

SECURE NETWORK DESIGN, FIREWALL CONFIGURATION, AND ATTACK SIMULATION IN GNS3 GROUP 11 – NETWORK DESIGN IMPLEMENTATION PROJECT

INTRODUCTION

The purpose of the project is to organize and introduce a secure segmented network based on the GNS3, FortiGate firewall, Cisco router, and several virtual machines in the form of Kali Linux, Windows 10, and Ubuntu. The main idea is to establish complete isolated cyber-range where internal attacks like port scans and DoS-type traffic can be safely modeled and studied.

It is made deliberately unreachable (not connected to the internet) to ensure all data within the lab is kept within the lab environment. This allows a secure practical introduction of various heavy-lifting offensive tools (nmap and hping3, etc.), and also to examine a range of defensive features including firewall policy, segmentation, logs and packet capture.

NETWORK TOPOLOGY OVERVIEW

The ultimate topology will be composed of the following components:

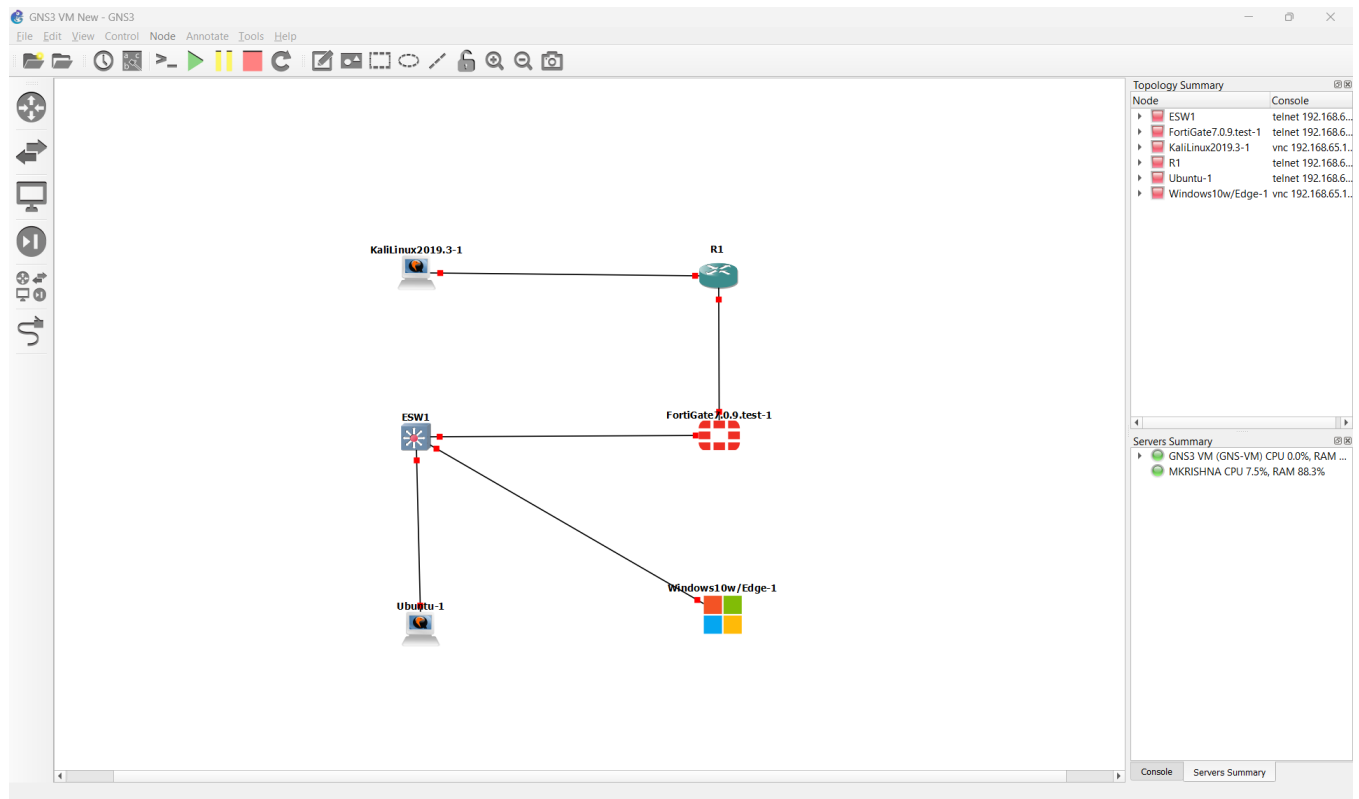
- Kali Linux – attacker machine.
- Windows 10 – victim workstation.
- Ubuntu -internal server/ normal host.
- FortiGate firewall- principal safety and observatory.
- Cisco R1 router- routing LAN segments to LAN segments.
- ESW1 switch - LAN predictable distribution switch.

