|  |  |
| --- | --- |
| **A close up of a logo  Description automatically generated** | **TO00BR91-3007**  **Enterprise Security and Practitioners** |
|
|
| Anup Nepal | 14.03.2024 |
|  |

**Practice Lab work- Report-1**

**(Anup Nepal)**

Table of Contents

[Lab work I 3](#_Toc161740180)

[Lab 1: Work With Remote Connectivity 4](#_Toc161740181)

[Lab 2: Perform Digital Forensics 5](#_Toc161740182)

[Lab 3: Configuring SCCM Configuration Items and Baselines 7](#_Toc161740183)

[Lab work II 8](#_Toc161740184)

[Lab 1: Integrate Network and Security Components 9](#_Toc161740185)

[Lab 2: Install and Configure Network Load Balancing 12](#_Toc161740186)

[Lab 3: Implement SSL VPN using ASA Device Manager 15](#_Toc161740187)

[Lab 4: Perform Firewall Rule-based Management 17](#_Toc161740188)

[Lab work III 19](#_Toc161740189)

[Lab 1: Configure Verify and Troubleshoot Port Security 20](#_Toc161740190)

[Lab 2: Scanning and Remediating Vulnerabilities with OpenVAS 22](#_Toc161740191)

[Lab 3: Analyze network traffic with Wireshark 24](#_Toc161740192)

[Lab 4: Configure and Verify IPv4 and IPv6 Access Lists for Traffic Filtering 26](#_Toc161740193)

[Lab work IV 28](#_Toc161740194)

[Lab 1: Configuring Endpoint 29](#_Toc161740195)

[Lab 2: Encryption and Hashing 33](#_Toc161740196)

[Lab 3: Performing security assessment using various tools. 35](#_Toc161740197)

[Conclusion 38](#_Toc161740198)

# 

# Lab work I

## 

## Lab 1: Work With Remote Connectivity

This lab focused on remote connectivity working with both RDP and Windows PowerShell remoting to establish the connection between the systems.

### Walkthrough of the lab exercise:

|  |  |
| --- | --- |
| Exercise 1 : Working with RDP | |
| In this first exercise, the task was to establish a Remote Desktop connection between two systems. Using the RDC desktop, I attempt to connect to remote computer as shown in the picture on the right. Upon successful connection, I was able to access the system remotely. Also, there were other configuration which I did during this exercise such as audio setting and the drive to access. | Fig 1. Remote Desktop Connection-Login dialog box |
| Exercise 2: Administering Windows with PowerShell Remoting | |
| After trying out the RDP desktop, this exercise guided me through configuring and enabling Windows PowerShell remoting between PLABWIN10 and PLABWIN801. I disabled public network connections on both machines to establish PowerShell Remoting. Using Windows PowerShell, I enabled WinRM, configured trusted hosts, and established a remote connection with PLABWIN801 from PLABWIN10. I successfully managed remote computers by creating a new folder on PLABWIN801's C: drive using PowerShell commands from PLABWIN10. | Fig 2. Enabling Window Powershell Remoting (WinRM)    Fig 3. Remote connection successfull |
| Learning Reflection | |
| During this lab, I had an opportunity to learn and understand remote desktop connection in greater details both with desktop version and powershell version. I learned different ways to establish the remote connection, however, it was fascinating to observe that I needed to turn off the firewall rule, which for me was little weired, as it might increase the security threat. Anyhow, good lab session on building my confidence working with Remote desktop connection. | |

## Lab 2: Perform Digital Forensics

This lab focused on remote connectivity working with Linux File system, tools and methodologies used in digital forensics.

### Walkthrough of the lab exercise:

|  |  |
| --- | --- |
| Exercise 1: Preparing a Target Drive for Acquisition in Linux | |
| In this exercise, I was preparing a target drive for acquisition in Linux. I connected to the Kali Linux workstation, used GParted to format a disk volume as FAT32, and mounted the partition to /mnt/sdb1. This process involved creating a new partition, formatting it, and configuring the /etc/fstab file to ensure the partition is mounted automatically on boot. Finally, I verified the successful mounting of the partition by checking for the "key" icon next to the FAT32 partition in GParted. | Fig 1. Kali Linux GUI File Manager |
| Exercise 2: Deleting NTFS Files | |
| The first task was to get the SID of admin, I did so by using command prompt.  In the second task, I used OSForensics to view deleted files. Then, I deleted a folder using File Explorer and observed how OSForensics reflected the changes in the Recycle Bin.  Finally, I examined the properties of a deleted file such as File Viewer, Hex/String Viewer, Text Viewer, and Metadata tab. Additionally, I used the search feature to search for specific patterns and visually locate the deleted content of the file. | Fig 2. Viewing the property of deleted file using file\_system\_browsing tool provided by OS Forensic |
| **Exercise 3 - Using Sleuth Kit and Autopsy** | |
| I started by getting disk images from a file on Windows 10. Then, I restarted the Apache service on Kali Linux. After that, I used Sleuth Kit and Autopsy on Kali Linux to analyze disk images and find important data. I also downloaded USB drive images on Windows 10 for analysis. Finally, I used Autopsy on Windows 10 to examine disk images and look at email files. | Fig 3. Autopsy    Fig 4. Locating files using Slueth Kit and Autopsy |
| **Learning Reflection** | |
| This lab provided me with practical experience in working with disk management, disk partitioning, and so on in a Linux environment. By using tools like GParted and understanding the process of formatting partitions and configuring automatic mounting, I gained valuable skills in managing storage resources. Additionally, the lab offered insight into forensic tasks using tools like OSForensics, Sleuth Kit, and Autopsy. These tasks allowed me to explore techniques for collecting and analyzing digital evidence, enhancing my understanding of forensic procedures and methodologies. | |

