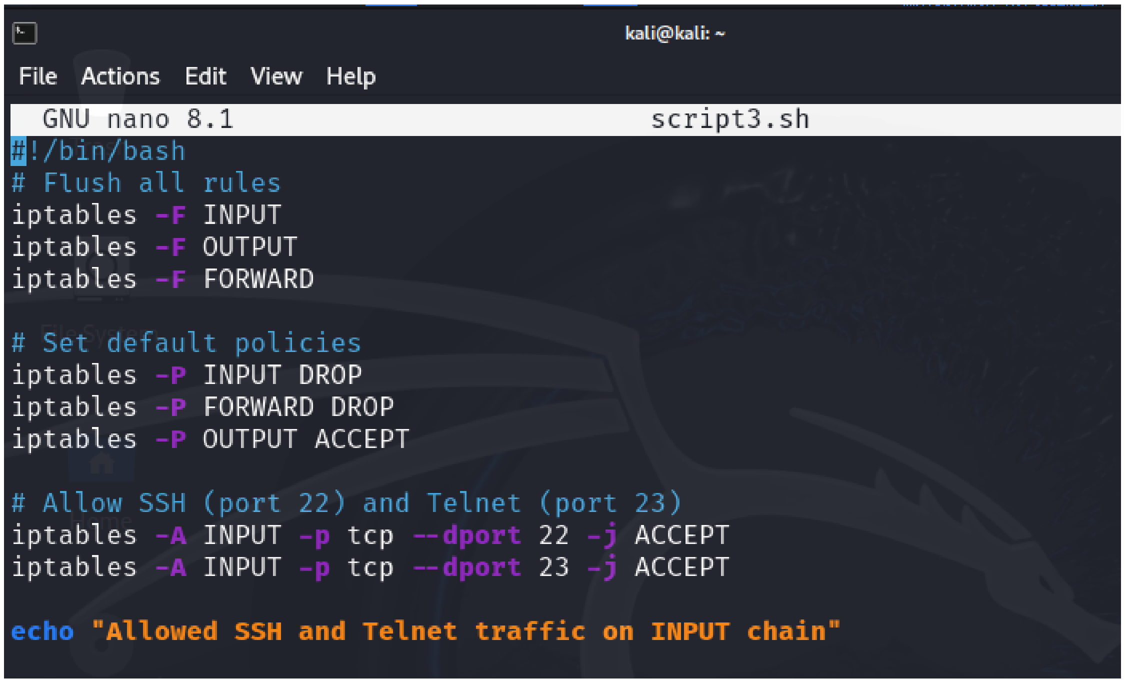
```
#!/bin/bash
# Flush all rules
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD

# Set default policies
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

echo "Flushed iptables and set policies: INPUT and FORWARD to DROP, OUTPUT to ACCEPT"
```

Script 2: The second script was made to flush all the rules and also to set the policy for all chains to ACCEPT. This script ensures that there are no restrictions on traffic flow.



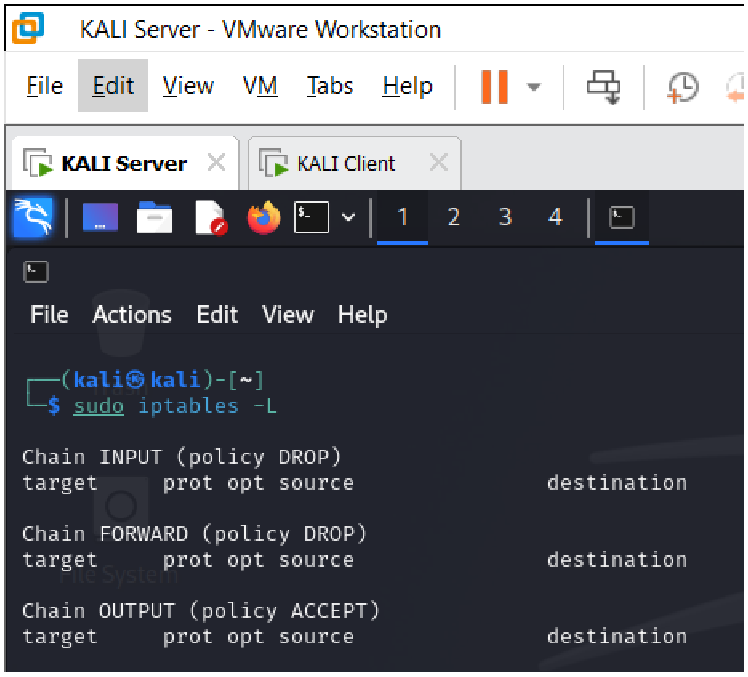
```
  GNU nano 8.1                                script2.sh
#!/bin/bash
# Flush all rules
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD

# Set default policies
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT

echo "Flushed iptables and set policies to ACCEPT"
```

Script 3: The third script was like the first but with the addition of allowing Telnet (port 23) as well as SSH (port 22) traffic on the INPUT chain.



