

INT303 WEEK TWO LAB ASSIGNMENT

INT303: Networking Fundamentals – Lab 4

Lab 4: Simulating Network Routing and VLAN Configuration in Linux

Objective:

In this lab, students will simulate network routing and VLAN configuration using Linux-based tools. They will use the OWASP Broken Web Application VM's IP address to test connectivity and routing principles.

Learning Outcomes:

By the end of this lab, students will:

- Understand how to set up routing and VLANs on Linux.
 - Be able to simulate network device behavior using Linux tools.
 - Gain hands-on experience in network troubleshooting.
 - Learn how to configure `iptables`, `ip route`, and network namespaces to manage traffic.
-

Materials Needed:

- Linux machine or VM (e.g., Ubuntu, Kali).
 - OWASP Broken Web Application VM (reachable on the network).
 - IP address of the OWASP Broken Web Application VM (e.g., 192.168.X.X).
 - Terminal access.
-

Lab Exercises:

Exercise 1: Setting Up Static Routing in Linux

1. **Task:** Set up static routes in Linux to simulate routing behavior. Ensure the Linux machine can route traffic to the OWASP Broken Web Application VM.
 - **Steps:**
 - a. Use `ip route` to add a static route to reach the OWASP Broken Web Application VM.
 - b. Ensure that your machine routes traffic correctly through the simulated network.
 - **Commands:**
 - `sudo ip route add <destination_network> via <gateway_IP> dev <interface>`
Example: `sudo ip route add 192.168.4.0/24 via 192.168.1.1 dev eth0`
 - `ip route show` – Verify your routing table.
 - **Verification Command:**
 - `ping <OWASP_IP>` – Ensure that the Linux machine can reach the OWASP Broken Web Application VM.
 2. **Question:**
 - How does static routing work on a Linux system?
 - What challenges could arise when setting up routes manually?
-

Exercise 2: VLAN Configuration Using Network Namespaces

3. **Task:** Create virtual LANs (VLANs) using Linux network namespaces to segment traffic. Assign the OWASP Broken Web Application VM to a separate namespace and test connectivity between namespaces.
 - **Steps:**
 - a. Create two network namespaces to simulate VLANs.
 - b. Assign virtual network interfaces to each namespace.
 - c. Route traffic between the namespaces.
 - d. Test communication between namespaces and the OWASP Broken Web Application VM.
 - **Commands:**
 - `sudo ip netns add vlan1` – Create a network namespace for VLAN 1.
 - `sudo ip netns add vlan2` – Create a network namespace for VLAN 2.

- `sudo ip link add veth1 type veth peer name veth2` – Create virtual network interfaces.
- `sudo ip link set veth1 netns vlan1` – Assign veth1 to vlan1.
- `sudo ip link set veth2 netns vlan2` – Assign veth2 to vlan2.
- `sudo ip netns exec vlan1 ip link set lo up` – Enable loopback interface in the namespace.
- `sudo ip netns exec vlan1 ip addr add 192.168.10.1/24 dev veth1` – Assign IP.
- `sudo ip netns exec vlan1 ping <OWASP_IP>` – Test communication.

4. **Question:**

- How do network namespaces simulate VLANs in Linux?
- What are the benefits of using namespaces in network isolation?

Exercise 3: IP Address Assignment and Subnetting in Linux

5. **Task:** Subnet your network and assign IP addresses to each namespace and the OWASP VM. Ensure the correct routing between subnets using static routes.
- **Steps:**
 - a. Subnet the given network (e.g., 192.168.0.0/24).
 - b. Assign IP addresses to network namespaces and devices.
 - c. Configure static routes to enable communication between the subnets.
 - **Commands:**
 - `ip addr add <subnet_IP> dev <interface>` – Assign IP addresses.
 - `ip route add <subnet>` – Add static routes.
6. **Question:**
- How does subnetting work in Linux environments?
 - What challenges arise when configuring subnets manually?

