Red Team – Network Analysis and Pen Test, RSA Authentication, Securing Data in Transit

Yogaraj Govindarajalu Prabagaran

# Table of Contents

[Table of Contents 2](#_Toc159698319)

[Lab 2- Red Team – Network Analysis and Pen Test 3](#_Toc159698320)

[Description 3](#_Toc159698321)

[Preparation 3](#_Toc159698322)

[Observations 3](#_Toc159698323)

[Screenshots 4](#_Toc159698324)

[Reflection 7](#_Toc159698325)

[Lab 6- RSA Authentication 8](#_Toc159698326)

[Description 8](#_Toc159698327)

[Preparation 8](#_Toc159698328)

[Observations 8](#_Toc159698329)

[Screenshots 9](#_Toc159698330)

[Reflection 14](#_Toc159698331)

[Lab 3- Securing Data in Transit 15](#_Toc159698332)

[Description 15](#_Toc159698333)

[Preparation 15](#_Toc159698334)

[Observations 16](#_Toc159698335)

[Screenshots 17](#_Toc159698336)

[Reflection 31](#_Toc159698337)

[References 32](#_Toc159698338)

# Lab 2- Red Team – Network Analysis and Pen Test

## Description

In this lab, I am examining vulnerabilities inherent in our network's FTP (File Transfer Protocol) services to understand their potential risks. I capture and analyze traffic generated during an FTP login attempt utilizing Wireshark, a network protocol analyzer. This provides insight into how plaintext usernames and passwords are transmitted over the network, underscoring the susceptibility to interception by malicious entities. Following the activity's outlined steps, I gained practical experience in identifying and addressing these vulnerabilities, emphasizing the necessity of secure authentication methods and encryption protocols to safeguard sensitive data transmission. This practical exercise revolves around network security principles and tools like Wireshark for vulnerability assessment.

## Preparation

I deployed a Windows Server 2019 Virtual machine utilizing a template with an IP address of 10.172.94.35.  
Downloaded and installed the “Wireshark” application in the Windows Virtual machine.  
I deployed a Rocky Linux Virtual machine with an IP address of 10.172.94.33.  
Enabled FTP services on the Rocky Linux to simulate a real-world scenario.  
Username- YG9578 is set for the Linux VM.

## Observations

I enabled FTP services on a designated Rocky Linux virtual machine (VM) to simulate a real-world scenario. Specific commands were collectively used to configure and set up a vsftpd FTP server on a Linux system. Initially, vsftpd is installed using the DNF package manager, and its service is started and enabled to ensure it runs automatically on system boot. The status of the vsftpd service is checked to verify successful activation. The net-tools package is installed to provide additional networking utilities, and netstat is utilized to confirm that vsftpd is listening on the expected ports. A backup of the vsftpd configuration file is created before modifications. A new user, "YG9578," is added and given access to vsftpd by appending the name to the vsftpd user list. Firewall rules are then adjusted to allow traffic on FTP ports 20-21, and the firewall configuration is reloaded. Finally, the vsftpd service is restarted to apply the changes, and its status is checked for confirmation. These steps collectively establish a functioning vsftpd FTP server with appropriate configurations and user access permissions.

Then, I installed and configured Wireshark on the Windows workstation to intercept network traffic. Next, I initiated an FTP login attempt to the target server, replicating the actions of a legitimate user. I captured the network traffic using Wireshark, capturing packets exchanged during the login process.

Afterward, I analyzed the captured packets to extract plaintext credentials, demonstrating the vulnerability of FTP to credential interception. I utilized Wireshark's features, such as packet filtering and flow analysis, to streamline the identification of credential data within the captured traffic. Finally, I applied built-in filters within Wireshark to isolate and extract the credentials used for FTP authentication, providing concrete evidence of the vulnerability.

## Screenshots

A screenshot of a computer

Description automatically generated

Figure 1.1: VM Configuration - This screenshot shows the initial configuration settings of the virtual machine. It displays the network settings and services enabled on the VM.

A screenshot of a computer

Description automatically generated

Figure 1.2: VM Configuration   
This screenshot shows the configuration settings of the virtual machine where FTP services are enabled. It displays the network settings and services promoted on the VM.

A screenshot of a computer

Description automatically generated

Figure 1.3: Final VM Configuration   
This screenshot shows the final configuration settings of the virtual machine where FTP services are enabled after adding the user.