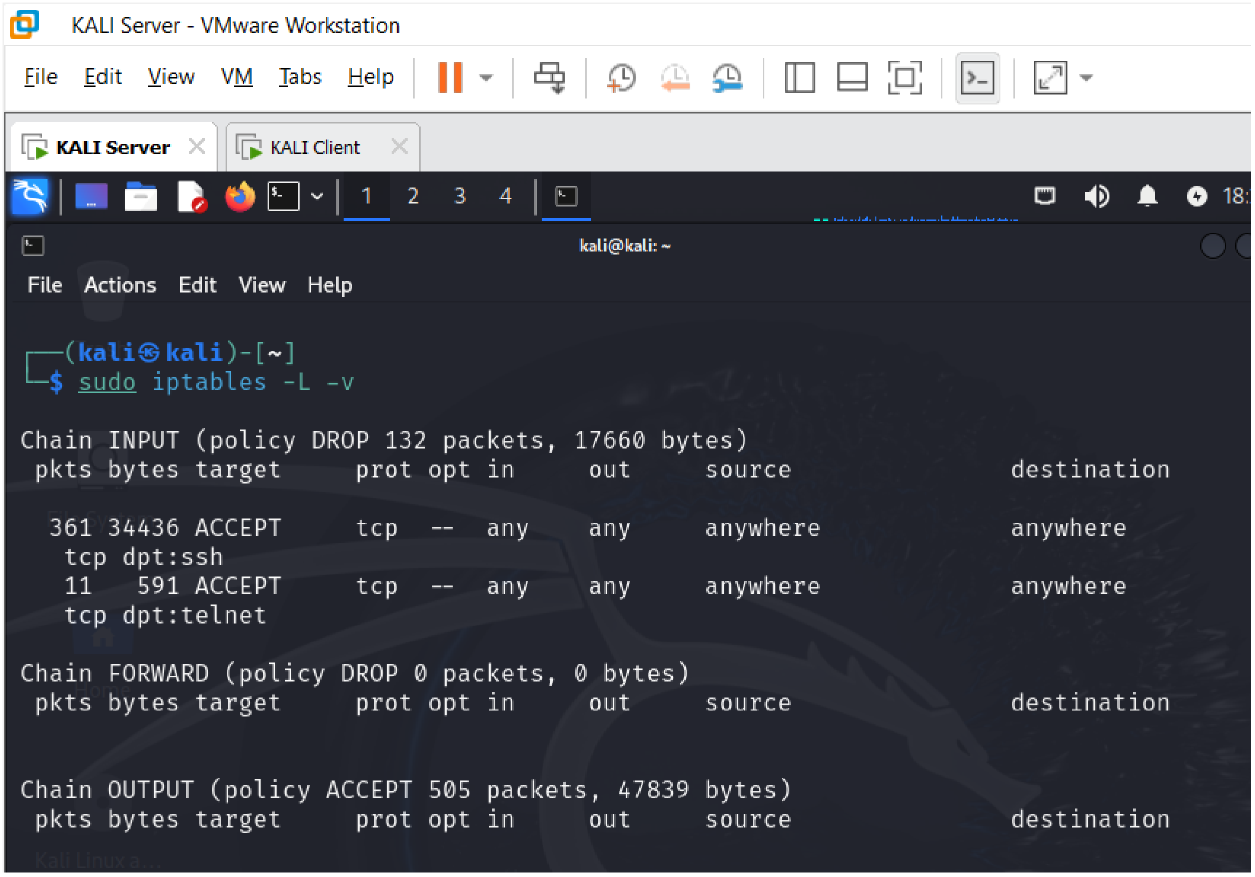
```
  GNU nano 8.1                                script3.sh
#!/bin/bash
# Flush all rules
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD

# Set default policies
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

# Allow SSH (port 22) and Telnet (port 23)
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 23 -j ACCEPT

echo "Allowed SSH and Telnet traffic on INPUT chain"
```

After running Script 1, I used the iptables -L command to list the current rules as well as verified that all chains were set as expected.


KALI Server - VMware Workstation
File Edit View VM Tabs Help
KALI Server
KALI Client
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -L

Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

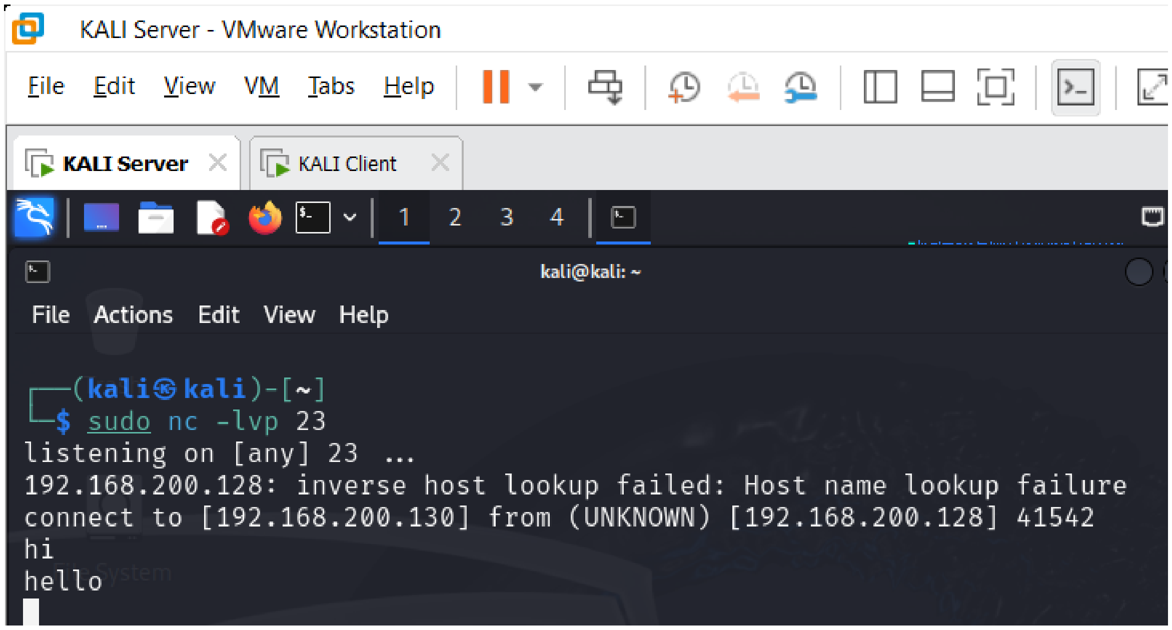
Then, after running Script 3, I ran iptables -L again to verify that only Telnet and SSH traffic were allowed.


