Secure Network Design and Implementation

Son Dang, Prakiti Dhungel, Ashwini Gondhi

University of North Texas

CSCE 5585 Advanced Network Security

Dr. Ali Zarafshani

11/25/2025

Abstract

This project aims to design a secure network environment using GNS3 and Fortigate firewalls. It involves network topology with complete segmentation for the DMZ and LAN network, VPN services for remote connections, and IDS/IPS integrated functions utilizing Wireshark and Suricata software as well as the integrated IDS/IPS capabilities on FortiGate firewalls. The research also shows how Wireshark and Suricata can work together to protect a network by simulating a SYN flooding attacks. Additionally, by conducting a vulnerability assessment using Nexus, we can identify potential weaknesses and propose mitigation strategies to enhance the security defense of the system.

1. Introduction

This project focuses on designing and implementing a secure network environment using GNS3 as our simulation software. We will use FortiGate firewall version 7.0.9 to demonstrate the functionality of VPN, IDS, and IPS systems. We have configured the network to demonstrate secure segmentation between the Demilitarized Zone (DMZ) and the private local area network (LAN). Additionally, we implemented IDS/IPS on our virtual machine, showcasing threat detection and prevention using Wireshark and Suricata at the host level. We will also conduct a security assessment/penetration testing to identify any potential vulnerabilities/weaknesses within our network, and propose mitigation strategies to enhance the overall network security of our network using the network vulnerability scanning tool, Nessus.

2. Network Environment (GNS3)

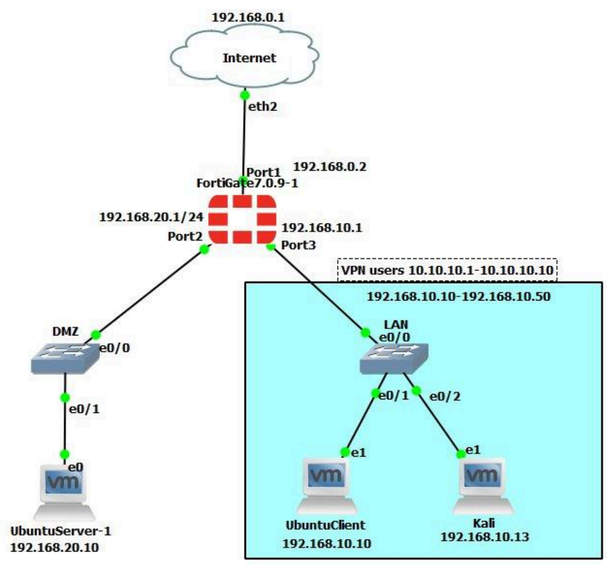
- FortiGate firewall (v7.0.9): provides routing, firewall policies, VPN access

- DMZ: Public-facing network, a separate network zone from the LAN to provide isolation

- LAN switch: Private network, connecting internal devices for communications

- Ubuntu server, Ubuntu/Kali clients: Simulating network attacks

- Suricata & Wireshark: IDS system to capture and detect, and prevent using IPS



3. FortiGate Firewall Policies:

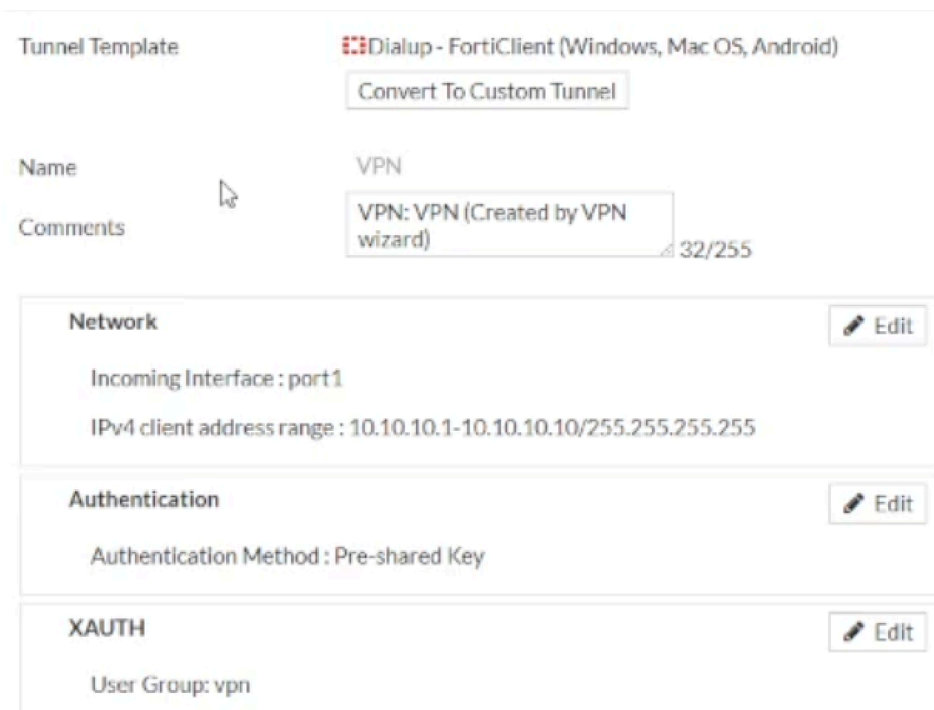
For our network firewall policies, we are allowing all traffic except from the DMZ to the LAN, since DMZ is a public-facing server, we do not want it to interact with our private local network. Firewall policy rules:

