Project Title: Setup and Secure a Virtual network

Group Number: 8

Team Members: Baban Poonia and Sukhjot Singh

Course Information: INFO 3171 System Security (S10)

PROJECT OVERVIEW

Objective: The main objective of this project is to create a virtualized network environment and apply security measures to secure it from unauthorized access. The setup addresses the need for a secure network to tackle threats like network intrusion, Denial of Service attack. By implementing a firewall with pfSense, and IDS with Snort, the project demonstrates the methods for securing the communication channels and proactively monitoring the traffic to identify potential threats. This approach is to both detect and prevent attacks, and to improve overall security of the network.

Problem Statement: The core problem this project addresses is the importance of setting up a network in a secured method. It focuses on basic things but those are very important to secure the network. It's often the basic things that some people don't focus on and ignore a vulnerability in a network. Attackers can exploit those vulnerabilities, and these can be data breaches, loss of sensitive information. This project shows how a basic firewall and IDS can reduce such risk.

BACKGROUND INFORMATION

Cybersecurity Topic

Targeting typical weaknesses in network access and traffic management, this project focuses on network security in a virtualized context. A crucial component of cybersecurity is network security which guards against threats including Denial of Service assaults, illegal access, and possible invasions. Using a tiered strategy that combines a firewall and an Intrusion Detection System to safeguard data flow and monitor for possible threats, this project showcases

fundamental security approaches by mimicking an organizational network within a virtual machine environment.

In simple terms, pfSense acts as a firewall controlling which traffic can enter or leave the network while Snort, the Intrusion Detection System, monitors everything to catch any suspicious activity. Together these tools provided a setup that could filter, detect, and respond to potential security risks within the network.

Existing Solutions

Network security may be achieved in many other methods, from more versatile software-based solutions like pfSense and Snort which we used here, to hardware firewalls which are used in bigger companies. Advanced security is provided by traditional hardware firewalls such as those made by Cisco, although they may be expensive and occasionally difficult to adjust. We used pfSense and Snort because we needed a virtual solution. PfSense became the main hub for traffic flow management in our configuration. To handle IP addresses and control DNS requests we configured it as the network's gateway and DHCP server. We set up firewall rules to only allow necessary and needed traffic, such as DNS and web surfing. While blocking other potentially dangerous traffic to maintain network security in our virtualized network environment. By continuously monitoring for indications of assaults, such as anomalous access attempts or spikes in network traffic that may point to a DoS attack, Snort provides an extra degree of protection. By combining the two tools we were able to create a network that was secure and adaptable to many requirements.

Practical Application

This project replicates the type of network configuration that many small enterprises or educational labs may use, this project is practical for that reason. To imitate a real-world situation, we constructed a network with a Windows Server serving as the domain controller and DNS server and a Windows 10 client as the user's endpoint. This configuration improved our comprehension of how fundamental security technologies contribute to network security. Our project demonstrated firsthand how intrusion detection and firewall rules may cooperate to monitor and safeguard a network. For example, we might regulate the traffic flow by configuring rules in pfSense and we could receive notifications from Snort anytime something questionable occurred. We showed how this strategy may make a network much more difficult to attack by learning how to handle these signals and adjust the security settings. This opened our eyes in many ways when it comes to cyber security for example, even simple security measures may massively lower risks and stop data breaches.

IMPLEMENTATIONS

Technical Approach

The approach for this project is to implement a virtual network using a virtual machine. We make it work like it’s a network in a small organization with some devices connected to it, and we implemented the security measures on that. The project aims to build a security framework using multiple layers of defense like pfSense (firewall) and Snort (IDS). The primary object is to create a network that could have secure communication and monitor the traffic for anything unusual. The firewall and IDS both have some rules implemented in them which restrict unnecessary traffic and allow only necessary traffic. There is also a firewall rule for limiting the bandwidth from one source IP address to prevent DOS attacks. We also added a firewall rule for WAN interface to avoid any external device to send any ping or packet to any

internal device on our network, this is to make sure that no one tries to use things like *Nmap* to scan our network and attack it.

