FreeBSD Network Security Exercise

Networks and Systems Security II (CSE354/554)

Table of Contents

Lab Overview

This lab configures a **FreeBSD-based firewall (VM2)** to perform **bidirectional NAT** and traffic filtering using `pf`, while **VM3** hosts a web server with strict file permissions enforced by **ACLs**. Key objectives:

1. Isolate VM1 (client) and VM3 (server) into separate subnets.
2. Route traffic through VM2 (firewall) with NAT.
3. Restrict VM3's web server to ports 80/443.
4. Use Linux ACLs to control access to the web directory.

Virtual Machine Setup

VM Specifications

VM	Role	OS	RAM	CPU	Interfaces
VM1	Client	FreeBSD 14	1GB	1	`iface1` # em0 (LAN)
VM2	Firewall/NAT	FreeBSD 14	1GB	1	`iface1` # em0, `iface2` # em1
VM3	Web Server	Linux/FreeBSD	1GB	1	`iface2` # em1, `em1` (WAN)

Network Topology

```
[VM1: 10.0.1.10]
    │
    ├── (iface1: # em0 10.0.1.1)
    │
[VM2: Firewall/NAT]
    │
    ├── (iface2: # em1 10.0.2.1)
        │
        └── [VM3: 10.0.2.10]
               │
               └── [External: 192.168.8.150]
```

- **Subnet 1**: `10.0.1.0/24` (VM1 ↔ VM2)
- **Subnet 2**: `10.0.2.0/24` (VM2 ↔ VM3)
- **External**: `192.168.8.0/24` (VM3's internet access)

Task 1: Firewall & NAT Configuration

Step 1: Assign Static IP Addresses

On VM1 (Client):

```
ifconfig iface1  # em0inet 10.0.1.10 netmask 255.255.255.0
route add default 10.0.1.1  # Set VM2 as gateway
```

On VM2 (Firewall):

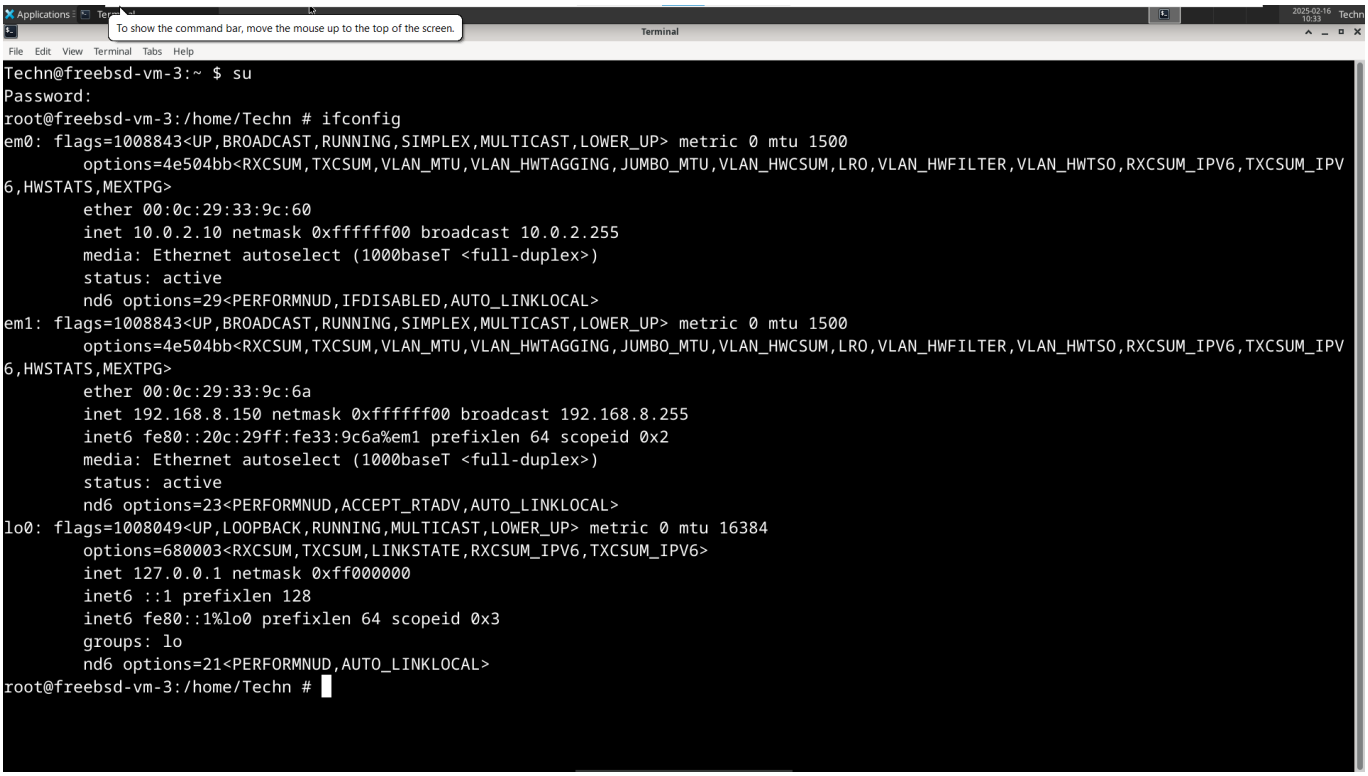
```
ifconfig iface1  # em0inet 10.0.1.1 netmask 255.255.255.0
ifconfig iface2  # em1inet 10.0.2.1 netmask 255.255.255.0
```