## Lab 3: Configuring SCCM Configuration Items and Baselines

This lab dealt with server administration tools such as compliance setting, configuring, and deploying configuration baseline and so on using SCCM.

### Walkthrough of the lab exercise:

|  |  |
| --- | --- |
| Exercise 1 & 2 - Create Windows Configuration Items for Compliance Settings / Create Configuration Baseline and Deploy the Baseline | |
| In Exercise 1, I followed steps to create a configuration item in System Center 2012 R2 Configuration Manager, configuring settings related to operating system compliance. Then, in Exercise 2, I created and deployed a configuration baseline using System Center Configuration Manager on the PLABSYS01 server, ensuring compliance with specified settings across the network. | Fig 1. Completion status of the compliance rule |
| Exercise 3 - Create Windows Configuration Items for Compliance Settings | |
| In Exercise 3, I imported configuration data into SCCM Configuration Manager Packs from the web on the PLABSYS01 server. After navigating to the Practice Labs Intranet website and downloading the ConfigMgr2012ConfigPack.msi file, I ran the installer and completed the installation. Then, in the Configuration Manager Console, I imported the downloaded configuration pack file CM2012ServerRolesConfigpack, successfully adding the configuration data to SCCM Configuration Manager Packs. | Fig 2. Newly imported configuration baseline had been added to configuration pack |
| **Learning Reflection** | |
| Although I completed the tasks of creating a configuration baseline and deploying it using SCCM, I feel uncertain about how much I truly learned. I followed the instructions without fully understanding the process. However, I hope that in future labs, I will gain a deeper understanding of the concepts and actions I performed. | |

# Lab work II

## Lab 1: Integrate Network and Security Components

During this lab session, I focused on integrating network and security components, following the Percipio lab order. Before diving into the practical exercises, I carefully reviewed the learning outcomes and studied the lab setup diagram to grasp a clear understanding of the tasks ahead.

### Walkthrough of the lab exercise:

|  |  |
| --- | --- |
| Exercise 1: Install Snort | |
| In this exercise, I followed the instructions to install Snort on a Windows 10 Domain Member. The process was straightforward, and I also saved the blacklist.rules file from the intranet to local storage. | Fig 1. Installation of Snort completed |
| Exercise 2: Test Snort | |
| After successful installation and having downloaded the required file, it was time to test the snort tool. Since Snort being command line tool, it was accessed through the CMD terminal. Snort being a network detection and intrusion tool, we needed to assign a network interface so that Snort could listen to it using the command:  snort -i 1 -v  The output of this command is shown in the picture. Next task was to run Snort in IDS mode, and it required few more steps such as creating rules files and folders into Snort directory to configure before it started to work. | Fig 1. Assigning network interface for Snort to listen    Fig 2. Created rules file in Snort directory |
| Exercise 3: Configure and Re-Test Snort | |
| After setting up the necessary files and directories, I proceeded to modify the Snort configuration file and rerun the Snort tool.  The modifications involved substituting "ipvar" with "var" and adjusting the paths for directories such as dynamicprocessor, dynamicengine, and dynamicdetension, along with updating the paths for the previously created black and white rules lists.  These adjustments resulted in numerous changes to the configuration file. Because of these modifications, I successfully executed the Snort IDS | Fig 1. Making changes to snort.config file    Fig. 2. Successful execution of Snort IDS |
| Exercise 4: Configure Iptables | |
| This task was completed using the Kali machine. Initially, I installed Putty using Q-terminal. Additionally, to enable Telnet services, I activated the Telnet feature on a Windows 8 server and initiated the Telnet service to establish bidirectional communication. Subsequently, I conducted tests using both Windows 10 and Kali, demonstrating connectivity via Telnet and SSH on ports 23 and 22, respectively. Following this, I configured the IP tables on the Kali machine to prevent Telnet connections from exiting the Kali machine and connecting to external devices, as well as to block SSH connections to Kali from external sources. After configuration, as depicted in Figure 3, ports 22 and 23 rejected all ICMP packets. This configuration was validated using the Zen map GUI, which confirmed that ports 22 and 23 were filtered, preventing Nmap from scanning them. | Fig 1. Testing Telnet connection from Kali to Windows 8    Fig 2. SSH connection to Kali machine from windows 10 |
| Fig 3. Blocking the incoming and outgoing connection from port 22 and 23 | |

### Learning Reflection

Overall, I found the module to be enjoyable. However, I faced some confusion during the second exercise where I was merely changing the configuration file as per the instructions. This didn't provide me with a clear understanding of Snort in IDS mode, something I plan to explore further after completing the lab. The key lessons I learned from this lab include getting acquainted with Snort, configuring IP tables to block specific ports, and using Zen map to some extent. In a broader sense, I gained knowledge of new techniques and tools for implementing network security components, which enhances my skill set in this domain.