Exercise 4: Testing Connectivity Using Ping and Traceroute

7. **Task:** Use `ping` and `traceroute` to test connectivity between the Linux machine, network namespaces, and the OWASP Broken Web Application VM. Analyze the routing path.
- **Commands:**

- `ping <OWASP_IP>` – Check connectivity.
- `traceroute <OWASP_IP>` – Trace the packet path.

8. Question:

- What can `traceroute` reveal about network issues?
- How does packet routing affect network performance?

Exercise 5: Configuring `iptables` for Routing and Firewall Rules

9. **Task:** Use `iptables` to simulate a router by controlling the flow of traffic between different networks and the OWASP VM. Block certain traffic while allowing other traffic.

- **Steps:**
 - a. Enable IP forwarding on the Linux machine.
 - b. Set up `iptables` rules to allow or block specific traffic between namespaces and the OWASP VM.
 - c. Test the firewall by pinging the OWASP VM and other devices.
- **Commands:**
 - `echo 1 > /proc/sys/net/ipv4/ip_forward` – Enable IP forwarding.
 - `sudo iptables -A FORWARD -s 192.168.10.0/24 -d 192.168.20.0/24 -j ACCEPT` – Allow forwarding.
 - `sudo iptables -A FORWARD -s 192.168.10.0/24 -d <OWASP_IP> -j DROP` – Block traffic to OWASP VM.

10. Question:

- How can `iptables` be used to simulate routing and firewall functionality?
- What are common mistakes when configuring `iptables` rules?

Exercise 6: Monitoring Traffic Using `tcpdump`

11. **Task:** Use `tcpdump` to monitor traffic between your Linux namespaces and the OWASP VM. Capture and analyze packets to understand traffic flow.

- **Commands:**
 - `sudo tcpdump -i <interface>` – Capture traffic on a specific interface.
 - `sudo tcpdump -i veth1` – Monitor traffic on the virtual network interface.
 - `sudo tcpdump -i eth0 src <OWASP_IP>` – Monitor traffic to/from OWASP VM.

12. Question:

- How can `tcpdump` be used to diagnose network issues?
 - What types of traffic do you observe during the simulation?
-

Submission Requirements:

- Submit a report including:
 - Screenshots of the static routing, namespace creation, and `iptables` configurations.
 - A brief explanation of each step and the results from `ping`, `traceroute`, and `tcpdump` commands.
 - Troubleshooting steps for any encountered issues.
-

Reflection:

This lab demonstrates how Linux-based tools can effectively simulate routing and VLAN behavior. Students gain critical insights into routing, network segmentation, and traffic control without needing physical network hardware.

INT303: Networking Fundamentals – Lab 5

Lab 5: IP Address Analysis and Network Report

Objective:

This lab aims to deepen your understanding of IP addressing, network scalability, and subnetting. You will analyze real-world websites, classify IP addresses, calculate device capacities, determine subnet masks, and perform advanced IP operations. Additionally, you will compile a detailed report showcasing your findings and understanding of IP networking.

Learning Outcomes:

Upon completing this lab, you will:

- Master IP address classification.
 - Calculate the number of devices supported by each class of IP address.
 - Identify and work with subnet masks.
 - Gain experience with both basic and advanced IP tools.
 - Develop a comprehensive network report.
 - Learn hands-on exercises with advanced concepts like subnetting, network summarization, and IP analysis.
-

Instructions:

Step 1: Select 10 Websites

1. Choose **10 popular websites** (e.g., `apple.com`, `twitter.com`, etc.) to analyze.
2. **Bonus Exercise:** Include a variety of domains, such as `.com`, `.org`, `.edu`, and country-specific domains (e.g., `.co.uk`, `.de`). Reflect on the differences in the IP address ranges assigned to these domains.

Step 2: Ping the Websites

2. Use the **ping** command in your terminal to retrieve the IP addresses of each website.
 - **Example Command:**
 - `ping apple.com`
3. **Bonus Exercise:** Identify if the website resolves to multiple IP addresses (i.e., if it uses a content delivery network or load balancing). Record multiple IPs where applicable and compare the geographical locations of each.