Tools & Technologies: To create the virtual network, we have used the **Oracle VirtualBox 7.1.2.** On which we have three different machines. The first one is the firewall, and it is **pfSense 2.7.1 (AMD64).** This acts as the default gateway for other machines in this network and it will also act as the DHCP Server and provide/lease IP addresses to all the connected devices on the internal network. Only this machine is connected to the internet, the rest of the machines are only connected to the internal network, which in this case is named 'intnet' (internal network). All other machines get the internet through the pfSense. On pfSense, we have implemented the Intrusion Detection System called **Snort.** Specifically, we are using **Snort 2.9.20.8** with GPLv2 Community rules implemented on that. Those rules help detect most common attacks and threats for the networks. The second machine is a **Windows Server 2019** machine. This machine is used as the domain controller to manage all the user accounts and devices. This helps in managing permissions for the files and it is essential for having a functional client machine, which is the 3rd machine on this network. The Windows Server Machine also works as the DNS Server. For client/host machine, we are using **Windows 10 Pro edition**. Windows Server 2019 has an Active Directory running on it, it provides some accounts that can be used on the client machine by users. The client machine is for end users. We are also using Wireshark 4.2.8 to monitor the traffic on these machines.

Development Process: First the pfSense was installed so that it can be used as the default gateway. Then the Windows Server machine was installed and configured for Active Directory. Then, a Windows 10 client machine was installed to test if the network connection is working

and the user accounts made in Active Directory are working on the client machine. All the machines were connected to each other by adjusting the adapter setting on the VirtualBox to have all machines connected to the internal network ‘intnet’ and only pfSense was connected to the internet/ethernet. All machines were able to ping each other which helped us know that connectivity is good, and we moved on to the next step which was to install Snort on the pfSense by using the web interface of pfSense. Also, at this stage we added some rules in the firewall to allow essential traffic and block the rest. Only essential ports were allowed in the pfSense rules like port 80 and 443 for web interfaces. Also ports for DNS, LDAP and SMB were allowed so Windows Server could communicate with Client machine and allow file sharing as well. After all this we tested the network. All essential things were working and the IDS Snort was sending alerts for anything unusual like if we visited any unsafe webpage, it would send an alert on the pfSense firewall under Snort - > Alerts.

Security Measures: Several key security measures were implemented throughout the project to ensure the network remained secure:

- **Firewall Rules**: The pfSense firewall was configured to only allow traffic from trusted sources and block all other unauthorized traffic. Rules were set for source/destination IP and port numbers to minimize exposure to threats.
- **Intrusion Detection System**: Snort was configured with predefined rules to detect common network-based attacks like port scans, DoS attacks, and unauthorized access attempts. It was set up to log any suspicious activities and generate alerts, which were monitored through the pfSense web interface.

- **Rate Limiting**: To mitigate the risk of DoS attacks, rate limiting was configured in pfSense to limit the amount of traffic from any single IP address, helping to prevent a single source from overwhelming the network with excessive requests.
- **Traffic Monitoring**: Traffic is monitored all the time by IDS and firewall and we also do manual monitoring by using Wireshark to see all the traffic on the network. This can help us get data of all the packets on the network, which we can look into if we need to.

By implementing these security measures, the project ensured that the network was protected against common threats while also providing tools for monitoring and alerting in the case of an attack.

PROJECT TEAM CONTRIBUTIONS

Roles and responsibilities: For this, Sukhjot first made the design of the virtual network and installed Oracle Virtualbox to install all the required machines. Baban did research for the firewalls and chose pfSense and configured all the rules for the pfSense firewall. Baban installed and implemented a firewall, made and added all the required rules. Sukhjot researched for IDS and finalized Snort that was added on top of the pfSense and then Baban added the GPLv2 community rules in that. Sukhjot installed Windows Server and Client machines and attached all of them to one internal network. Sukhjot set up the server as the domain controller, configured that for Active directory and for DNS server as well. Sukhjot added some user accounts in Active Directory. Baban tested all the machines and verified that they can all communicate with each other. He tested the IDS system by visiting an untrusted website on the internet using a client machine and saw that IDS sent an alert about it. We also used Wireshark to monitor the

network traffic. Then, we both documented everything in this project report.