## Lab 2: Install and Configure Network Load Balancing

During this lab session, I focused on integrating network and security components, following the Percipio lab order. Before diving into the practical exercises, I carefully reviewed the learning outcomes and studied the lab setup diagram to grasp a clear understanding of the tasks ahead.

### Walkthrough of the lab exercise:

|  |  |
| --- | --- |
| Exercise 1: Installing Network Load Balancing Nodes | |
| In this exercise, there were two tasks to perform, the first one was to install IIS on servers using server manager and the second task was to install NLB on servers using Windows PowerShell. I was able to perform these tasks smoothly, however, to understand the context, I did some background study of IIS and NLB which helped me to understand the objective of this lab work. | Fig 1. Installation of IIS on server |
| Fig 2. Successfully installed the NLB component | |
| Exercise 2: Configuring a New NLB Cluster | |
| In the first task of this exercise, I created the initial NLB cluster node using the Network Load Balancing Manager. This process involved several steps, including selecting the host interface, assigning an IP address to the cluster within the host IP range, and providing an internet name for the cluster.  The second task involved configuring the DNS server to establish a hostname-to-IP address mapping that directs to the newly created cluster name using the domain controller server management. Once again, I followed a series of steps to input information about the new host, including its hostname/domain name and IP address. Upon successful addition of the host, I could observe it listed in the DNS Manager console. | Fig 1. Assigning parameters to a new NLB cluster    Fig 2. Adding new host in DNS manager |
| Exercise 3: Adding a Secondary Node to an NLB Cluster | |
| In this exercise, the objective was to expand the NLB cluster by adding a secondary node to the existing configuration established in the previous step. The process for creating the additional node was like the earlier task, however with a different host this time. Upon completing the task, the Network Load Balancing Manager console displayed the newly updated cluster configuration, as illustrated in Fig 1. | Fig 1. NLB Manager console showing cluster information and cluster configuration for all know NLB clusters. |
| Exercise 4: Examining the Working of an NLB Cluster | |
| This task was completed using the Kali machine. Initially, I installed Putty using Q-terminal. Additionally, to enable Telnet services, I activated the Telnet feature on a Windows 8 server and initiated the Telnet service to establish bidirectional communication. Subsequently, I conducted tests using both Windows 10 and Kali, demonstrating connectivity via Telnet and SSH on ports 23 and 22, respectively. Following this, I configured the IP tables on the Kali machine to prevent Telnet connections from exiting the Kali machine and connecting to external devices, as well as to block SSH connections to Kali from external sources. After configuration, as depicted in Figure 3, ports 22 and 23 rejected all ICMP packets. This configuration was validated using the Zen map GUI, which confirmed that ports 22 and 23 were filtered, preventing Nmap from scanning them. | A computer screen with a black background  Description automatically generated  Fig 1. Testing Telnet connection from Kali to Windows 8  A computer screen shot of a computer  Description automatically generated  Fig 2. SSH connection to Kali machine from windows 10 |
| A screenshot of a computer program  Description automatically generated  Fig 3. Blocking the incoming and outgoing connection from port 22 and 23 | |

### Learning Reflection

This lab work taught me the concept of NLB, its installation and configuration. I also learned how to start an IIS server in windows. I have previously studied load balancing; it was good to get hands-on experience on how it works in real scenarios. Moreover, I also got exposure working with telnet using SSH, understanding different connection patterns that exist.

## Lab 3: Implement SSL VPN using ASA Device Manager

This lab focuses on configuring and managing VPN solutions using the Cisco ASA Device Manager interface and implement both Clientless SSL VPN and AnyConnect VPN configurations, enabling secure remote access to network resources.

