Lab 2 – Networking Fundamentals and Routing Configuration

Week 2 - Systems Integration

Learning Objectives

Upon completion of this lab, you will be able to understand and configure networking at multiple levels of the TCP/IP stack, implement routing between network interfaces, configure Network Address Translation for internet connectivity, and secure network communications using iptables firewall rules.

Prerequisites

Before beginning this lab, ensure you have completed Lab 1 successfully, your Ubuntu Server VM is operational with network connectivity, and you have created a snapshot of your working VM as a restore point. You will need sudo privileges on your VM for all network configuration tasks.

Part 1: Understanding Your Network Configuration

Begin by examining your current network setup to establish a baseline understanding. Log into your Ubuntu Server VM and execute the following commands, documenting the output for reference.

Network Interface Discovery

```
# Display all network interfaces
ip link show

# Show detailed interface configuration
ip addr show

# Display routing table
ip route show

# Check DNS configuration
resolvectl status
```

Document the following information from your system:

- Primary network interface name (likely enp0s3 or similar)
- Current IP address and subnet mask
- Default gateway address
- DNS server addresses

Understanding Network Layers

To observe how data moves through the network stack, we'll use tcpdump to capture packets. Install it if not already present:

```
sudo apt update
sudo apt install tcpdump
```

Capture packets while pinging an external host:

```
# In terminal 1, start packet capture
sudo tcpdump -i any -w capture.pcap -c 20

# In terminal 2, generate traffic
ping -c 5 8.8.8.8
```

Examine the captured packets:

```
# View packet summary
sudo tcpdump -r capture.pcap -nn

# View packet details including Ethernet headers
sudo tcpdump -r capture.pcap -e -nn
```

Part 2: Configuring a Second Network Interface

For routing configuration, we need multiple network interfaces. We'll add a second adapter to your VM.

VirtualBox Configuration

Shutdown your VM and modify its settings in VirtualBox:

1. In VirtualBox Manager, select your VM and click Settings
2. Navigate to Network → Adapter 2
3. Enable the adapter and set it to "Host-only Adapter"
4. Ensure "Cable Connected" is checked
5. Start your VM

Interface Configuration

After booting, verify the new interface exists:

```
ip link show
```

You should see a new interface (likely enp0s8). Configure it with a static IP address by creating a new netplan configuration:

```
sudo nano /etc/netplan/99-second-interface.yaml
```

Add the following configuration, adjusting interface names as needed:

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s8:
      addresses:
        - 192.168.56.10/24
```

Apply the configuration:

```
sudo netplan apply
ip addr show enp0s8
```

Part 3: Enabling IP Forwarding and Routing

Transform your VM into a router capable of forwarding packets between networks.

Enable IP Forwarding

First, check the current forwarding status:

```
sysctl net.ipv4.ip_forward
```

Enable forwarding temporarily:

```
sudo sysctl -w net.ipv4.ip_forward=1
```

Make it permanent by editing sysctl.conf:

```
sudo nano /etc/sysctl.conf
```

Uncomment or add:

```
net.ipv4.ip_forward=1
```

Configure NAT with iptables

Set up Network Address Translation to allow the internal network to access the internet:

```
# Enable NAT on the external interface
sudo iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE

# Allow forwarding from internal to external
sudo iptables -A FORWARD -i enp0s8 -o enp0s3 -j ACCEPT

# Allow established connections back
sudo iptables -A FORWARD -i enp0s3 -o enp0s8 \
  -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Verify the rules:

```
sudo iptables -t nat -L -n -v
sudo iptables -L FORWARD -n -v
```

Part 4: Implementing Firewall Security

Secure your router with appropriate firewall rules while maintaining functionality.

Basic Security Configuration

Create a firewall script for consistent configuration:

```
sudo nano /root/firewall.sh
```

Add the following comprehensive firewall configuration:

```
#!/bin/bash
# Firewall configuration for router

# Flush existing rules
iptables -F
iptables -t nat -F
iptables -X

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

# Allow loopback
iptables -A INPUT -i lo -j ACCEPT

# Allow established connections
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

# Allow SSH from internal network only
iptables -A INPUT -p tcp --dport 22 -i enp0s8 -j ACCEPT

# Allow DNS
iptables -A INPUT -p udp --dport 53 -j ACCEPT
iptables -A INPUT -p tcp --dport 53 -j ACCEPT

# Allow DHCP
iptables -A INPUT -p udp --dport 67:68 -j ACCEPT

# Allow ping from internal network
iptables -A INPUT -p icmp --icmp-type echo-request -i enp0s8 -j ACCEPT

# Allow routing from internal to external
iptables -A FORWARD -i enp0s8 -o enp0s3 -j ACCEPT

# Configure NAT
iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE

echo "Firewall configured successfully"
```

Make the script executable and run it:

```
sudo chmod +x /root/firewall.sh
sudo /root/firewall.sh
```

Testing Security Rules

Test that the firewall is working correctly:

```
# Check that rules are active
sudo iptables -L -n -v

# Test that SSH still works from host machine
# From your host, try: ssh user@192.168.56.10

# Verify external connectivity still works
ping -c 2 8.8.8.8
```

Part 5: Making Configuration Persistent

Ensure all configurations survive reboots.

```
# Install persistence tools
sudo apt install iptables-persistent netfilter-persistent

# Save current rules
sudo netfilter-persistent save

# Create systemd service for firewall
sudo nano /etc/systemd/system/firewall.service
```

Add the following service definition:

```
[Unit]
Description=Configure Firewall
After=network.target

[Service]
Type=oneshot
ExecStart=/root/firewall.sh
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

Enable the service:

```
sudo systemctl enable firewall.service
sudo systemctl start firewall.service
```

Part 6: Testing Your Router Configuration

Clone your VM to create a client machine for testing:

1. Shutdown your current VM
2. In VirtualBox, right-click the VM and select "Clone"
3. Name it "Client-VM" and choose "Full clone"
4. Modify Client-VM settings: disable Adapter 1, keep only Adapter 2 (Host-only)
5. Start Client-VM and configure it to use your router

On the Client-VM:

```
# Configure network to use router as gateway
sudo nano /etc/netplan/99-client.yaml
```

Add:

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      addresses:
        - 192.168.56.20/24
      routes:
        - to: default
          via: 192.168.56.10
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

Apply and test:

```
sudo netplan apply
ping -c 2 192.168.56.10  # Test router connectivity
ping -c 2 8.8.8.8        # Test internet via router
```

Important Security Note

If internet connectivity through the router fails, verify IP forwarding is enabled on the router, NAT rules are properly configured, and the firewall isn't blocking FORWARD chain traffic. Use `sudo iptables -L -n -v` to check packet counters and identify where packets are being dropped.

Assessment Questions

Answer the following questions to verify your understanding:

1. Explain the difference between the INPUT, OUTPUT, and FORWARD chains in iptables.
2. Why is NAT necessary for the client VM to access the internet through your router?
3. What would happen if you set the default FORWARD policy to ACCEPT instead of DROP?
4. How does the state module (–state ESTABLISHED,RELATED) improve both security and functionality?
5. What is the purpose of enabling IP forwarding, and what happens when it's disabled?

Submission

While this lab requires no formal submission, ensure you can demonstrate:

- Working routing between two network interfaces
- Functional NAT configuration allowing internet access
- Secure firewall rules that protect the router while maintaining functionality
- Persistent configuration that survives reboots

Save your firewall script and netplan configurations for reference.

Information

Assignment 1 will be distributed next week during the DNS lab. Ensure your networking foundation is solid as the DNS resolver you'll implement will rely on proper network configuration.