On VM3 (Server):

```
ifconfig iface2  # em1inet 10.0.2.10 netmask 255.255.255.0
ifconfig em1 inet 192.168.8.150 netmask 255.255.255.0
route add default 192.168.8.1  # External gateway
```

```
technogenius@freebsd-vm-1:/ $ su
Password:
root@freebsd-vm-1:/ # ifconfig
em0: flags=1008843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,LOWER_UP> metric 0 mtu 1500
        options=4e504bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,LRO,VLAN_HWFILTER,VLAN_HWTSO,RXCSUM_IPV6,TXCSUM_IPV
6,HWSTATS,MEXTPG>
        ether 00:0c:29:f8:e9:0e
        inet 10.0.1.10 netmask 0xffffff00 broadcast 10.0.1.255
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
em1: flags=1008843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,LOWER_UP> metric 0 mtu 1500
        options=4e504bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,LRO,VLAN_HWFILTER,VLAN_HWTSO,RXCSUM_IPV6,TXCSUM_IPV
6,HWSTATS,MEXTPG>
        ether 00:0c:29:f8:e9:18
        inet 192.168.8.148 netmask 0xffffff00 broadcast 192.168.8.255
        inet6 fe80::20c:29ff:fef8:e918%em1 prefixlen 64 scopeid 0x2
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
        nd6 options=23<PERFORMNUD,ACCEPT_RTADV,AUTO_LINKLOCAL>
lo0: flags=1008049<UP,LOOPBACK,RUNNING,MULTICAST,LOWER_UP> metric 0 mtu 16384
        options=680003<RXCSUM,TXCSUM,LINKSTATE,RXCSUM_IPV6,TXCSUM_IPV6>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x3
        groups: lo
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
root@freebsd-vm-1:/ #
```

```
Techg@freebsd-vm-2:/ $ su
Password:
root@freebsd-vm-2:/ # ifconfig
em0: flags=1008843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,LOWER_UP> metric 0 mtu 1500
        options=4e504bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,LRO,VLAN_HWFILTER,VLAN_HWTSO,RXCSUM_IPV6,TXCSUM_IPV6,HWSTATS,MEXTPG>
        ether 00:0c:29:65:2f:d2
        inet 10.0.1.1 netmask 0xffffff00 broadcast 10.0.1.255
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
em1: flags=1008843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,LOWER_UP> metric 0 mtu 1500
        options=4e504bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,LRO,VLAN_HWFILTER,VLAN_HWTSO,RXCSUM_IPV6,TXCSUM_IPV6,HWSTATS,MEXTPG>
        ether 00:0c:29:65:2f:dc
        inet 10.0.2.1 netmask 0xffffff00 broadcast 10.0.2.255
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
em2: flags=1008843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,LOWER_UP> metric 0 mtu 1500
        options=4e504bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,LRO,VLAN_HWFILTER,VLAN_HWTSO,RXCSUM_IPV6,TXCSUM_IPV6,HWSTATS,MEXTPG>
        ether 00:0c:29:65:2f:e6
        inet 192.168.8.149 netmask 0xffffff00 broadcast 192.168.8.255
        inet6 fe80::20c:29ff:fe65:2fe6%em2 prefixlen 64 scopeid 0x3
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
        nd6 options=23<PERFORMNUD,ACCEPT_RTADV,AUTO_LINKLOCAL>
lo0: flags=1008049<UP,LOOPBACK,RUNNING,MULTICAST,LOWER_UP> metric 0 mtu 16384
        options=680003<RXCSUM,TXCSUM,LINKSTATE,RXCSUM_IPV6,TXCSUM_IPV6>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x4
        groups: lo
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
pflog0: flags=1000141<UP,RUNNING,PROMISC,LOWER_UP> metric 0 mtu 33152
        options=0
        groups: pflog
root@freebsd-vm-2:/ #
```



