

CSCE 5585: Secure Network Design and Implementation project

Teammates:

Gyaneswar Sai Bandaru

Rakesh Reddy Jammuladinne

Avinash Reddy Gangapuram

Virtual Lab Setup

Tools Used:

GNS3: In network emulation and topology design.

VMware: As the hypervisor to support virtual machines needed for the project execution

Setup Details:

GNS3 Server Configuration: The GNS3 server was then installed by importing it on VMware as an OVA file. Stating used the GNS3 server linked with the GNS3 client for topological construction and control.

It also has web management capability that can be accessed and manage through its Web User Interface .

Virtual Machine (VM) Integration: Several VMware-based virtual machines were imported and integrated into the GNS3 environment:

Attack Machine: It is applied for security testing and penetration simulation practices.

Ubuntu Server: Used to install software firewall and IDS and database hosting service provider

Remote Machine: Emulated an external entity that needs to connect to the network resources from outside.

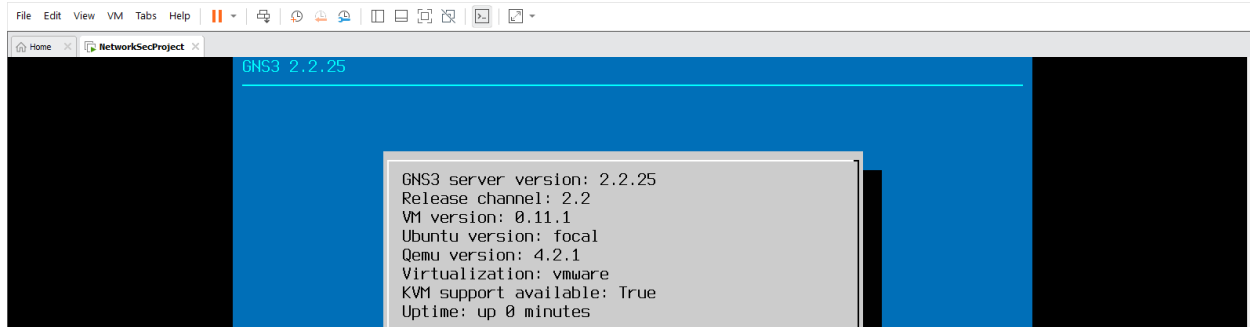
Network Emulation:

The network topologies that were emulated in GNS3 were connected with the virtual machines that were being run in the VMware to make the results of the two interfaces compatible.

Each of the machines was connected to the correct network interface to be able to communicate with those VLANs, and the subnetworks.

Verification: Ensured that the GNS3 server and client were working by checking that they could both connect. Used the ping and traceroute command to make sure that VMware machines that were imported

into the GNS3 topology work by connecting them. Opened GNS3 Web UI to check connection of the server and to follow the changes in the topology.



Network Design and Segmentation

The proposed architecture of the logical network was to have internet connectivity, internal departmental networks, the DMZ for public sectors, VPN for remote connections, and connection to the external world.

Network Segmentation

The network was divided into the following segments:

VLANs for Internal Departments and Guests:

- VLAN 10: IT Department
- VLAN 20: Finance Department
- VLAN 30: Guest Network

DMZ (Demilitarized Zone): responsible for hosting include Web and mail servers that require face interface with the outside public.

Internal Network: Special for the security of critical systems in enterprise such as the database server.

Connectivity Details: It was necessary to provide for a safe VLAN mechanism that allows for communication and isolates departments and the Guest network.

The DMZ was intended to be used to keep outside services away from the internal networks.

Configuring VLANs

Step by step VLAN was configured on the layer three switch for segmentation where each VLAN was to represent a given department or network.

VLAN Configuration Steps

Create VLANs on the Layer 3 Switch:

The VLANs were assigned proper IDs and were also given proper names.

vlan 10

name IT

vlan 20

name Finance

vlan 30

name Guest

Assign VLANs to Devices:

IT devices (desktops and servers) had topology access ports that were in VLAN 10.

Finance devices (desktops and laptops) were connected to access ports in VLAN 20.

The Guest network was created and placed in VLAN30 for those people who only have limited access on your network.

interface ethernet 1/0

switchport mode access

switchport access vlan 10

description IT_Desktop

interface ethernet 1/1

switchport mode access

switchport access vlan 20

description Finance_Laptop

interface ethernet 1/2

switchport mode access

switchport access vlan 30

description Guest_Device

Configure Inter-VLAN Routing:

Sub-Interface were configured to enable inter-VLAN routing using Switched Virtual Interface (SVIs). They all have their own gateway IP address assigned for every VLAN.

interface vlan 10

ip address 192.168.10.1 255.255.255.0

no shutdown

interface vlan 20

ip address 192.168.20.1 255.255.255.0

no shutdown

interface vlan 30

ip address 192.168.30.1 255.255.255.0

no shutdown

Control Inter-VLAN Communication:

VLANs isolation was done through the implementation of the Access Control List (ACL). For instance, the Guest network (VLAN 30) was limited from accessing IT (VLAN 10) and Financial (VLAN 20) IT resources.

```
ip access-list extended GUEST_RESTRICTIONS
```

```
deny ip 192.168.30.0 0.0.0.255 192.168.10.0 0.0.0.255
```

```
deny ip 192.168.30.0 0.0.0.255 192.168.20.0 0.0.0.255
```

```
permit ip any any
```

```
interface vlan 30
```

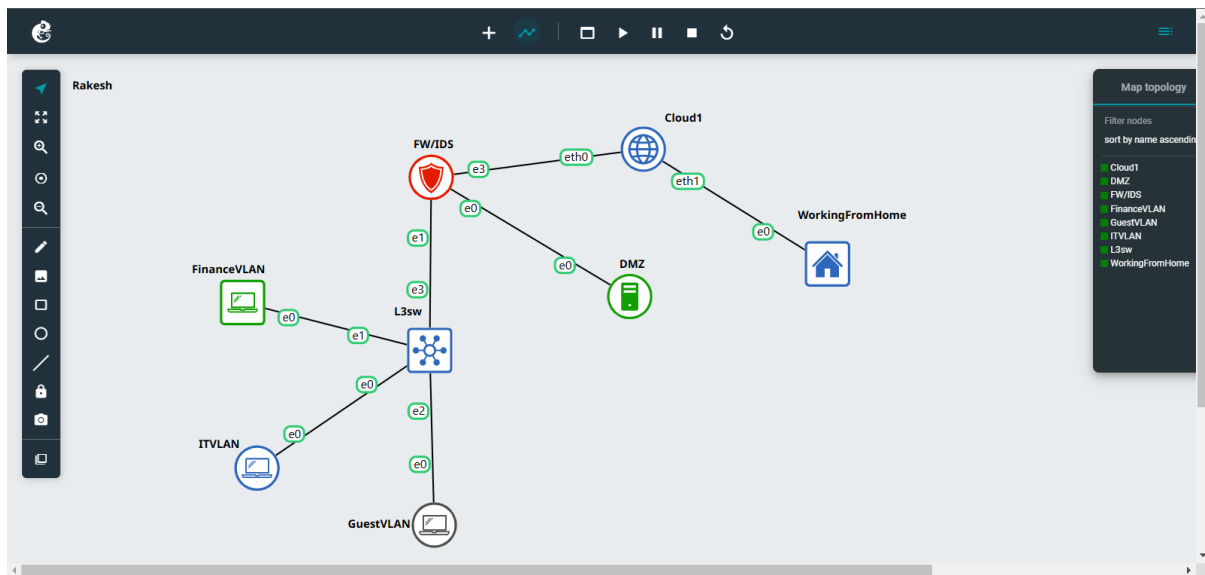
```
ip access-group GUEST_RESTRICTIONS in
```

Three VLANs: There are four VLAN created namely, IT departments (VLAN 10), Finance (VLAN 20) and Guest (VLAN 30).

The DMZ is located at the familiar area where internal network and connections to external sources are made.

Fully identified as to the VLAN to which they belong each properly bipolar bit range.