KALI Server - VMware Workstation
File Edit View VM Tabs Help
KALI Server
KALI Client
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -L -v

Chain INPUT (policy DROP 132 packets, 17660 bytes)
 pkts bytes target     prot opt in     out     source               destination

  361 34436 ACCEPT     tcp  --  any    any     anywhere             anywhere
   tcp dpt:ssh
   11   591 ACCEPT     tcp  --  any    any     anywhere             anywhere
   tcp dpt:telnet

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain OUTPUT (policy ACCEPT 505 packets, 47839 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

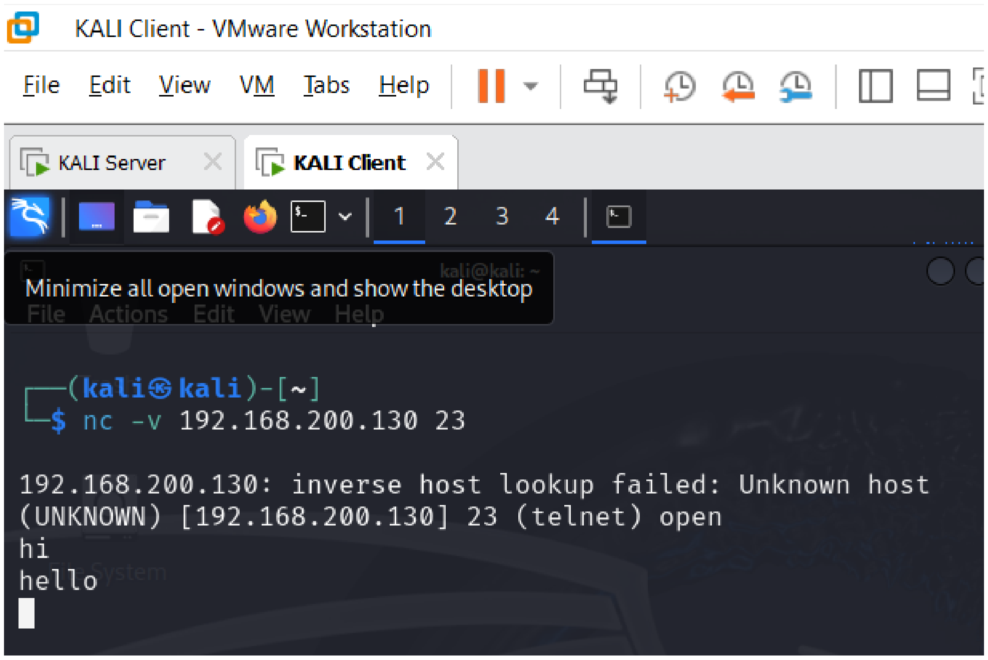
After That, I tested the connection to the server using both SSH as well as Telnet. For this, I used netcat as it simplifies the process. After connecting, I checked the firewall's packet counters using iptables -L to confirm that the firewall had processed Telnet and SSH packets.

telnet (server) :


KALI Server - VMware Workstation
File Edit View VM Tabs Help
KALI Server
KALI Client
kali@kali: ~
File Actions Edit View Help

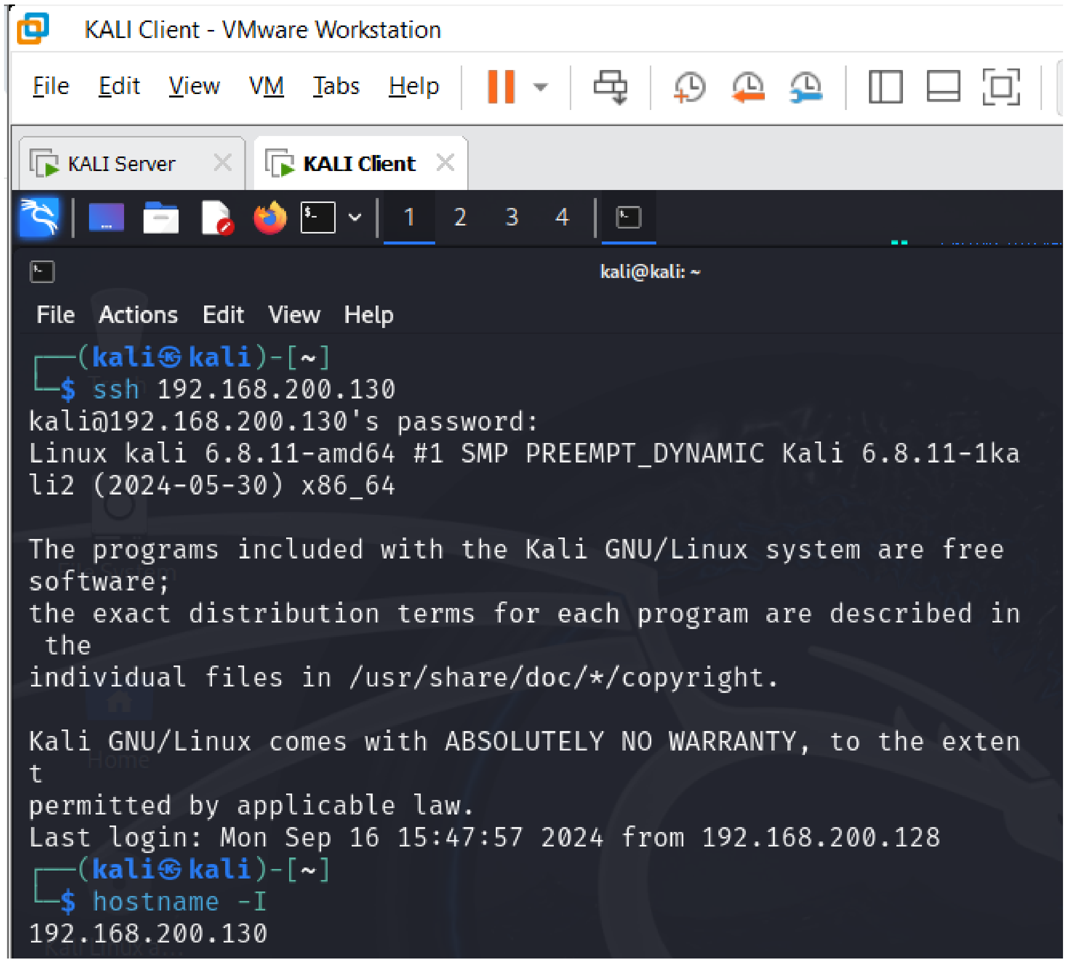
```
┌──(kali㉿kali)-[~]
└─$ sudo nc -lvp 23
listening on [any] 23 ...
192.168.200.128: inverse host lookup failed: Host name lookup failure
connect to [192.168.200.130] from (UNKNOWN) [192.168.200.128] 41542
hi
hello
```

Telnet (client) :


KALI Client - VMware Workstation
File Edit View VM Tabs Help
KALI Server
KALI Client
Minimize all open windows and show the desktop
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ nc -v 192.168.200.130 23

