VLAN Segmentation and Policy Enforcement using FortiGate Firewall in VMware Workstation Pro

1. Introduction

Network segmentation is a vital technique in cybersecurity to isolate different departments, users, or services to enhance security, performance, and manageability. This project is a continuation of the initial FortiGate firewall implementation and focuses on VLAN segmentation using dedicated physical ports for VLANs.

This lab implementation utilizes a FortiGate VM with three network interfaces in VMware Workstation Pro. One port (port1) is used for WAN access through a bridged connection to the host laptop. The other two ports (port2 and port3) are dedicated to internal VLAN segments for two different networks: VLAN10 for Linux Mint VM and VLAN20 for Kali VM. Each VLAN is configured with its own IP subnet, DHCP scope, and firewall policies to enforce access control.

2. Objective

- To implement VLAN segmentation using dedicated interfaces on FortiGate Firewall.
- To configure DHCP servers for VLAN 10 and VLAN 20.
- To create firewall policies to manage traffic between VLANs and enable internet access.
- To simulate a secure virtual network with segmented departments.
- Scheduling Policy for VLAN10 from 10:00 to 18:00.
- To restrict VLAN20 access to a specific domain (cisco.com) using firewall policies.

3. Tools and Technologies Used

- **VMware Workstation Pro** – Virtualization platform for hosting FortiGate and VMs.
- **FortiGate Firewall VM** – Next-generation firewall used for VLAN configuration.
- **Linux Mint VM (VLAN10)** – Acts as a client in the Staff network.
- **Kali Linux VM (VLAN20)** – Acts as a client in the student network.
- **Web GUI** – Used for all FortiGate configurations.

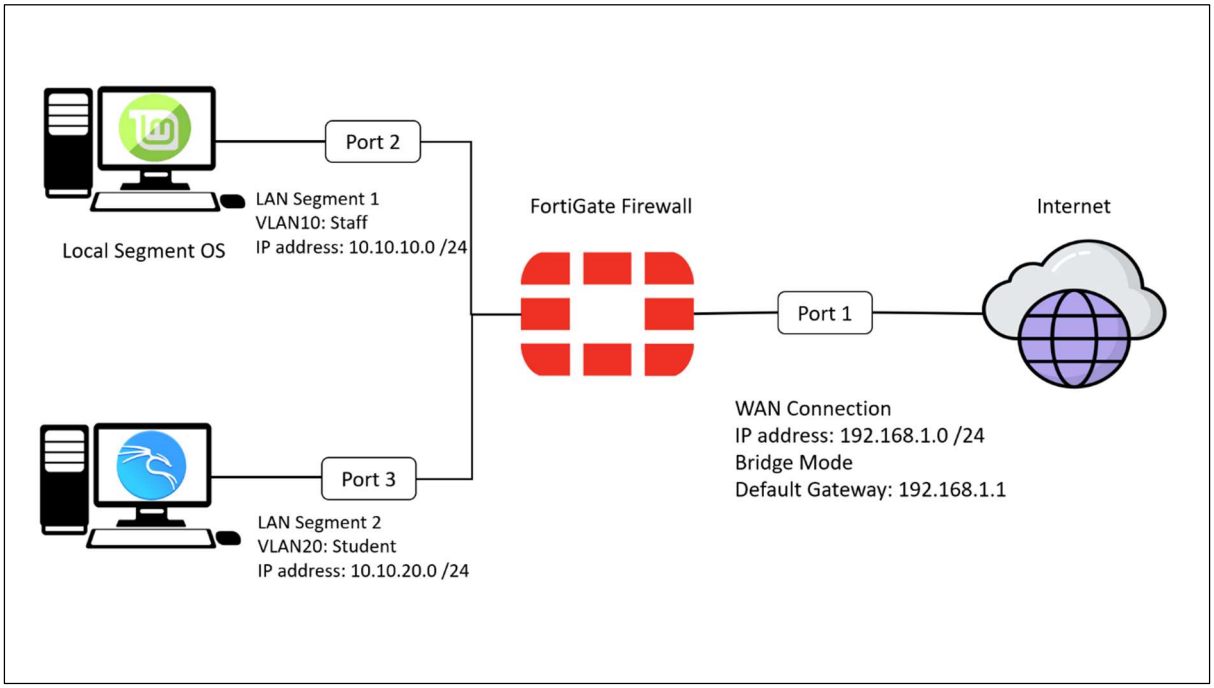
4. Network Design and Topology

The network topology designed in this lab environment emulates a small enterprise scenario using VLANs for segmentation. A FortiGate firewall VM acts as the central security gateway with three network interfaces. The first interface (port1) is configured for internet connectivity through a bridged connection with the host machine. The second interface (port2) is assigned

to VLAN10 and connected to a Linux Mint VM, representing the Staff network. The third interface (port3) is assigned to VLAN20 and linked to a Kali Linux VM, representing the student network. Each VLAN operates in an isolated subnet with its own DHCP configuration. This logical separation allows for distinct traffic control and policy enforcement. FortiGate enforces firewall policies between the VLANs and controls internet access, demonstrating a secure and manageable network architecture.

The FortiGate VM has three physical interfaces:

INTERFACE	SUBNET	CONNECTED DEVICE	VLAN ROLE	DHCP RANGE
PORT1	192.168.1.0/24	Host Laptop (WAN bridge)	WAN	Provided by host DHCP
PORT2	10.10.10.0/24	Linux Mint VM	VLAN10 (Staff)	10.10.10.10 - .100
PORT3	10.10.20.0/24	Kali Linux VM	VLAN20 (Student)	10.10.20.10 - .100



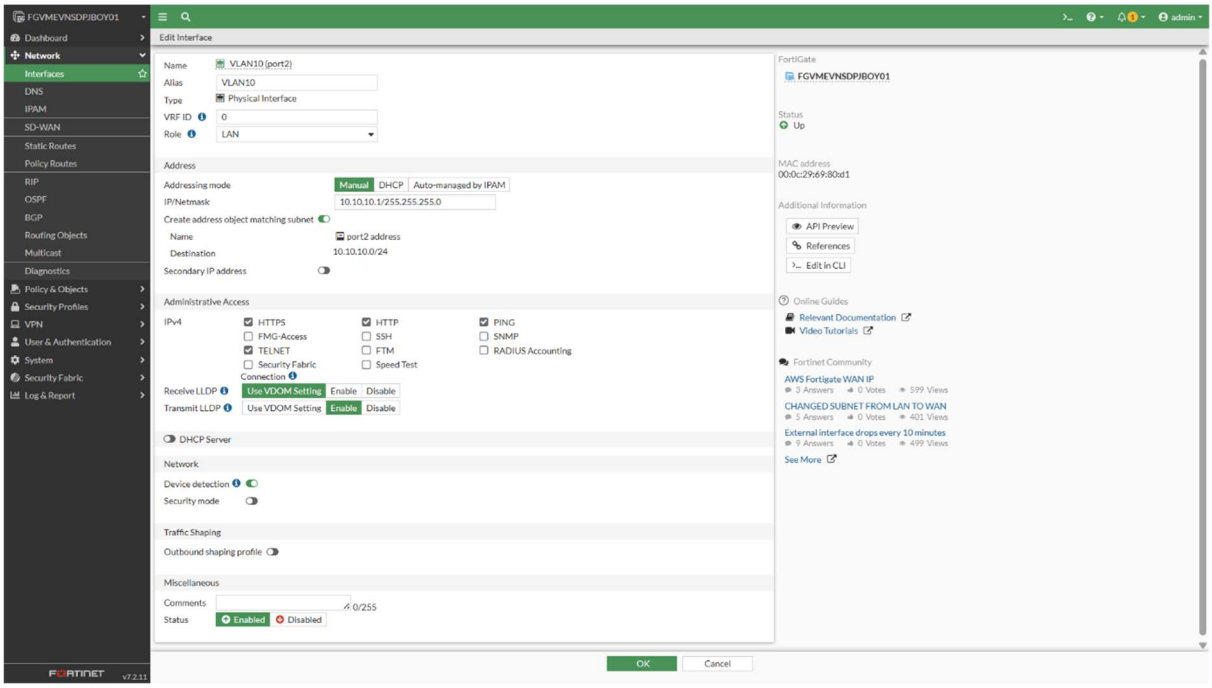
5. Implementation and Configuration

This section outlines the step-by-step implementation process using the FortiGate firewall's graphical user interface. Each phase of the configuration, from interface setup to policy enforcement, is handled through FortiGate's intuitive GUI, ensuring ease of use for beginners and professionals alike. The process starts by configuring the LAN interfaces (port2 and port3) with static IP addresses and enabling administrative access. Following this, DHCP servers are activated to automatically assign IPs to the client VMs. A static route is created to facilitate outbound internet traffic through port1. The client virtual machines