### Walkthrough of the lab exercise:

|  |  |
| --- | --- |
| Exercise 1: Implement a Clientless SSL VPN using the Cisco ASA Device Manager | |
| The task starts with launching Cisco ASDM on the Windows server.  Next step was to configure the clientless SSL VPN connection using the configuration tab for the ASDM program. After following multiple steps of configuration which included assigning new user, defining VPN types, in this case “Clientless”, assigning group policy, creating a bookmark list with title and URL, the summarylooks like this:  Next step was to try to connect to this web server in which we configured the SSL VPN. The webpage loads successfully and displays the bookmark we created earlier, however on hovering the bookmark, the popup came with a message “Cannot find server or DNS error” leading us to configure the firewall which is not recognizing the host.  After the configuration was completed, it was possible to access the web DMZ using clientless VPN. | A screenshot of a computer  Description automatically generated  Fig 1. Summary of Clientless VPN setup  A screenshot of a computer  Description automatically generated  Fig 2. Configure router to bypass firewall.  A screenshot of a computer  Description automatically generated  Fig 3. Successfully accessing the page using VPN |
| Exercise 2: Implement AnyConnect using Cisco ASA Device manager | |
| The first step in this exercise was to clear existing identity key pairs that exist in firewall. The configuration commands look like fig 1.  In the process of creating AnyConnect VPN, I got the following error, because of which I could the proceed to the exercise any further. Pity that I really wanted to see this VPN types working. | A screenshot of a computer  Description automatically generated  Fig 1. Configuring the firewall terminal to zeroize the crypto keys |
| A screenshot of a computer  Description automatically generated  Fig 2. Error in sending command while trying to configure AnyConnect VPN | |
| Learning Reflection This lab helped to improve my understanding of VPNs, especially how they work behind the scenes. Firstly, I learned about configuring SSL VPNs and observed several steps to follow to establish a connection. It was a good addition of having DMZ in our lab setup, allowing me to learn how to configure router in such scenarios. The second task was to configure AnyConnect and I was really interested in completing it but there occurred an error and I could not proceed forward. To conclude, I think I got the overview and technical knowhow on VPNs after completing this lab. | |

## Lab 4: Perform Firewall Rule-based Management

This lab focuses on configuring firewall Rules using Window Firewall, using Remote Desktop and Command Line Interface.

### Walkthrough of the lab exercise:

|  |  |
| --- | --- |
| Exercise 1 - Configuring Firewall Rules Using Windows Firewall | |
| Firstly, I verified the live connection between two computers by pinging PLABWIN10 from PLABDC01. The second task involved creating a firewall rule to block incoming ping packets to PLABWIN10, which was accomplished by adjusting the Windows Firewall settings on PLABWIN10. In in third task, I tested the effectiveness of the firewall rule by attempting to ping PLABWIN10 from PLABDC01, resulting in timed-out requests as expected. Finally, Task 4 involved reconfiguring the firewall rule to allow incoming ping packets, and after verifying the changes, I successfully received responses from PLABWIN10, confirming the reconfigured firewall rule's effectiveness | Fig 1. Ping command to test the connectivity |
| Exercise 2 - Configuring Firewall Rules using Windows Firewall with Advanced Security | |
| Like the previous exercise I started this exercise with verifying the live connection between two computers using PING. Then, I created a firewall rule to block the ping response from PLABDC01 by configuring the Windows Firewall with Advanced Security settings on PLABDC01. Then I tested the effectiveness of the firewall rule by attempting to ping PLABDC01 from PLABWIN10, resulting in timed-out requests as expected due to the blocked rule. Finally, I did the reconfiguring of the firewall rule to allow incoming ping packets, and as result successfully enabling PLABDC01 to respond to ping requests. | Fig 2. Implementing advanced security feature: in this case blocking the ICMP Echo request from PLABC01 |
| Exercise 3 - Configuring Firewall Rules using Remote Desktop | |
| This exercise started with setting up a remote session with PLABWIN10 accessing PLABDC01's desktop. In task 2 I created a firewall rule to block ping responses from PLABDC01, achieved through the remote desktop connection. After that I verified the effectiveness of the firewall rule by testing ping requests, confirming the blocked connection. Finally, I practiced reconfiguring the firewall rule using the remote desktop connection, allowing ping responses from PLABDC01. | Fig 3. Enabling the firewall rule using remote desktop connection |
| Exercise 4 - Configuring Firewall Rules from the Command Line Interface | |
| In this exercise, I did a lot of things to manipulate the firewall setting on PLABDC01 using Command Line Interface (CLI) commands. First, I checked that it was stopping ping requests, which it was. After that, I tried pinging again, and this time it worked because the firewall was off. Next, I used more commands to turn the firewall back on, and when I tried pinging again, it didn't work because the firewall was blocking it again. | Fig 4. Disabling the firewall rul for all the profiles using CLI |
| Learning Reflection | |
| Each exercises in this lab taught me different ways to handle firewall settings, starting from basic setups to more advanced methods. I learned how to check active connections, create and adjust firewall rules, test their effectiveness, and change them as needed. It also showed me the flexibility and effectiveness of firewall management in safeguarding computer systems. Overall, these exercises boosted my confidence in managing network security through firewall configuration. | |

# 

# Lab work III

## Lab 1: Configure Verify and Troubleshoot Port Security

This lab focused on configuring static and dynamic port security with additional port security configuration settings and finally configure Err-disable revovery.