Configuration tools for inter VLAN communication routing paths and access controls.



```

MultilayerSW - PuTTY
L3sw(config)#
L3sw(config)#vlan 10
L3sw(config-vlan)#name Finance
L3sw(config-vlan)#exit
L3sw(config)#vlan 20
L3sw(config-vlan)#name IT
L3sw(config-vlan)#exit
L3sw(config)#vlan 30
L3sw(config-vlan)#name Guest
L3sw(config-vlan)#exit
L3sw(config)#interface vlan 10
L3sw(config-if)#ip add
*Nov 20 06:39:30.159: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to down
L3sw(config-if)#ip add
L3sw(config-if)#ip address 192.168.10.1 255.255.255.0
L3sw(config-if)#no shu
L3sw(config-if)#no shutdown
L3sw(config-if)#
*Nov 20 06:39:47.787: %LINK-3-UPDOWN: Interface Vlan10, changed state to down
L3sw(config-if)#exit
L3sw(config)#interface vlan 20
L3sw(config-if)#ip address 192.168.10.1 255.255.255.0
*Nov 20 06:40:15.757: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to down
L3sw(config-if)#ip address 192.168.20.1 255.255.255.0
L3sw(config-if)#no shutdown
L3sw(config-if)#exit
L3sw(config)#
*Nov 20 06:40:20.700: %LINK-3-UPDOWN: Interface Vlan20, changed state to down
L3sw(config)#interface vlan 30
L3sw(config-if)#ip address 192.168.20.1 255.255.255.0
*Nov 20 06:40:38.837: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan30, changed state to down
L3sw(config-if)#ip address 192.168.30.1 255.255.255.0
L3sw(config-if)#no shutdown
L3sw(config-if)#
*Nov 20 06:40:49.982: %LINK-3-UPDOWN: Interface Vlan30, changed state to down
L3sw(config-if)#exit
L3sw(config)#copy run
L3sw(config)#exit
L3sw#
*Nov 20 06:41:10.905: %SYS-5-CONFIG_I: Configured from console by console
L3sw#