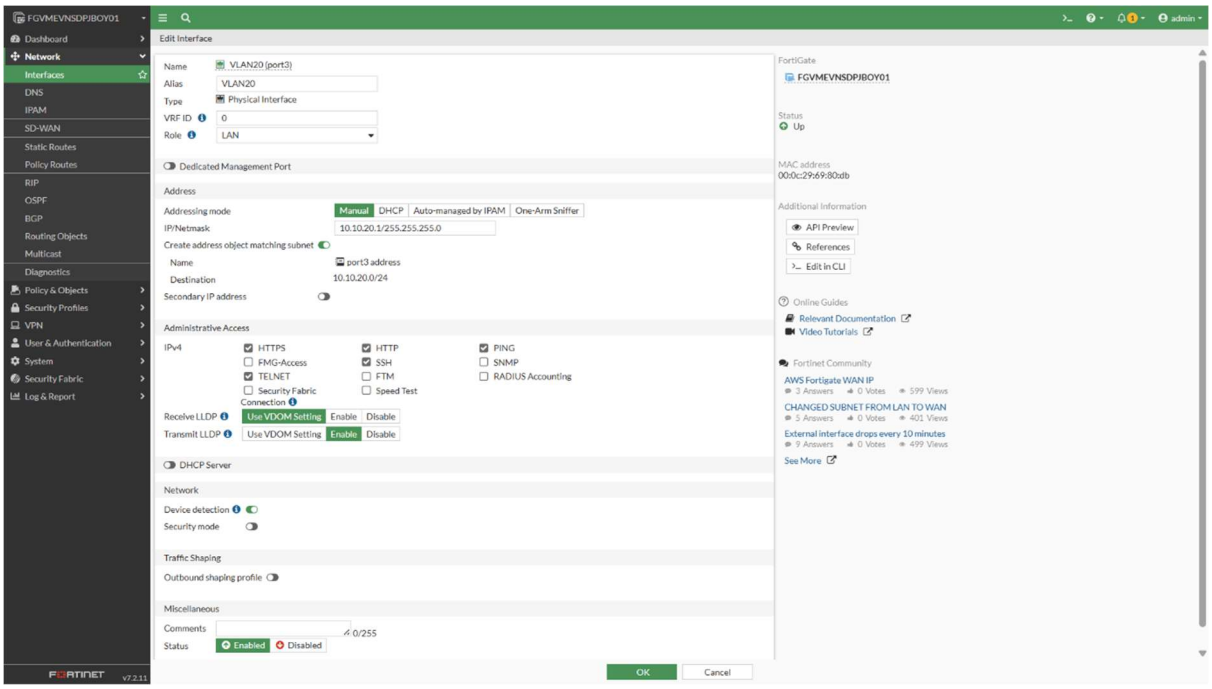
are connected to the appropriate VLAN interfaces via VMware's custom network configuration [LAN Segments: VLAN10 and VLAN20]. Specific firewall policies are then defined to allow, restrict, or deny traffic based on VLAN origin and destination. Finally, verification steps confirm successful segmentation, policy implementation, and controlled access to internal and external resources.

Step 1: Configure Interfaces (GUI)

- Open the FortiGate Web GUI via the IP assigned to port1 (through bridge mode).
- Navigate to **Network → Interfaces**.
- Click on port2: VLAN10
    - Assign IP: 10.10.10.1/24
    - Set Interface Role: LAN
    - Enable administrative access: **HTTP, HTTPS**, **PING**, **TELNET**
    - Click OK

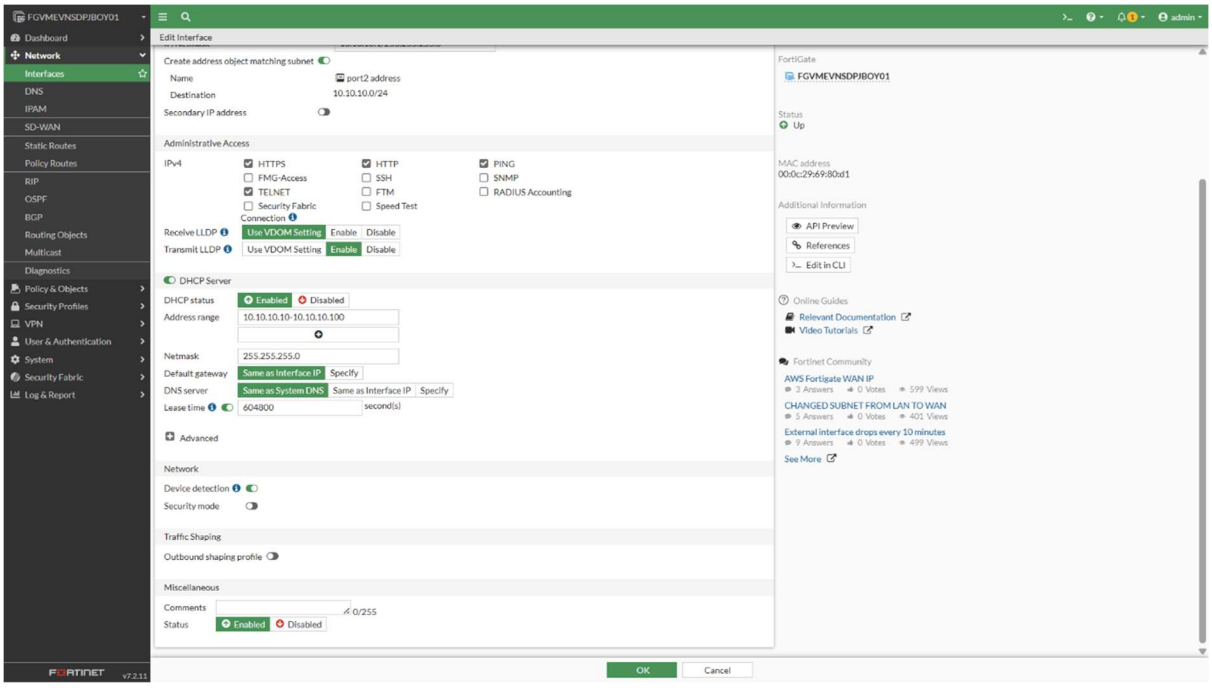


- Click on port3: VLAN20
    - Assign IP: 10.10.20.1/24
    - Set Interface Role: LAN
    - Enable administrative access: **HTTP, HTTPS**, **PING**, **TELNET**
    - Click OK