- Allowed: LAN→DMZ, LAN→Internet, DMZ→Internet, VPN Remote→:LAN
- Blocked: DMZ→LAN: Public-facing server cannot have access to the internal network

Name	Source	Destination	Schedule	Service	Action	NAT	Security Profiles	Log	Bytes
DMZ (port2) → LAN (port3) 1									
Block DMZ to LAN	all	all	always	ALL	DENY			Disabled	0 B
DMZ (port2) → port1 1									
DMZ to Internet	all	all	always	ALL	ACCEPT	Enabled	IPS all_default SSL certificate-inspection	UTM	0 B
LAN (port3) → DMZ (port2) 1									
LAN to DMZ	all	all	always	ALL	ACCEPT	Enabled	IPS all_default SSL certificate-inspection	UTM	1.42 MB
LAN (port3) → port1 1									
LAN to Internet	all	all	always	ALL	ACCEPT	Enabled	IPS default SSL certificate-inspection	UTM	612.25 kB
VPN → LAN (port3) 1									
vpn_VPN_remote_0	VPN_range	lan	always	ALL	ACCEPT	Enabled	IPS high_security SSL certificate-inspection	UTM	0 B
Implicit 1									

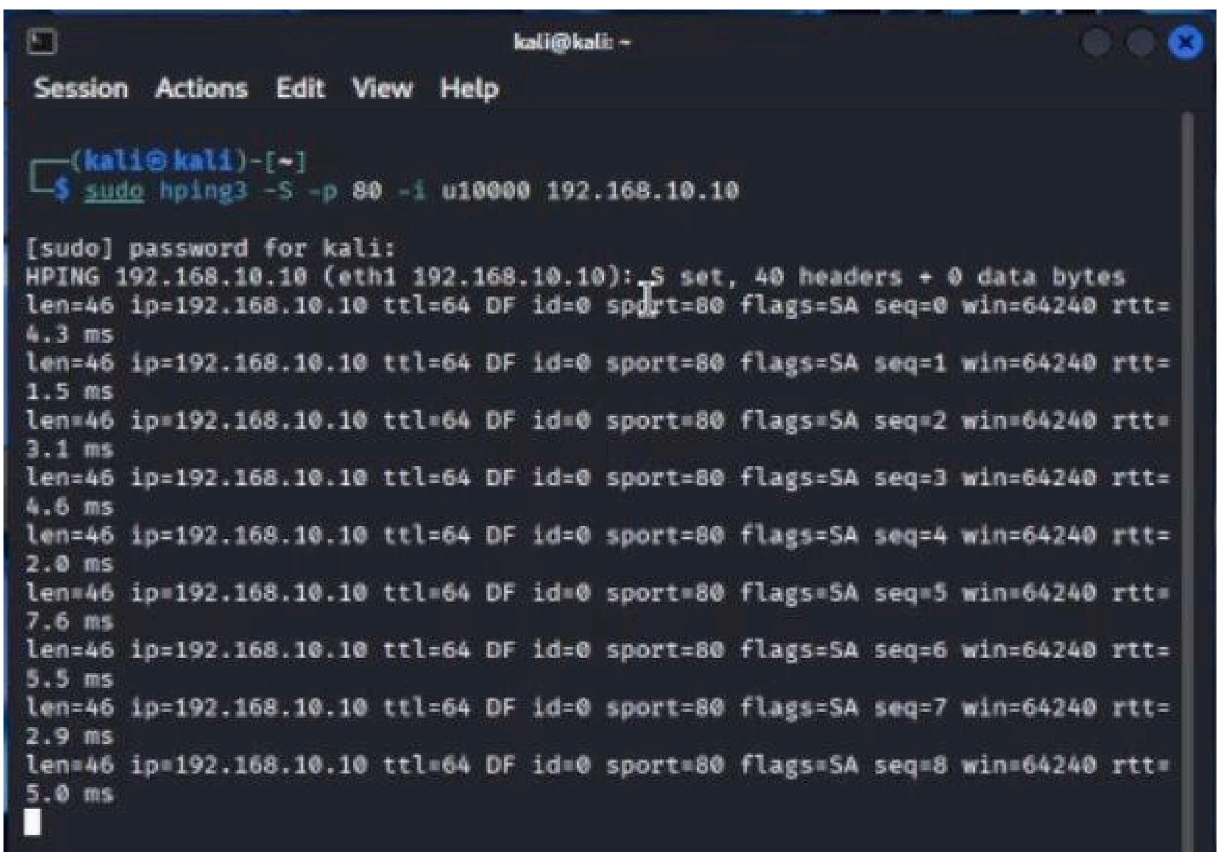
4. VPN Remote Access

We have configured remote access for external users to access the private Local Area Network (LAN). New users will be assigned an IP address from 10.10.10.1 to 10.10.10.10, allowing external users to securely connect through the VPN tunnel and ensuring that only authenticated users have access to the resources.