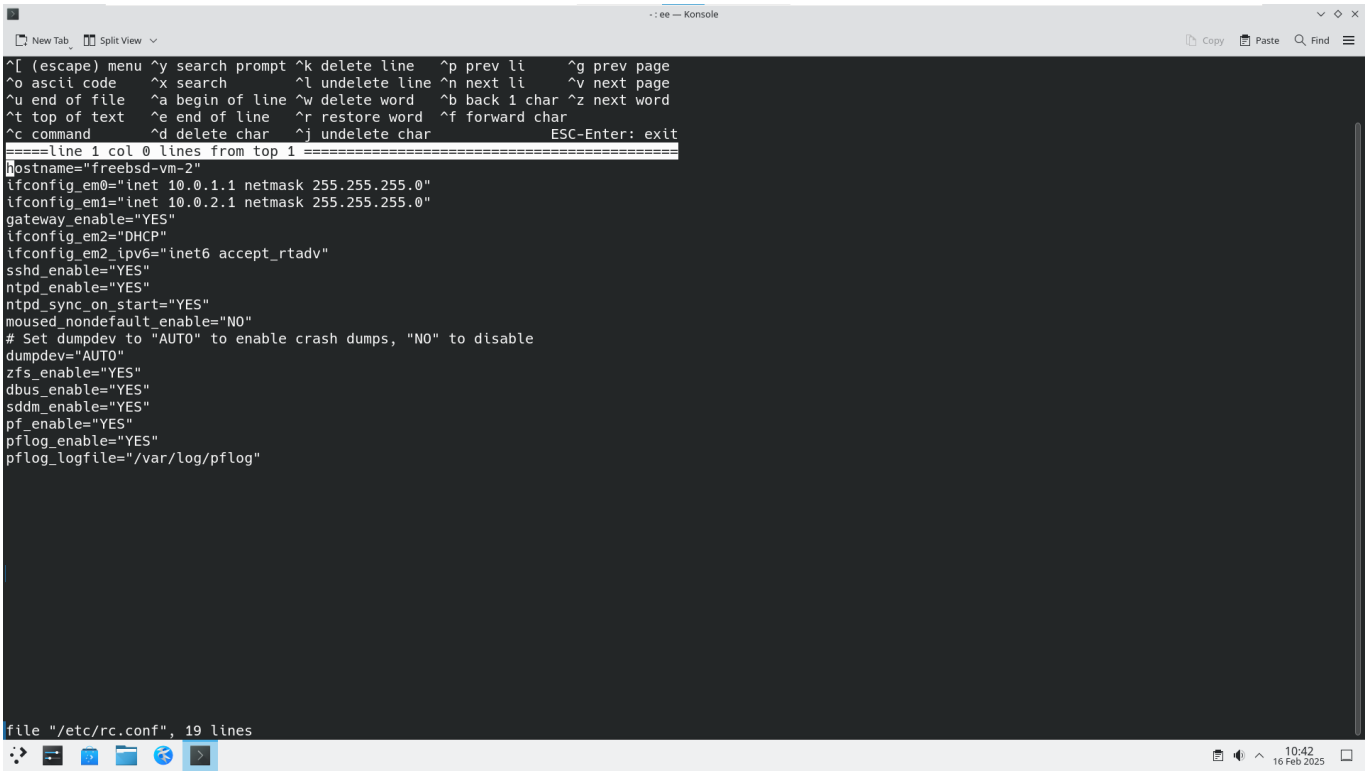
Step 2: Enable IP Forwarding on VM2

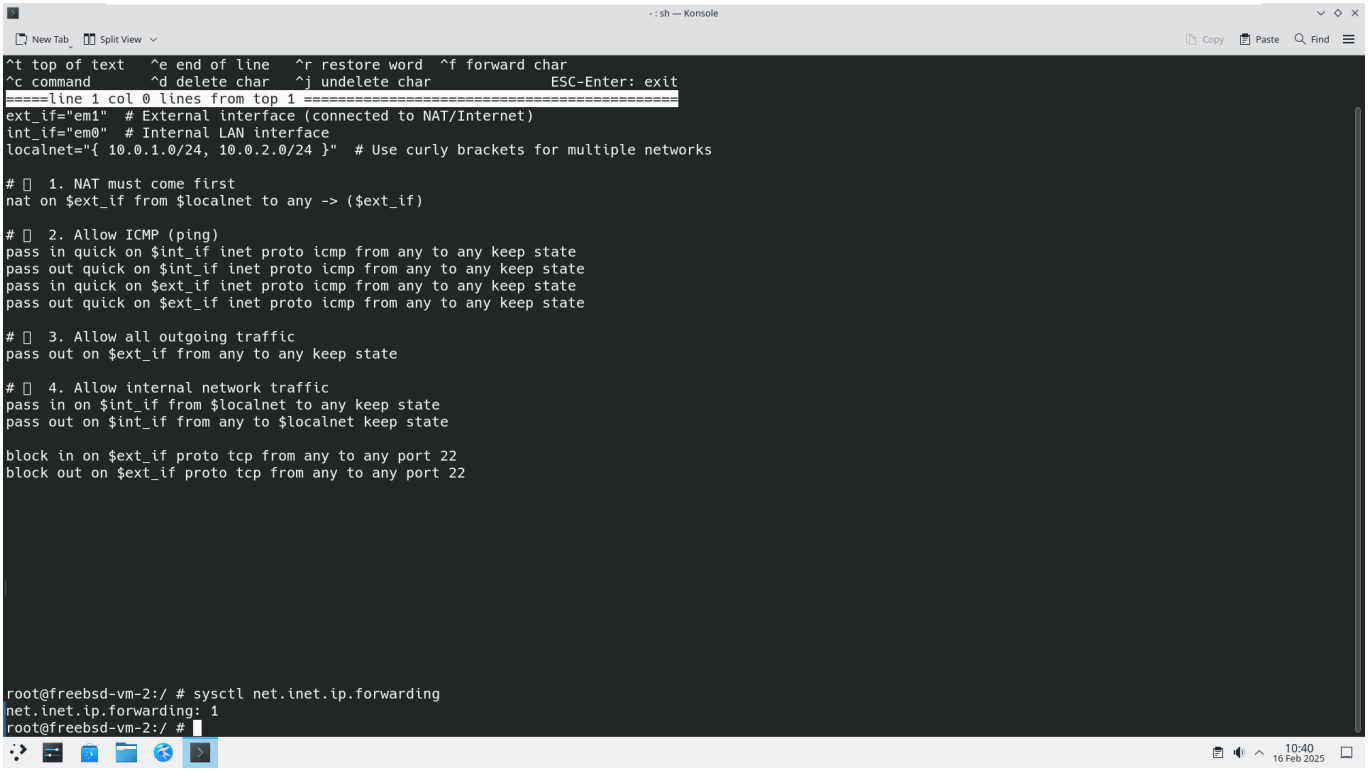
Enable packet forwarding to allow VM2 to route traffic:

```
sysctl net.inet.ip.forwarding=1
echo 'net.inet.ip.forwarding=1' >> /etc/sysctl.conf  # Persist on reboot
```