A screenshot of a computer

Description automatically generated

Figure 1.4: FTP Login Attempt - This screenshot captures the attempt to log in to the FTP server from the workstation using WinSCP. It shows the login interface and the input of username and password.

A screenshot of a computer

Description automatically generated

Figure 1.5: Credential Identification - This screenshot demonstrates using built-in filters in Wireshark to extract user credentials from the captured network traffic. It displays the application interface with the filtered results showing the extracted credentials.

## Reflection

The network penetration activity targeting FTP services provided invaluable insights into the inherent vulnerabilities associated with the File Transfer Protocol. By deploying a Linux VM with vsftpd configured as the FTP server and a Windows VM equipped with Wireshark for traffic analysis, we embarked on a simulated attack scenario to exploit these vulnerabilities.

Throughout the exercise, the fundamental insecurity of FTP became evident as data transmission occurred in plaintext, leaving usernames and passwords vulnerable to interception. This highlighted the significance of establishing encryption and secure authentication techniques to reduce hazards in real-world network systems.

Wireshark's use was crucial in recording and examining network traffic as it enabled me to see the unencrypted transfer of credentials during the FTP login procedure. By following packet flows and applying filters within Wireshark, we efficiently identified and extracted the credentials used for FTP authentication.

Moreover, the practical knowledge obtained from this exercise offered valuable insights into carrying out vulnerability assessments and network penetration tests. To protect against possible attacks, it emphasized the need for proactive security measures and ongoing monitoring and strengthening of security standards.

# Lab 6- RSA Authentication

## Description

In this lab, I explored the intricacies of implementing RSA authentication for secure shell (SSH) connections between Windows and Linux systems. Following a detailed set of instructions, I learned how to generate an RSA key pair on my Windows platform using PuttyGen, securely transfer the public key to the Linux environment, and configure the Linux system to authenticate users using RSA keys. This practical exercise deepened my understanding of cryptographic techniques for ensuring secure communication and enhanced my skills in setting up secure network connections across different computing platforms. Through completing this lab, I have gained valuable proficiency in essential aspects of network security, including deploying secure authentication mechanisms and seamlessly integrating cryptographic protocols within a network infrastructure.

## Preparation

Before beginning the lab, I ensured that I had all the necessary components set up and configured correctly:

Virtual Machines Deployment: Using my vSphere client, I deployed two virtual machines: Windows Server 2019 and Rocky Linux.

Static IP Assignment: To ensure consistent connectivity, I assigned static IP addresses to both virtual machines. Verifying that both VMs were on the same virtual network and could successfully ping each other was crucial to establishing communication between them.  
I deployed a Windows Server 2019 Virtual machine utilizing a template with an IP address of 10.172.94.35.  
I deployed a Rocky Linux Virtual machine with an IP address of 10.172.94.34.

Putty Application Installation: I downloaded all Putty applications onto the Windows VM to facilitate SSH connections and RSA key generation. I opted for the zip file at the bottom of the download list on the Putty website to ensure compatibility and ease of installation.

By completing these preparatory steps diligently, I laid a solid foundation for the successful execution of the lab tasks and ensured a smooth experience throughout the configuration process.

## Observations

Step 1: RSA Key Pair Generation (Windows VM)

I launched the PuttyGen utility and generated an RSA key pair, ensuring that RSA was selected. After entering a passphrase, I saved only the private key. Then, I copied the content of the generated public key in PuttyGen.

Step 2: User Configuration (Linux VM)

I logged in and switched to the root user on the Linux VM. Next, I created a user account without a password. Changing to the user's home directory, I made a hidden directory named .ssh. Setting permissions for the .ssh directory and adjusting to it, I used the echo command to append the public key to the authorized\_keys file. Finally, I returned to the root user account.

Step 3: SSH Authentication (Windows VM)

I launched the Putty authentication agent utility pageant.exe to authenticate without a password prompt. Adding the private key to the agent, I launched Putty and entered the Host Name in the format ygprabagaran@10.172.94.34. With this setup, I was able to authenticate seamlessly.

## Screenshots

A screenshot of a computer

Description automatically generated

Figure 2.1: RSA Key Pair Generation

This screenshot captures the Windows VM's RSA key pair generation process using the PuttyGen utility. It demonstrates the selection of RSA, the key's generation, and the private key's saving in the directory containing the Putty apps.