### Walkthrough of the lab exercise:

|  |  |
| --- | --- |
| Exercise 1 - Configuring Static and Dynamic Port Security | |
| In this exercise, I configured static port security on a Cisco switch using the NYCORE1 device. By examining the MAC address table and setting up the switchport to only allow a specific MAC address (NYEDGE1 router), I learned how to restrict unauthorized devices from accessing the network.  In the next task, I performed dynamic port security and its configuration on the NYCORE1 switch. By removing the statically assigned MAC address and enabling dynamic port security, I observed how the switch autonomously learns and secures MAC addresses in real-time. | Fig 1. MAC address table of NYCORE1 switch    Fig 2. Enabling port security on Fast Ethernet interface using mac-address |
| Exercise 2: Configure Additional Port Security Configuration Settings | |
| In this session, I explored advanced port security setups on NYCORE2 to strengthen network protection. By using commands like "switchport port-security mac-address sticky" and "switchport port-security maximum," I ensured only authorized devices could access the port. Additionally, I implemented the "switchport port-security violation shutdown" command to automatically shut down the port upon any security breaches.  Next, I simulated a security breach scenario by changing the allowed MAC address on the port. Following the steps to clear the sticky MAC address and set a new one triggered a security violation. As a result, the port was promptly shut down, showcasing the effectiveness of the configured violation action. Through the verification process, I confirmed the port's status as "Secure-shutdown," affirming the successful implementation of port security measures. | Fig 3. Configuring fastEhthernet 12 interface on NYCORE2 with port security enableled and verifying the status on the same interface    Fig 4. Simlulating a violation and the verifying the status of port: secure-shutdown |
| Exercise 3 - Configuring Err-disable Recovery | |
| In Task 1, I configured err-disable recovery on NYCORE2 to automatically recover from port security violations. I examined the current configuration and enabled recovery for **psecure-violation**, setting the recovery interval to 30 seconds.  Moving on to Task 2, I verified the err-disable recovery setup by reviewing the settings and observed the interface status and recovery countdown. The verification confirmed the successful activation of err-disable recovery for port security violations. Additionally, I monitored syslog messages indicating the interface's reactivation and power reception, followed by another violation leading to the port re-entering the err-disabled state. | Fig 5. list of reason for err-diabled state of router |
| Learning Reflection | |
| Throughout these exercises, I gained hands-on experience in configuring various aspects of port security on Cisco switches, ranging from static and dynamic port security to advanced configuration settings and err-disable recovery. It was interesting to learn about the measures that can be implied with router to harden the network security. I did the exercise as instructed, however I have to accept if I understand them fully. | |

## Lab 2: Scanning and Remediating Vulnerabilities with OpenVAS

The lab included the exercises related to OpenVAS Scanning, LDAP and Validating Security Changes with OpenVAS.

### Walkthrough of the lab exercise:

|  |  |
| --- | --- |
| Exercise 1: OpenVAS scanning | |
| In Task 1, I started OpenVAS on Kali Linux using the PLABKALI01 device. By accessing OpenVAS through the Applications menu and initiating the service, I gained practical experience in setting up this vulnerability assessment tool.  Moving on to Task 2, I explored the configuration options within OpenVAS and reviewed different scan configurations. By examining port lists and scan configurations, I gained insights into the scanning techniques employed by OpenVAS and the significance of selecting appropriate configurations based on network requirements.  Finally, in Task 3, I performed an actual vulnerability scan using OpenVAS, targeting the PLABDC01 device. By initiating the scan and monitoring its progress through the Greenbone dashboard, I witnessed the tool in action, detecting and reporting vulnerabilities within the target environment. | Fig 1. Creating a task specifying the scan config and target host    Fig 2. Scan result from PLABCD01. |
| Exercise 2 - Applying Windows Secure Updates | |
| In Task 1, I initiated the process of applying a crucial Windows update on the PLABDC01 device to address identified vulnerabilities from previous exercise. Using Internet Explorer, I accessed the Intranet page and located the specific update file, within the Tools section.  Moving on to Task 2, I proceeded to install the update by opening the selected file and confirming the action through the Windows Update Standalone Installer prompt. Throughout the installation process, I diligently monitored its progress until completion. After the installation was finished, I initiated a restart of PLABDC01 to finalize the update process. | Fig. 3. Detail report of the found vulnerability  A yellow rectangle with black text  Description automatically generated  Fig 2. Installing the patch for the previous found vulnerability |
| Exercise 3 - Validating Security Changes with OpenVAS | |
| After applying the patches to the previously identified vulnerability, I initiated a rescan of PLABDC01 with OpenVAS to confirm the effectiveness of recent security changes. The main objective of this rescan was to validate whether the previously identified vulnerability had been successfully addressed following the installation of the patch. As expected, the vulnerability has been removed, as evidenced by the screenshot on the right. | Fig 1. Screenshot shows the vulnerability has been eliminated after rescanning post applying the pathces |
| Learning Reflection | |
| Throughout this exercise, I gained valuable insights into OpenVAS scanning using Greenbone. I learned how to utilize Greenbone to initiate scans and analyze the results effectively. Additionally, I acquired knowledge about patching and its implementation to address identified vulnerabilities. Overall, this exercise provided a comprehensive understanding of vulnerability assessment, analysis of scan results, and the implementation of security measures based on the findings. | |

## Lab 3: Analyze network traffic with Wireshark

This lab dealt with Wireshark and its attributes such as download and install, capturing packets, packets analysis, verifying output logs and perform packet jumping.

