Stateful Firewall Rules

Prepare VirtualBox Image

Change the MAC Address

1. Start VirtualBox.
2. Navigate to **Settings > Network > Adapter 1 > Advanced > MAC Address**.
3. Generate a new random MAC address.

Change the Network Adapter

- **Home/University Ethernet Network**:
  - **Settings > Network > Adapter 1 > Attached to: Bridged**
- **Eduroam**:
  - Connect to a university Ethernet network or create a new NAT network.

  - **File > Preferences > Networks > NAT Networks > Add new NAT network** (leave settings to defaults).
  - Set Adapter 1 to use the NAT network created earlier.

Disable IPv6

1. Start the image and login as `isp/isp`.
2. Open `/etc/sysctl.conf` and add:

   ```
   net.ipv6.conf.all.disable_ipv6 = 1
   net.ipv6.conf.default.disable_ipv6 = 1
   net.ipv6.conf.lo.disable_ipv6 = 1
   ```

3. Apply changes:

   ```
   sudo sysctl -p
   ```

4. Verify IPv6 is disabled:

   ```
   cat /proc/sys/net/ipv6/conf/all/disable_ipv6
   ```

   Output should be `1`.

Install Apache2 and SSH Server

1. Install required packages:

   ```
   sudo apt install openssh-server apache2 git curl
   ```

2. Generate default digital certificates for Apache2:

   ```
   sudo make-ssl-cert generate-default-snakeoil --force-overwrite
   ```

3. Enable Apache2 SSL Site:

   ```
   sudo a2ensite default-ssl
   sudo a2enmod ssl
   ```

4. Restart Apache server:

   ```
   sudo service apache2 restart
   ```

5. Test Apache2:
   - Open `http://localhost` and `https://localhost` in a browser.
   - Alternatively, test with curl: `curl http://localhost`.
6. Test SSH server:

   ```
   ssh localhost
   ```

   Answer `yes`, provide password `isp`, and press `Ctrl+D` to exit.

Download the Script Template

1. Clone the repository:

   ```
   git clone https://github.com/lem-course/isp-iptables.git
   ```

2. Change execution permissions:

   ```
   chmod +x iptables2.sh
   ```

Solve Assignments (INPUT and OUTPUT Chains Only)

1. Edit the script `iptables2.sh`.

2. For each task:
    - Write a solution.
    - Start the script:

      ```
      sudo ./iptables2.sh start
      ```

    - Check active rules:

      ```
      sudo iptables --list -nv
      ```

    - Test rules using appropriate programs:
        - ICMP: `ping`.
        - DNS: `dig www.fri.uni-lj.si`.
        - HTTP: `curl google.com`.
        - SSH: `ssh isp@<machine-IP>`.
    - Restart the script after modifications:

      ```
      sudo ./iptables2.sh restart
      ```

Firewall Forwarding Rules

Network Setup

- Use three virtual machines: **router**, **client**, and **server**.
- **Router**:
    - Interfaces: client_subnet, server_subnet, and Internet connectivity.
- **Client**:
    - Subnet: client_subnet.
- **Server**:
    - Subnet: server_subnet.

Set Up VirtualBox Images

1. Clone the existing image to create **client** and **server**.
2. Generate new MAC addresses for the clones.
3. Configure NICs:
    - **Router (isp):**
        - Adapter 1: NAT, Bridged, or NAT Network.
        - Adapter 2: Internal Network, `client_subnet`.
        - Adapter 3: Internal Network, `server_subnet`.
    - **Client:** Internal Network, `client_subnet`.
    - **Server:** Internal Network, `server_subnet`.

Prepare Router Machine (isp)

1. Assign IPs to `enp0s8` and `enp0s9`:
    - Edit `/etc/netplan/01-network-manager-all.yaml`:

      ```
      network:
        version: 2
        ethernets:
          enp0s3:
            dhcp4: true
            dhcp-identifier: mac
          enp0s8:
            addresses: [10.0.0.1/24]
          enp0s9:
            addresses: [172.16.0.1/24]
      ```

    - Apply changes:

      ```
      sudo netplan apply
      ```

2. Enable IPv4 routing:

   ```
   echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
   ```

3. Set up NAT:

   ```
   sudo iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
   ```

Prepare the Client Machine

1. Configure `/etc/netplan/01-network-manager-all.yaml`:

```
network:
  version: 2
  ethernets:
    enp0s3:
      addresses: [10.0.0.2/24]
      routes:
        - to: default
          via: 10.0.0.1
      nameservers:
        addresses: [8.8.8.8]
```

2. Apply changes:

```
sudo netplan apply
```

3. Test connectivity by pinging the router and public Internet:

```
ping 8.8.8.8
```

Prepare the Server Machine

1. Configure `/etc/netplan/01-network-manager-all.yaml`:

```
network:
  version: 2
  ethernets:
    enp0s3:
      addresses: [172.16.0.2/24]
      routes:
        - to: default
          via: 172.16.0.1
      nameservers:
        addresses: [8.8.8.8]
```

2. Apply changes:

```
sudo netplan apply
```

Filtering

1. Edit `iptables2.sh`.
2. Add rules to the FORWARD chain to permit ICMP, DNS, SSH, HTTP, and HTTPS traffic.
3. Test rules by launching requests from the client machine.

Additional Tasks

1. **Allow SSH between client_subnet and server_subnet; block SSH to the Internet.**
2. **Block access to facebook.com:**
    - Find IP address:

      ```
      dig +noall +answer facebook.com | cut -f6 | xargs | tr " " ,.
      ```

    - Add a rule to block the IP address.
3. **Limit ping requests to the firewall:**
    - Allow only 10 ping requests per minute from the public Internet.