192.168.200.130: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.200.130] 23 (telnet) open
hi
hello
```

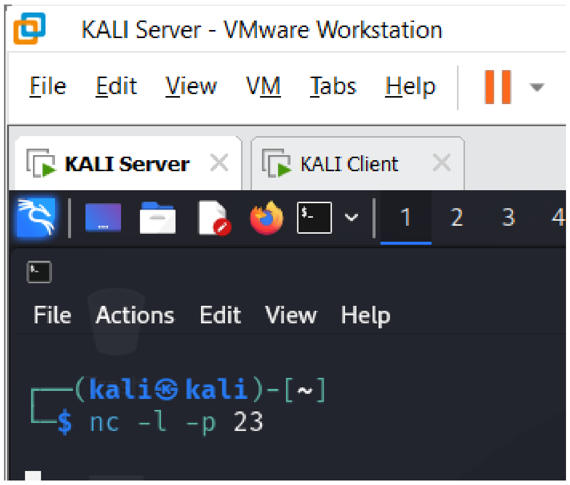
SHH:


KALI Client - VMware Workstation
File Edit View VM Tabs Help
KALI Server
KALI Client
1 2 3 4
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ssh 192.168.200.130
kali@192.168.200.130's password:
Linux kali 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1ka
li2 (2024-05-30) x86_64

The programs included with the Kali GNU/Linux system are free
software;
the exact distribution terms for each program are described in
 the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the exten
t
permitted by applicable law.
Last login: Mon Sep 16 15:47:57 2024 from 192.168.200.128
┌──(kali㉿kali)-[~]
└─$ hostname -I
192.168.200.130
```

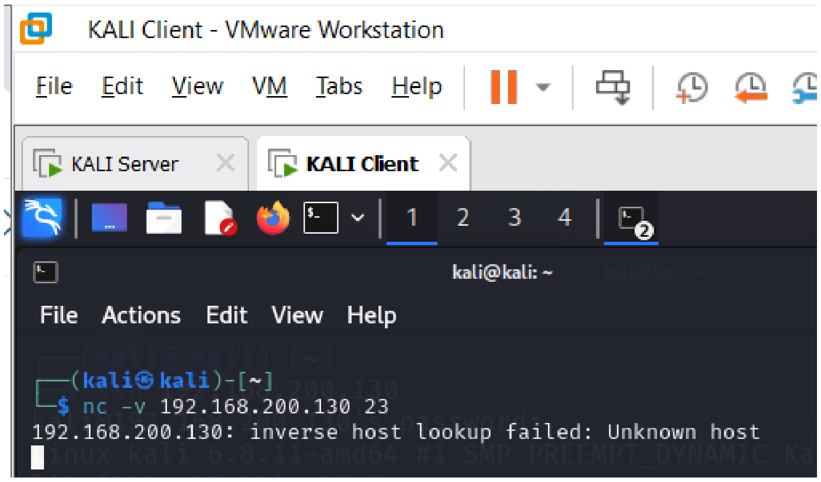
Finally, I ran again Script 1, which dropped all connections again. I then attempted to connect using SSH and Telnet and confirmed that the connections were blocked.

Server (telnet) :


KALI Server - VMware Workstation
File Edit View VM Tabs Help
KALI Server
KALI Client
1 2 3 4
File Actions Edit View Help

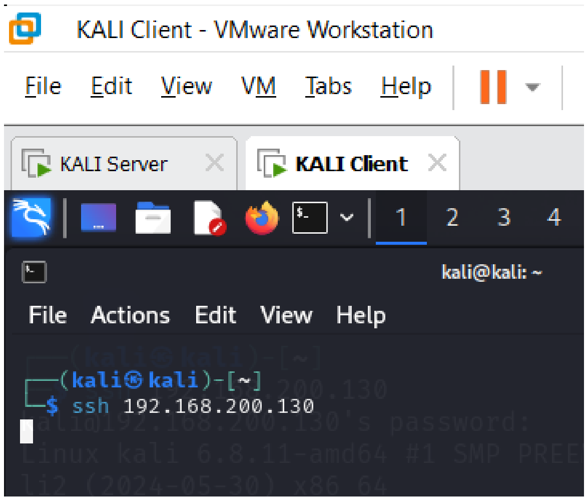
```
┌──(kali㉿kali)-[~]
└─$ nc -l -p 23
```

Client (telnet) :