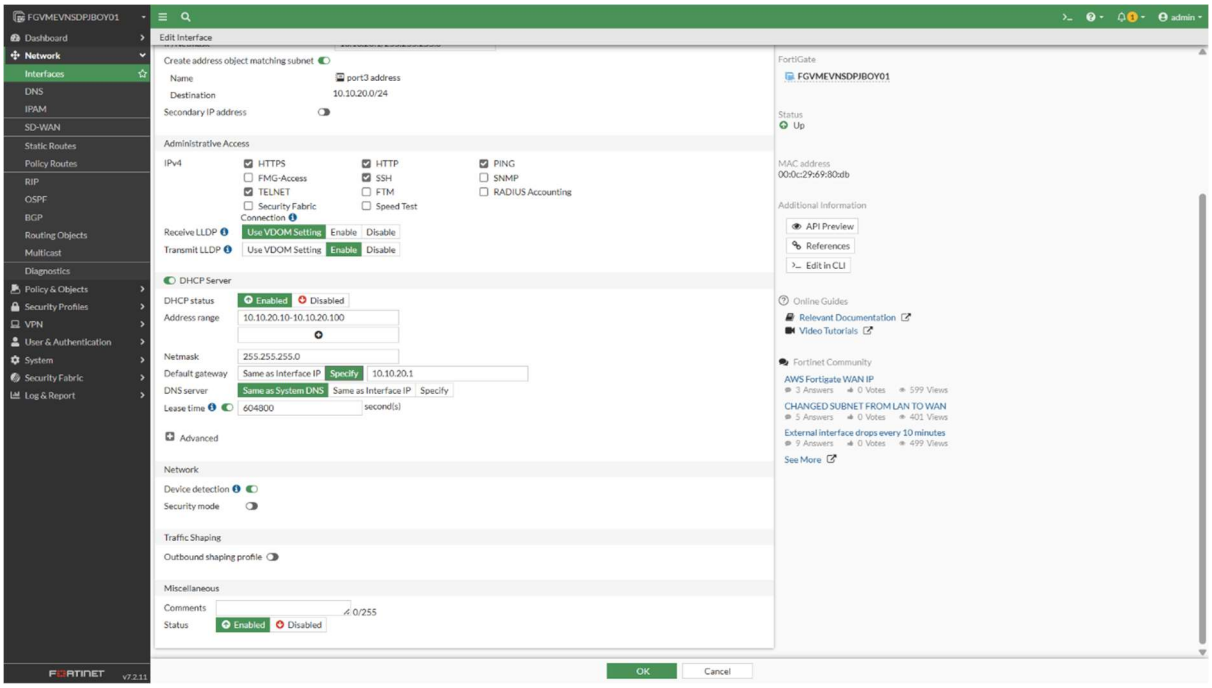


Step 2: Enable DHCP Servers (GUI)

- Go to **Network → Interfaces → Edit port2**
  - Enable DHCP Server
  - IP Range: 10.10.10.10 to 10.10.10.100
  - Default Gateway: Same as Interface IP
  - DNS Server: 8.8.8.8
  - Save changes

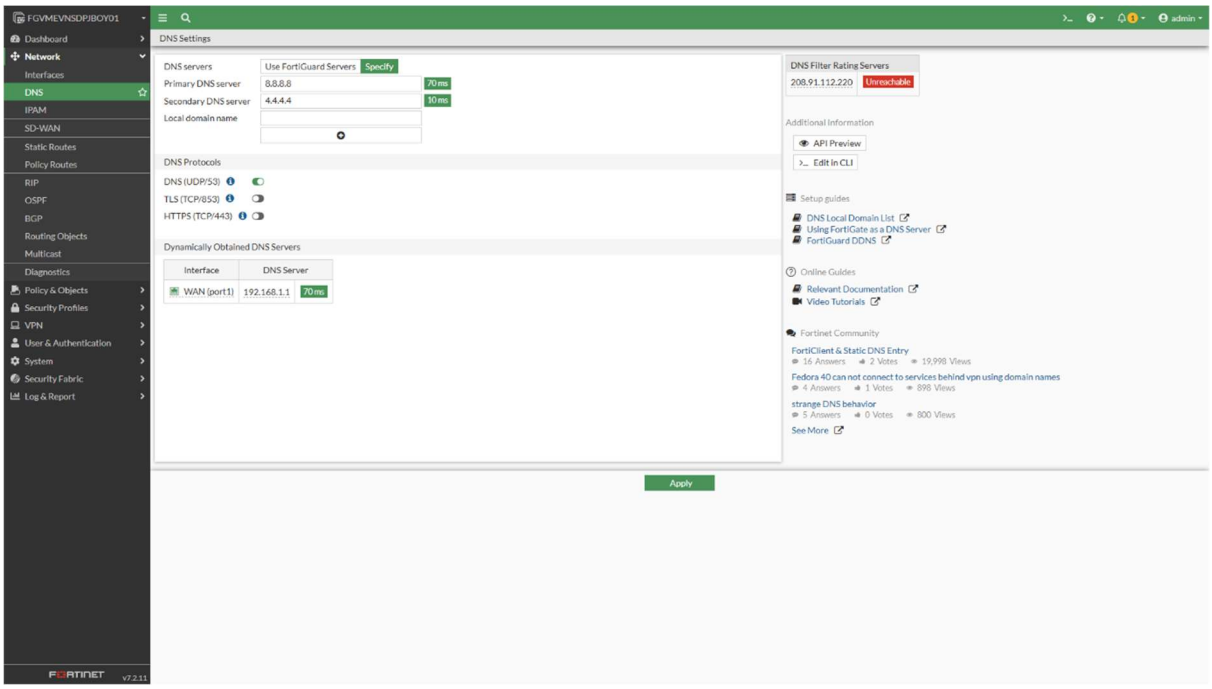


- Go to **Network → Interfaces → Edit port3**
    - Enable DHCP Server
    - IP Range: 10.10.20.10 to 10.10.20.100
    - Default Gateway: 10.10.20.1
    - DNS Server: 8.8.8.8
    - Save changes



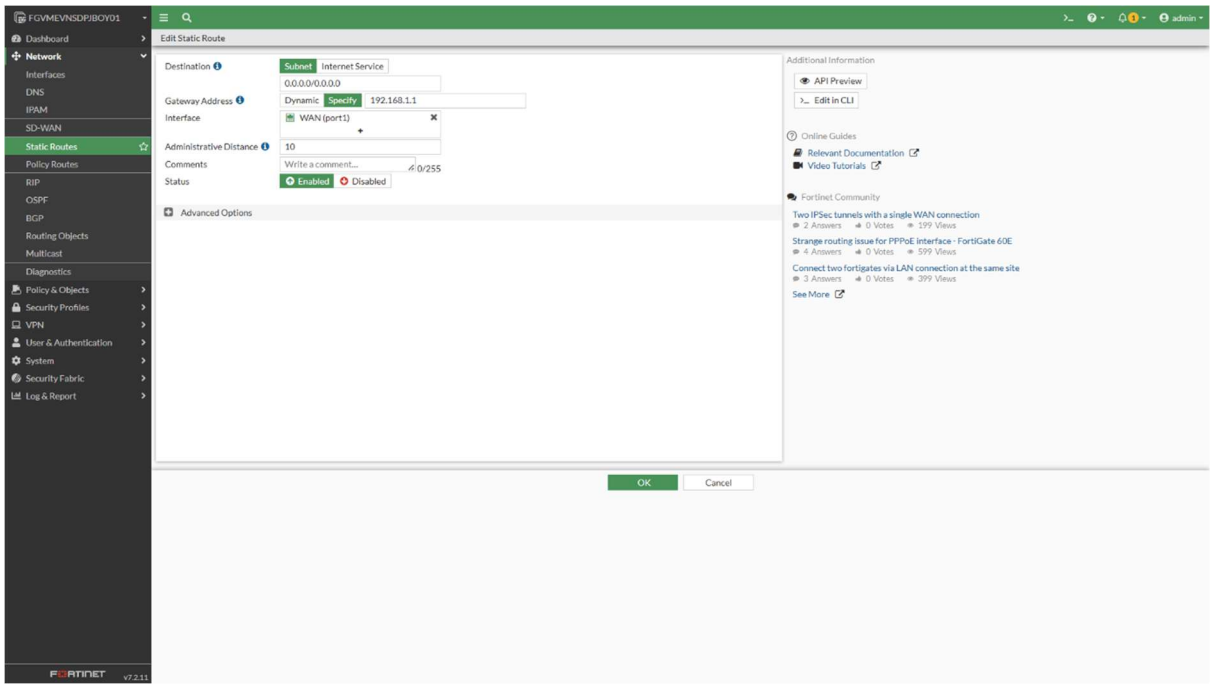
Step 3: Configure Static Route to Internet

DNS was configured through the FortiGate GUI under Network → DNS. The configuration included setting the Primary DNS server to 8.8.8.8 and the Secondary DNS server to 4.4.4.4. This ensures that the firewall and internal LAN clients can resolve domain names to IP addresses, allowing access to external websites and services. Additionally, DNS caching was enabled to improve resolution performance. This GUI-based setup supports services like FortiGuard, Web Filtering, and policy-based domain controls essential for network operations and security.