```

```
MultilayerSW - PuTTY
Destination filename [startup-config]?
Warning: Attempting to overwrite an NVRAM configuration previously written
by a different version of the system image.
Overwrite the previous NVRAM configuration?[confirm]
Building configuration...
Compressed configuration from 1525 bytes to 903 bytes[OK]
L3sw#show vlan brif
^
% Invalid input detected at '^' marker.

L3sw#config t
Enter configuration commands, one per line. End with CNTL/Z.
L3sw(config)#interface e0/0
L3sw(config-if)#swi
L3sw(config-if)#switchport m
L3sw(config-if)#switchport mode a
L3sw(config-if)#switchport mode access
L3sw(config-if)#swi
L3sw(config-if)#switchport acc
L3sw(config-if)#switchport access vlan 20
L3sw(config-if)#no shu
L3sw(config-if)#no shutdown
L3sw(config-if)#exit
L3sw(config)#interface e0/1
L3sw(config-if)#switchport mode access
*Nov 20 06:43:26.199: %LINK-3-UPDOWN: Interface Vlan20, changed state to up
L3sw(config-if)#switchport mode access
L3sw(config-if)#
*Nov 20 06:43:27.199: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to up
L3sw(config-if)#switchport access vlan 10
L3sw(config-if)#no shutdown
L3sw(config-if)#exit
L3sw(config)#interface e0/2
L3sw(config-if)#switchport mode access
L3sw(config-if)#switchport access vlan 30
L3sw(config-if)#switchport mode access
*Nov 20 06:44:09.858: %LINK-3-UPDOWN: Interface Vlan10, changed state to up
*Nov 20 06:44:10.864: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to up
L3sw(config-if)#no shutdown
L3sw(config-if)#exit
L3sw(config)#
```

```
MultilayerSW - PuTTY
L3sw(config-if)#no shu
L3sw(config-if)#no shutdown
L3sw(config-if)#exit
L3sw(config)#interface e0/1
L3sw(config-if)#switchport mode access
*Nov 20 06:43:26.199: %LINK-3-UPDOWN: Interface Vlan20, changed state to up
L3sw(config-if)#switchport mode access
L3sw(config-if)#
*Nov 20 06:43:27.199: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to up
L3sw(config-if)#switchport access vlan 10
L3sw(config-if)#no shutdown
L3sw(config-if)#exit
L3sw(config)#interface e0/2
L3sw(config-if)#switchport mode access
L3sw(config-if)#switchport access vlan 30
L3sw(config-if)#switchport mode access
*Nov 20 06:44:09.858: %LINK-3-UPDOWN: Interface Vlan10, changed state to up
*Nov 20 06:44:10.864: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to up
L3sw(config-if)#no shutdown
L3sw(config-if)#exit
L3sw(config)#
L3sw(config)#
L3sw(config)#e
*Nov 20 06:44:38.724: %LINK-3-UPDOWN: Interface Vlan30, changed state to up
*Nov 20 06:44:39.732: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan30, changed state to up
L3sw(config)#exit
L3sw#cop
*Nov 20 06:44:41.127: %SYS-5-CONFIG_I: Configured from console by console
L3sw#copy run
L3sw#copy running-config st
L3sw#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
Compressed configuration from 1678 bytes to 984 bytes[OK]
L3sw#
L3sw#
L3sw#
```

```
L3sw#show vlan brief

VLAN Name                Status    Ports
-----
1    default                active    Et0/3, Et1/0, Et1/1, Et1/2
                                           Et1/3, Et2/0, Et2/1, Et2/2
                                           Et2/3, Et3/0, Et3/1, Et3/2
                                           Et3/3
10   Finance                active    Et0/1
20   IT                    active    Et0/0
30   Guest                 active    Et0/2
1002 fddi-default          act/unsup
1003 token-ring-default    act/unsup
1004 fddinet-default       act/unsup
1005 trnet-default         act/unsup
L3sw#
```

```

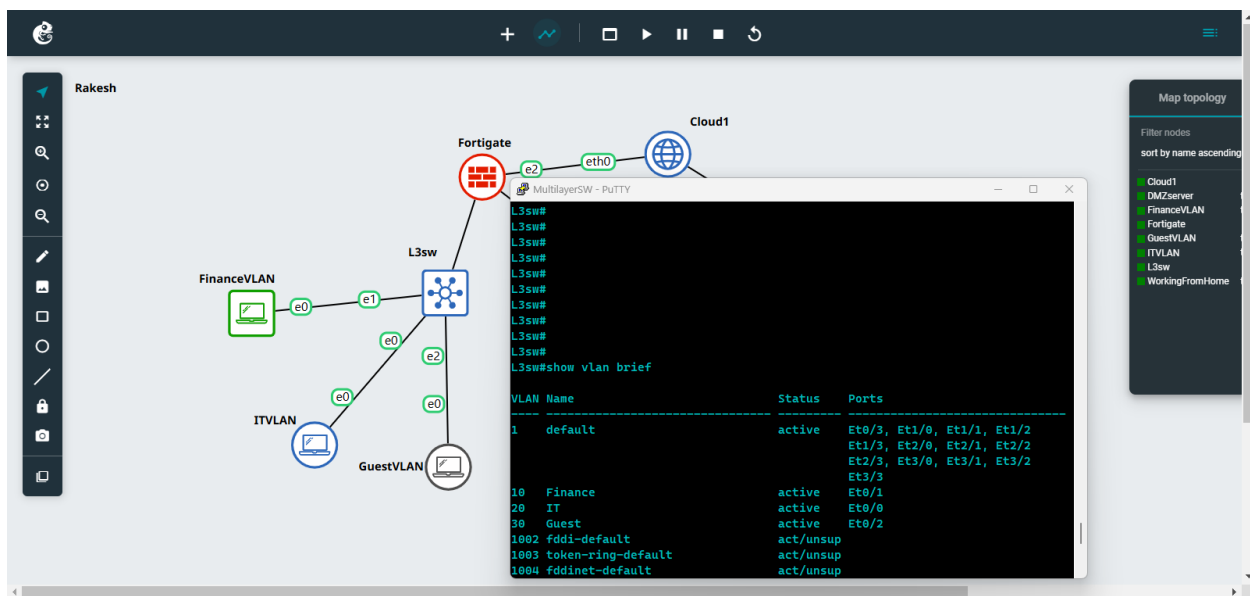
MultilayerSW - PuTTY
L3sw#
L3sw#
L3sw#
L3sw#show vlan brief

VLAN Name                Status    Ports
-----
1    default                active    Et0/3, Et1/0, Et1/1, Et1/2
                                           Et1/3, Et2/0, Et2/1, Et2/2
                                           Et2/3, Et3/0, Et3/1, Et3/2
                                           Et3/3
10   Finance                active    Et0/1
20   IT                    active    Et0/0
30   Guest                 active    Et0/2
1002 fddi-default         act/unsup
1003 token-ring-default   act/unsup
1004 fddinet-default      act/unsup
1005 trnet-default        act/unsup

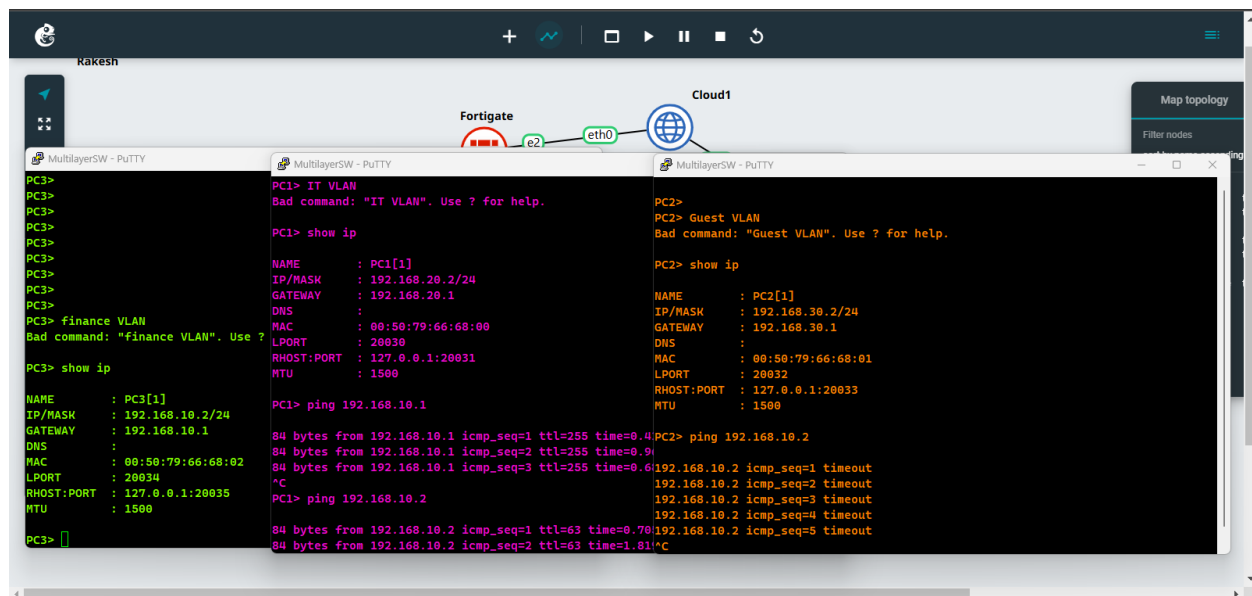
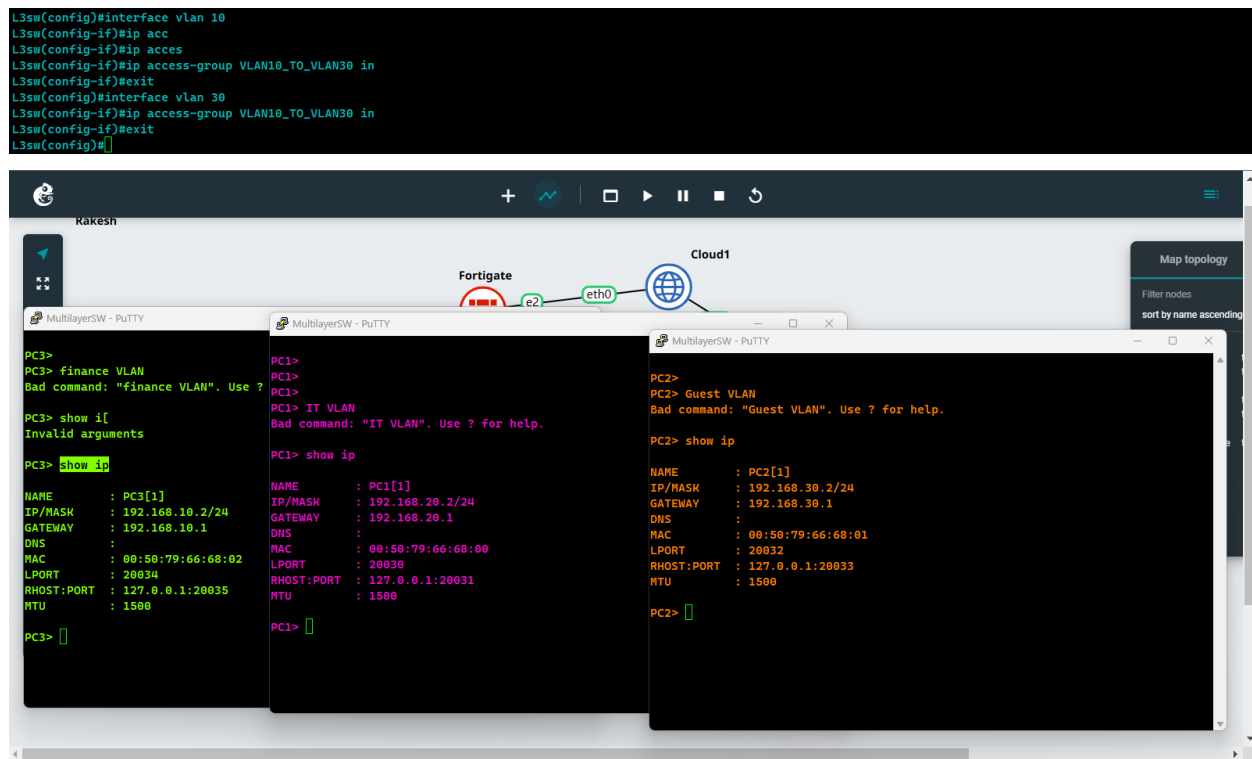
L3sw#show ip interface brief

Interface    IP-Address    OK? Method Status    Protocol
Ethernet0/0   unassigned    YES unset   up        up
Ethernet0/1   unassigned    YES unset   up        up
Ethernet0/2   unassigned    YES unset   up        up
Ethernet0/3   unassigned    YES unset   up        up
Ethernet1/0   unassigned    YES unset   up        up
Ethernet1/1   unassigned    YES unset   up        up
Ethernet1/2   unassigned    YES unset   up        up
Ethernet1/3   unassigned    YES unset   up        up
Ethernet2/0   unassigned    YES unset   up        up
Ethernet2/1   unassigned    YES unset   up        up
Ethernet2/2   unassigned    YES unset   up        up
Ethernet2/3   unassigned    YES unset   up        up
Ethernet3/0   unassigned    YES unset   up        up
Ethernet3/1   unassigned    YES unset   up        up
Ethernet3/2   unassigned    YES unset   up        up
Ethernet3/3   unassigned    YES unset   up        up
Vlan1        unassigned    YES unset   administratively down down
Vlan10       192.168.10.1  YES manual up        up
Vlan20       192.168.20.1  YES manual up        up
Vlan30       192.168.30.1  YES manual up        up

```

```
L3sw(config) t
Enter configuration commands, one per line. End with CNTL/Z.
L3sw(config)#ip access-list extended VLAN10_TO_VLAN30
L3sw(config-ext-nacl)#deny ip 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255
L3sw(config-ext-nacl)#deny ip 192.168.30.0 0.0.0.255 192.168.10.0 0.0.0.255
L3sw(config-ext-nacl)#permit ip any any
L3sw(config-ext-nacl)#exit
L3sw(config)#
L3sw(config)#
```



Firewall Deployment

To implement the processes, machines were configured such that an Ubuntu server served as the firewall and offered the DMZ for the honeypot and the web server. The network configuration

specifically entails creation of WAN, LAN and DMZ interfaces, firewall policies and safe access to the LAN interface via SSH.

Network Configuration:

Managing Network Interfaces

The server has three network interfaces:

WAN of network connected to the internet is encompassed in the scenario as ens33.

ens34 – Local Area Network (internal organization network)

hrhhmrols35 – Proving grounds (for honeypot and web server)

It was used by Netplan, a tool for wilting up of Ubuntu network configuration.

Netplan Configuration File

The /etc/netplan/01-netcfg.yaml file was customized by changing the settings on the interface eth1 to have static IPs for both LAN and DMZ while the WAN interface would be set to DHCP.