KALI Client - VMware Workstation
File Edit View VM Tabs Help
KALI Server
KALI Client
1 2 3 4
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ nc -v 192.168.200.130 23
192.168.200.130: inverse host lookup failed: Unknown host
```

SSH:


KALI Client - VMware Workstation
File Edit View VM Tabs Help
KALI Server
KALI Client
1 2 3 4
kali@kali: ~
File Actions Edit View Help

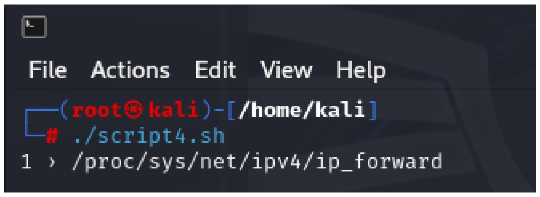
```
┌──(kali㉿kali)-[~]
└─$ ssh 192.168.200.130
```

Part 2

Routing :

For this part, the setup required a client, a server, as well as a gateway system. configuring the gateway to have two network interfaces: one connected to the client network and the other to the server network.

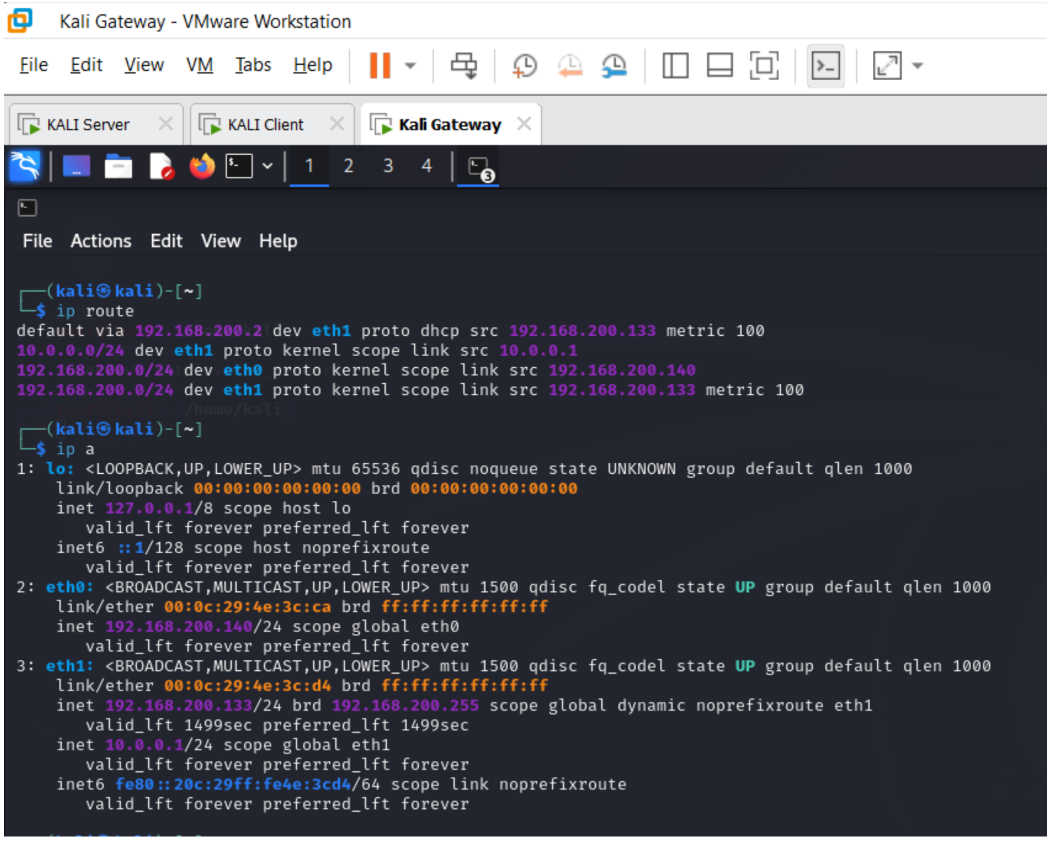
I enabled IP forwarding on the gateway using the command:
echo 1 > /proc/sys/net/ipv4/ip_forward



```
┌──(root㉿kali)-[/home/kali]
└─# ./script4.sh
1 › /proc/sys/net/ipv4/ip_forward
```

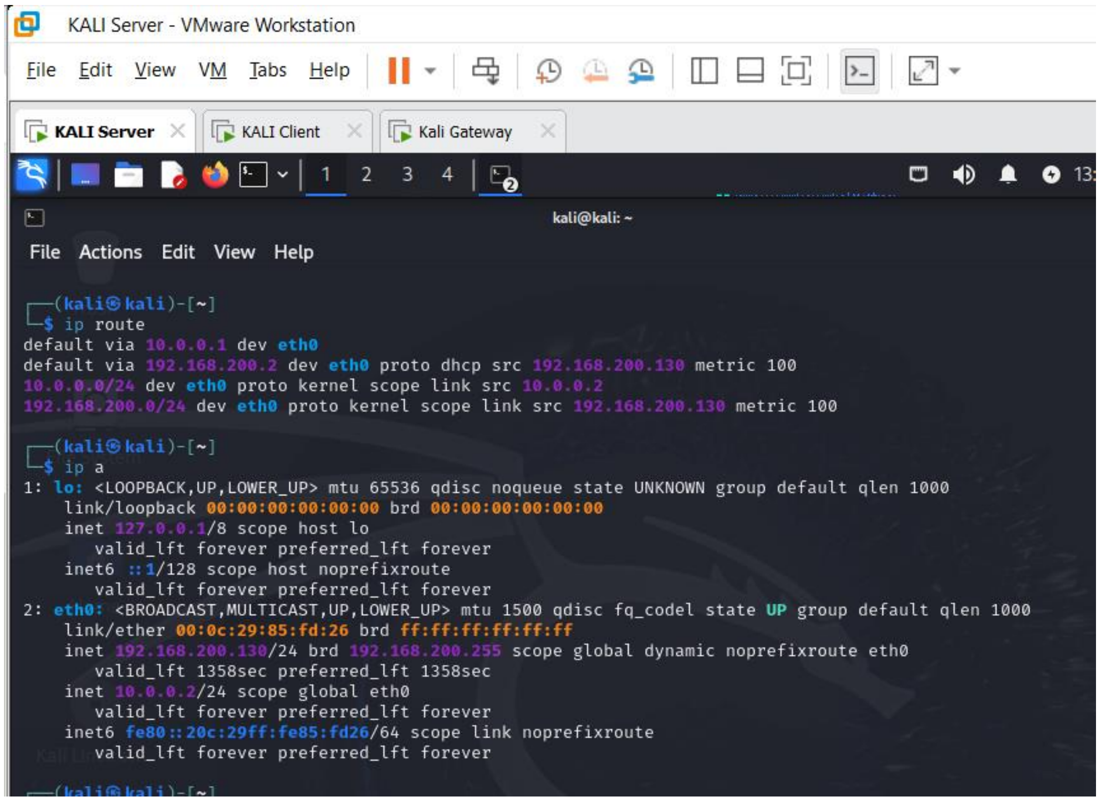
This allows the gateway to forward packets between the client and server networks.

IP route and IP address of Server, Client, and Gateway.
Gateway:


Kali Gateway - VMware Workstation
File Edit View VM Tabs Help
KALI Server
KALI Client
Kali Gateway
1 2 3 4
File Actions Edit View Help

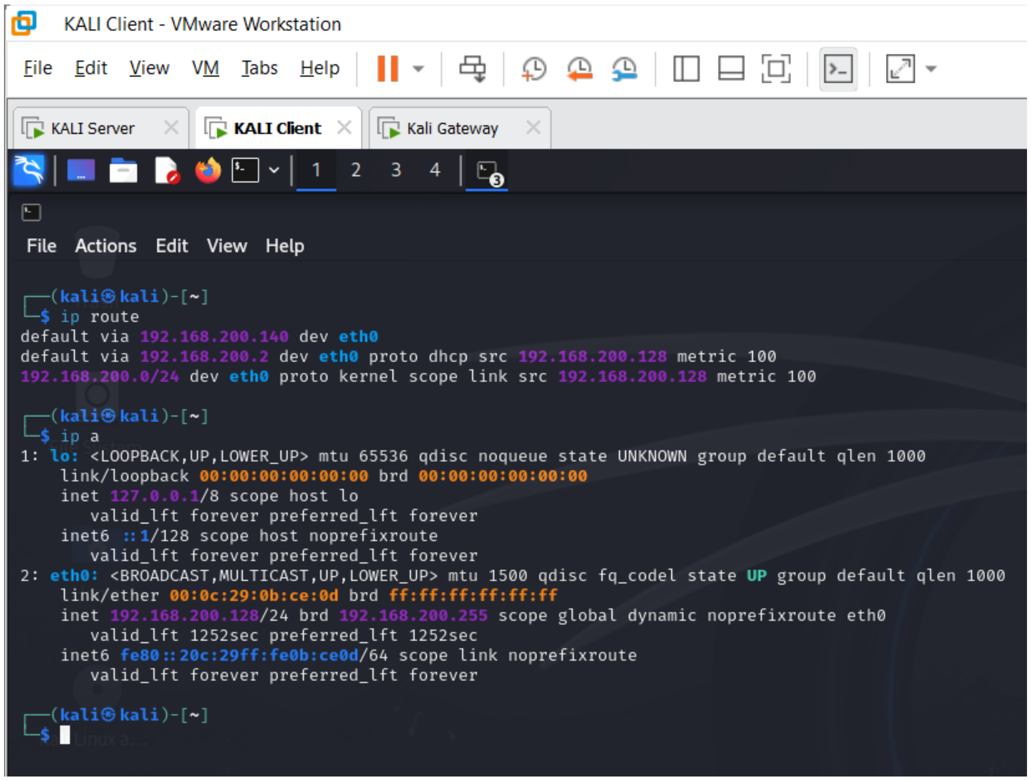
```
┌──(kali㉿kali)-[~]
└─$ ip route
default via 192.168.200.2 dev eth1 proto dhcp src 192.168.200.133 metric 100
10.0.0.0/24 dev eth1 proto kernel scope link src 10.0.0.1
192.168.200.0/24 dev eth0 proto kernel scope link src 192.168.200.140
192.168.200.0/24 dev eth1 proto kernel scope link src 192.168.200.133 metric 100