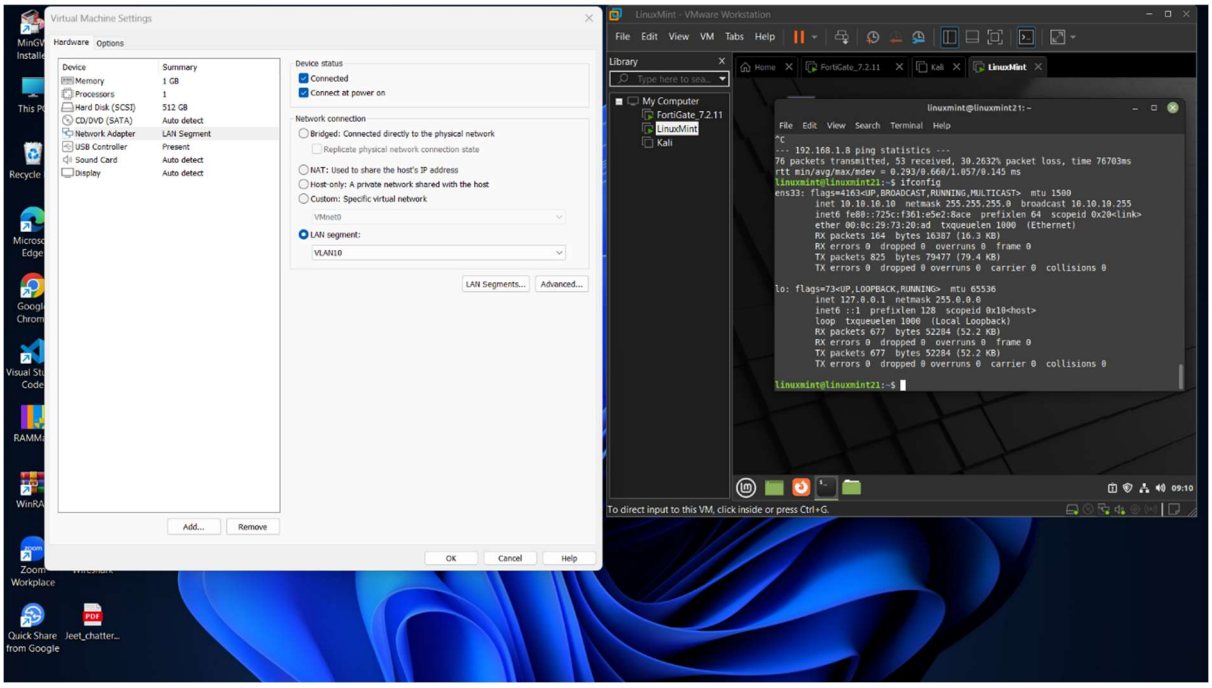
Step 4: Static Route Configuration

- Go to **Network → Static Routes → Create New**
    - Destination: 0.0.0.0/0
    - Gateway: 192.168.1.1 (Host router IP)
    - Interface: port1
    - Save and apply

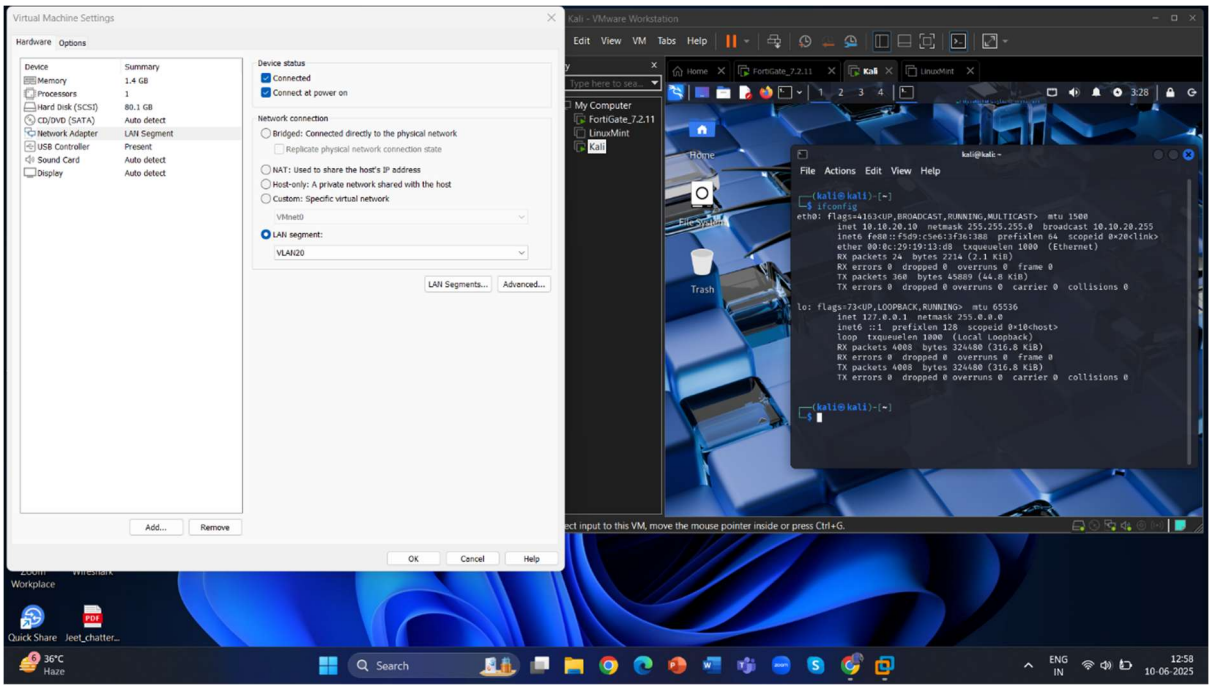


Step 5: Connection with Client VMs

- **Configure VMware Network Settings:**
  - Ensure FortiGate port2 is connected to **LAN segment VLAN10** and port3 to **LAN segment VLAN20**
  - In Linux Mint VM settings, set Network Adapter to **LAN segment VLAN10**.
  - In Kali Linux VM settings, set Network Adapter to **LAN segment VLAN20**.
- **Linux Mint VM (VLAN10):**
  1. Power on the Linux Mint VM.
  2. Open terminal and run ip a to verify IP assignment.
  3. Ensure the IP is in the 10.10.10.x range.
  4. Try ping 10.10.10.1 to verify connection to FortiGate.