The final network configuration is:

network:

version: 2

ethernets:

ens33: # WAN interface

dhcp4: true

ens34: # LAN interface

addresses:

- 192.168.1.1/24

dhcp4: false

ens35: # DMZ interface

addresses:

- 192.168.2.1/24

dhcp4: false

Applying the Configuration

After editing the file, we applied the new network configuration:

sudo netplan apply

This allowed for verification of the configuration of the interfaces (ens33, ens34, ens35) for the required static IPs and obtained through DHCP.

Installing and Configuring Firewall (iptables):

Installing iptables

We used another service called iptables to manage the firewall and made sure it would still load on reboot.

sudo apt update

Additionally, for changing the rules you need to run the following command to install iptables and iptables-persistent permanently: *sudo apt install iptables iptables-persistent -y*

This assures the users that the settings of the firewall are initialized for use as soon as the computer has booted.

Basic iptables Configuration

We created the following rules:

Default Policies: In the AD MX record, accept no traffic by default for any new connections or forwarded ones.

sudo iptables -P INPUT DROP

sudo iptables -P FORWARD DROP

sudo iptables -P FORWARD ACCEPT

Allow loopback interface (localhost):

Because existing as a root user, the cycle is performed as follows; `sudo iptables -A INPUT -i lo -j ACCEPT` According to me super user, `sudo iptables -A OUTPUT -o lo -j ACCEPT`

Allow established and related connections:

There is the following command to allow all incoming traffic which is coming from abroad through ESTABLISHED,RELATED state: `sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT`

`sudo iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT`

Allow SSH on WAN (ens33):

`sudo iptables -A INPUT -i ens33 -p tcp --dport 22 -j ACCEPT`

Allow LAN (ens34) to communicate with the DMZ (ens35):

New rule please type: `sudo iptables -A FORWARD -i ens34 -o ens35 -j ACCEPT`

The forwarding rule is as follows :

`sudo iptables -A FORWARD -i ens34 -o ens33 -j ACCEPT`

Allow DMZ (ens35) to access the internet via the WAN interface (ens33) for HTTP/HTTPS traffic:

`sudo iptables -A FORWARD -i ens35 -o ens33 -p tcp --dport 80 -j ACCEPT`

`sudo iptables -A FORWARD -i ens35 -o ens33 -p tcp --dport 443 -j ACCEPT`

Enable NAT (Network Address Translation) for LAN and DMZ to access the internet:

Next we need to change some parameters, in order to do that we need to have the root access as follows:

`sudo iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE`

Testing and Verification:

Verifying iptables Rules

We used the following command to list the current firewall rules:

`sudo iptables -L -v -n`

This enabled us to confirm the \$RULE\$ where by we witnessed that traffic on port 22 was allowed to access the machine via SSH while other traffic was prohibited in as much as we implemented the other rules.

Connectivity Testing

Using ping, curl and wget we tested the internet connection from the LAN and confirm that it was live from the DMZ.

We also used Nmap to verify that the firewall rules were effectively blocking unwanted traffic and allowing the necessary services:

showing ports 22,80,443 scan them using

```
nmap -sS -p 22,80,443 192.168.244.128
```

Logging Suspicious Activity:

We added logging rules to monitor any dropped packets:

```
sudo iptables -A INPUT -j LOG --log-prefix "IPTables-Dropped: " --log-level 4
```

Logs are stored in /var/log/syslog and can be monitored using:

```
sudo tail -f /var/log/syslog
```

Nmap Scan Testing

As for the firewall configuration, we used Nmap port scanner in order to check if only the ports that should be opened are opened. All the testing was performed from a Kali Linux platform to probe the WAN interface, which is the IP address of the server (192.168.244.128).

Nmap Command Used

This can be done by using the following command;

```
sudo nmap -p 22,443,80 192.168.244.128 -v -p 22,443,80
```

The scan targets are defined to be SSH on port 22, HTTPS on port 443, HTTP on port 80.

-v: Enables verbose mode which provide output in more detail.

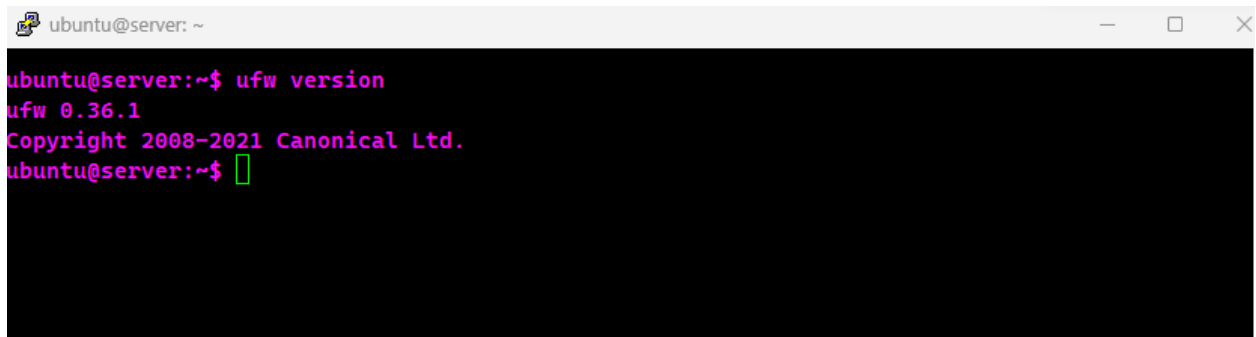
Key Observations:

Port 22 (SSH): Utilizing the online scanner, the port scan probe found that port 22 is open, therefore SSH is permitted through the firewall on the WAN side. This is expected because when configuration the Linux server we only opened port number 22 for remote management.

Port 80 (HTTP): The scan also revealed that port 80 was filtered which meant that the firewall is preventing HTTP traffic, which accredits the setting of only allowing some traffic.

Port 443 (HTTPS): In the same way, port 443 was also filtered, therefore HTTPS is also banned, in compliance to DMZ and LAN firewall settings.

MAC Address: Also, the MAC address linked with the server is provided, which also shows that the server is being run on a VMware virtual machine.

A terminal window titled 'ubuntu@server: ~' with standard window controls. The terminal shows the command 'ufw version' being executed, with the following output: 'ufw 0.36.1' and 'Copyright 2008-2021 Canonical Ltd.'. The prompt 'ubuntu@server:~\$' is followed by a green cursor.

```
ubuntu@server:~$ ufw version
ufw 0.36.1
Copyright 2008-2021 Canonical Ltd.
ubuntu@server:~$
```

ubuntu@server: /etc/netplan

```
ubuntu@server:~$ ufw version
ufw 0.36.1
Copyright 2008-2021 Canonical Ltd.
ubuntu@server:~$ sudo nano /etc/netplan/01-netcfg.yaml
ubuntu@server:~$ cd /etc/netplan
ubuntu@server:/etc/netplan$ ls | grep 01-netcfg.yaml
ubuntu@server:/etc/netplan$ ls
00-installer-config.yaml
ubuntu@server:/etc/netplan$ sudo nano 00-installer-config.yaml
ubuntu@server:/etc/netplan$ sudo netplan apply
/etc/netplan/00-installer-config.yaml:9:5: Invalid YAML: inconsistent indentation:
  ens34:
    ^
ubuntu@server:/etc/netplan$ sudo nano 00-installer-config.yaml
ubuntu@server:/etc/netplan$ sudo netplan apply
ubuntu@server:/etc/netplan$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:29:09:38 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.244.128/24 metric 100 brd 192.168.244.255 scope global dynamic ens33
        valid_lft 1777sec preferred_lft 1777sec
    inet6 fe80::20c:29ff:fe29:938/64 scope link
        valid_lft forever preferred_lft forever
3: ens37: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 00:0c:29:29:09:42 brd ff:ff:ff:ff:ff:ff
    altname enp2s5
4: ens38: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 00:0c:29:29:09:4c brd ff:ff:ff:ff:ff:ff
    altname enp2s6
ubuntu@server:/etc/netplan$
```

```
ubuntu@server:/etc/netplan$ iptables
iptables v1.8.7 (nf_tables): no command specified
Try 'iptables -h' or 'iptables --help' for more information.
ubuntu@server:/etc/netplan$
```

```
ubuntu@server:/etc/netplan$ sudo iptables -F
sudo iptables -t nat -F
ubuntu@server:/etc/netplan$
```