┌──(kali㉿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:29:4e:3c:ca brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.140/24 scope global eth0
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:4e:3c:d4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.133/24 brd 192.168.200.255 scope global dynamic noprefixroute eth1
       valid_lft 1499sec preferred_lft 1499sec
    inet 10.0.0.1/24 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe4e:3cd4/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Server:


KALI Server - VMware Workstation
File Edit View VM Tabs Help
KALI Server
KALI Client
Kali Gateway
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ip route
default via 10.0.0.1 dev eth0
default via 192.168.200.2 dev eth0 proto dhcp src 192.168.200.130 metric 100
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.2
192.168.200.0/24 dev eth0 proto kernel scope link src 192.168.200.130 metric 100

┌──(kali㉿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:29:85:fd:26 brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.130/24 brd 192.168.200.255 scope global dynamic noprefixroute eth0
       valid_lft 1358sec preferred_lft 1358sec
    inet 10.0.0.2/24 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe85:fd26/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
```

Client :

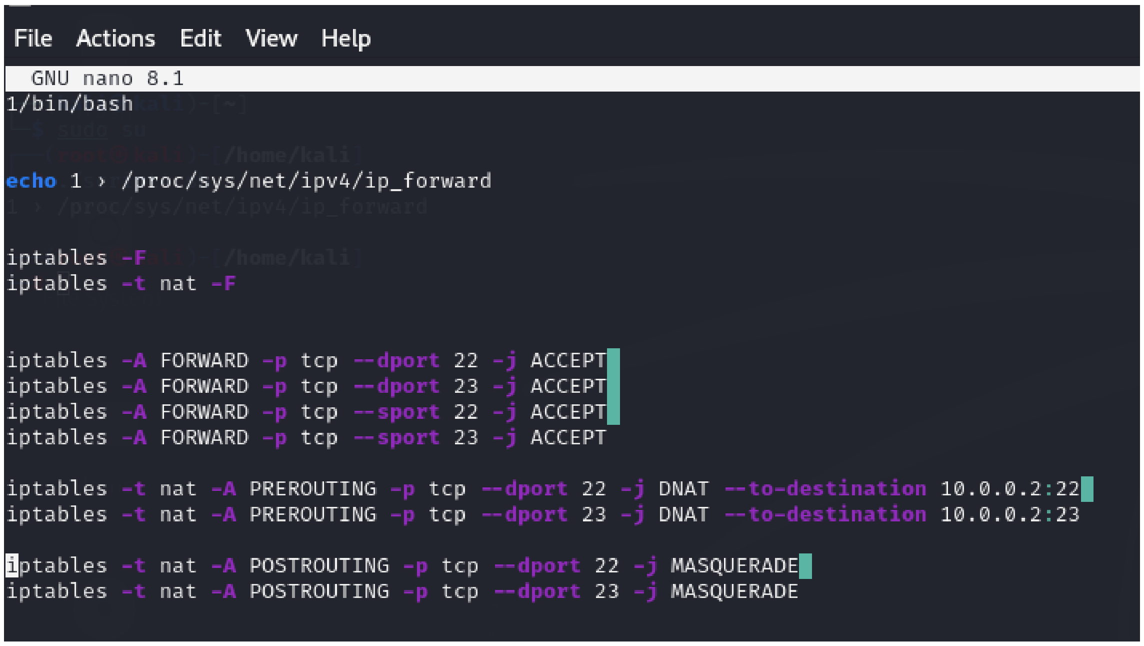

KALI Client - VMware Workstation
File Edit View VM Tabs Help
KALI Server
KALI Client
Kali Gateway
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ip route
default via 192.168.200.140 dev eth0
default via 192.168.200.2 dev eth0 proto dhcp src 192.168.200.128 metric 100
192.168.200.0/24 dev eth0 proto kernel scope link src 192.168.200.128 metric 100

┌──(kali㉿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:29:0b:ce:0d brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.128/24 brd 192.168.200.255 scope global dynamic noprefixroute eth0
       valid_lft 1252sec preferred_lft 1252sec
    inet6 fe80::20c:29ff:fe0b:ce0d/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ 
```

Forwarding :

To forward SSH and Telnet traffic through the gateway, I used the following script. This script set up DNAT to forward incoming SSH as well as Telnet traffic to the server (IP: 10.0.0.2) and also uses SNAT for outgoing traffic.