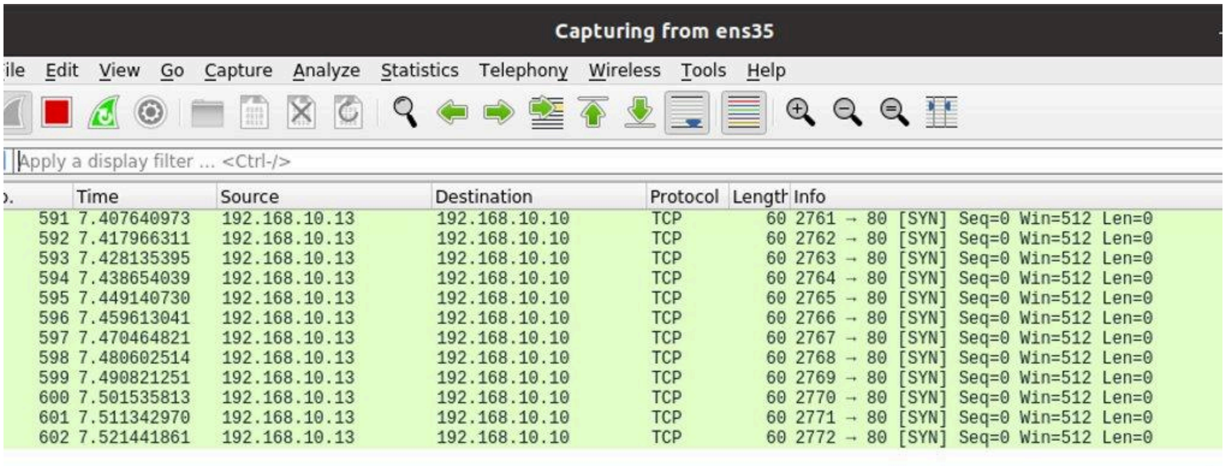


5. IDS/IPS Config/Simulating SYN flood attack

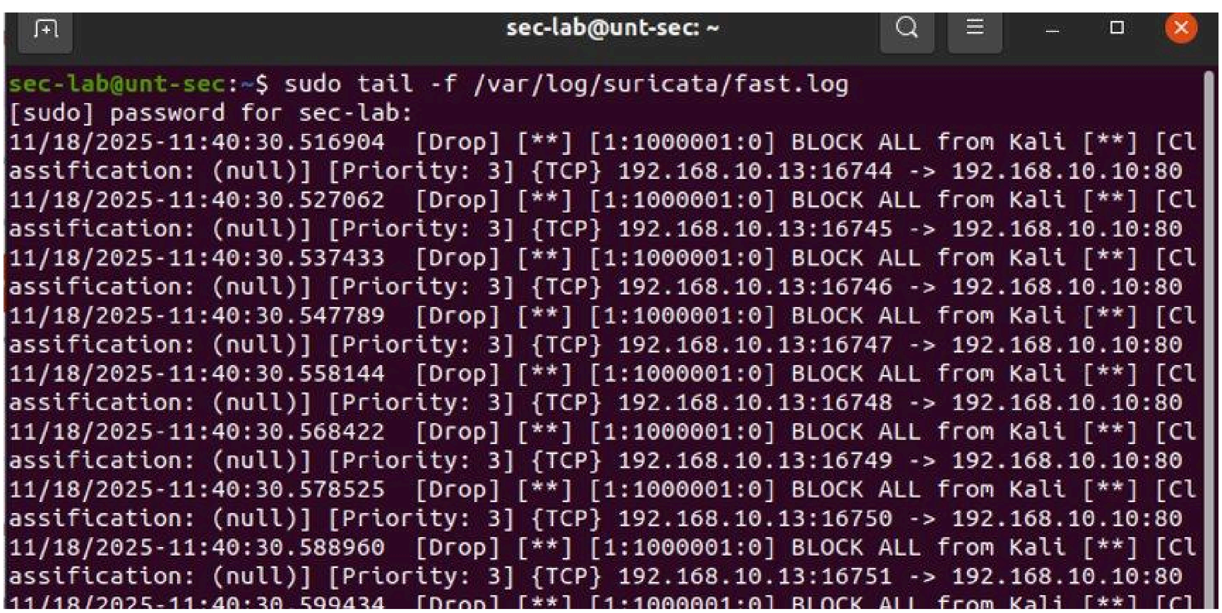
Along with IPS in firewall functionality, we also implement additional IDS and IPS in the Ubuntu Client to demonstrate the effectiveness of both systems. First, we use the hping3 attack to flood the Ubuntu with SYN packets from the Kali client to the Ubuntu Client. From the Ubuntu client, we detected the attacks using Wireshark and prevented them using Suricata.
For the attack, we are using the command “sudo hping3 -S -p 80 -i u10000 192.168.10.10”. This command will allow us to send SYN packets, denoted by -S, through port 80, denoted by -i. The u10000 value represents the time interval between each packet sent to the Ubuntu client, which has the IP address 192.168.10.10.



In the Ubuntu Client, we will use Wireshark to listen to the traffic. By doing this, we can see that there are numerous SYN packets from the Kali client to the Ubuntu client using port 80.