A screenshot of a computer

Description automatically generated

Figure 2.2: User Account Creation on Linux VM

This screenshot illustrates the creation of a user account and preparation for RSA key-based SSH authentication on the Linux VM. It shows the steps to create a user account and set up the .ssh directory.

A screenshot of a computer screen

Description automatically generated

Figure 2.3: Adding Public key to the authorized\_keys file.

This screenshot shows the steps to set up the .ssh directory and add the public key to the authorized\_keys file.

A screenshot of a computer

Description automatically generated

Figure 2.4: Launching Putty Authentication Agent

This screenshot exhibits the launch of the Putty authentication agent utility, pageant.exe, on the Windows VM. It demonstrates adding the saved private key file to the agent and entering the passphrase when prompted.

A computer screen with a white box

Description automatically generated

Figure 2.5: Putty Configuration for SSH Authentication

In this screenshot, the configuration of Putty for SSH authentication into the Linux VM without using credentials is displayed. It shows the entry of the Host Name in the format [ygprabagaran@10.172.94.34](mailto:ygprabagaran@10.172.94.34).

A computer screen shot of a black rectangle

Description automatically generated

Figure 2.6: Putty Configuration for SSH Authentication

This screenshot clearly shows the successful authentication without a password prompt.

## Reflection

This lab provided valuable insights into the intricacies of configuring RSA authentication for SSH connections between Windows and Linux systems. Throughout the process, I gained practical experience in generating RSA key pairs, configuring user accounts for RSA authentication on the Linux system, and authenticating without a password prompt from the Windows VM.

One aspect that stood out to me was securely transferring the public key to the Linux system. This highlighted the critical role of cryptographic techniques in ensuring secure communication across networked environments.

Additionally, configuring user accounts on the Linux system underscored the significance of proper user management practices and file permissions, particularly in the context of SSH authentication.

Overall, this lab enhanced my understanding of RSA authentication and reinforced the importance of security best practices in network administration.

# Lab 3- Securing Data in Transit

## Description

In Lab 3 on Securing Data in Transit, my lab partner Fiona and I embarked on a comprehensive hands-on exercise to equip us with the knowledge and skills necessary to ensure secure data transmission across networked environments. The lab was structured into two parts, each configuring different network aspects.

For the first part, I was tasked with designing and configuring the network infrastructure for the main office, Toronto Site 1. This involved setting up NGWF (Palo Alto Firewall) and routers. I carefully assigned interfaces to different network segments, configured routing, created network zones, and implemented network address translation (NAT) policies to facilitate secure communication within the network.

In the second part of the lab, we established an IPSec VPN tunnel between Toronto Site 1 and our peer site located in British Columbia (BC\_SiteR1). Fiona and I collaboratively validated IP connectivity, created tunnel interfaces, and configured crypto profiles, IKE gateways, and IPSec tunnels. We defined policies to allow traffic between the subnets of the two sites and troubleshoot VPN tunnel connectivity issues using CLI commands and GUI tools.

Together, Fiona and I could apply our theoretical understanding of network security concepts to practical situations, which was an invaluable educational experience. By finishing this lab, we have improved our abilities to build, configure, and secure network infrastructures and create secure communication routes between geographically separated locations.

## Preparation

Before starting the lab, Fiona and I ensured we had all the necessary resources and access to complete the tasks effectively. We followed these steps to prepare for the lab:

Patch cable connection on Kit 99 and Kit 100:

* Kit 99 Palo Alto port one to fe 0/1 in the router.
* Kit 100 Palo Alto port one to fe 0/1 in the router.
* Kit 99 Fe1/1 router to Kit 100 Fe1/1 router utilizing cross connections.

Review Lab Requirements: We carefully reviewed the lab instructions to understand the objectives, tasks, and components of configuring the network infrastructure and establishing the IPSec VPN tunnel.

Allocate Responsibilities: Fiona and I discussed and allocated responsibilities for different lab parts. We decided that I would focus on configuring the network infrastructure for Toronto Site 1. At the same time, Fiona will establish the configuration with BC\_SiteR1, and we will work on the IPSec VPN tunnel.

Gather Required Materials: We gathered all the required materials, including access credentials for the network devices and firewall GUI. Additionally, we ensured we had access to the software tools and documentation needed for the configuration tasks.

By carefully preparing and allocating responsibilities, Fiona and I ensured we were ready to tackle the lab tasks efficiently and effectively. This proactive approach enabled us to maximize our lab time and achieve the desired learning outcomes.