DNAT Script:

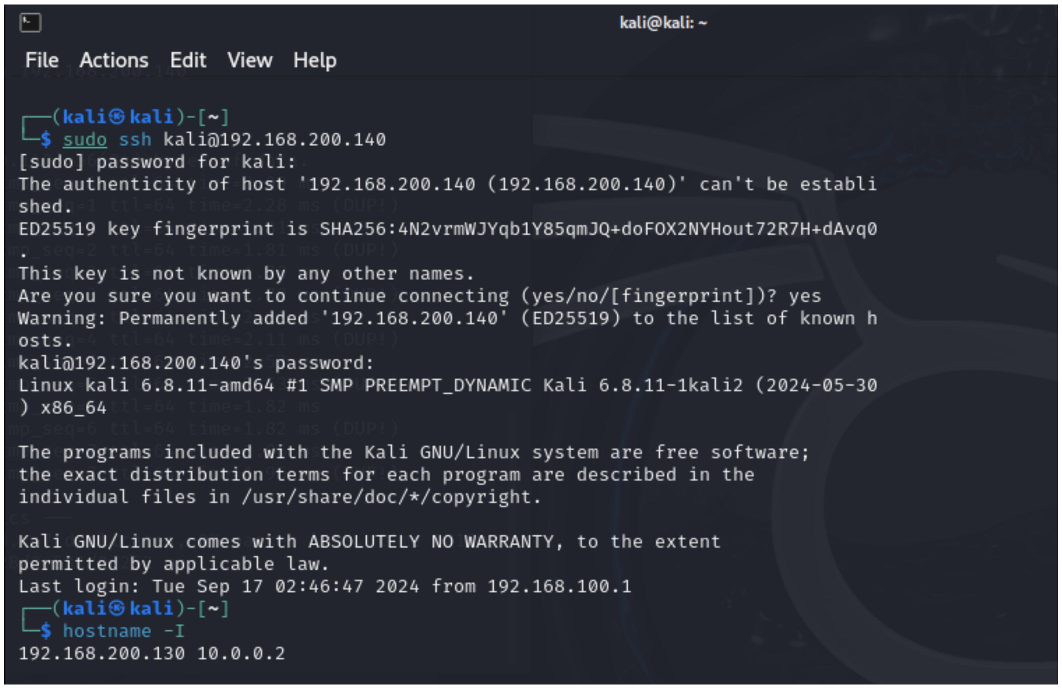


```
1/bin/bash

echo 1 › /proc/sys/net/ipv4/ip_forward

iptables -F
iptables -t nat -F


iptables -A FORWARD -p tcp --dport 22 -j ACCEPT
iptables -A FORWARD -p tcp --dport 23 -j ACCEPT
iptables -A FORWARD -p tcp --sport 22 -j ACCEPT
iptables -A FORWARD -p tcp --sport 23 -j ACCEPT

iptables -t nat -A PREROUTING -p tcp --dport 22 -j DNAT --to-destination 10.0.0.2:22
iptables -t nat -A PREROUTING -p tcp --dport 23 -j DNAT --to-destination 10.0.0.2:23

iptables -t nat -A POSTROUTING -p tcp --dport 22 -j MASQUERADE
iptables -t nat -A POSTROUTING -p tcp --dport 23 -j MASQUERADE
```

Connection after running script

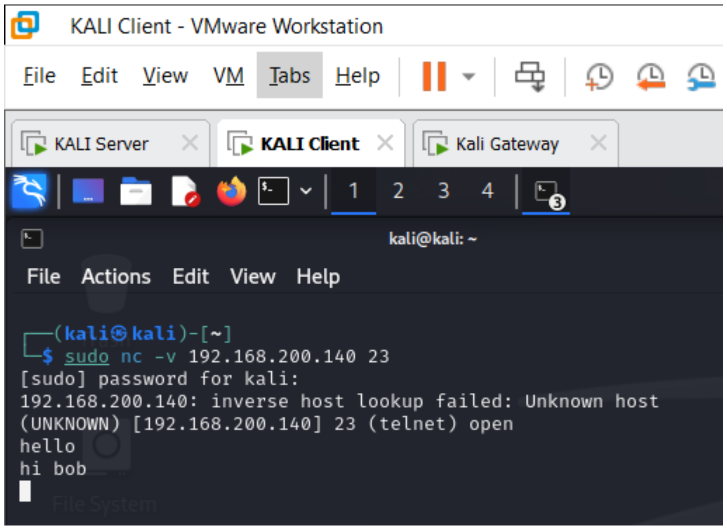
I connected to the gateway using SSH and Telnet from the client. The gateway forwarded the connections to the server, as verified through successful SSH and Telnet logins. I used netcat for the Telnet connection.