Collaboration: For making the project report, we made two google docs files which we could both edit. We added all the details of everything we did there, including some screenshots that we will use for the presentation (not for this report). After class, we met in the library to work on the project. We did planning in the first week and started implementing the project next week. We had a brief meeting after each class to discuss the project and update each other about the progress on the given parts of the project. We used 1 laptop for hosting this virtual network and managed everything there. We did research for the project in our own time and then contributed to the project as needed.

RESULTS AND OUTCOMES

Result: In the end, we have a virtual network with machines which are able to communicate with each other and they get their IP Address from the pfSense firewall. The pfSense firewall also acts as their default gateway and filters the traffic for all the internal machines. Moreover, the IDS Snort is able to detect an unusual activity done by any device in the network. For example, if someone visits an unsecure website using the client machine, the IDS will send an alert about it (this is also shown in the demo video). Overall, we were able to set up a secure and isolated network on the VirtualBox which includes a Windows Server for Active Directory, a Windows 10 client, and a pfSense firewall with Snort for intrusion detection.

Outcomes: The goal was to have a secure network where anything unusual could be detected and the outcome was the same. This project shows the importance of having a firewall and Intrusion Detection System in a network. Moreover, the firewall rules can be added as per the

requirement of the network and the IDS can be activate to work as Intrusion Prevention System but we did not do that here because there can be a lot of false positives if the IPS is always active. The project illustrated that a properly configured firewall combined with IDS can effectively reduce the impact of network vulnerabilities, thereby reinforcing the importance of layered security measures. There is no perfectly secure network but having some preventive measure like firewall and IDS can make a huge difference in keeping the network safe in most cases.

Demo Video: **Here's the link for the demo video**

CONCLUSION AND REFLECTIONS

Project Leanings: Through this project, we gained a deeper understanding of network security concepts such as firewall configuration, and the practical application of intrusion detection systems. Also, we learned how to make the Windows Server work as the DNS server and also make it work as domain controller and do Active Directory related stuff while the DHCP server job and Default Gateway job is given to the firewall pfSense because it is the first point of contact for external devices to communicate with this network.

Challenges and Solutions: Setting up the initial network was challenging. Initially, we aimed to have the Windows Server function as the DHCP server, but this caused issues, with the client machine unable to access the internet. To resolve this, we configured pfSense as the DHCP server and default gateway. With this setup, pfSense also automatically became the DNS server.

However, since we needed Active Directory services on the Windows Server for user account management, we wanted it to act as the DNS server as well.

After testing various configurations, we found a solution: on the client machine, we set the Windows Server as the DNS server. On the Windows Server itself, we used the loopback address (127.0.0.1) as its DNS server. Then, within the DNS settings in Windows Server Manager, we set a forwarder to Google's external DNS server (8.8.8.8). This way, whenever the client machine queried the Windows Server, it could access user accounts and other resources from Active Directory. For requests that the Windows Server couldn't resolve, it forwarded them to the external DNS server at 8.8.8.8.

Future work: There are several ways to extend this project further. We could add a VPN to remotely access this network. We could also do network segmentation but this project was not big enough to do that. It could be expanded by implementing automated response protocols within the IDS to actively block suspicious IPs but we did not want that because of false positives.

REFERENCES

Tools

Offensive Security. (n.d.). *Kali Linux*. Retrieved from https://www.kali.org

pfSense. (n.d.). *pfSense Documentation*. Retrieved from https://www.pfsense.org

Oracle. (n.d.). *VirtualBox Documentation*. Retrieved from https://www.virtualbox.org

Supplemental Learning Resources

Computer Security Division, I. T. L. (n.d.). *CSRC category: Security and privacy*. CSRC. https://csrc.nist.gov/Topics/Security-and-Privacy

Fhabte. (2024, January 31). *What is network security? the different types of protections*. Check Point Software. https://www.checkpoint.com/cyber-hub/network-security/what-is-network-security/

Lawrence Systems. (n.d.). *YouTube channel*. YouTube. Retrieved [October, 2024], from https://www.youtube.com/@LAWRENCESYSTEMS