GNS3 Simple Network Topology Lab

- ★ **Be Careful with Kali Linux:**
- ★ Kali Linux is a powerful tool for penetration testing, and it's essential to use it responsibly. I cannot emphasize enough the importance of ensuring that you are disconnected from the internet when practicing your penetration testing skills. This precaution is necessary to avoid unintended consequences or potential legal issues.
- ★ **Course Content Overlap:**
- ★ Some of the hands-on session content overlaps with CSCI6706, a graduate course taught by Professor Nur Zincir-Heywood, who specializes in cybersecurity topics. Professor Zincir-Heywood, my supervisor, is an excellent source of knowledge. I have had the honor of working with her as a Research Assistant (RA) and Teaching Assistant (TA) for four years. Her insights and expertise have significantly influenced this course material.

Introduction

In this practical session, we'll set up a simple network topology using GNS3. We will configure network devices, set up Docker containers, and use tools like tc and iperf to emulate network conditions and measure performance. Additionally, we will configure an FTP server within a Docker container.

Prerequisites

Before you begin, ensure that Docker is installed and running on your system. Start the Docker service with the following command:

```
sudo systemctl start docker
# or
sudo service docker start
```

1. Access Control Lists (ACL) in Routers and Switches

Access Control Lists (ACLs) are used in routers and switches to control traffic flow based on defined rules. They allow or deny traffic based on various criteria such as source/destination IP address, port numbers, protocol types, etc. In this section, we'll learn how to configure ACLs on routers and switches in our network topology.

1.1. Standard ACL Configuration

1. **Define ACL Rules:**
    - Determine the access rules you want to implement, such as permitting or denying specific types of traffic.
    - Example ACL rule:

```
access-list 1 permit 192.168.1.0 0.0.0.255
```

2. **Apply ACL to Interface:**
    - Apply the ACL to the router interface in the desired direction (inbound or outbound).
    - **Inbound (In)**: This refers to traffic that is coming into a network interface. An inbound ACL is used to control and filter packets that are entering the interface from an external source before they reach the internal network. Inbound ACLs are applied to incoming traffic to determine whether to allow or deny the packets as they enter the interface.
    - **Outbound (Out)**: This refers to traffic that is leaving a network interface. An outbound ACL is used to control and filter packets that are exiting the interface towards an external destination. Outbound ACLs are applied to outgoing traffic to determine whether to allow or deny the packets as they leave the interface.

- Example:

```
interface GigabitEthernet0/1
ip access-group 1 in
```

3. **Verify ACL Configuration:**
   - Check the ACL configuration and applied interfaces.
   - Example:

```
show access-lists
show interfaces GigabitEthernet0/1
```

1.2. Extended ACL Configuration

1. **Define ACL Rules:**
   - Determine the access rules for controlling traffic within VLANs or on specific Router ports.
   - Example ACL rule:

```
access-list 101 permit tcp any host 192.168.1.10 eq 80
```

2. **Apply ACL to VLAN or Interface:**
   - Apply the ACL to the VLAN interface or individual switch ports.
   - Example:

```
interface Vlan10
ip access-group 101 in
```

3. **Verify ACL Configuration:**
   - Check the ACL configuration and applied interfaces.
   - Example:

```
show access-lists
show interfaces vlan 10
```

By configuring ACLs on routers and switches, you can control traffic flow within your network, improve security, and enforce network policies effectively.

1.3. Extended Access Control Lists (ACLs):

Type of ACL used in networking to provide more granular control over traffic filtering. They can filter network traffic based on a wide range of criteria, including:

- **Access-list #(>100) permit/deny Protocol Source(any/ host/rang of addresses) Destination(Host/any/range) eq Destination port**

Example

A rule to permit HTTP traffic from any host in the 192.168.1.0/24 subnet to a web server at 10.0.0.1 would look like this:

```
access-list 100 permit tcp 192.168.1.0 0.0.0.255 host 10.0.0.1 eq 80
```