### Walkthrough of the lab exercise:

|  |  |
| --- | --- |
| Exercise 1: Download and Install Wireshark | |
| The first and second tasks of the exercise dealt with navigating to the Wireshark web application and downloading the Wireshark application.  Upon completion of installation, the next task was to launch the program, apparently there are two modes of launching Wireshark, i.e. with limited privileges and with administrative privileges.  Then I practice launching Wireshark with the administrative privileges. However, I did not proceed any further, rather I launched the legacy version of Wireshark as per the instructions.  I became aware that many network administrators prefer to use the legacy version even though there is newer UI. | Fig 1. Wireshark installation walkthrough |
| Exercise 2: Capture Packets with Wireshark | |
| The main objective of this task was to analyze the network ARP traffic when one system in the network is rebooting. To achieve this, I launched Wireshark in administrative privilege mode selecting Ethernet 2 as an interface to capture with a domain computer “PLABWIN10” and observe the traffic when other domain computer “PLABWIN801” was in booting state. The key observation for me was that how all systems in network responds to “who has <Ip address>? ” | Fig 2. Wireshark capture with ARP filter |
| Exercise 3 & 4 - Perform Packet Analysis | |
| In this task, I explored Wireshark's interface and options to analyze captured traffic flows comprehensively. I learned to interpret various details listed under different tabs, such as Protocol, and applied basic filtering techniques to refine the displayed information. By examining ARP sessions, I identified devices communicating and confirming their addresses. Additionally, I observed color-coded traffic types and filtered specific protocols like DNS and HTTP to focus on relevant data. This exercise equipped me with essential skills to navigate Wireshark effectively, interpret network protocols, and analyze traffic patterns for troubleshooting and security purposes. | Fig 3. Using filter: screenshot showing the TCP stream of http filter |
| Exercise 5 & 6 - Perform Packet Jumping | |
| Exercise 5 was how to save the packet to a file, which was quite straightforward. Moving on to exercise 6, I learned how to perform packet jumping in Wireshark, which allows for seamless navigation between related packets within a capture. By applying a filter to display ICMP packets, I identified ping requests and their corresponding replies. Using Wireshark's packet details pane, I traced the link between request and response frames, enabling efficient analysis of network traffic. Through double-clicking on referenced frames, I practiced jumping between related packets, gaining insights into their correlation. | Fig 4. Observing the frame showing the link between frame 914 and 915-one being ping request and other being response |
| Learning Reflection The Wireshark lab exercises were really helpful for learning about network packet analysis. I got to know how to use Wireshark's interface and what permissions are needed to run it. By capturing packets, I could study ARP traffic when devices restart, which gave me insights into how they communicate. Exploring Wireshark's features and filters helped me understand network protocols and how to analyze traffic better.. Overall, these exercises taught me crucial skills in analyzing network packets, which will help me improve network performance and security. | |

## Lab 4: Configure and Verify IPv4 and IPv6 Access Lists for Traffic Filtering

This lab focused on configuring and Verifying Access Lists for Traffic Filtering implementing both IPV4 and IPV6.