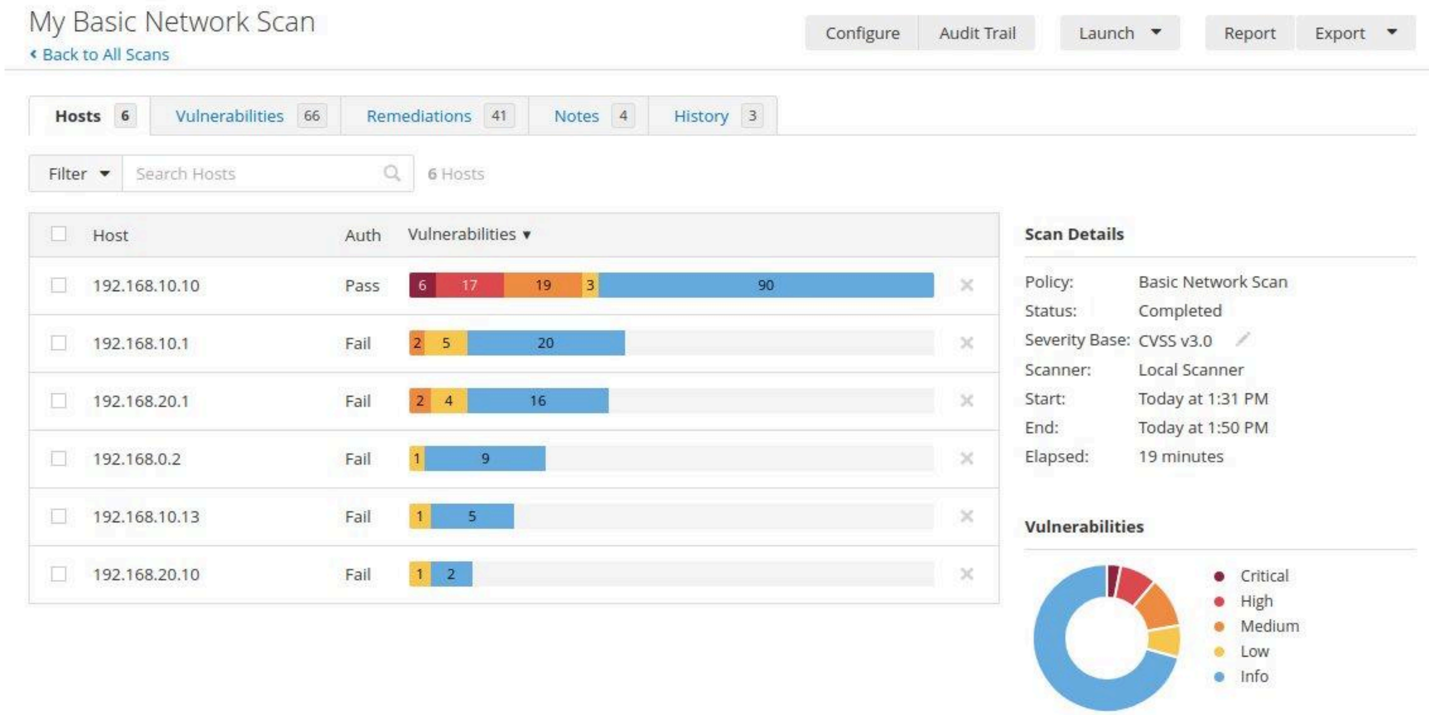


We then run the command “sudo nano /etc/suricata/rules/local.rules” to identify the rules to block all traffic from the Kali client. We will activate Suricata by running the command “sudo suricata -c /etc/suricata/suricata.yaml -i ens35”. Finally, we run the command “sudo tail -f /var/log/suricata/fast.log” to check the logs related to the attack simulation, and we can observe that many packets from the Kali client (192.168.10.13) have been dropped/blocked.



6. Security Assessment Host-Level Findings

For our network security assessment, we utilize Nessus, a vulnerability assessment tool. The goal of the vulnerability assessment is to identify weaknesses within the system and propose mitigation strategies to enhance network security. By identifying these new vulnerabilities as soon as possible, we can demonstrate a critical cybersecurity routine, such as regularly updating and monitoring the system. After performing a host scan and a vulnerability assessment, we identified several critical vulnerabilities at the Host level and two minor vulnerabilities at the network level. At the host level, the majority of our vulnerabilities stem from outdated packages in many applications. Most importantly, the Ubuntu host machine is an end-of-life (EOL) device, meaning it can no longer receive security patches from the manufacturer. For our network vulnerabilities, we found out that the SSH algorithms we use for the VPN access are very weak, and the ICMP timestamp request, which potentially leaks the mapping of the network to the attackers. To mitigate the vulnerabilities, we will address the host-level vulnerabilities by updating