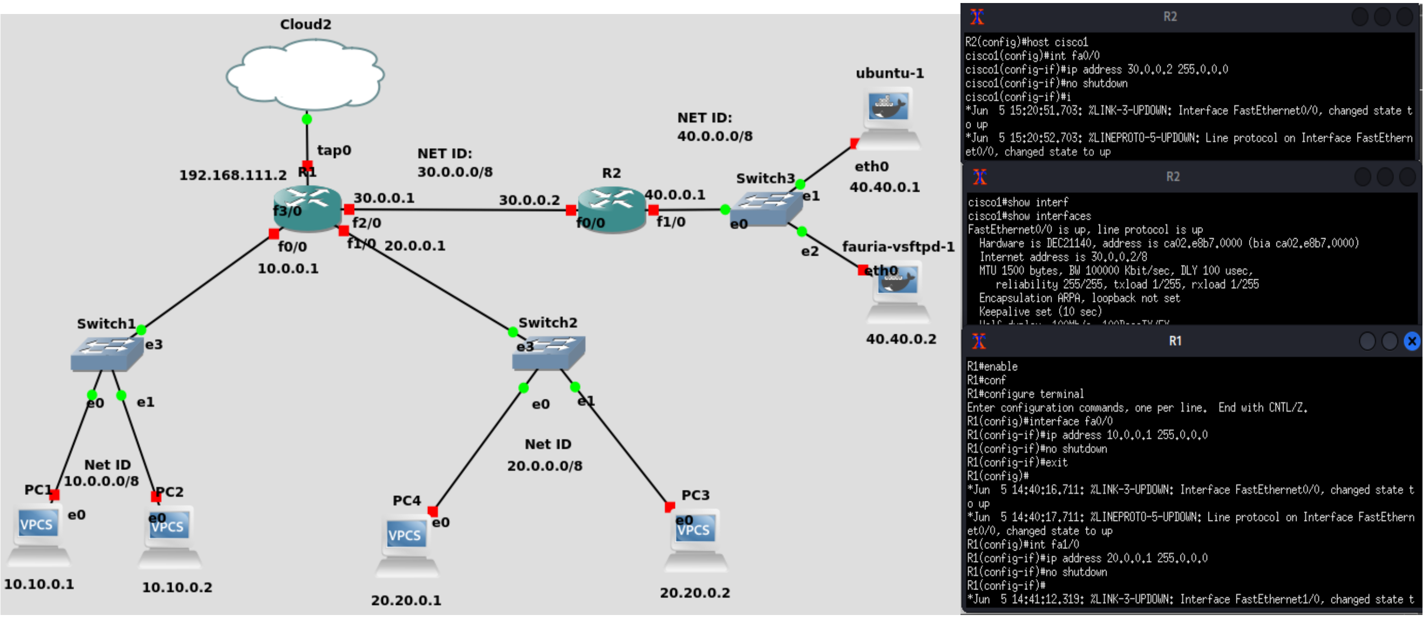
This rule specifies:

- **TCP traffic**
- **From any IP in the 192.168.1.0/24 subnet**
- **To the IP address 10.0.0.1**
- **On port 80 (HTTP)**

Extended ACLs provide powerful and flexible traffic control capabilities in network security management.

4. **Practice 2:**

Practice 2 involves extending the project from Practice 1 based on Figure 2. In addition to adding routers, the task requires restricting access to the Ubuntu server to all nodes in the 10.0.0.0/8 subnet and preventing 20.20.0.1 from accessing the FTP server. You'll need to configure the routers, assign IP addresses accurately, verify connectivity, and then properly configure the Access Control Lists (ACLs).



```
R2(config)#host cisco1
cisco1(config)#int fa0/0
cisco1(config-if)#ip address 30.0.0.2 255.0.0.0
cisco1(config-if)#no shutdown
cisco1(config-if)#i
*Jun  5 15:20:51.703: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state t
o up
*Jun  5 15:20:52.703: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/0, changed state to up
```

```
cisco1#show interf
cisco1#show interfaces
FastEthernet0/0 is up, line protocol is up
  Hardware is DEC21140, address is ca02.e8b7.0000 (bia ca02.e8b7.0000)
  Internet address is 30.0.0.2/8
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
```

```
R1#enable
R1#conf
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface fa0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#
*Jun  5 14:40:16.711: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state t
o up
*Jun  5 14:40:17.711: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/0, changed state to up
R1(config)#int fa1/0
R1(config-if)#ip address 20.0.0.1 255.0.0.0
R1(config-if)#no shutdown
R1(config-if)#
*Jun  5 14:41:12.319: %LINK-3-UPDOWN: Interface FastEthernet1/0, changed state t
```

Figure2

By following this document, you will set up a simple network topology in GNS3, configure devices, and use Docker containers.