## Observations

Throughout our lab completion, my lab partner Fiona and I made several noteworthy observations regarding the configuration process, network behaviors, and troubleshooting methodologies. These insights provided a valuable understanding of network security practices and protocols. Here are some key observations:

Network Configuration Complexity: We noted the intricate nature of configuring the network infrastructure, including NGWF (Palo Alto Firewall) and routers. Each task, from assigning interfaces to establishing network zones and defining routing policies, contributed to the overall complexity of the network setup.

Collaborative Approach: Collaboration played a pivotal role in efficiently completing lab tasks. By dividing responsibilities based on our strengths and expertise, Fiona and I could work concurrently, leading to expedited progress. Additionally, sharing insights and jointly troubleshooting challenges facilitated smoother problem resolution.

Understanding of IPSec VPN Tunneling: Establishing the IPSec VPN tunnel between Toronto Site 1 and BC\_SiteR1 afforded us a more profound comprehension of VPN technologies and protocols. We observed the step-by-step process of configuring IKE gateways and IPSec tunnels and defining security policies to ensure secure site communication.

Verification and Troubleshooting Strategies: We learned the significance of meticulously verifying each configuration step and employing effective troubleshooting techniques to identify and resolve issues. Leveraging CLI commands and GUI tools enabled us to validate IP connectivity, monitor VPN tunnel status, and diagnose connectivity problems proficiently.

Real-world Applicability: The lab provided a realistic simulation of network security challenges encountered in enterprise environments. By applying theoretical knowledge to practical scenarios, we gained valuable hands-on experience.

In summary, our observations underscored the intricacies of network security tasks, emphasized the importance of collaboration and adept troubleshooting, and highlighted the practical application of theoretical concepts in securing data transmission across networked environments.

## Screenshots

A screenshot of a computer

Description automatically generated

Figure 3.1: Firewall Interface Assignment and Layer 3 Interface Configuration.

This screenshot showcases the assignment of the interface to the network topology designed for the firewall configuration. It displays the mapping of Ethernet interfaces to specific network segments, such as the Internet. It shows the successful assigning of layer three interfaces and IP addresses to that interface according to the designated network segments.

A screenshot of a computer

Description automatically generated

Figure 3.2: Virtual Router Configuration

This screenshot captures the configuration of a virtual router to enable routing between the different networks created on the firewall. It shows the addition of interfaces to the virtual router and the setup of a static route for default gateway routing.

A screenshot of a computer

Description automatically generated

Figure 3.4: Network Zone Creation

This screenshot depicts the creation of network zones for the Internet. It showcases the configuration of network zones with specific interfaces assigned to enforce the segmentation and isolation of network traffic.

A screenshot of a computer

Description automatically generated

Figure 3.5: Firewall Connectivity configuration

This screenshot displays configuring Management profiles for the interface to unblock ICMP packets. It demonstrates the successful configuration of firewall management profiles to allow ping.

A screenshot of a computer

Description automatically generated

Figure 3.6: Source NAT Policy Creation

This screenshot illustrates the creation of a Source NAT Policy for translating private network addresses to routable addresses on the Internet. It showcases the configuration of NAT parameters and verifies the created policy.

A screenshot of a computer

Description automatically generated

Figure 3.7a: VPN Tunnel Configuration

This screenshot showcases the configuration of a VPN IPSec tunnel between the two sites. It displays the setup of tunnel interfaces, crypto profiles, IKE gateways, and IPSec tunnels to establish secure communication between the sites.

A screenshot of a computer

Description automatically generated

Figure 3.7b: VPN Tunnel Configuration

This screenshot showcases the configuration of a VPN IPSec tunnel between the two sites. It displays the setup of tunnel interfaces, crypto profiles, IKE gateways, and IPSec tunnels to establish secure communication between the sites.

A screenshot of a computer

Description automatically generated

Figure 3.7c: VPN Tunnel Configuration

This screenshot showcases the configuration of a VPN IPSec tunnel between the two sites. It displays the setup of tunnel interfaces, crypto profiles, IKE gateways, and IPSec tunnels to establish secure communication between the sites.

A computer screen shot of a computer screen

Description automatically generated

Figure 3.8a: VPN Tunnel Verification

The traffic flow and VPN tunnel status are verified in this snapshot. The processes to verify IP connectivity, troubleshoot VPN tunnel problems, and verify the status of each VPN tunnel phase using the GUI and command-line interface (CLI) are demonstrated.