Kali machine exists within a single subnet and windows and Ubuntu also exist within a subnet behind the firewall. The router links the attacker network to a firewall through a transit network. This is a segmented design that enables us to have control and monitoring of all cross-LAN traffic using the firewall.

IP ADDRESSING AND DEVICE ROLES

The addressing scheme is organized into three main networks:

Windows/Ubuntu LAN: 192.168.10.0/24



Kali, Windows, Ubuntu, FortiGate, R1, and ESW1 make up the GNS3 network topology

Windows 10: 192.168.10.30;

Ubuntu: 192.168.10.40;

default gateway: 192.168.10.1 (FortiGate port3)

Kali LAN: 192.168.20.0/24

-Gateway: 192.168.20.1 (R1 Fa0/1)

10.0.0.0/24 is the router-firewall transit network.

-FortiGate port 2: 10.0.0.2;

-R1 Fa0/0: 10.0.0.1

The FortiGate firewall therefore has:

port3 – inside LAN (192.168.10.1/24), connecting to Windows and Ubuntu via ESW1.

port2 – outside/transit (10.0.0.2/24), connected to R1.

port1 – disabled (no internet access by design).

This setup will guarantee that any traffic between Kali (attacker network) and Windows (victim) will go through the router and the firewall and it will be visible to security controls and logs.

CONFIGURATION STEPS

Router R1:

The Cisco R1 router will have two interfaces and a static route:

FastEthernet0/0 = 10.0.0.1 (toward FortiGate)

FastEthernet0/1 = 192.168.20.1 (toward Kali)

A static route was used to reach the Windows/Ubuntu LAN:

```
ip route 192.168.10.0 255.255.255.0 10.0.0.2
```

The NAT is switched off in such a way that all the environment is internal and offline.

FortiGate Firewall:

On the FortiGate, interfaces were configured as:

port3: 192.168.10.1/24 (inside LAN for Windows and Ubuntu).

port2: 10.0.0.2/24 (connected to R1).

The fundamental IPv4 policy is developed to permit the LAN-to-LAN communication using the firewall.

Source: all internal networks (e.g., 192.168.20.0/24, 192.168.10.0/24).

Destination: all internal networks.

Action: ACCEPT.

NAT: Disabled.

Logging: Enabled for all sessions.

This policy may also have optional DoS protection or IPS profiles in order to record high traffic patterns or suspicious traffic patterns.

```
FortiGate7.0.9.test-1 - PuTTY
config firewall policy
edit 1
    set name "LAN-to-R1"
    set uuid dc7b31ca-c030-51f0-4994-bed53b78a9ff
    set srcintf "port1"
    set dstintf "port2"
    set action accept
    set srcaddr "all"
    set dstaddr "all"
    set schedule "always"
    set service "ALL"
next
edit 2
    set name "R1-to-LAN"
    set uuid cc97729a-c185-51f0-ed70-b9eaf536418b
    set srcintf "port2"
    set dstintf "port1"
    set action accept
    set srcaddr "all"
    set dstaddr "all"
    set schedule "always"
    set service "ALL"
next
edit 10
    set name "LAN20-to-LAN10"
    set uuid 303ae642-c18b-51f0-6793-90ef03c74c26
    set srcintf "port2"
    set dstintf "port1"
    set action accept
    set srcaddr "all"
    set dstaddr "all"
    set schedule "always"
    set service "ALL"
next
edit 11
    set name "LAN10-to-LAN20"
    set uuid 8c844fc2-c18d-51f0-182c-ad33d17113af
    set srcintf "port1"
    set dstintf "port2"
    set action accept
    set srcaddr "all"
    set dstaddr "all"
    set schedule "always"
    set service "ALL"
next
edit 20
    set name "LAN20-inside"
    set uuid 1f573fa2-c36f-51f0-a3d2-37638c9f616a
    set srcintf "port1"
    set dstintf "port1"
    set action accept
    set srcaddr "all"
    set dstaddr "all"
    set schedule "always"
    set service "ALL"
```

FortiGate IPv4 policy allowing LAN-to-LAN traffic with logging enabled.

Switch ESW1

ESW1 switch works at Layer 2 and contains local connections in the LAN:

- Dynamic ports to windows and Ubuntu hosts associated with VLAN 10.
- Port3 of FortiGate linked to uplink port.