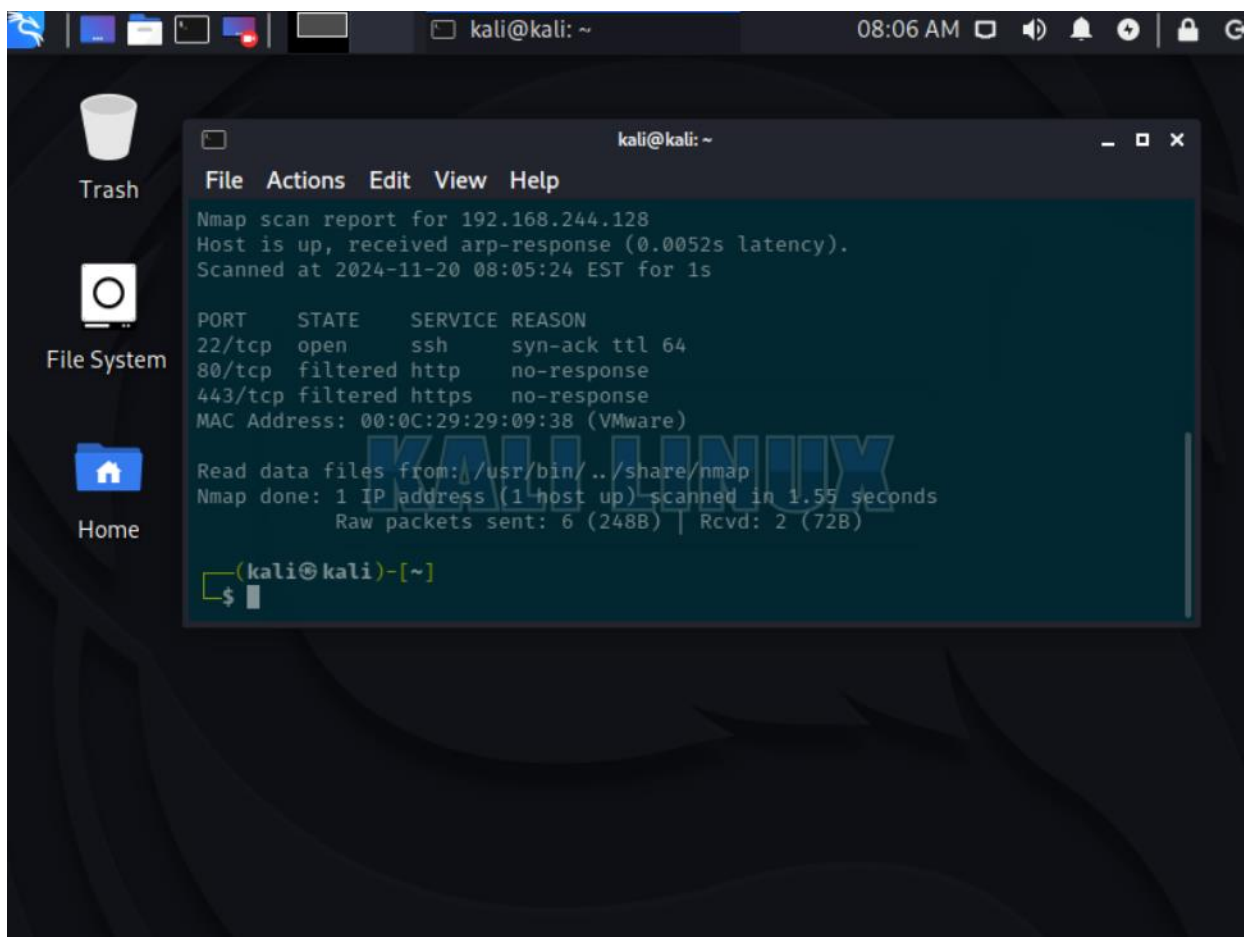


```
ubuntu@server: /etc/netplan
link/ether 00:0c:29:29:09:4c brd ff:ff:ff:ff:ff:ff
altname enp2s6
ubuntu@server:/etc/netplan$ iptables
iptables v1.8.7 (nf_tables): no command specified
Try 'iptables -h' or 'iptables --help' for more information.
ubuntu@server:/etc/netplan$ sudo iptables -F
sudo iptables -t nat -F
ubuntu@server:/etc/netplan$ sudo iptables -A INPUT -i ens33 -p tcp --dport 22 -j ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables-save > /etc/iptables/rules.v4
-bash: /etc/iptables/rules.v4: No such file or directory
ubuntu@server:/etc/netplan$ sudo iptables -L -v -n
Chain INPUT (policy ACCEPT 139K packets, 202M bytes)
  pkts bytes target     prot opt in     out     source                   destination
  11 1160 ACCEPT     tcp  --  ens33  *      0.0.0.0/0                0.0.0.0/0                tcp
dpt:22

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source                   destination

Chain OUTPUT (policy ACCEPT 54356 packets, 2256K bytes)
  pkts bytes target     prot opt in     out     source                   destination

Chain f2b-sshd (0 references)
  pkts bytes target     prot opt in     out     source                   destination
ubuntu@server:/etc/netplan$ sudo iptables -P INPUT DROP
sudo iptables -P FORWARD DROP
sudo iptables -P OUTPUT ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables -A INPUT -i lo -j ACCEPT
sudo iptables -A OUTPUT -o lo -j ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables -A INPUT -i ens33 -p tcp --dport 22 -j ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables -A FORWARD -i ens34 -o ens33 -j ACCEPT
sudo iptables -A FORWARD -i ens34 -o ens35 -j ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables -A FORWARD -i ens35 -o ens33 -p tcp --dport 80 -j ACCEPT
sudo iptables -A FORWARD -i ens35 -o ens33 -p tcp --dport 443 -j ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables -A FORWARD -i ens34 -o ens35 -j ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables -A FORWARD -i ens35 -o ens33 -j DROP
ubuntu@server:/etc/netplan$
```



```

ubuntu@server:/etc/netplan
iptables v1.8.7 (nf_tables): no command specified
Try 'iptables -h' or 'iptables --help' for more information.
ubuntu@server:/etc/netplan$ sudo iptables -F
sudo iptables -t nat -F
ubuntu@server:/etc/netplan$ sudo iptables -A INPUT -i ens33 -p tcp --dport 22 -j ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables-save > /etc/iptables/rules.v4
-bash: /etc/iptables/rules.v4: No such file or directory
ubuntu@server:/etc/netplan$ sudo iptables -L -v -n
Chain INPUT (policy ACCEPT 139K packets, 202M bytes)
  pkts bytes target     prot opt in     out     source            destination
    11 1160 ACCEPT     tcp  --  ens33  *      0.0.0.0/0         0.0.0.0/0         tcp
dpt:22

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source            destination

Chain OUTPUT (policy ACCEPT 54356 packets, 2256K bytes)
  pkts bytes target     prot opt in     out     source            destination

Chain f2b-sshd (0 references)
  pkts bytes target     prot opt in     out     source            destination
ubuntu@server:/etc/netplan$ sudo iptables -P INPUT DROP
sudo iptables -P FORWARD DROP
sudo iptables -P OUTPUT ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables -A INPUT -i lo -j ACCEPT
sudo iptables -A OUTPUT -o lo -j ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables -A INPUT -i ens33 -p tcp --dport 22 -j ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables -A FORWARD -i ens34 -o ens33 -j ACCEPT
sudo iptables -A FORWARD -i ens34 -o ens35 -j ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables -A FORWARD -i ens35 -o ens33 -p tcp --dport 80 -j ACCEPT
sudo iptables -A FORWARD -i ens35 -o ens33 -p tcp --dport 443 -j ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables -A FORWARD -i ens34 -o ens35 -j ACCEPT
ubuntu@server:/etc/netplan$ sudo iptables -A FORWARD -i ens35 -o ens33 -j DROP
ubuntu@server:/etc/netplan$ sudo iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE
ubuntu@server:/etc/netplan$ sudo iptables-save > /etc/iptables/rules.v4
-bash: /etc/iptables/rules.v4: No such file or directory
ubuntu@server:/etc/netplan$