the required packages on the Ubuntu local machine. To defend against ICMP timestamp requests, we can simply filter ICMP requests and outgoing timestamp replies to ensure the system does not reveal any information to external sources. For weak SSH algorithms, we can strengthen them by disabling the weak algorithms and enforcing a stronger algorithm, such as Advanced Encryption Standard (AES), via the FortiGate command line interface. By implementing the required mitigation strategies, we can demonstrate the practice of system hardening, reducing the attack surfaces while ensuring all vulnerabilities are properly addressed. This significantly strengthens the overall network security, demonstrating a secure network design and implementation.



9. Conclusion

In conclusion, the project showcases a secure network design by evaluating the functionality as well as the effectiveness of the IDS/IPS, Firewall policies, and remote access VPN. By using a simulated SYN flood attack using hping, we can see how the system works as intended by detecting and preventing the malicious traffic. FortiGate firewall filtering of traffic was also demonstrated with complete segmentation from the DMZ to the LAN switch using a firewall policy. In addition, the vulnerability assessment highlights the importance of keeping a system up to date and under constant observation. These practices combined will greatly reduce the number of vulnerabilities, demonstrating strengths in network security.

References

CVE-1999-0524 Detail. NVD. (2004, January 1). https://nvd.nist.gov/vuln/detail/CVE-1999-0524

CVE-2019-0053 Detail. NVD. (2019, July 24). https://nvd.nist.gov/vuln/detail/CVE-2019-0053