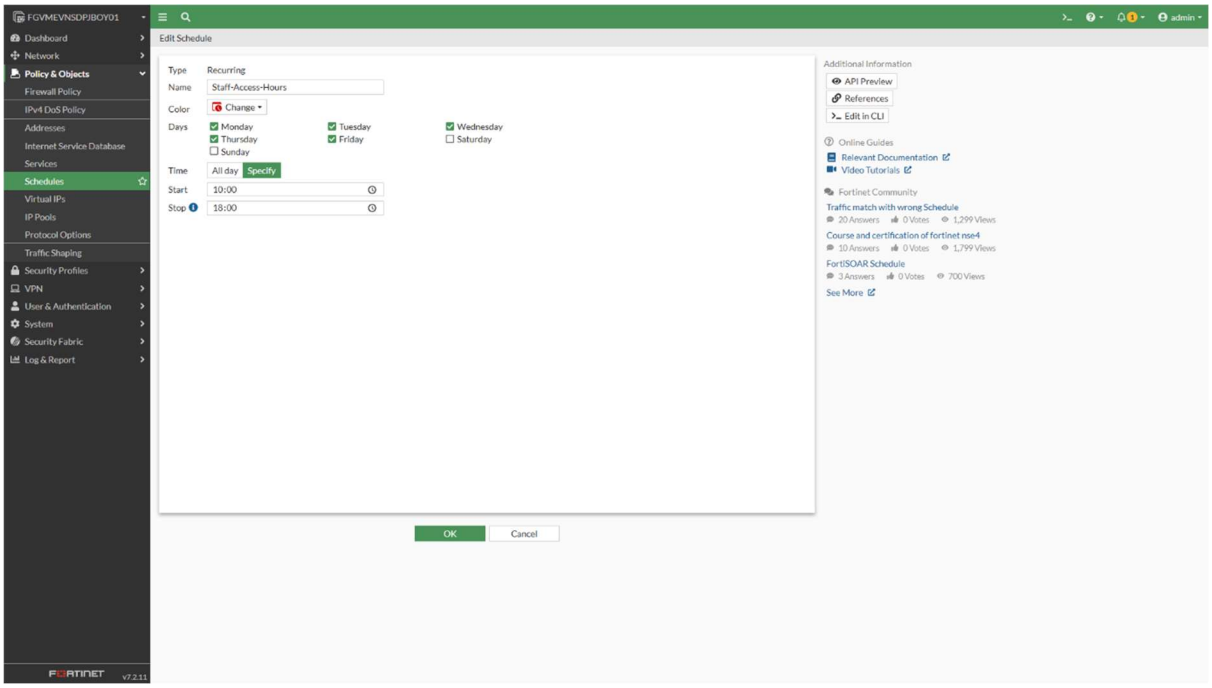


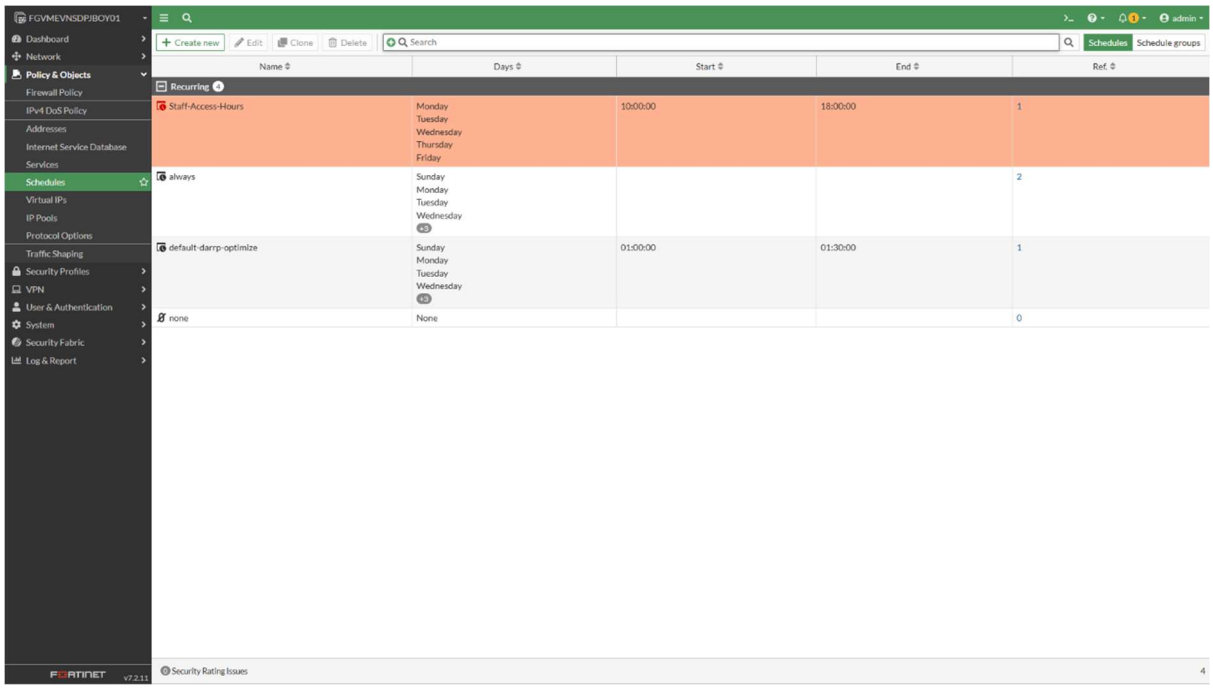
- **Kali Linux VM (VLAN20):**
  1. Power on the Kali VM.
  2. Use ip a to check IP address assigned by DHCP.
  3. Confirm IP is in the 10.10.20.x subnet.
  4. Ping 10.10.20.1 for gateway connectivity check.



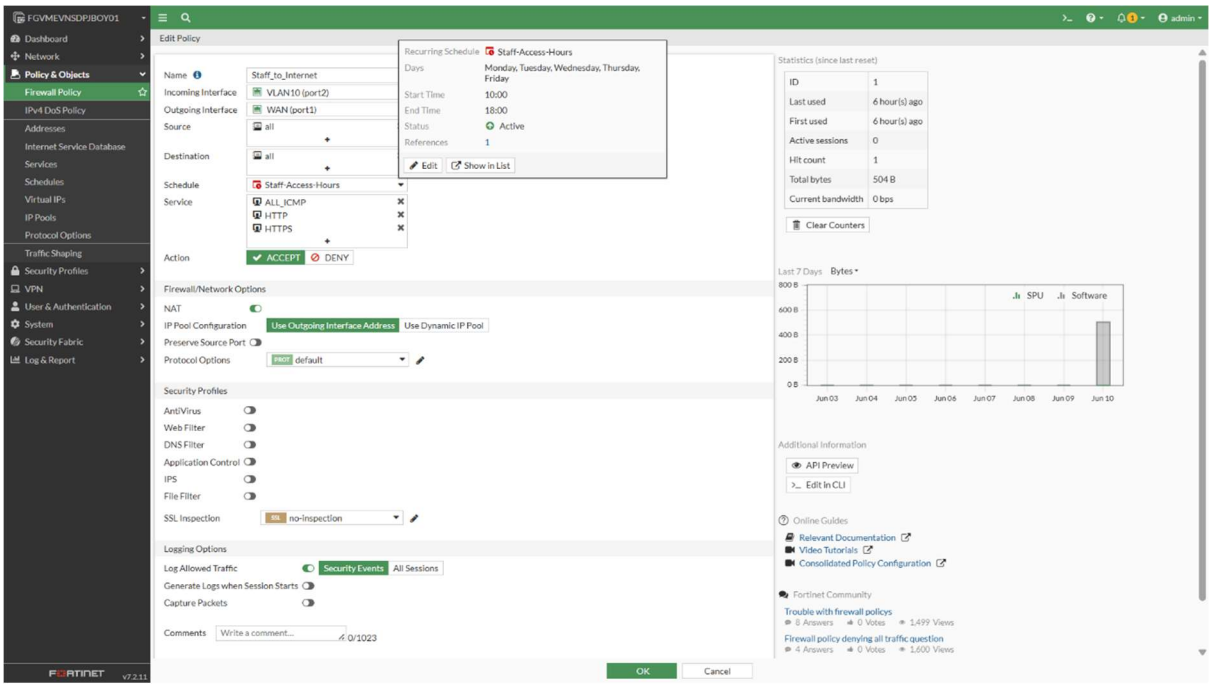
Step 6: Create Specific Firewall Policies

- Go to **Policy & Objects → Firewall Policy → Create New**
- **Policy 1 (Staff_to_Internet):**
- Incoming: port2, Outgoing: port1
- Source: all, Destination: all
- Service: ALL, Action: ACCEPT
- Schedule: Create a new schedule (10AM-6PM)
    - Go to **Objects → Schedules → Recurring → Create New**
    - Name: Staff-Access-Hours
    - Days: Monday to Friday
    - Time: 10:00 to 18:00
    - Save