```

```

(kali㉿kali)-[~]
└─$ sudo nmap -p 22,443,80 192.168.244.128 -vv
[sudo] password for kali:
Starting Nmap 7.91 ( https://nmap.org ) at 2024-11-20 08:05 EST
Initiating ARP Ping Scan at 08:05
Scanning 192.168.244.128 [1 port]
Completed ARP Ping Scan at 08:05, 0.10s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 08:05
Completed Parallel DNS resolution of 1 host. at 08:05, 0.01s elapsed
Initiating SYN Stealth Scan at 08:05
Scanning 192.168.244.128 [3 ports]
Discovered open port 22/tcp on 192.168.244.128
Completed SYN Stealth Scan at 08:05, 1.24s elapsed (3 total ports)
Nmap scan report for 192.168.244.128
Host is up, received arp-response (0.0052s latency).
Scanned at 2024-11-20 08:05:24 EST for 1s

PORT      STATE      SERVICE REASON
22/tcp    open      ssh      syn-ack ttl 64
80/tcp    filtered  http      no-response
443/tcp   filtered  https     no-response
MAC Address: 00:0C:29:29:09:38 (VMware)

Read data files from: /usr/bin/../../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.55 seconds
Raw packets sent: 6 (248B) | Rcvd: 2 (72B)

```

Open VPN Deployment in Ubuntu server

The establishment of a secure OpenVPN for the purpose of allowing remote connection on an Ubuntu server is described. The concern was to establish and provide secure connection from client systems to the organization's internal server, where valuable assets reside. Installation and configuration remain the most critical processes since they enable the software to meet the intended users' needs fully. To create an effectively protected VPN answer, OpenVPN was integrated into Easy-RSA software – for certificate and encryption management. Originally, the process involved upgrading the server so that it would work well with the current version and be secure.

Thus, there was established a Certificate Authority (CA) which would sign both server and client certificates. After that it generated the certificates and keys for the server, Diffie-Hellman parameters for encryption and an HMAC to add an increased layer of security. These elements made make sure that there was a secure method of authentication or the encryption of the VPN.

In the server configuration file (server.conf), the port (udp) was set as the protocol, the encryption system (AES-256-CBC) and the SHA256 authentication was used and the DNS forwarding for the VPN clients. Network forwarding was set to 1 for traffic routing and firewall rules were modified with UFW to add VPN (port 1194/UDP) and SSH. Finally the OpenVPN service was started and meant to start at boot time.

Client Access Setup

Client configuration was to create .ovpn files that include the appropriate certificate and key, as well as the connection profile. As for the other files, these files are to be copied onto the clients' computers for their easy use in program design. It also needed settings capable of providing encryption, secure keys retention, as well as server connection parameters.

Testing and Verification

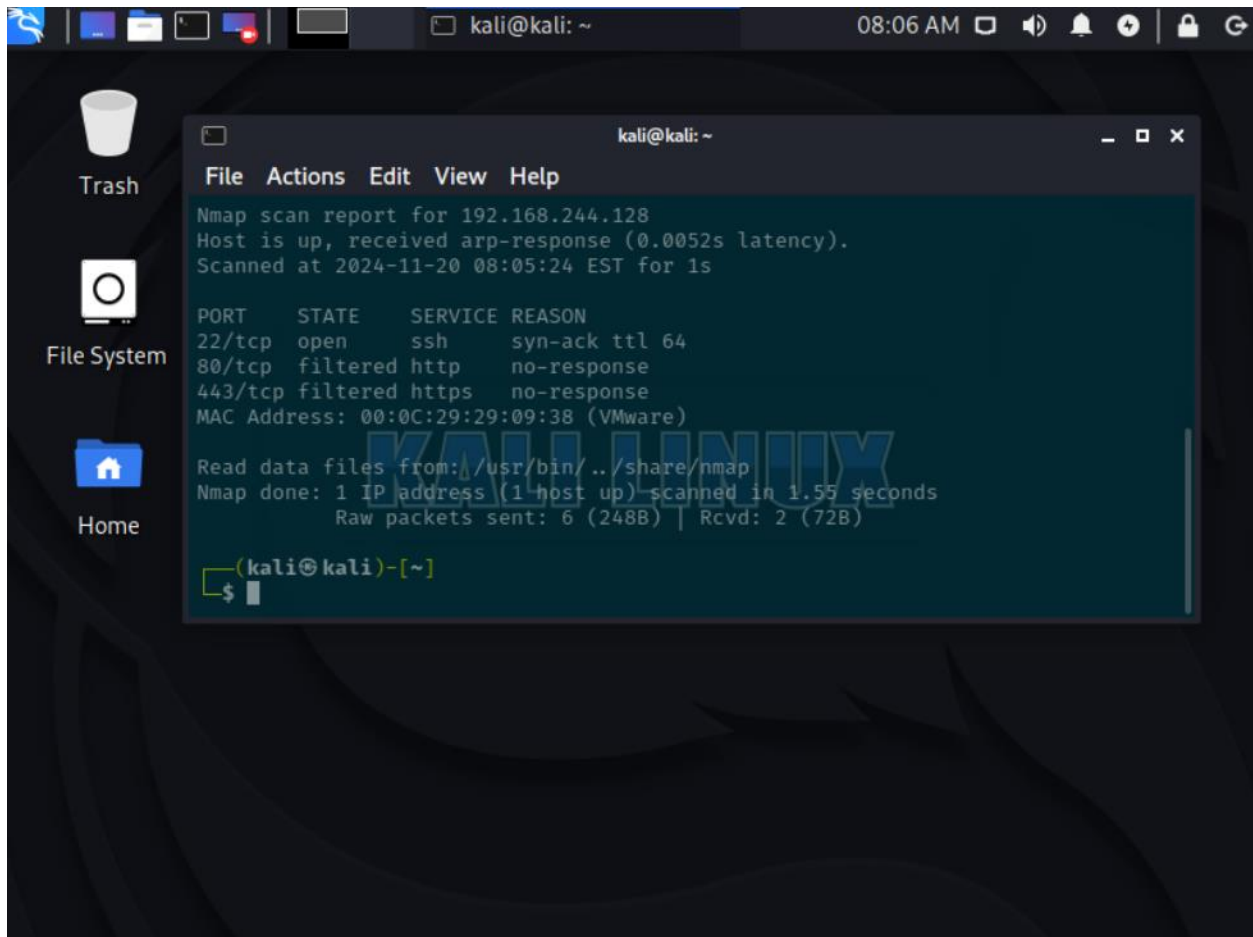
The employed VPN was checked for its functionality and its protection capability. For OpenVPN client configuration from a client device installation was conducted and then use the configuration file in .ovpn format to connect. Finally, the connection was successful by browsing the file server and other private web sites within the network.