This basic switch setup will guarantee that anything attempting to traverse windows/ubuntu would be aggregated and sent to the firewall to be examined.

Host Configuration

The Windows and Ubuntu hosts are given the hostname: 192.168.10.1 and 192.168.10.1 respectively and with 192.168.10.1 as default gateway. The Kali machine has been set with the router as its gateway and the 192.168.20.0/24 IP address set.

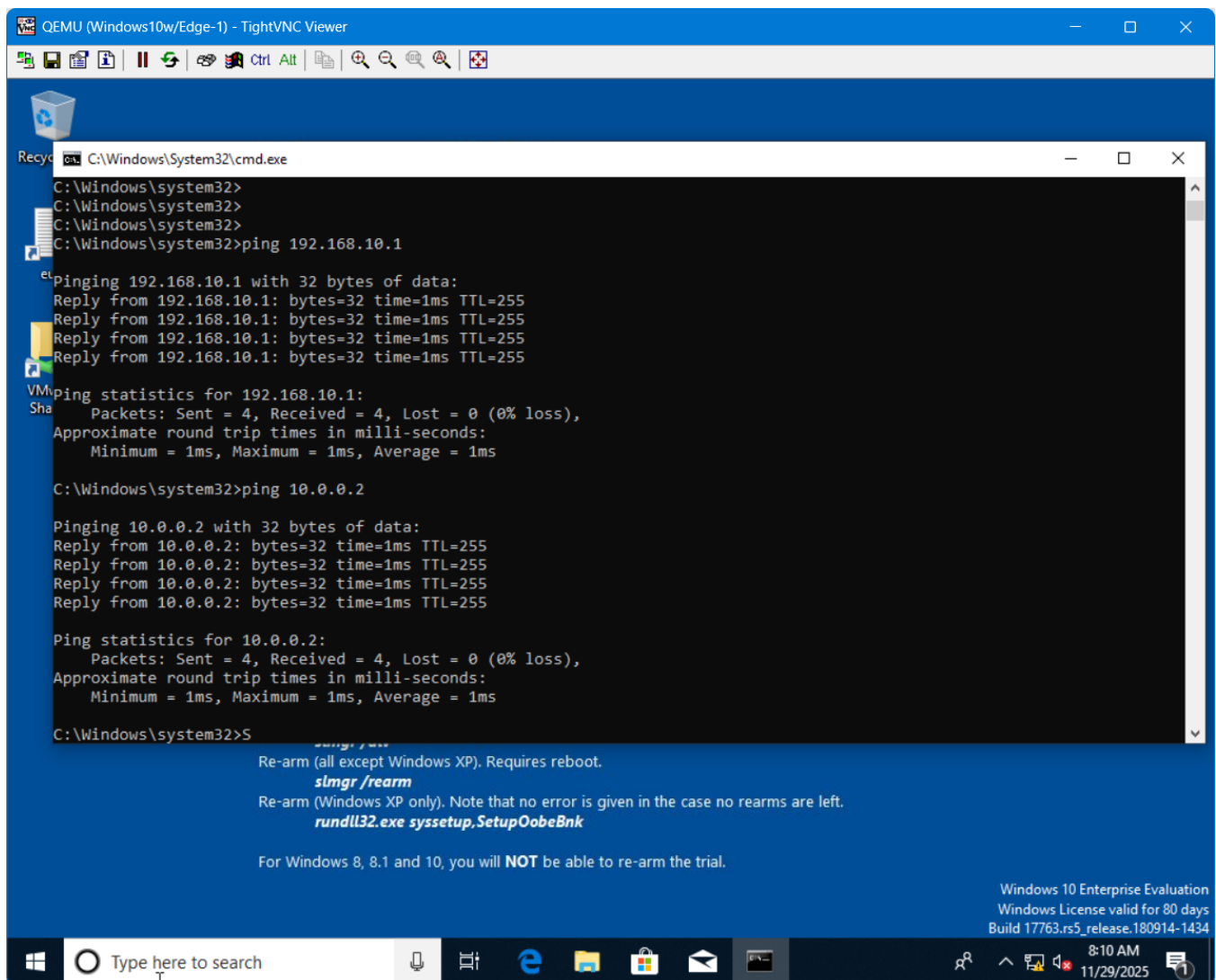
Basic connectivity testing is performed using ping:

Windows → FortiGate (192.168.10.1)

Kali → R1 (192.168.20.1)

Ubuntu → Windows and FortiGate

These tests have ensured that routing and firewall policies are also operating.