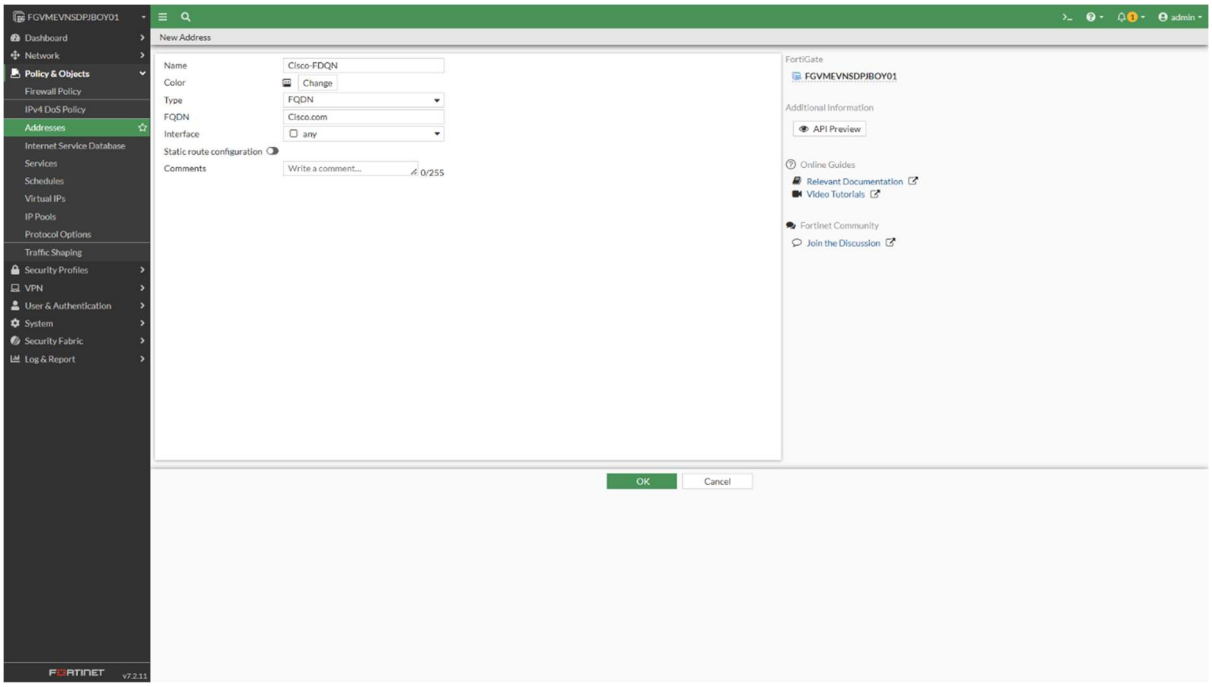


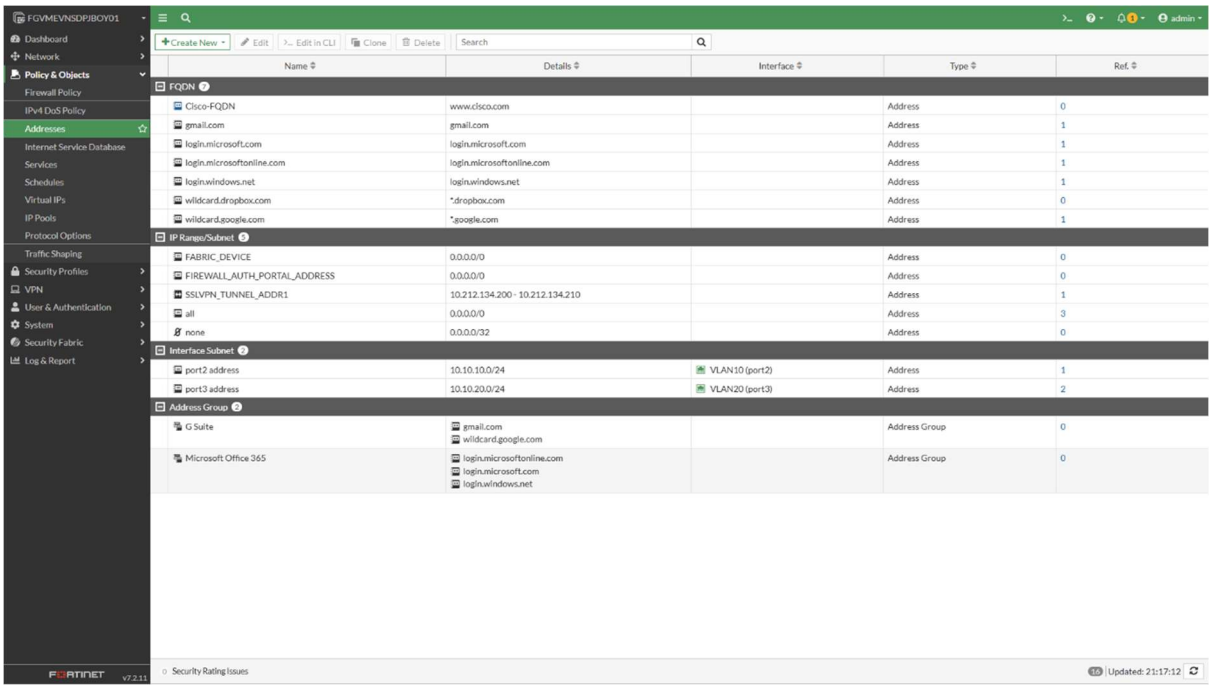
- Apply Staff-Access-Hours to the policy
- NAT: Enable



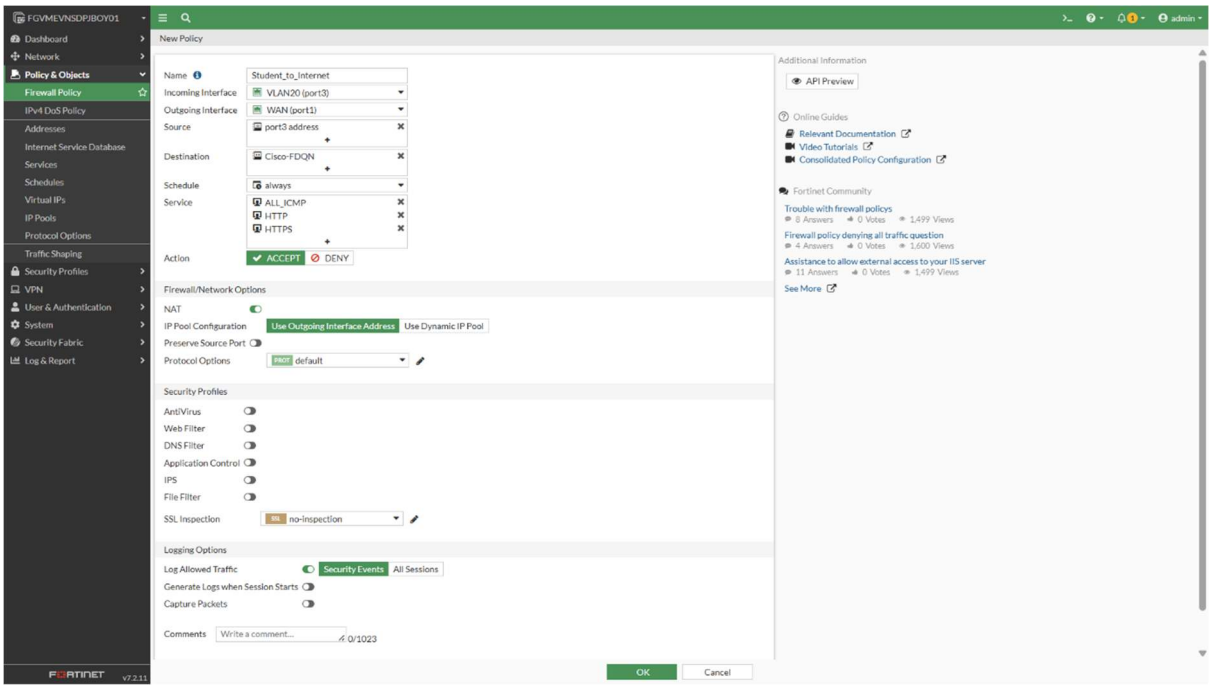
- **Policy 2 (VLAN20 to Internet - Restricted):**
  - Create FQDN address object:
    - Go to **Policy & Objects → Addresses → Create New → FQDN**
    - Name: Cisco-FQDN