Step 3: Document the IP Addresses

3. Create a table to record the IP addresses for each website you ping.

Step 4: Classify the IP Addresses

4. Determine the **class** (A, B, or C) of each IP address based on its first octet.
 - **Bonus Exercise:**
 - Research the reserved IP ranges (e.g., private IP ranges like 10.0.0.0/8) and see if any of your IP addresses fall into these special ranges.
 - Discuss the importance of these reserved IPs in network configurations.

Step 5: Calculate the Number of Devices Each IP Can Support

5. For each IP class, calculate how many devices the network can accommodate. This will give you an understanding of network size and scalability.
 - **Device Capacity Formula:**
 - **Class A:** $2^{24} - 2$
 - **Class B:** $2^{16} - 2$
 - **Class C:** $2^8 - 2$
 - **Bonus Exercise:**
 - Calculate the actual number of usable hosts by performing subnetting on these networks. For instance, consider subdividing a Class B network into smaller subnets and calculate the device capacity for each subnet.

- Consider subnet masks like /26 or /28 for Class C networks and explore how they impact the number of devices.
-

Step 6: Determine the Subnet Mask

6. Identify the **default subnet mask** for each IP address based on its class.
 - **Bonus Exercise:**
 - Explore scenarios where custom subnetting is required (e.g., for network segmentation). How does altering the subnet mask affect the device capacity and overall network organization?
 - Try experimenting with subnet masks like /25, /27, and explain the difference in the number of networks and hosts.
-

Step 7: Advanced IP Tools (Bonus Exercises)

- **Traceroute (Network Path Analysis):** Use `traceroute` (Linux/Mac) or `tracert` (Windows) to discover the network path between your system and one of the websites.
 - `traceroute apple.com`
 - **Analyze the hops** between your system and the destination. Identify the location and IP address of each hop.
 - **Bonus:** Explain the significance of each hop, and analyze if any of them belong to private networks or are hosted on cloud infrastructure.
 - **GeoIP Location:** Use online tools like **IPinfo.io** or **GeoIP** to check the geographical location of the IP addresses.
 - Identify if the IP addresses belong to data centers, ISPs, or any specific regions.
 - **Bonus:** Compare the latency (ping time) with the geographic distance.
-

Step 8: Compile a Professional Report

Create a report that includes the following sections:

1. **Introduction:**
 - Outline the purpose of the lab, your objectives, and the significance of IP address analysis and subnetting.
2. **Website List and IP Addresses:**

- Present the list of websites and their corresponding IP addresses.
 - 3. **IP Address Classification:**
 - Classify each IP address and explain the method used for classification (Class A, B, or C). Discuss the characteristics of each class and their practical applications.
 - 4. **Number of Devices Supported:**
 - Present your device capacity calculations for each IP address. Include a detailed explanation of how the number of devices is determined based on the IP class and subnetting.
 - 5. **Subnet Masks:**
 - List the subnet masks for each IP address and discuss any custom subnetting scenarios.
 - 6. **Advanced Tools and Analysis (Optional):**
 - Present your findings from the traceroute, GeoIP analysis, and any other advanced IP tools you explored. Discuss how these tools can be applied in network troubleshooting, performance optimization, or security audits.
 - 7. **Conclusion:**
 - Summarize your findings and reflect on the importance of IP address management, subnetting, and network scalability in real-world networking.
-

Reflection:

This lab provides a comprehensive understanding of IP address analysis, subnetting, and the usage of advanced network tools. By working with real-world data, you'll gain practical skills applicable to network configuration, security auditing, and system administration. Compiling the information into a professional report will further enhance your technical documentation skills.

Additional Exercises for Further Practice:

- **Subnetting Challenge:** For each IP address you pinged, create multiple subnets and calculate the network address, first usable IP, last usable IP, and broadcast address.
- **IP Address Conflict Detection:** Simulate a scenario where IP address conflicts occur within a network and outline troubleshooting steps.
- **Ping Flood Analysis:** Use tools like `hping3` to simulate a flood of ICMP requests and analyze the impact on a network using tools like Wireshark.