Verify:

```
sysctl net.inet.ip.forwarding  # Should return "1"
```





Step 3: Configure PF Firewall Rules

Edit `/etc/pf.conf` on VM2:

```
# Define interfaces
ext_if = "iface2" # em1
int_if = "iface1" # em0
localnet = "{ 10.0.1.0/24, 10.0.2.0/24 }"

# NAT Configuration
nat on $ext_if from $localnet to any -> ($ext_if)  # Outbound NAT
```

```
nat on $int_if from any to $ext_if -> ($int_if)    # Inbound NAT

# Traffic Rules
block all  # Default deny

# Allow internal traffic
pass in quick on $int_if from $localnet to any
pass out quick on $ext_if from any to $localnet

# Allow HTTP/HTTPS from external networks
pass in on $ext_if proto tcp from any to any port { 80, 443 }
pass out on $ext_if proto tcp from any to any port { 80, 443 }

# Block SSH (Port 22)
block in quick on $ext_if proto tcp from any to any port 22
```

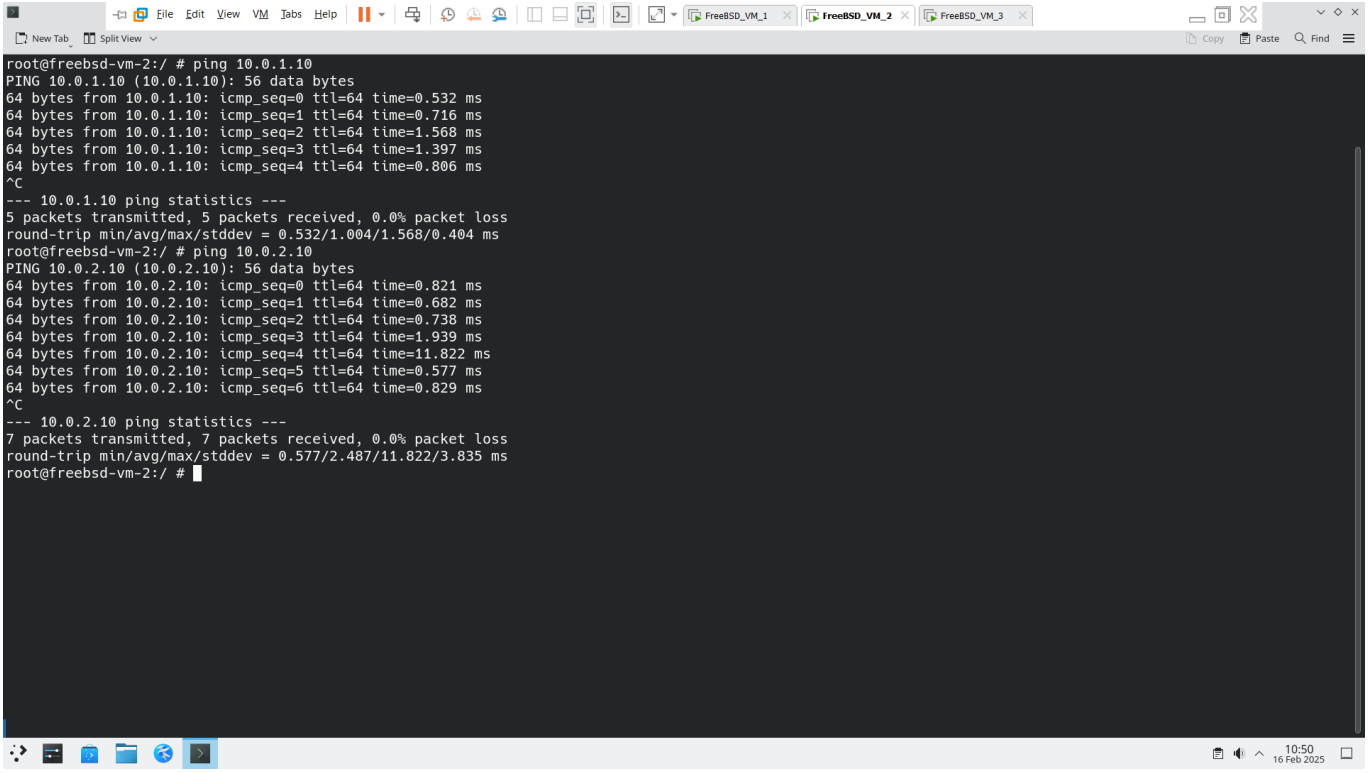
Activate Rules:

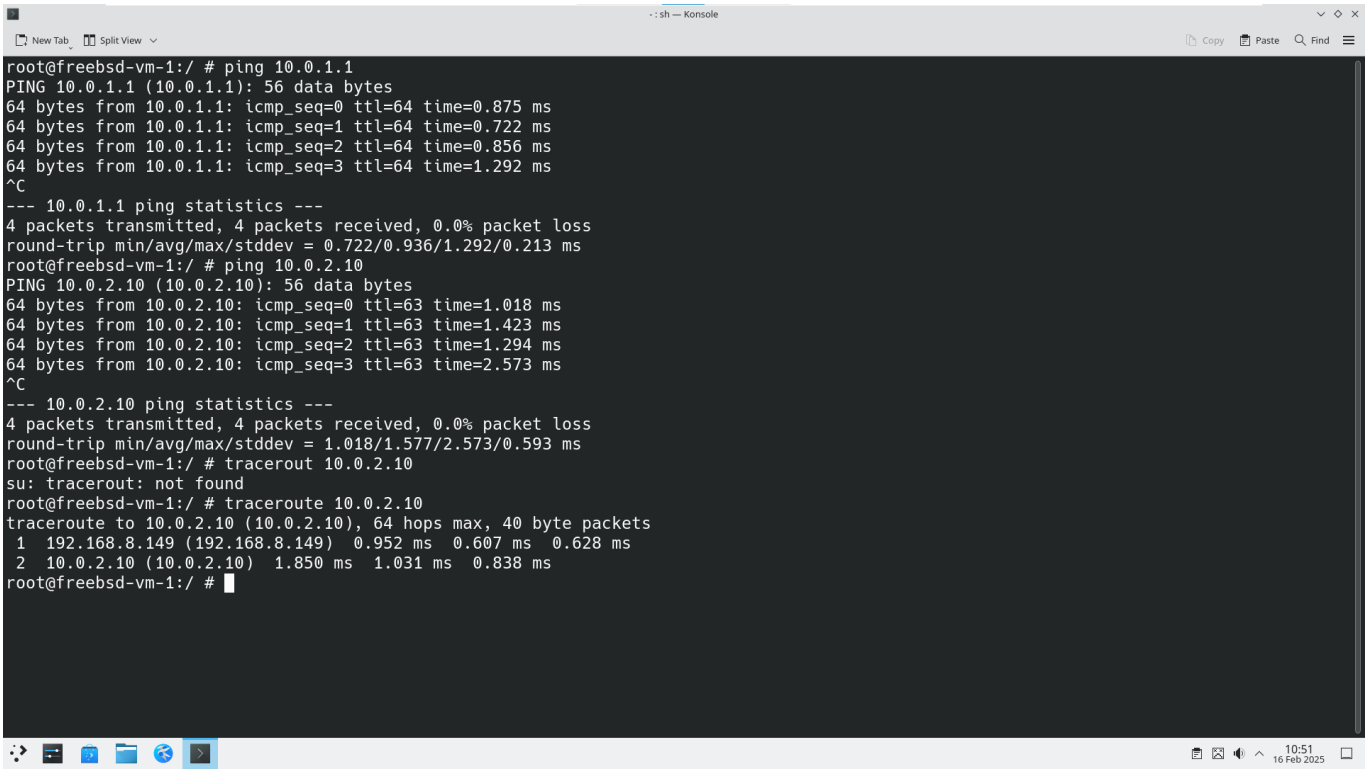
```
pfctl -f /etc/pf.conf  # Load rules
pfctl -s rules  # Verify rules (screenshot this!)
```

Step 4: Test NAT & Connectivity

From VM1:

```
ping 10.0.2.10  # Should succeed
traceroute 10.0.2.10  # Traffic must route through VM2 (10.0.1.1 → 10.0.2.1)
```