- FQDN: www.cisco.com
- Interface: Any

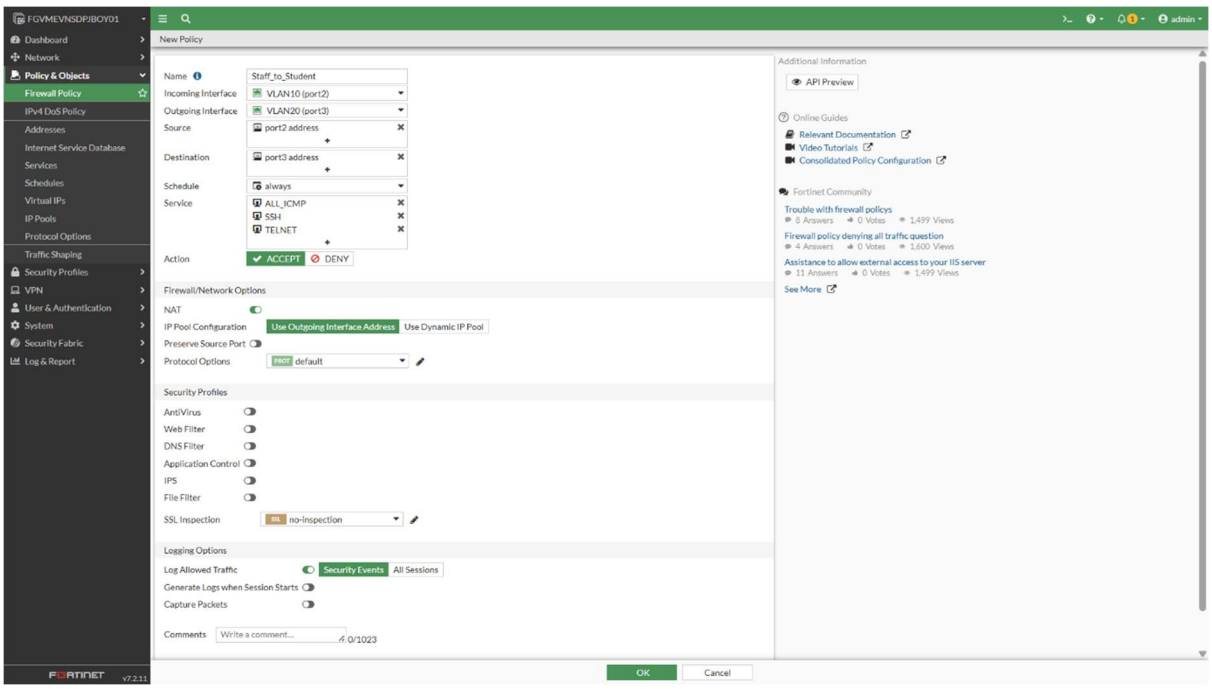




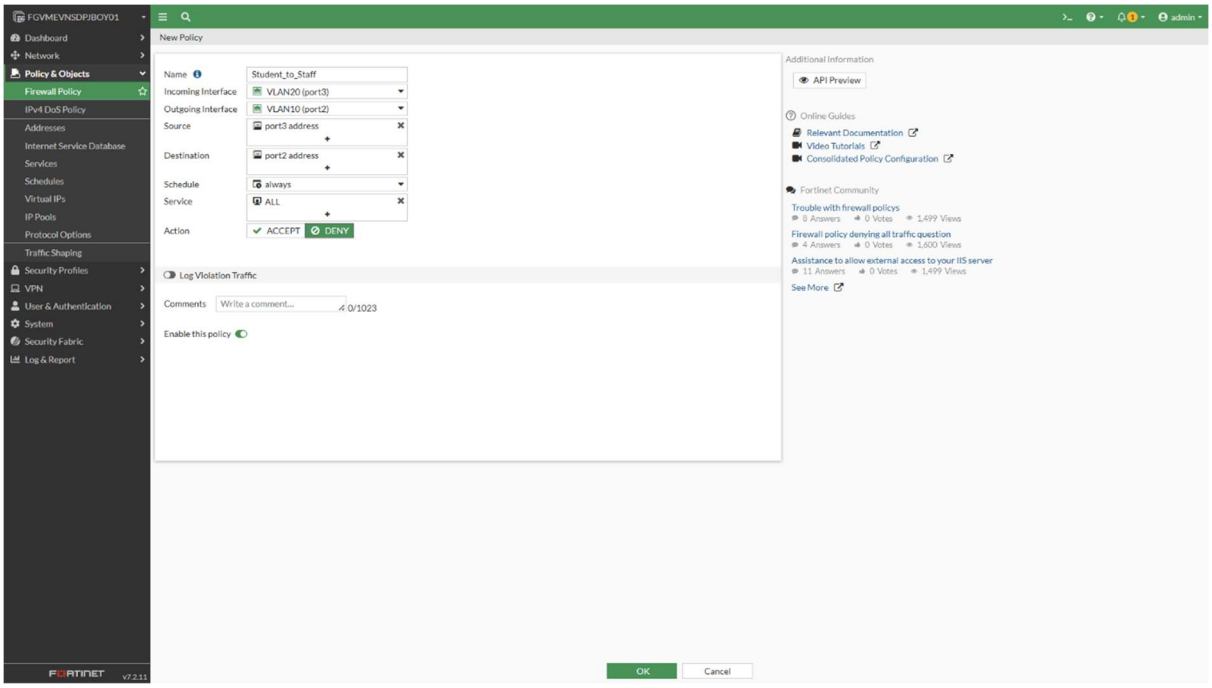
- Incoming: port3, Outgoing: port1
- Source: VLAN20 address range (10.10.20.0/24)
- Destination: Cisco-FQDN
- Service: ALL_ICMP, HTTP, HTTPS, Action: ACCEPT
- NAT: Enable



- **Policy 3 (VLAN10 to VLAN20):**
  - Incoming: port2, Outgoing: port3
  - Source: VLAN10 address range
  - Destination: VLAN20 address range
  - Service: ICMP, SSH, or custom ports as needed
  - Action: ACCEPT (allow selected services for monitoring or shared access)
  - NAT: Disable



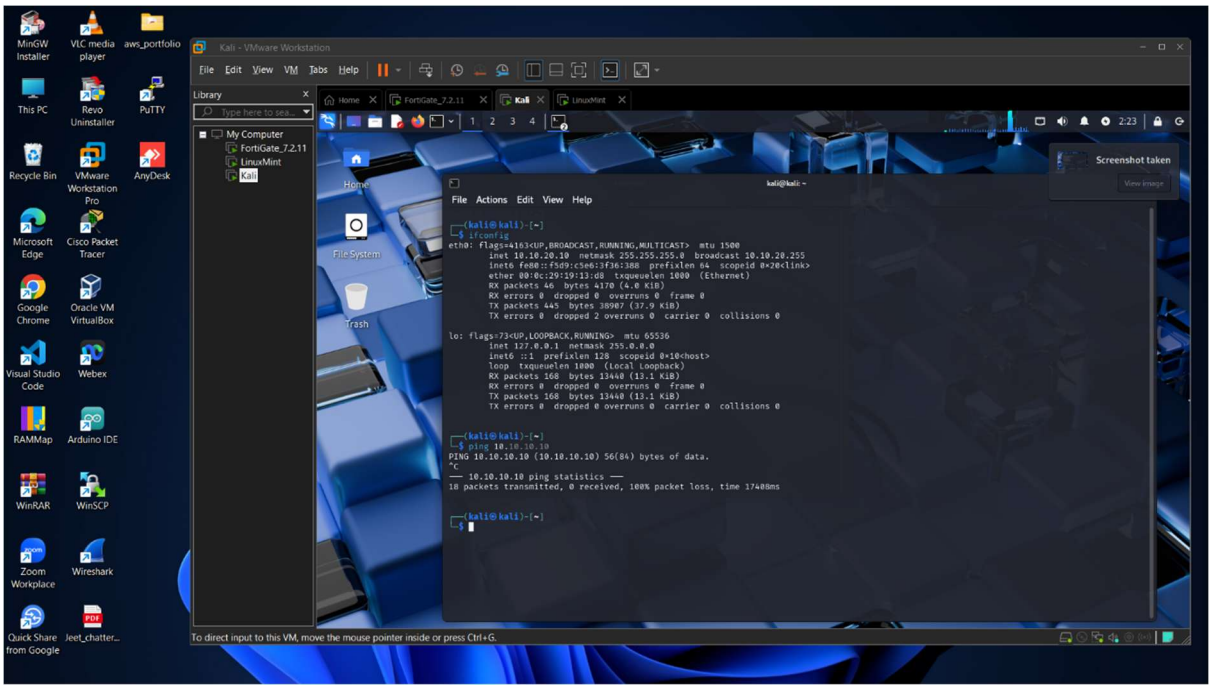
- **Policy 4 (VLAN20 to VLAN10):**
    - Incoming: port3, Outgoing: port2
    - Source: VLAN20 address range
    - Destination: VLAN10 address range
    - Service: ALL
    - Action: DENY (to restrict student VLAN from accessing staff VLAN)