A screenshot of a computer

Description automatically generated

Figure 3.8b: VPN Tunnel Verification

The traffic flow and VPN tunnel status are verified in this snapshot. The processes to verify IP connectivity, troubleshoot VPN tunnel problems, and verify the status of each VPN tunnel phase using the GUI and command-line interface (CLI) are demonstrated.

A screenshot of a computer

Description automatically generated

Figure 3.8c: VPN Tunnel Verification

The traffic flow and VPN tunnel status are verified in this snapshot. The processes to verify IP connectivity, troubleshoot VPN tunnel problems, and verify the status of each VPN tunnel phase using the GUI and command-line interface (CLI) are demonstrated.

A screenshot of a computer

Description automatically generated

Figure 3.8d: VPN Tunnel Verification

The traffic flow and VPN tunnel status are verified in this snapshot. The processes to verify IP connectivity, troubleshoot VPN tunnel problems, and verify the status of each VPN tunnel phase using the GUI and command-line interface (CLI) are demonstrated.

A screenshot of a computer

Description automatically generated

Figure 3.9a: GUI Verification of IPSec Tunnels

This screenshot demonstrates verifying the status of IPSec tunnels using the graphical user interface (GUI). This GUI verification allows for a comprehensive assessment of the IPSec tunnel configuration and ensures that all phases function as intended for secure communication between the two sites.

A screenshot of a computer

Description automatically generated

Figure 3.9b: GUI Verification of IPSec Tunnels

This screenshot demonstrates verifying the status of IPSec tunnels using the graphical user interface (GUI. This GUI verification allows for a comprehensive assessment of the IPSec tunnel configuration and ensures that all phases function as intended for secure communication between the two sites.

## Reflection

Undertaking this lab on securing data in transit provided me with invaluable practical experience in designing and configuring a network spanning two sites and establishing an IPSec VPN tunnel between them. Throughout the lab, I encountered various hurdles and gained insights into network security practices and protocols.

In the initial segment of the lab, I meticulously crafted and executed the network infrastructure for the main office (Toronto Site 1), incorporating NGWF (Palo Alto Firewall) and Cisco routers. Assigning interfaces to different network segments and configuring routing, zones, and network address translation (NAT) policies were pivotal steps in ensuring secure and efficient data transmission within the network.

The subsequent lab segment posed the challenge of setting up an IPSec VPN tunnel between Toronto Site 1 and the peer site in British Columbia (BC\_SiteR1). This involved validating IP connectivity, creating tunnel interfaces, configuring crypto profiles, IKE gateways, and IPSec tunnels, and defining policies to facilitate traffic between the sites' subnets. The effective setup and functioning of the VPN tunnel greatly depended on troubleshooting techniques, such as checking traffic flow and VPN tunnel status.

Overall, this lab was a thorough educational experience that enabled me to apply my theoretical understanding of network security concepts to actual situations. It underscored the importance of meticulous planning, configuration, and troubleshooting in securing data in transit across networked environments.

# References

1. Panopto. (n.d.). Panopto. <https://conestogac.ca.panopto.com/Panopto/Pages/Viewer.aspx?id=4199c21d-ebb4-49e1-b0cc-afa3015aec70>
2. *Playlist: In-demand content selected by experts and peers*. (n.d.). O’Reilly Online Learning. <https://learning.oreilly.com/playlists/166d99d6-8d39-48de-bf77-f66b5f505b1c/>
3. *Login with a Linux FTP username and password*. (n.d.). Super User. <https://superuser.com/questions/396594/login-with-linux-ftp-username-and-password>
4. P. N. (2021, November 8). *RSA Algorithm - How does it work? - I’ll PROVE it with an Example! -- Cryptography - Practical TLS*. YouTube. <https://www.youtube.com/watch?v=Pq8gNbvfaoM>
5. T. W. T. (2021, January 22). *Linux for Programmers #7 - Public/Private Key Authentication (RSA) & FTP*. YouTube. <https://www.youtube.com/watch?v=XPZSkEagn68>
6. J. D. (2023, February 10). *Importing and Loading Palo Alto Config*. YouTube. <https://www.youtube.com/watch?v=JXumelUwAoY>
7. J. D. (2023, March 10). *Configuring Palo Alto Site-to-Site VPN*. YouTube. <https://www.youtube.com/watch?v=kWN_Rn-bY5s>