The screenshot shows a Windows 10 desktop environment running in a QEMU VM, viewed through a TightVNC Viewer. The desktop background is blue. A command prompt window is open, displaying the following text:

```
C:\Windows\system32>
C:\Windows\system32>
C:\Windows\system32>
C:\Windows\system32>ping 192.168.10.1
et
Pinging 192.168.10.1 with 32 bytes of data:
Reply from 192.168.10.1: bytes=32 time=1ms TTL=255
Reply from 192.168.10.1: bytes=32 time=1ms TTL=255
Reply from 192.168.10.1: bytes=32 time=1ms TTL=255
Reply from 192.168.10.1: bytes=32 time=1ms TTL=255
Ping statistics for 192.168.10.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\Windows\system32>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=1ms TTL=255
Reply from 10.0.0.2: bytes=32 time=1ms TTL=255
Reply from 10.0.0.2: bytes=32 time=1ms TTL=255
Reply from 10.0.0.2: bytes=32 time=1ms TTL=255
Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\Windows\system32>S
```

Below the command prompt, there is a blue banner with white text that reads:

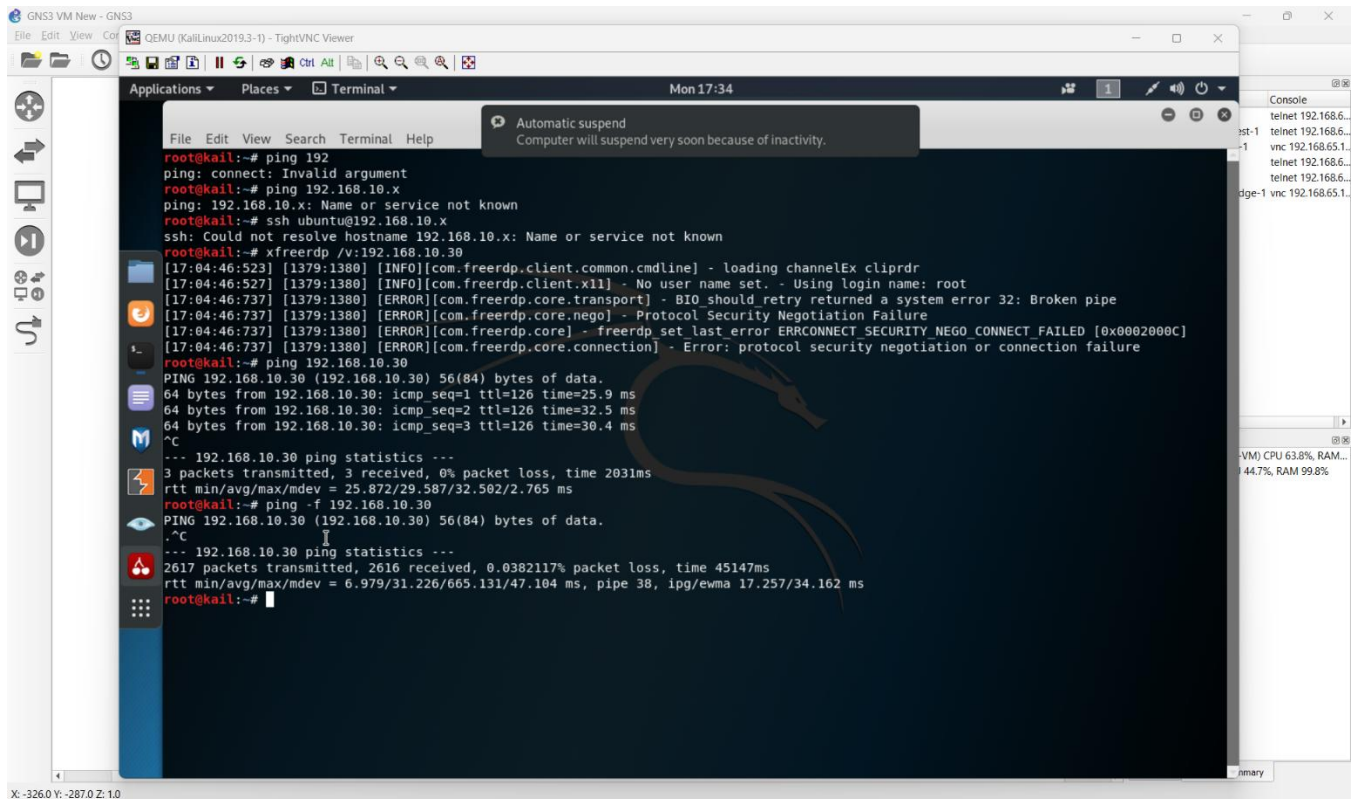
Re-arm (all except Windows XP). Requires reboot.
slmgr /rearm
Re-arm (Windows XP only). Note that no error is given in the case no rearms are left.
rundll32.exe syssetup.SetupOobeBnk
For Windows 8, 8.1 and 10, you will **NOT** be able to re-arm the trial.