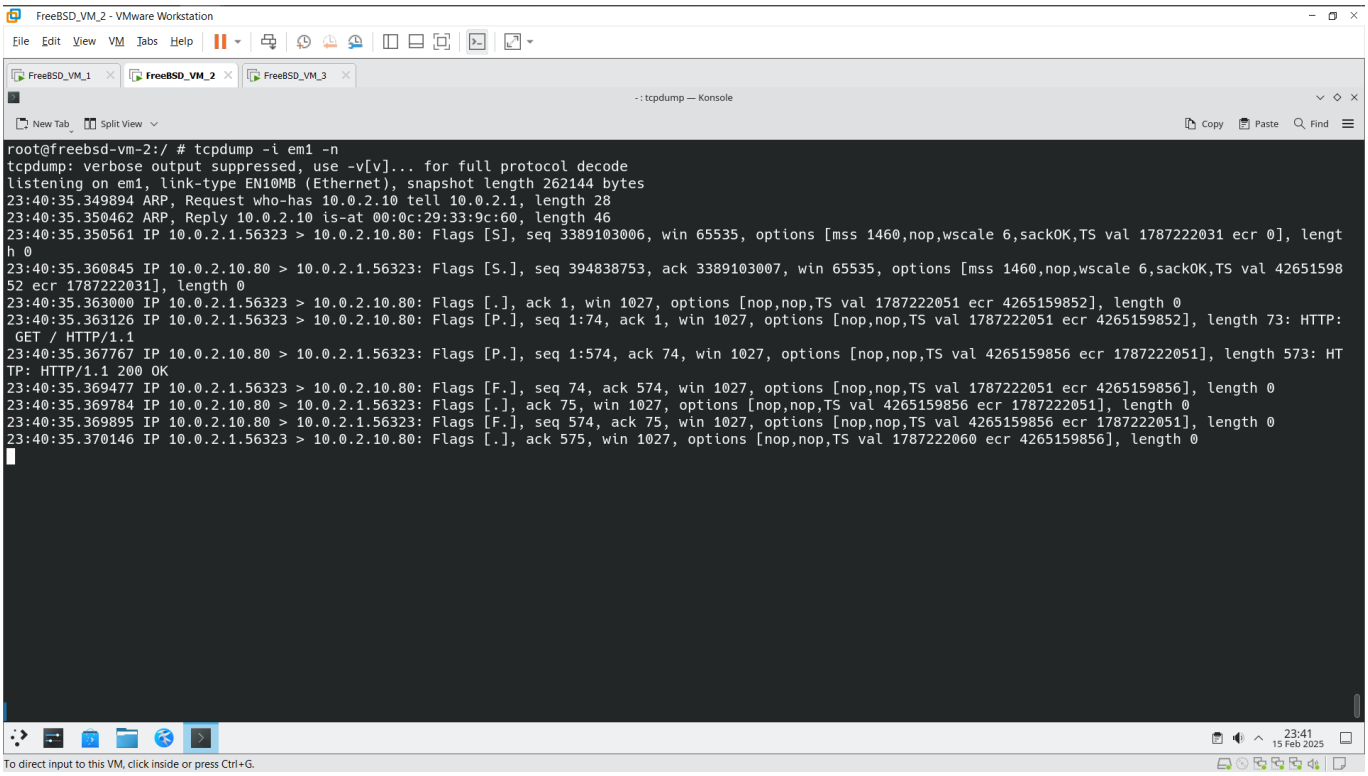


Capture NAT Traffic:

On VM2, run:

```
tcpdump -i iface2  # em1-n port 80  # Observe NAT translation
```

- **Expected Result**: Requests from VM1 (10.0.1.10) appear as coming from VM2 (10.0.2.1).



