### ****Network Security Setup Documentation****

### ****Instance Overview****

| **Instance** | **Details** |
| --- | --- |
| **Firewall** | **IP Address**: 192.168.20.6 **Role**: Manage and monitor traffic, protect the network, and act as a gateway between the external network and the internal systems. **Tools Used**: iptables |
| **Central Server** | **IP Address**: 192.168.20.7 **Role**: Host services (e.g., web server) and run intrusion detection systems (Snort and Suricata). **OS**: Ubuntu Desktop (configured for server applications). |
| **Attack Simulation** | **IP Address**: 192.168.20.8 **Role**: Simulate malicious traffic and generate test cases for the server and firewall. **Tools Used**: hping3. |

### ****Instance 1: Firewall (Ubuntu Server - 192.168.20.6)****

#### **Tasks Completed**

1. **Configured IP Forwarding**:
   * Enabled traffic forwarding between interfaces:

sudo sysctl -w net.ipv4.ip\_forward=1

* + Made the change permanent in /etc/sysctl.conf.

1. **Set Up iptables Rules**:
   * Allowed forwarding of traffic to and from the internal subnet:

sudo iptables -A FORWARD -s 192.168.20.0/24 -j ACCEPT

sudo iptables -A FORWARD -d 192.168.20.0/24 -j ACCEPT

* + Saved the rules for persistence:

sudo apt install iptables-persistent

sudo netfilter-persistent save

1. **Configured Traffic Filtering**:
   * Logged and dropped specific traffic (e.g., HTTP on port 80):

sudo iptables -A INPUT -p tcp --dport 80 -j LOG --log-prefix "Blocked Traffic: "

sudo iptables -A INPUT -p tcp --dport 80 -j DROP

### ****Instance 2: Central Server (Ubuntu Desktop - 192.168.20.7)****

#### **Tasks Completed**

1. **Installed Apache**:
   * Installed and started Apache to test HTTP traffic:

sudo apt install apache2

sudo systemctl start apache2

1. **Installed and Configured Snort**:
   * **Installation**:

sudo apt update

sudo apt install snort

* + **Configuration**:
    - Set HOME\_NET to monitor the local subnet:

plaintext

ipvar HOME\_NET 192.168.20.0/24

* + - Added custom rules to local.rules:
      * Detect ICMP traffic:

plaintext

alert icmp any any -> any any (msg:"ICMP traffic detected"; sid:1000001;)

* + - * Detect SYN Flood (DDoS):

plaintext

alert tcp any any -> any any (msg:"Possible SYN Flood Detected"; flags:S; threshold:type both, track by\_src, count 20, seconds 10; sid:1000002;)

* + **Testing**:
    - Verified alerts in /var/log/snort/snort.alert.fast:

sudo tail -f /var/log/snort/snort.alert.fast

1. **Installed and Configured Suricata**:
   * **Installation**:

sudo apt update

sudo apt install suricata

* + **Configuration**:
    - Defined HOME\_NET:

yaml

vars:

address-groups:

HOME\_NET: "[192.168.20.0/24]"

* + - Configured ens160 as the monitored interface:

yaml

af-packet:

- interface: ens160

cluster-id: 99

cluster-type: cluster\_flow

defrag: yes

* + **Custom Rules**:
    - ICMP Detection:

plaintext

alert icmp any any -> any any (msg:"ICMP traffic detected"; sid:1000001; rev:1;)

* + - SYN Flood Detection:

plaintext

alert tcp any any -> any any (msg:"Possible SYN Flood Detected"; flags:S; threshold:type both, track by\_src, count 20, seconds 10; sid:1000002; rev:1;)

* + **Testing**:
    - Verified alerts in /var/log/suricata/fast.log:

sudo tail -f /var/log/suricata/fast.log

### ****Instance 3: Attack Simulation (Ubuntu Desktop - 192.168.20.8)****

#### **Purpose**

* Generate test traffic, including both legitimate and malicious requests, to validate the IDS/IPS configuration on the central server.

#### **Tasks Completed**

1. **Network Testing**:
   * Sent ping requests and HTTP traffic to the server:

ping 192.168.20.7

curl http://192.168.20.7

1. **Simulated DDoS Attacks**:
   * Used hping3 to send SYN flood traffic:

sudo hping3 -S -p 80 --flood 192.168.20.7

* + Used ping -f to simulate ICMP flood attacks:

ping -f 192.168.20.7

1. **Verified Alerts**:
   * Confirmed that the attacks triggered alerts in both Snort (/var/log/snort/snort.alert.fast) and Suricata (/var/log/suricata/fast.log).