In the bottom right corner, there is a Windows 10 Enterprise Evaluation watermark that reads:

Windows 10 Enterprise Evaluation
Windows License valid for 80 days
Build 17763.rs5_release.180914-1434

The taskbar at the bottom shows the Windows Start button, a search bar with the text "Type here to search", and several application icons including the Edge browser, File Explorer, and the Mail app. The system tray on the right shows the date and time as 8:10 AM on 11/29/2025.

Basic connectivity testing is performed using ping from the Windows to other Appliances in the GNS3



Basic connectivity testing is performed using ping from the Kail Linux to other Appliances in the GNS3

Security Controls and Attack Simulation

Firewall Policy and Logging:

The FortiGate firewall offers the primary security enforcement in this topology. With the logging on the LAN-to-LAN policy, all traffic between Kali and Windows is logged, protocol, ports, and timestamps. This plays a critical role in the analysis of normal and malicious behavior. However, we cannot get the firewall logs since the License of the appliances that have been used by us has expired.

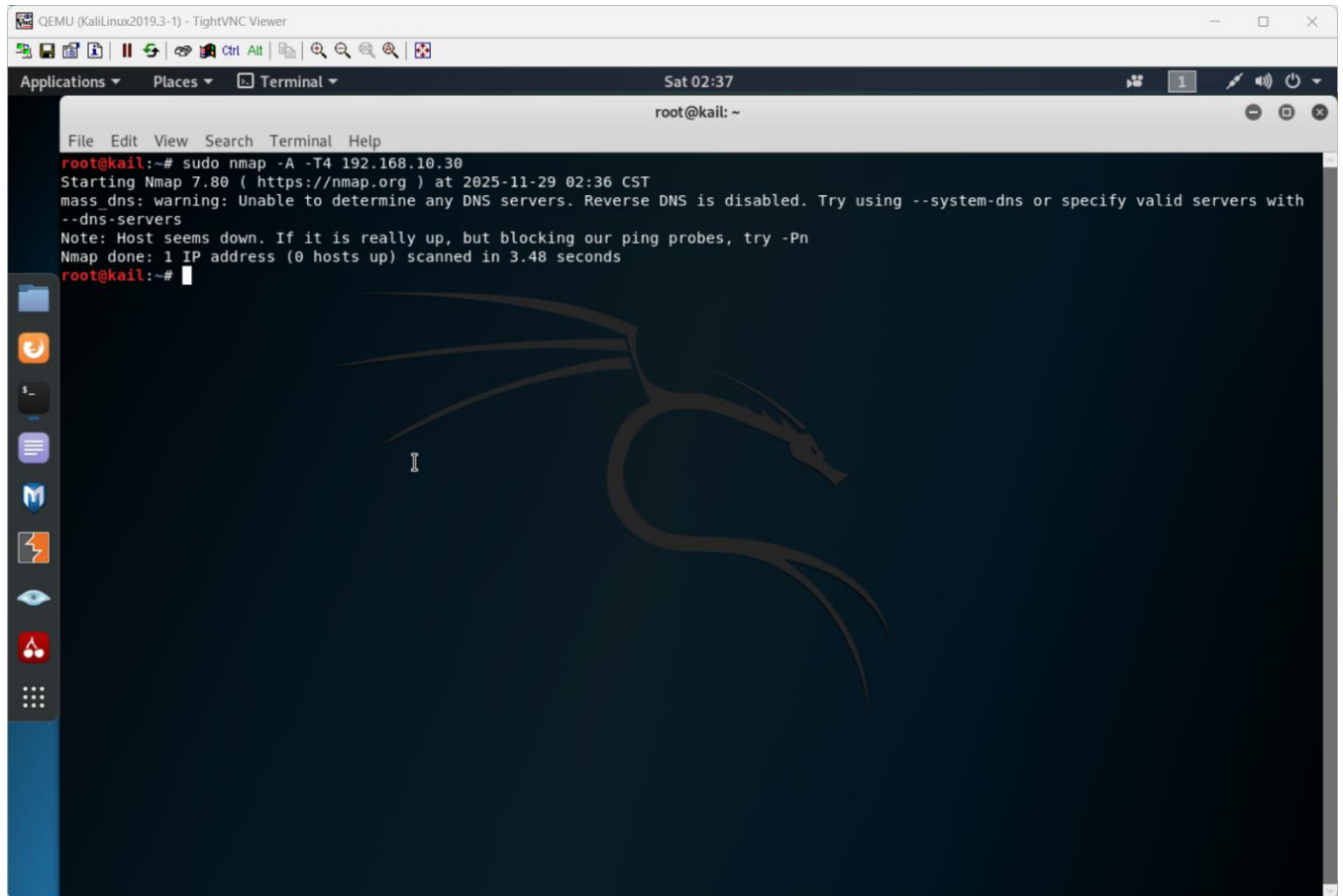
```
FortiGate7.0.9test-1 - PuTTY
config firewall policy
edit 1
set name "LAN-to-R1"
set uuid dc7b31ca-c030-51f0-4994-bed53b78a9ff
set srcintf "port1"
set dstintf "port2"
set action accept
set srcaddr "all"
set dstaddr "all"
set schedule "always"
set service "ALL"
next
edit 2
set name "R1-to-LAN"
set uuid cc97729a-c185-51f0-ed70-b9eaf536418b
set srcintf "port2"
set dstintf "port1"
set action accept
set srcaddr "all"
set dstaddr "all"
set schedule "always"
set service "ALL"
next
edit 10
set name "LAN20-to-LAN10"
set uuid 303ae642-c18b-51f0-6793-90ef03c74c26
set srcintf "port2"
set dstintf "port1"
set action accept
set srcaddr "all"
set dstaddr "all"
set schedule "always"
set service "ALL"
next
edit 11
set name "LAN10-to-LAN20"
set uuid 8c844fc2-c18d-51f0-182c-ad33d17113af
set srcintf "port1"
set dstintf "port2"
set action accept
set srcaddr "all"
set dstaddr "all"
set schedule "always"
set service "ALL"
next
edit 20
set name "LAN20-inside"
set uuid 1f573fa2-e36f-51f0-a3d2-37638c9f616a
set srcintf "port1"
set dstintf "port1"
set action accept
set srcaddr "all"
set dstaddr "all"
set schedule "always"
set service "ALL"
```

```
FortiGate7.0.9test-1 - PuTTY
FortiGate-VM64-KVM # show system interface
config system interface
edit "port1"
set vdom "root"