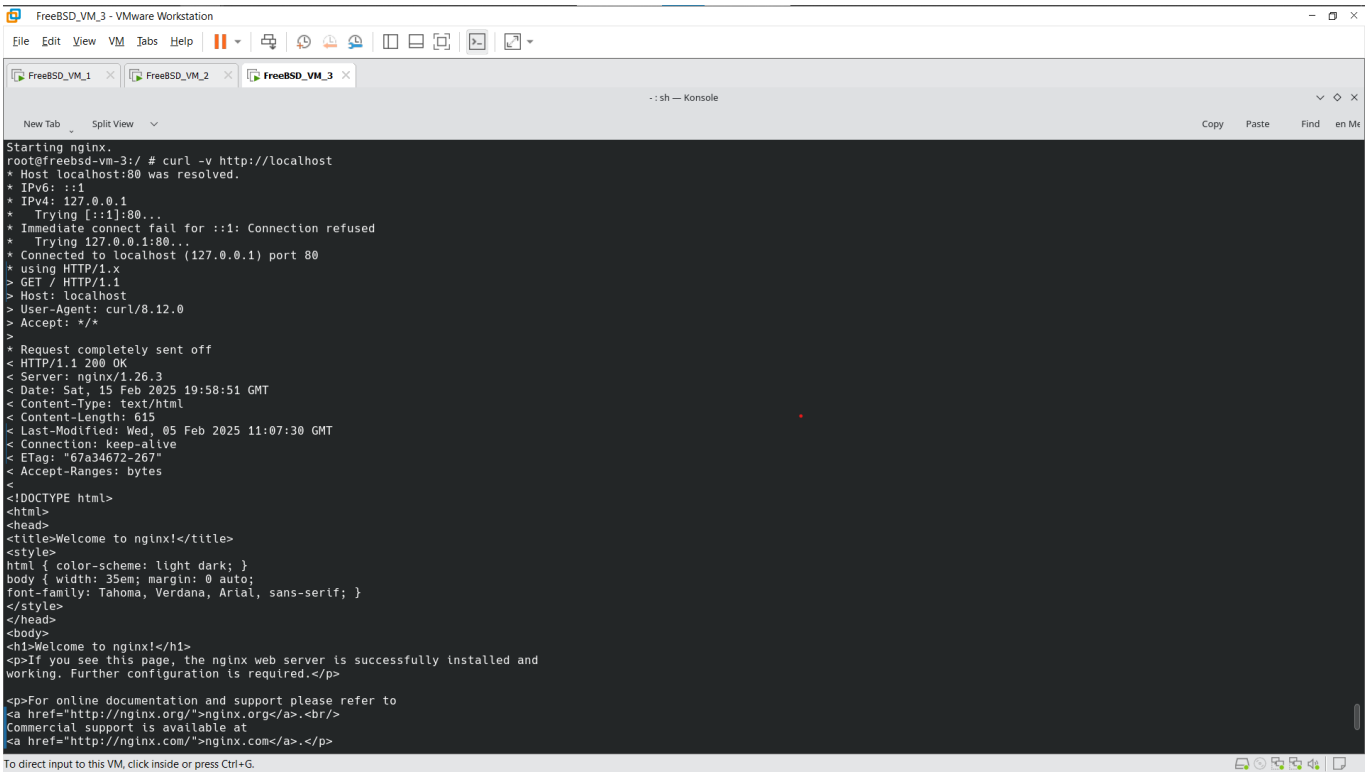
Task 2: Web Server & ACL Configuration

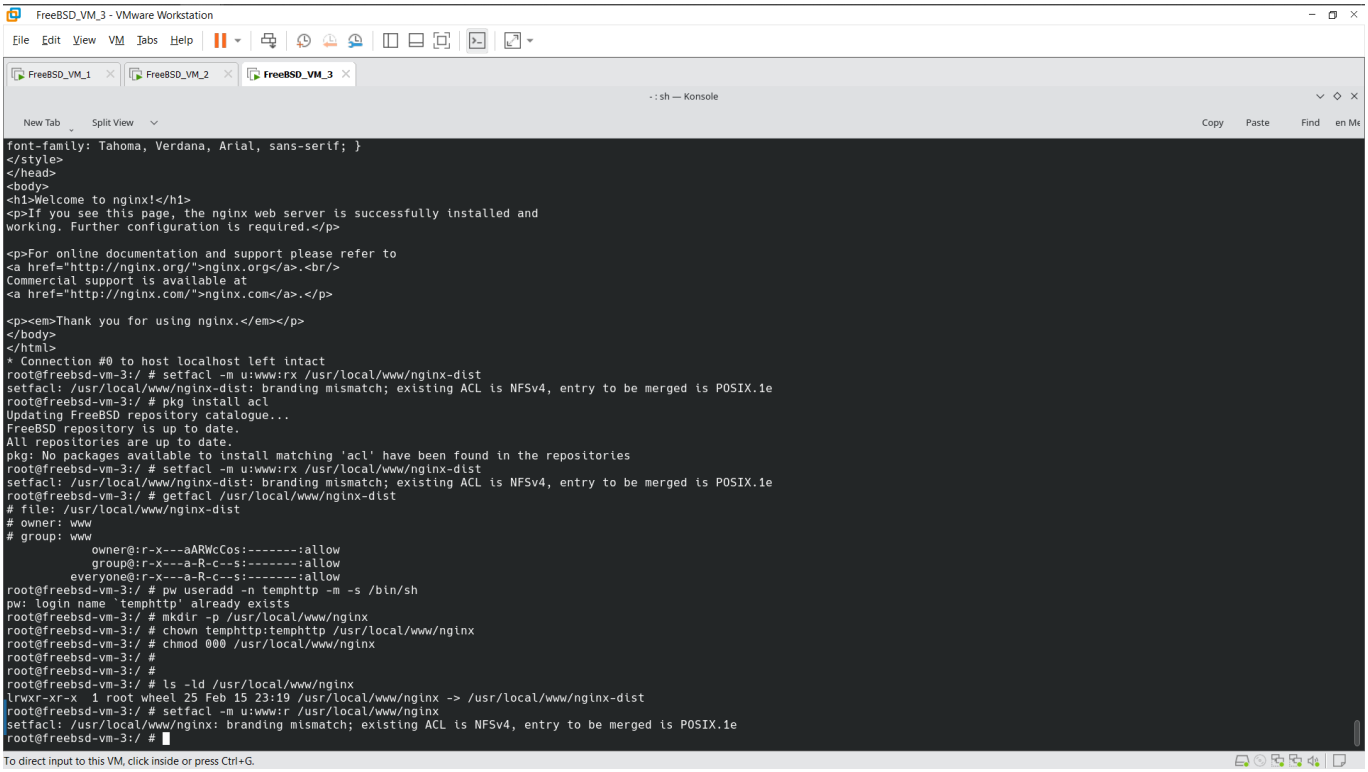
Step 1: Install & Configure Nginx

On VM3:

```
# Install Nginx (Linux example)
sudo apt update && sudo apt install nginx
sudo systemctl start nginx

# Create web directory
sudo mkdir -p /usr/local/www/nginx
```