### Walkthrough of the lab exercise:

|  |  |
| --- | --- |
| Exercise 1 - Configuring Standard and Extended Access Lists using IPv4 | |
| In the first task, I learned about access control lists (ACLs) and their significance in controlling traffic flows through a network. Specifically, I focused on creating standard access lists in IPv4 on the NYEDGE1 router.  By following the provided steps, I configured a standard access list to match a single device on the network, the PLABCSCO01 server, and denied traffic from it. Additionally, I learned about the implicit deny at the end of each access list, highlighting the importance of including a permit statement. Finally, I applied the access list to a specific interface, considering the directionality of the traffic flow.  In the second task, I learned how to create an extended access list, which gives more detailed control over network traffic compared to standard access lists. I configured access-list 110 on the NYEDGE1 router to permit FTP traffic but deny ICMP traffic. To achieve this, I followed a series of steps. First, I permitted TCP traffic for FTP using the "permit" command and specified the FTP ports. Then, I denied ICMP traffic to a specific IP address using the "deny" command. After configuring the access list, I applied it to interface GigabitEthernet 0/0 inbound.  Next, I tested the configuration by attempting to ping a website and use FTP from the PLABCSCO01 server. The ping failed, indicating that ICMP traffic was blocked, while the FTP connection was successful.  Finally, I reviewed the access list hits to confirm the matches on the permit and deny lines. | Fig 1. Access list configured on the router.    Fig 2. Access-list configuration permitting ftp and denying ICMP.    Fig 3. Verifying the access list - ICMP blocked whereas ftp allowed |
| Exercise 2 - Configuring Named Access Lists Using IPv4 | |
| In this task, I learned how to create a named extended access list called permit-ftp on the NYEDGE1 router. Through this process, I gained an understanding of the benefits of named access lists, such as their flexibility in configuration and management. By permitting FTP traffic from a specific source to a destination while blocking ICMP traffic, I practiced applying granular control over network traffic. This hands-on experience helped me grasp the practical aspects of access list configuration and its significance in network security and traffic management. | Fig. 3. Creating a named extended access list which allow ftp and deny icmp |
| Exercise 3 - Creating Access-Lists in IPv6 | |
| In this task, I learned how to set up IPv6 access lists on the NYEDGE1 router. First, I checked if the routers could communicate using IPv6. Then, I created an access list named block\_NYEDGE2 on NYEDGE1, specifying both source and destination addresses. After applying it to the outbound traffic on GigabitEthernet 0/1, I tested it by trying to ping NYEDGE2, which failed as expected. Looking at syslog messages helped me understand how the access list worked. I also added a rule to block traffic to an entire subnet. Finally, I made sure the access list was working correctly. | Fig 1. Defining the ipv6 access list and binding it to the interface |
| Learning Reflection | |
| These tasks helped me learn how to configure and check ACLs, which are essential for securing networks and managing traffic well. By creating standard and extended access lists, and named access lists, I got a better understanding of how ACLs work to control data flow based on specific rules. This hands-on practice not only improved my technical skills but also helped me grasp network security concepts like access control and policy enforcement. Moreover, I learned the importance of thorough testing to ensure ACL configurations work effectively in real-world networks. | |

# 

# Lab work IV

## Lab 1: Configuring Endpoint

This lab includes various exercises related to endpoint protection and security measures.

### Walkthrough of the lab exercise:

|  |  |
| --- | --- |
| Exercise 1 - Use Anti-virus Programs | |
| This task began with initiating a scan using Windows Defender, which identified some outdated configurations on the system. To address this, an update configuration was completed using the same Windows Defender application. Additionally, a custom scan was performed, with an exclusion added in Defender. I discovered Microsoft's online tool for scanning viruses, malware, and spyware, finding it straightforward to use.  However, I encountered difficulties completing this part of the task due to issues with Internet Explorer, which prevented me from following the instructions properly. Despite attempting to replicate the instructions using Mozilla, it was not an exact match. Nonetheless, I understood that the task aimed to implement additional security features such as extensions, popup blocking, and private browsing. | Fig 1. Running a scan for virus using defender    Fig. 2 Downloading Microsoft safety scanner online |
| Exercise 2 - Use Anti Spyware | |
| I was able to download the anti-spyware software from the internet, but I could run it in the system. The installation wizard never showed up, hence could not continue any further. | Fig 3. Trying to install superAntispyware free edition |
| Exercise 3 - Installing an Endpoint Protection Site System Role | |
| In Task 1, I installed the Endpoint Protection site system role on the PLABSYS01 server by following step-by-step instructions in the System Center 2012 R2 Configuration Manager console. This gave me hands-on experience in setting up Endpoint Protection in a network environment. In Task 2, I configured alerts for Endpoint Protection by adding new alert thresholds for malware detection and repeated detection patterns in the All-Systems Properties dialog box. This helped me learn how to set up alerts to notify administrators of important events in the Endpoint Protection system. | Fig 4. completion page of the Add Site System Roles Wizard in System Center 2012 R2 Configuration Manager. |
| Exercise 4 - Configure Definition Updates for Endpoint Protection | |
| In this exercise, I configured definition updates for Endpoint Protection using the System Center 2012 R2 Configuration Manager console. I created an automatic deployment rule named "ADR Endpoint Protection" and specified the target collection as All Systems. I set up schedules for evaluation and deployment, configured user experience options, specified alert settings and software update download behavior. Finally, I reviewed the settings and completed the wizard. This task provided hands-on experience in efficiently managing definition updates to ensure the security of client computers against malware threats. | Fig 5. Automatic deployment rule in configuration manager |
| Exercise 5 - Create and Deploy Antimalware policies for Endpoint Protection | |
| In this exercise, I created and deployed an antimalware policy for Endpoint Protection using the System Center 2012 R2 Configuration Manager console. The process involved configuring various settings such as scheduled scans, scan settings, default actions, real-time protection, exclusion settings, advanced options, threat overrides, Microsoft Active Protection Service settings, and definition updates. Once the policy was created, I deployed it to all systems by selecting the appropriate collection. This task provided practical experience in managing antimalware policies centrally to ensure consistent protection against malware and other threats across the network. | Fig 6. Created Antimalware policy in system center 2012 configuration manager |
| Exercise 6 - Configure Custom Client Settings for Endpoint Protection | |
| In this task, I set up client settings for Endpoint Protection using the System Center 2012 R2 Configuration Manager console. I created custom settings specifically for Endpoint Protection by accessing Client Settings in the Administration workspace. I named the new setting "Client Settings for Endpoint" and configured it to manage the Endpoint Protection client on client computers. Once created, I deployed these settings to the All Systems collection to ensure consistent configuration across all client devices. I verified the deployment by checking the Deployments tab. | Fig 7. Creating custom client device for Endpoint protection |
| Exercise 7 - Provision Endpoint Protection | |
| In the first task, the focus was on setting up the Endpoint Protection client on a reference computer, PLABSYS01. The steps involved navigating to the Client folder on the System Center 2012 Configuration Manager installation media, and then silently installing the Endpoint Protection client using the scepinstall.exe /s command.  Moving on to the second task, the objective was to verify the successful installation of the Endpoint Protection client on the reference computer. This was done by checking the System Center Endpoint Protection console, accessed by clicking on the PC status: Protected icon in the Windows notification area. The confirmation of successful installation was observed through the notification "For your protection, some settings are managed by your security administrator" displayed on the Settings tab within the Endpoint Protection console. | Fig 8 . Endpoint protection client installation completed    Fig 9. Verify the endpoint protection client installation |
| Learning Reflection | |
| In these exercises, I learned to use different security tools to improve network security and prevent malware attacks. Tasks included setting up anti-virus and anti-spyware programs, as well as configuring Endpoint Protection roles and policies. Despite some challenges, like Internet Explorer issues during scans, I realized the importance of strong security measures and proactive management. These exercises enhanced my technical skills and emphasized the need for continuous monitoring and configuration management for network security. | |