set ip 192.168.10.1 255.255.255.0
set allowaccess ping https ssh http
set type physical
set snmp-index 1
next
edit "port2"
set vdom "root"
set ip 10.0.0.2 255.255.255.0
set allowaccess ping ssh
set type physical
set snmp-index 2
next
edit "port3"
set vdom "root"
set type physical
set snmp-index 3
next
edit "port4"
set vdom "root"
set type physical
set snmp-index 4
next
edit "port5"
set vdom "root"
set type physical
set snmp-index 5
next
edit "port6"
set vdom "root"
set type physical
set snmp-index 6
next
edit "port7"
set vdom "root"
set type physical
set snmp-index 7
next
edit "port8"
set vdom "root"
set type physical
set snmp-index 8
next
edit "port9"
set vdom "root"
set type physical
set snmp-index 9
next
edit "port10"
set vdom "root"
set type physical
set snmp-index 10
```

Firewall Policy and PORT Connections

Port Scan Simulation with nmap

A nmap reconnaissance attack is simulated to Kali against the windows host: `nmap -A 192.168.10.30`
This scan tries to uncover the open ports, running services and the OS information of the windows machine. During the running of the scan, the firewall records several connection requests, made by the Kali IP, to different windows ports in the TCP connection.

A screenshot of a Kali Linux desktop environment viewed through a TIGHTVNC Viewer. The desktop has a dark blue background with a large, stylized dragon logo. A terminal window is open, showing the execution of an nmap scan. The terminal output indicates that the host is down, despite the scan being completed. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal prompt is 'root@kail: ~'. The output of the command 'sudo nmap -A -T4 192.168.10.30' is displayed in red and white text. The output includes the nmap version (7.80), the start time (2025-11-29 02:36 CST), a warning about DNS, a note about the host being down, and the scan completion message. The terminal window is titled 'root@kail: ~'. The desktop environment includes a sidebar with various application icons and a top bar with system status indicators.

```
QEMU (KaliLinux2019.3-1) - TightVNC Viewer
Applications Places Terminal Sat 02:37
root@kail: ~
File Edit View Search Terminal Help
root@kail:~# sudo nmap -A -T4 192.168.10.30
Starting Nmap 7.80 ( https://nmap.org ) at 2025-11-29 02:36 CST
mass dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with
--dns-servers
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.48 seconds
root@kail:~#
```