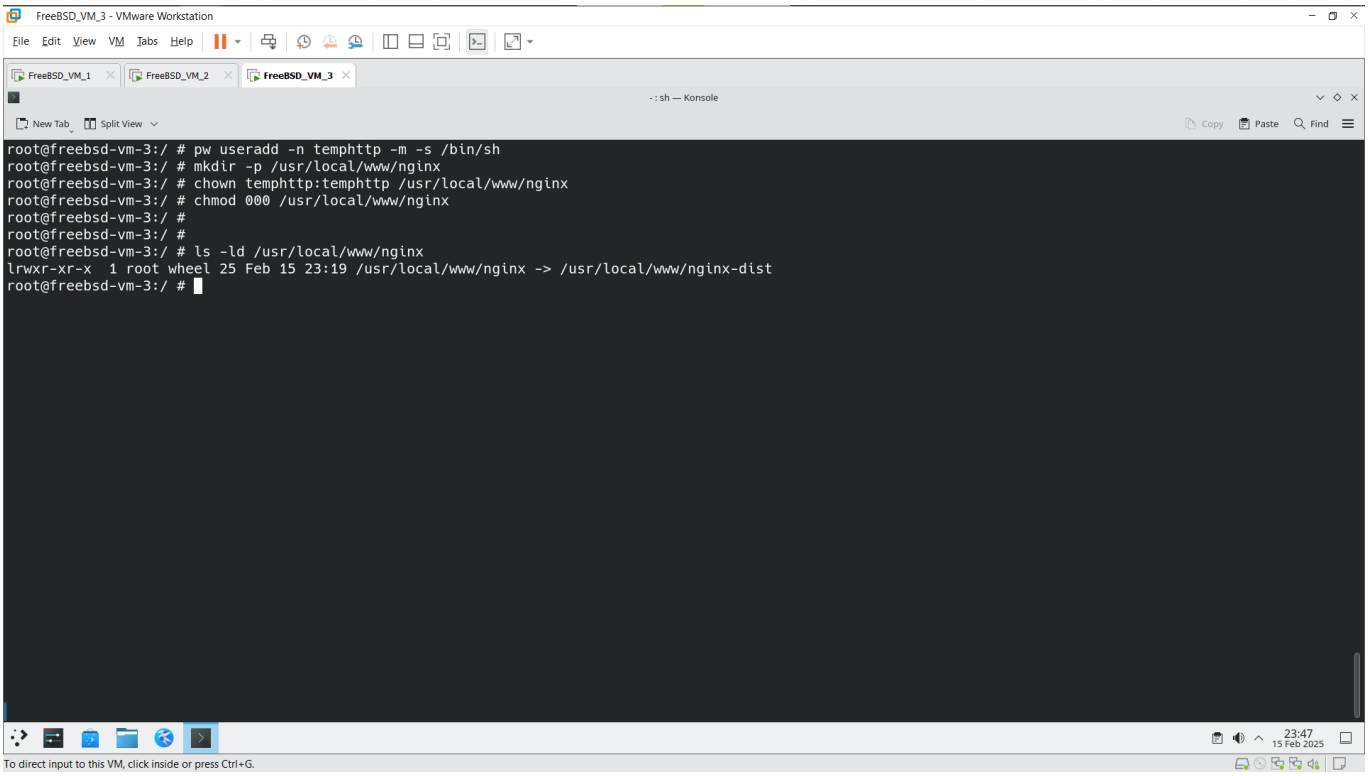




Step 2: Set File Permissions with `setuid`

Create User `templttp`:

```
sudo useradd templttp
sudo chown -R templttp:templttp /usr/local/www/nginx
sudo chmod 0700 /usr/local/www/nginx  # Deny all except owner
```



Attempt Access via Nginx:

```
curl http://10.0.2.10/nginx  # Should fail (403 Forbidden)
```

Step 3: Apply Linux ACLs

Grant read access to the `www-data` user/group (Nginx):

```
sudo setfacl -m u:www-data:r-x /usr/local/www/nginx
sudo getfacl /usr/local/www/nginx  # Verify ACLs (screenshot this!)
```

Retest Access:

```
curl http://10.0.2.10/nginx  # Should now succeed
```

Validation & Screenshots

Include the following in your report:

1. **VM2's Routing Table**: `netstat -rn`
2. **PF Rules**: `pfctl -s rules`
3. **NAT Verification**: `tcpdump` output showing translated IPs.
4. **ACL Configuration**: `getfacl /usr/local/www/nginx`

Troubleshooting

Issue	Solution
PF not loading	`kldload pf; service pf restart`
VM3 cannot reach the internet	Verify default route: `route -n get default`
Connection timeout on port 80	Check firewall rules: `pfctl -s rules`

References

1. FreeBSD PF Handbook
2. Linux ACL Guide