SSH connection



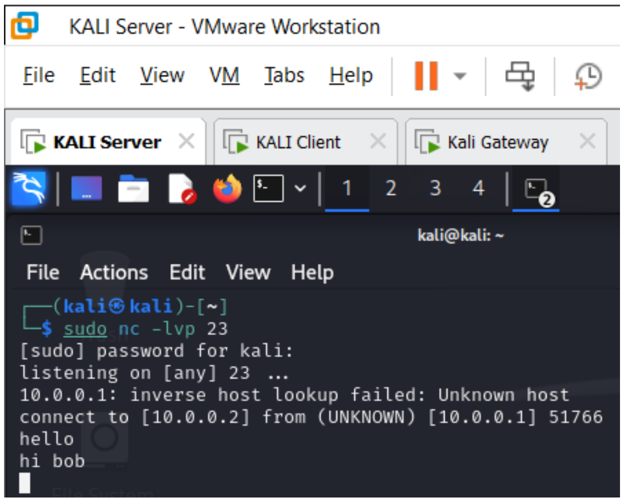
```
┌──(kali㉿kali)-[~]
└─$ sudo ssh kali@192.168.200.140
[sudo] password for kali:
The authenticity of host '192.168.200.140 (192.168.200.140)' can't be establi
shed.
ED25519 key fingerprint is SHA256:4N2vrmWJYqb1Y85qmJQ+doFOX2NYHout72R7H+dAvq0
.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.200.140' (ED25519) to the list of known h
osts.
kali@192.168.200.140's password:
Linux kali 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30
) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Sep 17 02:46:47 2024 from 192.168.100.1
┌──(kali㉿kali)-[~]
└─$ hostname -I
192.168.200.130 10.0.0.2
```

NC form client :


KALI Client - VMware Workstation
File Edit View VM Tabs Help
KALI Server
KALI Client
Kali Gateway
1 2 3 4
kali@kali: ~
File Actions Edit View Help

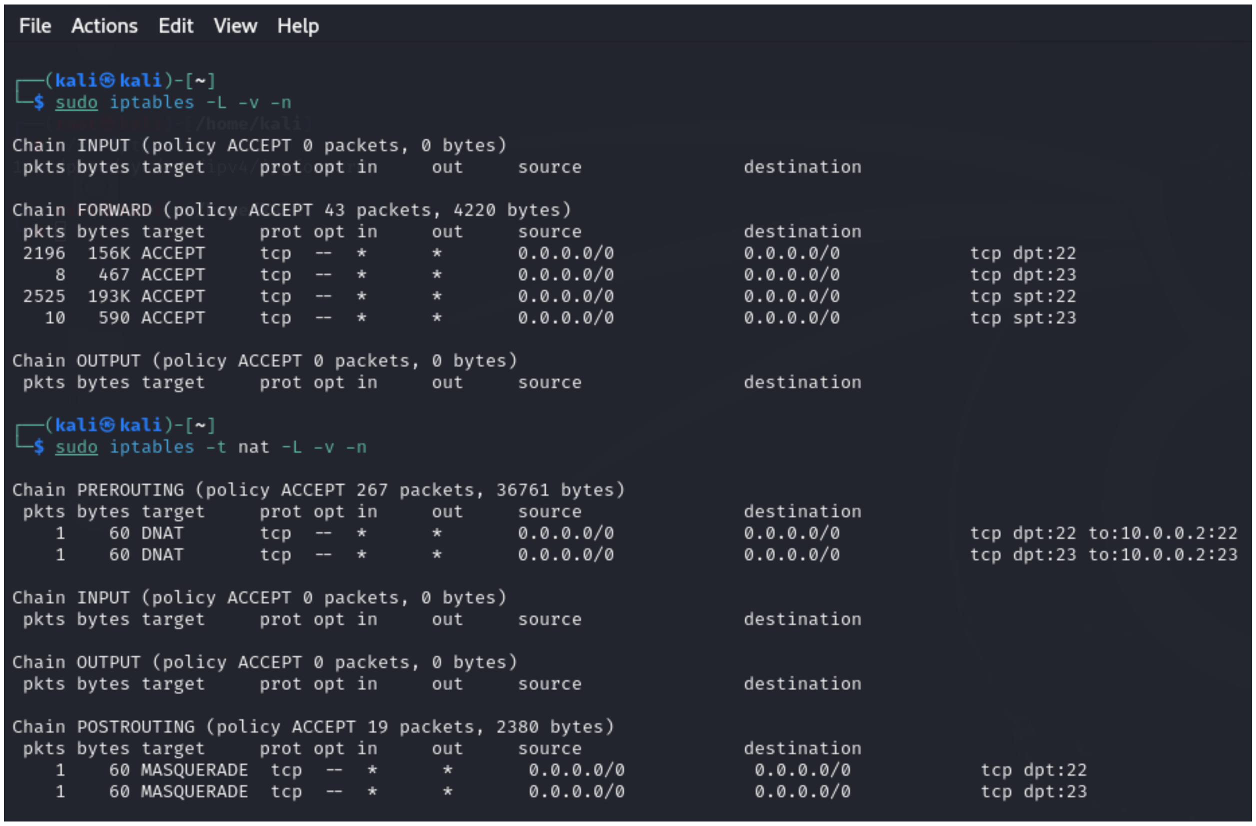
```
┌──(kali㉿kali)-[~]
└─$ sudo nc -v 192.168.200.140 23
[sudo] password for kali:
192.168.200.140: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.200.140] 23 (telnet) open
hello
hi bob
█
```

NC from server:


KALI Server - VMware Workstation
File Edit View VM Tabs Help
KALI Server
KALI Client
Kali Gateway
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ sudo nc -lvp 23
[sudo] password for kali:
listening on [any] 23 ...
10.0.0.1: inverse host lookup failed: Unknown host
connect to [10.0.0.2] from (UNKNOWN) [10.0.0.1] 51766
hello
hi bob
```

Verified server through IP tables and packets :



```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -L -v -n

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 43 packets, 4220 bytes)
 pkts bytes target     prot opt in     out     source               destination
 2196  156K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp dpt:22
    8   467 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp dpt:23
 2525  193K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp spt:22
   10   590 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp spt:23

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

┌──(kali㉿kali)-[~]
└─$ sudo iptables -t nat -L -v -n

Chain PREROUTING (policy ACCEPT 267 packets, 36761 bytes)
 pkts bytes target     prot opt in     out     source               destination
    1    60 DNAT       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp dpt:22 to:10.0.0.2:22
    1    60 DNAT       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp dpt:23 to:10.0.0.2:23

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 19 packets, 2380 bytes)
 pkts bytes target     prot opt in     out     source               destination
    1    60 MASQUERADE  tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp dpt:22
    1    60 MASQUERADE  tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp dpt:23
```

Conclusion:

Through the completion of this lab, I gained a practical understanding of configuring iptables firewall rules, enabling IP forwarding, and implementing DNAT and SNAT to route traffic across a gateway. The lab provided valuable insights into managing traffic flows and network security using iptables, and how these tools are critical for secure network operations.