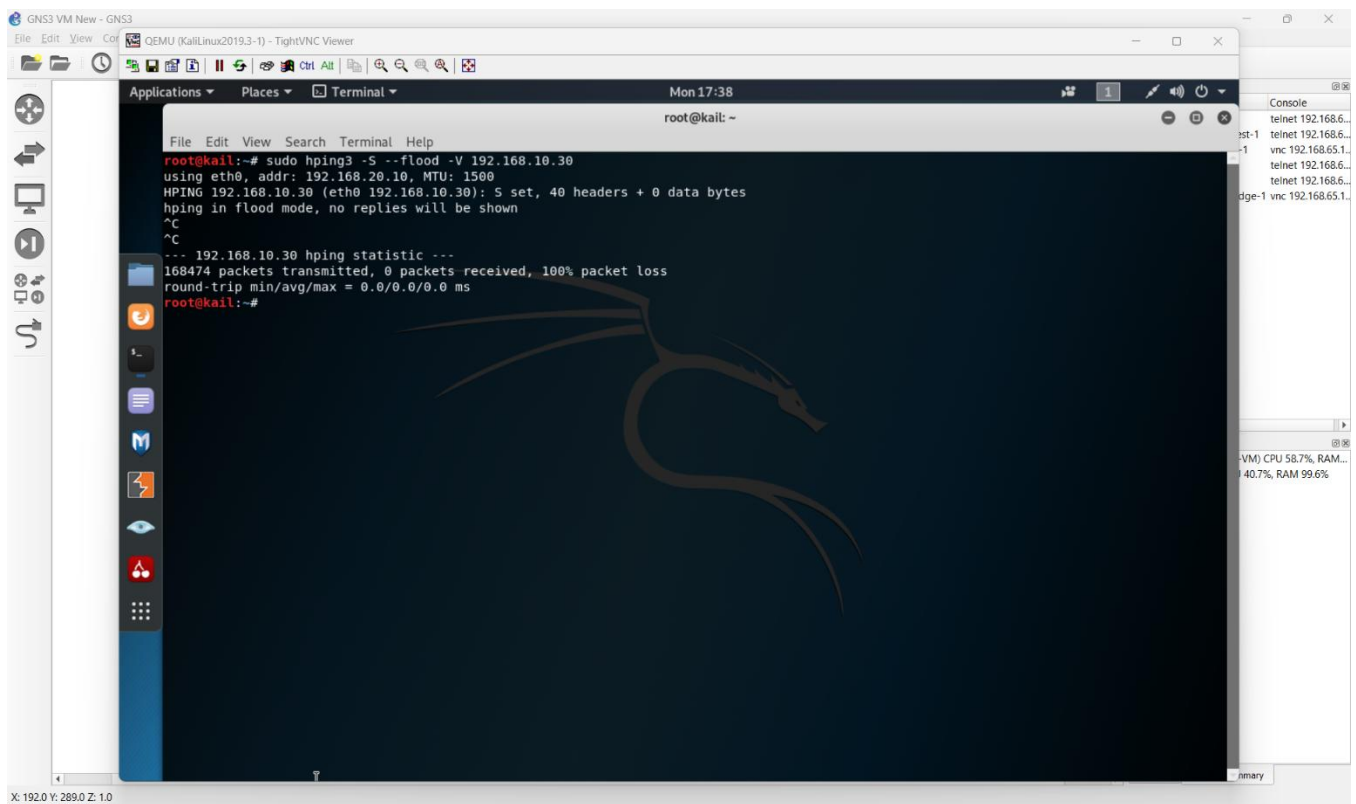
Port Scan Simulation with nmap in the Kail Linux in the GNS3

DoS-Style Traffic Simulation with hping3

Hping3 is used to do a safe DoS-type simulation, which sends a large number of SYN packets of Kali to Windows:

```
sudo hping3 -S --flood -V 192.168.10.30
```

This causes a SYN flood pattern to the isolated lab. Windows system could be delayed and the firewall logs indicates a peak in the connections made by the Kali host.



DoS-style SYN flood simulation from Kali to Windows using hping3.