Encryption was confirmed by auditing the client logs and it revealed that secure connection was initiated properly. Furthermore, tool for monitoring the traffic (tcpdump) was used to ensure that all the transmitted data was encrypted so that it could not be intercepted by a third party.

```
ubuntu@server: ~/openvpn-ca
.....*.....*.....
.....+.....+.....
.....+.....+.....
*****
DH parameters of size 2048 created at /home/ubuntu/openvpn-ca/pki/dh.pem

ubuntu@server:~/openvpn-ca$ sudo cp ~/openvpn-ca/pki/ca.crt /etc/openvpn
sudo cp ~/openvpn-ca/pki/private/server.key /etc/openvpn
sudo cp ~/openvpn-ca/pki/issued/server.crt /etc/openvpn
sudo cp ~/openvpn-ca/pki/dh.pem /etc/openvpn
sudo cp ~/openvpn-ca/ta.key /etc/openvpn
cp: cannot stat '/home/ubuntu/openvpn-ca/pki/private/server.key': No such file or directory
cp: cannot stat '/home/ubuntu/openvpn-ca/pki/issued/server.crt': No such file or directory
cp: cannot stat '/home/ubuntu/openvpn-ca/ta.key': No such file or directory
ubuntu@server:~/openvpn-ca$ sudo cp /pki/ca.crt /etc/openvpn
sudo cp pki/private/server.key /etc/openvpn
sudo cp pki/issued/server.crt /etc/openvpn
sudo cp pki/dh.pem /etc/openvpn
sudo cp ta.key /etc/openvpn
cp: cannot stat '/pki/ca.crt': No such file or directory
cp: cannot stat 'pki/private/server.key': No such file or directory
cp: cannot stat 'pki/issued/server.crt': No such file or directory
cp: cannot stat 'ta.key': No such file or directory
ubuntu@server:~/openvpn-ca$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/serve
er.conf.gz /etc/openvpn/
sudo gunzip /etc/openvpn/server.conf.gz
cp: cannot stat '/usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz': No suc
h file or directory
gzip: /etc/openvpn/server.conf.gz: No such file or directory
ubuntu@server:~/openvpn-ca$ sudo nano /etc/openvpn/server.conf
ubuntu@server:~/openvpn-ca$ sudo nano /etc/sysctl.conf
ubuntu@server:~/openvpn-ca$ sudo sysctl -p
net.ipv4.conf.default.rp_filter = 1
ubuntu@server:~/openvpn-ca$ sudo ufw allow 1194/udp
sudo ufw allow OpenSSH
sudo ufw enable
```

```
ubuntu@server: ~/openvpn-ca
sudo cp ~/openvpn-ca/pki/private/server.key /etc/openvpn
sudo cp ~/openvpn-ca/pki/issued/server.crt /etc/openvpn
sudo cp ~/openvpn-ca/pki/dh.pem /etc/openvpn
sudo cp ~/openvpn-ca/ta.key /etc/openvpn
cp: cannot stat '/home/ubuntu/openvpn-ca/pki/private/server.key': No such file or directory
cp: cannot stat '/home/ubuntu/openvpn-ca/pki/issued/server.crt': No such file or directory
cp: cannot stat '/home/ubuntu/openvpn-ca/ta.key': No such file or directory
ubuntu@server:~/openvpn-ca$ sudo cp /pki/ca.crt /etc/openvpn
sudo cp pki/private/server.key /etc/openvpn
sudo cp pki/issued/server.crt /etc/openvpn
sudo cp pki/dh.pem /etc/openvpn
sudo cp ta.key /etc/openvpn
cp: cannot stat '/pki/ca.crt': No such file or directory
cp: cannot stat 'pki/private/server.key': No such file or directory
cp: cannot stat 'pki/issued/server.crt': No such file or directory
cp: cannot stat 'ta.key': No such file or directory
ubuntu@server:~/openvpn-ca$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/serve
er.conf.gz /etc/openvpn/
sudo gunzip /etc/openvpn/server.conf.gz
cp: cannot stat '/usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz': No suc
h file or directory
gzip: /etc/openvpn/server.conf.gz: No such file or directory
ubuntu@server:~/openvpn-ca$ sudo nano /etc/openvpn/server.conf
ubuntu@server:~/openvpn-ca$ sudo nano /etc/sysctl.conf
ubuntu@server:~/openvpn-ca$ sudo sysctl -p
net.ipv4.conf.default.rp_filter = 1
ubuntu@server:~/openvpn-ca$ sudo ufw allow 1194/udp
sudo ufw allow OpenSSH
sudo ufw enable
Rules updated
Rules updated (v6)
Rules updated
Rules updated (v6)
Command may disrupt existing ssh connections. Proceed with operation (y|n)? n
Aborted
ubuntu@server:~/openvpn-ca$ sudo systemctl start openvpn@server
sudo systemctl enable openvpn@server
Job for openvpn@server.service failed because the control process exited with error code.
See "systemctl status openvpn@server.service" and "journalctl -xeu openvpn@server.service" for details.
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn@server.service → /lib/systemd/system/openvpn@.service.
ubuntu@server:~/openvpn-ca$
```



IDS/IPS Implementation

The IDS/IPS implemented is Snort, we download snort to the Ubuntu server and also installed the default rules and set up custom rule:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"SSH Brute Force"; flags:S; threshold:type both, track by_dst, count 5, seconds 60; sid:1000001;)
```

To test the IDS, we triggered attack traffic from the attacking machine, in this case, Kali Linux, and followed up with the Snort log to identify whether it was able to detect any rule violation, which it actually detected.

```

ubuntu@server: ~/openvpn-ca
ubuntu@server:~/openvpn-ca$ sudo apt install snort -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
libauthen-sasl-perl libclone-perl libdaq2 libdata-dump-perl libdumbnet1
libencode-locale-perl libfile-listing-perl libfont-afm-perl libhtml-form-perl
libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl
libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl
libhttp-negotiate-perl libio-html-perl libio-socket-ssl-perl liblwp-mediatypes-perl
liblwp-protocol-https-perl libmailtools-perl libnet-http-perl libnet-smtp-ssl-perl
libnet-ssleay-perl libtimedate-perl libtry-tiny-perl liburi-perl libwww-perl
libwww-robotrules-perl oinkmaster perl-openssl-defaults snort-common
snort-common-libraries snort-rules-default
Suggested packages:
libdigest-hmac-perl libgssapi-perl libcrypt-ssleay-perl libsub-name-perl
libbusiness-isbn-perl libauthen-ntlm-perl snort-doc
The following NEW packages will be installed:
libauthen-sasl-perl libclone-perl libdaq2 libdata-dump-perl libdumbnet1
libencode-locale-perl libfile-listing-perl libfont-afm-perl libhtml-form-perl
libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl
libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl
libhttp-negotiate-perl libio-html-perl libio-socket-ssl-perl liblwp-mediatypes-perl
liblwp-protocol-https-perl libmailtools-perl libnet-http-perl libnet-smtp-ssl-perl
libnet-ssleay-perl libtimedate-perl libtry-tiny-perl liburi-perl libwww-perl
libwww-robotrules-perl oinkmaster perl-openssl-defaults snort snort-common
snort-common-libraries snort-rules-default
0 upgraded, 37 newly installed, 0 to remove and 176 not upgraded.
Need to get 3,644 kB of archives.
After this operation, 14.7 MB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 snort-common-libraries amd64
2.9.15.1-6build1 [882 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 snort-rules-default all 2.9.15.1-6build1 [146 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 snort-common all 2.9.15.1-6build1 [49.7 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 libdumbnet1 amd64 1.12-10 [27.8 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 libdaq2 amd64 2.0.7-5 [83.5 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 snort amd64 2.9.15.1-6build1 [792 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libclone-perl amd64 0.45-1build3 [11.0 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libdata-dump-perl all 1.25-1 [25.9 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libencode-locale-perl all 1.05-1.1 [11.8 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libtimedate-perl all 2.3300-2 [34.0 kB]

```

```

ubuntu@server: ~/openvpn-ca
ubuntu@server:~/openvpn-ca$ snort -V

  __-  -> Snort! <*-
o"  )~  Version 2.9.15.1 GRE (Build 15125)
  ' ' '  By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
          Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
          Copyright (C) 1998-2013 Sourcefire, Inc., et al.
          Using libpcap version 1.10.1 (with TPACKET_V3)
          Using PCRE version: 8.39 2016-06-14
          Using ZLIB version: 1.2.11

ubuntu@server:~/openvpn-ca$ █

```

```
ubuntu@server: ~/openvpn-ca
GNU nano 6.2 /etc/snort/snort.conf
# /etc/snort/snort.$interface.conf (where '$interface' is the name of your
# network interface) and adjust the value there.
#
# The Debian init.d script is defined in such a way
# that you can run multiple instances.
#####
# Step #1: Set the network variables. For more information, see README.variables
#####
# Setup the network addresses you are protecting
#
# Note to Debian users: this value is overridden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET any

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
ipvar SMTP_SERVERS $HOME_NET