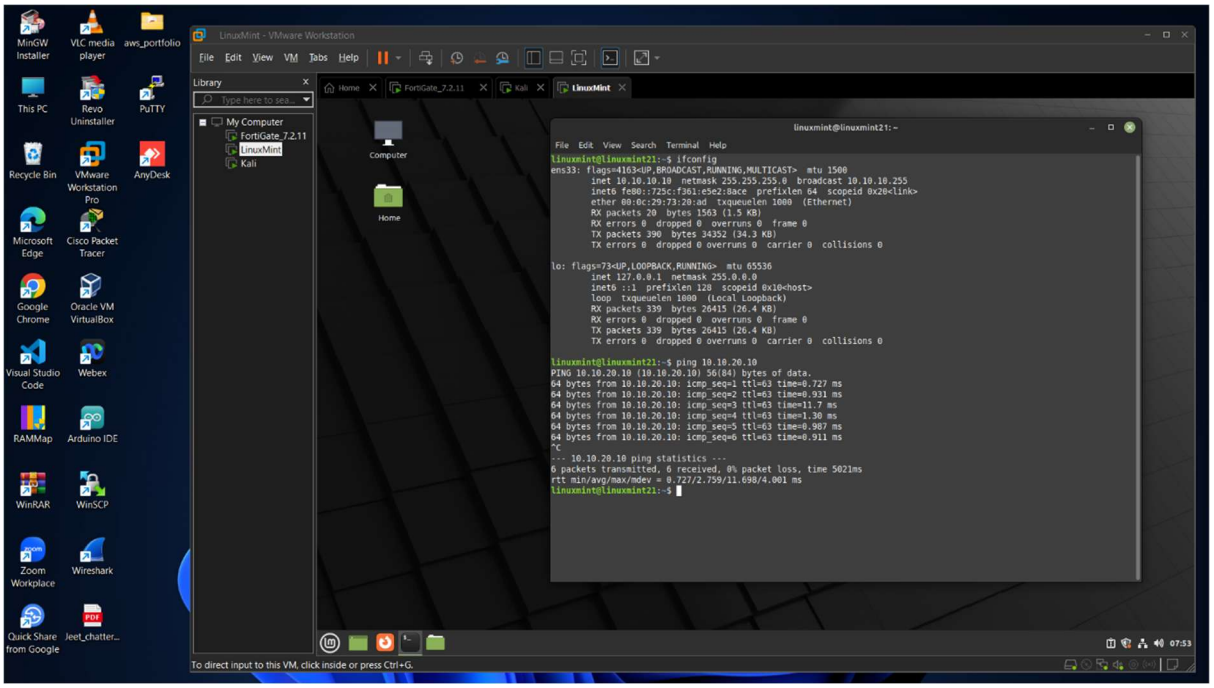


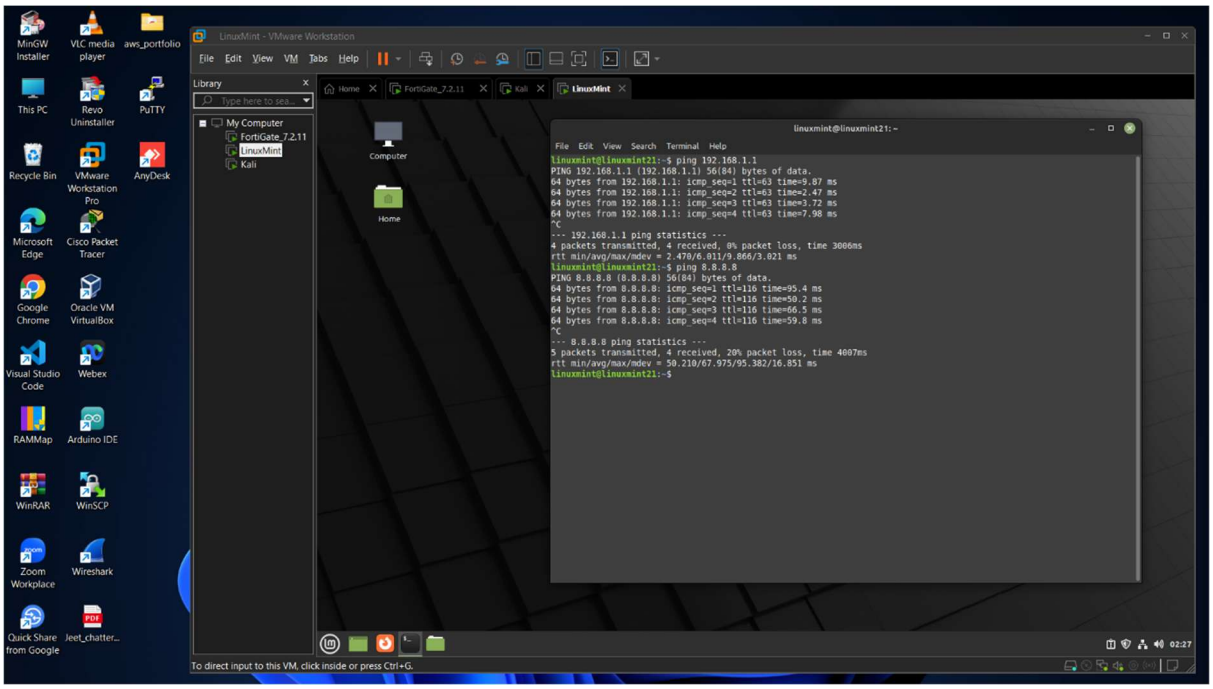
6. Results and Verification

- Verify the Policy with Ping
    - From Linux Mint VM (VLAN10), ping `8.8.8.8` to confirm internet access.
    - From Kali VM (VLAN20), ping `www.cisco.com` and verify success, try other websites to test restriction.
    - Test inter-VLAN communication using ping from Mint to Kali and vice versa Confirm that denied policies are blocking traffic as expected.

- DHCP servers assigned valid IP addresses to both Linux Mint and Kali VMs.
- Internet access confirmed on both VLANs with appropriate restrictions.
- Policy-based traffic flow verified using ping tests and domain-based access.
- Deny rules successfully blocked unauthorized inter-VLAN access.


Kali - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
FortiGate_7.2.11
LinuxMint
Kali
Home
File System
Trash
Screenshot taken
File Actions Edit View Help
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.10.20.10 netmask 255.255.255.0 broadcast 10.10.20.255
inet6 fe80::f5d9:c5e6:3f36:388 prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:19:13:d8 txqueuelen 1000 (Ethernet)
RX packets 46 bytes 4170 (4.0 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 445 bytes 38907 (37.9 KiB)
TX errors 0 dropped 2 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 168 bytes 13440 (13.1 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 168 bytes 13440 (13.1 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
(kali@kali)-[~]
$ ping 10.10.10.10
PING 10.10.10.10 (10.10.10.10) 56(84) bytes of data.
^C
— 10.10.10.10 ping statistics —
18 packets transmitted, 0 received, 100% packet loss, time 17408ms
(kali@kali)-[~]
$
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.


LinuxMint - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
FortiGate_7.2.11
LinuxMint
Kali
Computer
Home
linuxmint@linuxmint21: ~
File Edit View Search Terminal Help
linuxmint@linuxmint21:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.10.10.10 netmask 255.255.255.0 broadcast 10.10.10.255
inet6 fe80::725c:f361:e5e2:8ace prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:73:20:ad txqueuelen 1000 (Ethernet)
RX packets 20 bytes 1563 (1.5 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 390 bytes 34352 (34.3 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 339 bytes 26415 (26.4 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 339 bytes 26415 (26.4 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
linuxmint@linuxmint21:~$ ping 10.10.20.10
PING 10.10.20.10 (10.10.20.10) 56(84) bytes of data.
64 bytes from 10.10.20.10: icmp_seq=1 ttl=63 time=0.727 ms
64 bytes from 10.10.20.10: icmp_seq=2 ttl=63 time=0.931 ms
64 bytes from 10.10.20.10: icmp_seq=3 ttl=63 time=11.7 ms
64 bytes from 10.10.20.10: icmp_seq=4 ttl=63 time=1.30 ms
64 bytes from 10.10.20.10: icmp_seq=5 ttl=63 time=0.987 ms
64 bytes from 10.10.20.10: icmp_seq=6 ttl=63 time=0.911 ms
^C
--- 10.10.20.10 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5021ms
rtt min/avg/max/mdev = 0.727/2.759/11.698/4.001 ms
linuxmint@linuxmint21:~$
To direct input to this VM, click inside or press Ctrl+G.



7. Conclusion

This lab successfully demonstrates VLAN segmentation and policy enforcement using a FortiGate firewall in a VMware Workstation environment. By leveraging three physical interfaces, we effectively simulated real-world departmental isolation and access control. GUI-based configuration made implementation user-friendly and efficient. This setup serves as a strong foundation for advanced firewall practices, including deep inspection, logging, and security automation. Domain-based policy enforcement further enhances the precision of network security.