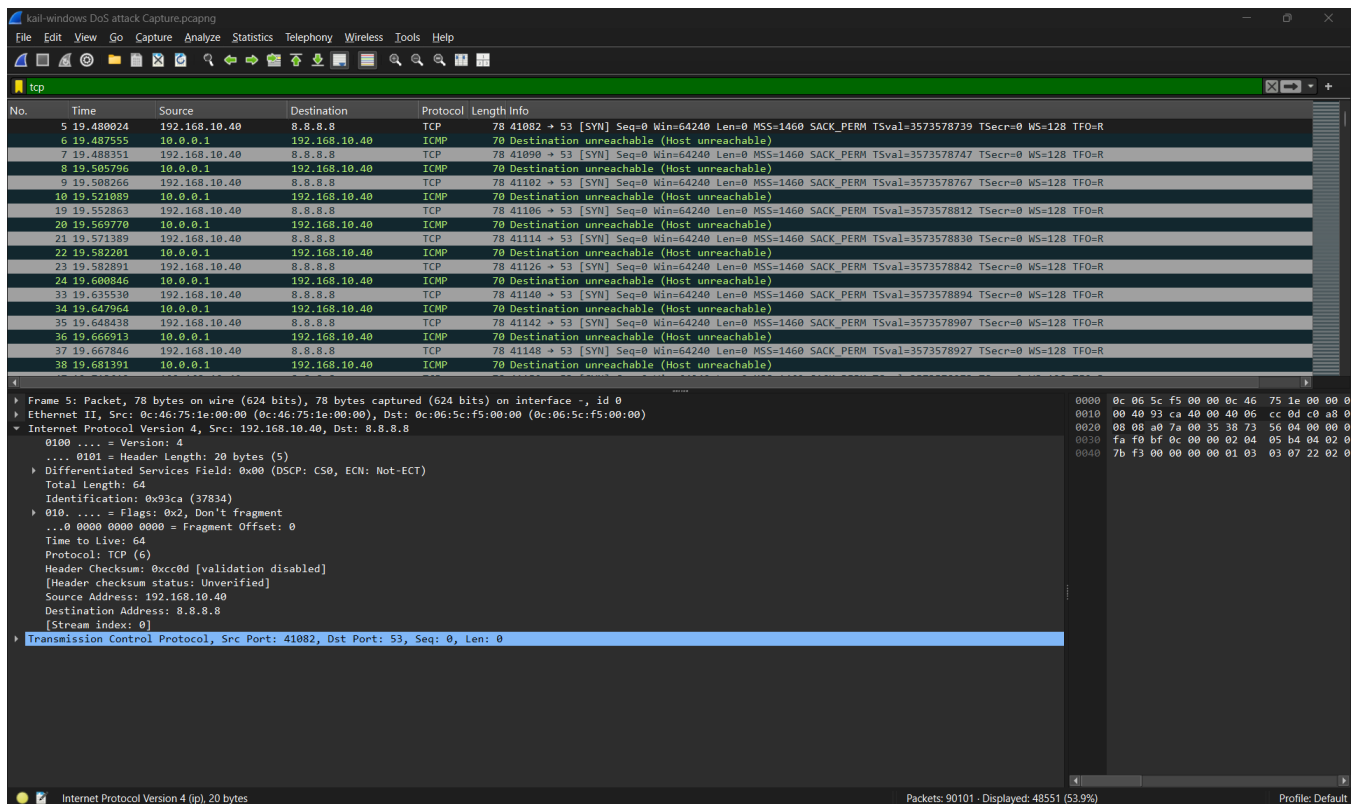
Packet Capture and Analysis

Packet capture is carried over within the GNS3 in tools like wireshark. Captured traffic includes:

- ICMP echo interacting with connectivity tests.
- TCP SYN packets in DoS style attack.

ARP broadcasting and responding inside of LAN.

The packet capture analysis is useful to visualize the way in which attacks spread across the network and ensures that all harmful traffic is redirected across the firewall to be inspected



Packet Capture and Analysis Using the Wireshark

CONCLUSION

This project shows that a segmented network with a central firewall is applicable in studying internal attacks within a safe, controlled environment. The design achieves this by ensuring that reconnaissance and DoS-type activities are not only visible but also restricted in scope by isolating Kali (attacker) and Windows and Ubuntu (victims) on different subnets and restricting all cross-LAN traffic to the FortiGate firewall.

The recorded topology, configuration procedures and security measures demonstrate that:

- Segmentation minimizes the number of targets to attack.
- Firewall policy gives necessary insight into bad traffic.
- Packet captures and logs are effective tools in reporting and comprehending attacks.

Overall, the offline GNS3-based lab provides a reusable platform for future experiments in network security, intrusion detection, and defensive configuration testing.