# List of web servers on your network
ipvar HTTP_SERVERS $HOME_NET

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```



```
kali@kali: ~
08:54 AM

File Actions Edit View Help

(kali@kali)-[~]
$ sudo nmap -sS -p- 192.168.244.128 -vv
Starting Nmap 7.91 ( https://nmap.org ) at 2024-11-20 08:52 EST
Initiating ARP Ping Scan at 08:52
Scanning 192.168.244.128 [1 port]
Completed ARP Ping Scan at 08:52, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 08:52
Completed Parallel DNS resolution of 1 host. at 08:52, 0.06s elapsed
Initiating SYN Stealth Scan at 08:52
Scanning 192.168.244.128 [65535 ports]
Discovered open port 22/tcp on 192.168.244.128
Completed SYN Stealth Scan at 08:52, 6.19s elapsed (65535 total ports)
Nmap scan report for 192.168.244.128
Host is up, received arp-response (0.0020s latency).
Scanned at 2024-11-20 08:52:51 EST for 6s
Not shown: 65534 closed ports
Reason: 65534 resets
PORT      STATE SERVICE REASON
22/tcp    open  ssh      syn-ack ttl 64
MAC Address: 00:0C:29:29:09:38 (VMware)

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 6.48 seconds
Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)

(kali@kali)-[~]
$
```

```
ubuntu@server:/var/log/snort$ sudo cat snort.alert.fast
11/20-10:50:16.135590 11/20-10:50:16.135590 [1:827:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
11/20-10:50:16.142784 11/20-10:50:16.142784 [1:827:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
11/20-10:50:16.798639 11/20-10:50:16.798639 [1:827:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
11/20-10:50:17.026784 11/20-10:50:17.026784 [1:827:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::1:ff05:7d2a
11/20-10:52:39.364476 11/20-10:52:39.364476 [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.244.130:37711 -> 192.168.244.128:161
11/20-10:52:39.372573 11/20-10:52:39.372573 [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.244.130:37711 -> 192.168.244.128:705
11/20-10:52:53.776519 11/20-10:52:53.776519 [1:1420:11] SNMP trap tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.244.130:41064 -> 192.168.244.128:162
11/20-10:52:54.597440 11/20-10:52:54.597440 [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.244.130:41064 -> 192.168.244.128:705
11/20-10:52:54.836432 11/20-10:52:54.836432 [1:249:8] DDOS mstream client to handler [**] [Classification: Attempted Denial of Service] [Priority: 2] {TCP} 192.168.244.130:41064 -> 192.168.244.128:15104
11/20-10:52:55.158503 11/20-10:52:55.158503 [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.244.130:41064 -> 192.168.244.128:161
ubuntu@server:/var/log/snort$
```

Testing and Security Assessment

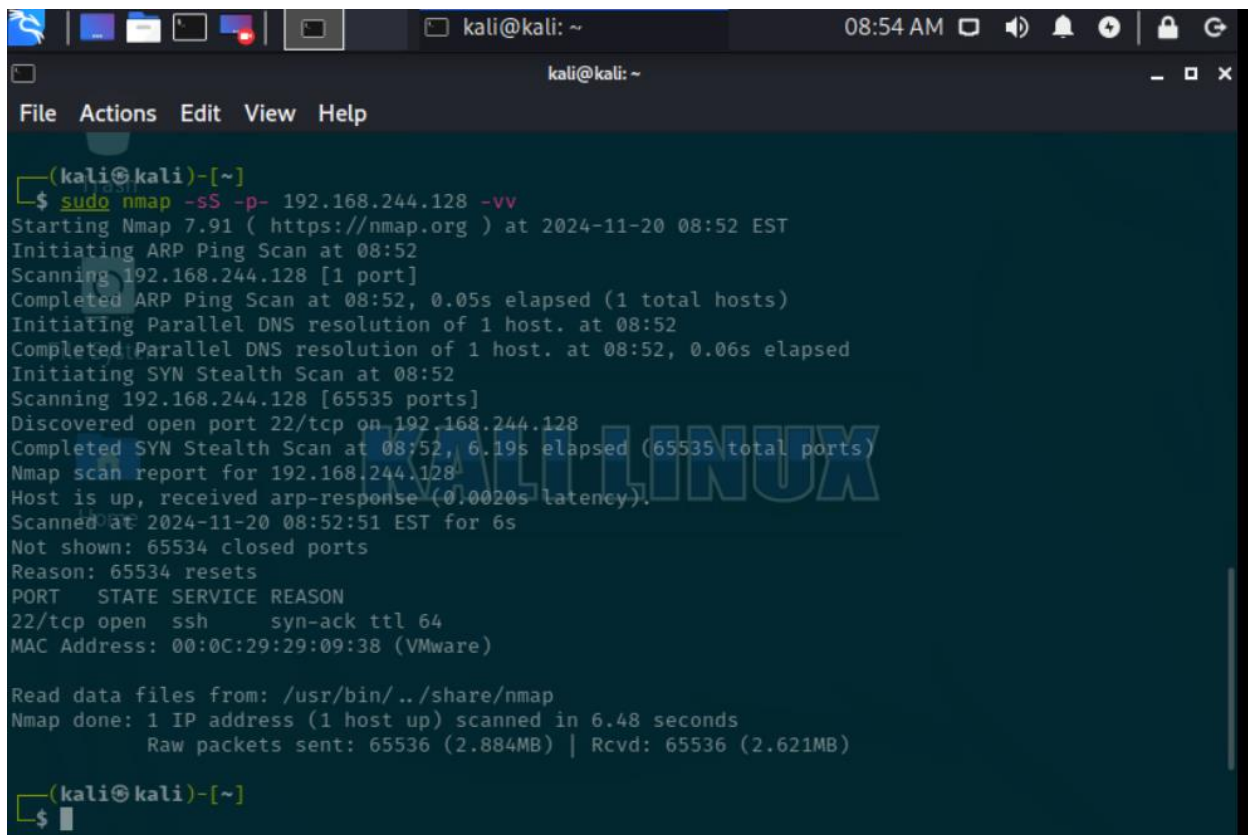
To test the implemented security in the network we simulated attacks from the network default gateway using both nmap and metasploit framework. The nmap scan showed negative result with all ports filtered by the firewall and but SSH port was luckily exposed which also was difficult to exploit through brute force and other attack vector due to strong ssh cyphers / encryption. We tried to utilize ssh exploit to obtain a reverse shell but it also didn't work. This simplified well implemented security but this some security misconfiguration can lead to compromise like exposing ssh to the internet.

The penetration testing of attempting to open the OpenSSH service by Metasploit was futile in the attempt. Although the service discovered was in port 22, the exploitation did not give the attacker a session. This could be due to several factors, including:

- A more secure version of OpenSSH, for instance use of more secure authentication methods (such as keys instead of passwords).
- The system likely having settings as fail2ban, rate-limiting or firewall that prevents brute force/exploiting.
- The particular Metasploit exploit used may not have been suitable for the particular SSH version in operation on the target system

Mitigation

Configuration optimization for security misconfiguration



```
(kali@kali)~[~]
$ sudo nmap -sS -p- 192.168.244.128 -vv
Starting Nmap 7.91 ( https://nmap.org ) at 2024-11-20 08:52 EST
Initiating ARP Ping Scan at 08:52
Scanning 192.168.244.128 [1 port]
Completed ARP Ping Scan at 08:52, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 08:52
Completed Parallel DNS resolution of 1 host. at 08:52, 0.06s elapsed
Initiating SYN Stealth Scan at 08:52
Scanning 192.168.244.128 [65535 ports]
Discovered open port 22/tcp on 192.168.244.128
Completed SYN Stealth Scan at 08:52, 6.19s elapsed (65535 total ports)
Nmap scan report for 192.168.244.128
Host is up, received arp-response (0.0020s latency).
Scanned at 2024-11-20 08:52:51 EST for 6s
Not shown: 65534 closed ports
Reason: 65534 resets
PORT      STATE SERVICE REASON
22/tcp    open  ssh     syn-ack ttl 64
MAC Address: 00:0C:29:29:09:38 (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 6.48 seconds
Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)

(kali@kali)~[~]
$
```

```
kali@kali: ~
09:08 AM
kali@kali: ~
File Actions Edit View Help
Name Current Setting Required Description
RHOSTS 192.168.244.128 yes The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT 22 yes The target port (TCP)
USERNAME ubuntu yes The username to login as

Payload options (cmd/unix/interact):
Name Current Setting Required Description
cmd CMD 'cat /etc/passwd' yes The command to execute via a remote process
interact CMD 'cat /etc/passwd' yes The command to execute via a remote process

Exploit target:
Home
Id Name
0 Unix-based Tectia SSH 6.3 or prior

msf6 exploit(unix/ssh/tectia_passwd_changereq) > set RHOSTS 192.168.244.128
RHOSTS => 192.168.244.128
msf6 exploit(unix/ssh/tectia_passwd_changereq) > set USERNAME ubuntu
USERNAME => ubuntu
msf6 exploit(unix/ssh/tectia_passwd_changereq) > run

[*] 192.168.244.128:22 - 192.168.244.128:22 - Sending USERAUTH Change request ...
[*] 192.168.244.128:22 - 192.168.244.128:22 - Auths that can continue: 51
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ssh/tectia_passwd_changereq) >
```