Feel free to reach out with any questions or concerns.

Let's dive in!

Cisco Routers Cheat sheet:

Show Commands:

- `show ip interface brief`: Displays brief information about all interfaces.
- `show running-config`: Displays the current running configuration.
- `show version`: Displays information about the router's hardware and software.
- `show ip route`: Displays the routing table.
- `show arp`: Displays the ARP table.
- `show interfaces`: Displays detailed information about all interfaces.

Configuration Commands:

- `enable`: Enters privileged EXEC mode.
- `configure terminal`: Enters global configuration mode.
- `interface [interface_name]`: Enters interface configuration mode for the specified interface.
- `ip address [ip_address] [subnet_mask]`: Assigns an IP address and subnet mask to an interface.
- `no shutdown`: Enables an interface.
- `hostname [hostname]`: Sets the hostname of the router.
- `enable password [password]`: Sets the password required to enter privileged EXEC mode.
- `enable secret [password]`: Sets the encrypted password required to enter privileged EXEC mode.
- `line console 0`: Enters console line configuration mode.
- `password [password]`: Sets the password required to access the console line.
- `exit`: Exits the current configuration mode.

Clear Commands:

- `clear arp`: Clears the ARP cache.
- `clear counters`: Clears interface counters.
- `clear ip route *`: Clears the routing table.
- **Save Configuration**:
- `write memory`: Saves the running configuration to the startup configuration (NVRAM).

Miscellaneous:

- `ping [ip_address]`: Sends ICMP echo requests to test connectivity to a specific IP address.
- `traceroute [ip_address]`: Traces the route to a destination IP address.
- `clock set [hh:mm:ss] [day month year]`: Sets the system clock

Access Control Lists (ACLs):

- `access-list [number] permit/deny [protocol] [source] [source_wildcard] [destination] [destination_wildcard] [eq/dynamic/established] [port]`: Creates an ACL entry to permit or deny traffic based on various criteria.
- `ip access-group [number] in/out`: Applies an ACL to inbound or outbound traffic on an interface.
- `ipv6 access-list [name] permit/deny [protocol] [source] [source_wildcard] [destination] [destination_wildcard] [eq/dynamic/established] [port]`: Creates an IPv6 ACL entry.

Password Encryption:

- `service password-encryption`: Encrypts plaintext passwords stored in the configuration.
- `enable secret [password]`: Sets an encrypted password required to enter privileged EXEC mode.

SSH Configuration:

- `ip domain-name [domain_name]`: Specifies the domain name used for generating RSA keys.
- `crypto key generate rsa [modulus_size]`: Generates RSA keys for SSH.
- `line vty 0 4`: Enters VTY line configuration mode for SSH access.
- `transport input ssh`: Allows SSH access to the router.
- `login local`: Enables local authentication for SSH access.
- `username [username] secret [password]`: Creates a local username/password for SSH authentication.

Virtual Private Networks (VPNs):

- `crypto isakmp policy [priority]`: Configures ISAKMP policy parameters for VPN negotiation.
- `crypto isakmp key [key] address [peer_address]`: Defines a pre-shared key for ISAKMP authentication.
- `crypto ipsec transform-set [name] [encryption_algorithm] [authentication_algorithm]`: Defines IPsec transform sets.
- `crypto map [name] [sequence_number] ipsec-isakmp`: Enters crypto map configuration mode.
- `match address [number]`: Specifies the ACL used to identify traffic to be encrypted.
- `set peer [peer_address]`: Sets the IP address of the remote VPN peer.

Secure Management:

- `login block-for [time] attempts [attempts] within [time_period]`: Blocks login attempts after a specified number of failed attempts within a time period.
- `login delay [seconds]`: Delays login responses after failed login attempts.
- `login quiet-mode access-class [acl_number] [period] [count]`: Restricts access during quiet mode to specific IP addresses.