## 

## Lab 2: Encryption and Hashing

This lab focused on different encryption techniques and calculating the hashes.

### Walkthrough of the lab exercise:

|  |  |
| --- | --- |
| Exercise 1 - Understand cryptographic basics | |
| The first task in this exercise was to install a called CryptoDemo. The installation file was in intranet/tools. It was straightforward to download and install the application.  After the installation, it was time to test the application. I chose 3DES encryption from the options and entered "Practice-Labs.com" in the Key box. Then, I typed the message "Welcome to Practice Labs, we hope you are really enjoying the course!!" in the Data box and clicked Encrypt. Each time I encrypted, I got a different result, showing how encryption changes data.  Lastly, I tried to implement the encryption algorithm using random key and time key as key and observed the encrypted data simultaneously. I tried a different hashing algorithm i.e. md5 and sha1 in this task and observed that the hash value remains the same for the given text. | Fig 1. Installing cryptoDemo    Fig 2. Encrypting with random key |
| Exercise 2 - Comparing Hashing Algorithms | |
| The task was to install a HashCalc application from E drive. It was easy to install.  I used HashCalc to observe and compare the hash value against a provided string text, HEX strings, and a file data format. Also tried other features such as HMAC during the exercise. | Fig. 3. Comparing hashvalue using HashCalc |
| **Exercise 3 - Comparing Hash Values** | |
| In this task, I checked if the Putty program was genuine by comparing its hash values from the official website with those generated using HashCalc. I visited the Putty website in Internet Explorer and found the hash values for version 0.67. I specifically looked for the MD5 and SHA-1 hash values for the putty.exe file and compared them with HashCalc's results to confirm the program's integrity. Unfortunately, I couldn't complete the task to compare hashes using the website "md5online.com" as I couldn't find it. | Fig 4. Comparing a md5 hashvalue with the checksum from a webpage |
| Learning Reflection | |
| Through these tasks I built upon my existing knowldege of hashes. I have used the tools previously so it went easy with the exercises. The key takeaway for me from this exercise was that I became aware HashCalc could generate the hash of the files also. Moreover, I became familier with the presence of online tools. Amidts the complicated other labs, this was one really easy for me. | |

## Lab 3: Performing security assessment using various tools.

This lab focused on different encryption techniques and calculating the hashes.

### Walkthrough of the lab exercise:

|  |  |
| --- | --- |
| Exercise 1 Identifying Live Systems | |
| The first task was to boot up Kali, after successfully booting up, I perfomed nmap and hping3 to identify the live hosts in a network. Everything went as planned. | Fig 1. Nmap scan result |
| Exercise 2 - Performing Zone Transfers | |
| I used Nslookup to perform a zone transfer in this task.I specified the DNS server's IP address with the server 192.168.0.1 command. Next, I set the query type to retrieve all records from the server using set type=any. Finally, I attempted to transfer the zone practise-labs.com to PLABWIN10 with the ls -d practise-labs.com command. However, since the DNS server used in the demonstration isn't vulnerable to zone transfers and the domain practise-labs.com isn't available in the environment, no successful output was displayed.  To perform a zone transfer using the DIG tool, using kali terminal, I entered the command dig axfr practise-labs.com 192.168.0.1 at the command prompt to request the complete listing of domain records from the DNS server. However, since the server used isn't vulnerable to zone transfers and the domain practise-labs.com isn't available in the environment, no successful output was displayed. | Fig. 2. Nslookup to perfrom zone transfer    Fig. 3. Using DIG to perfrom zone transfer |
| Exercise 3 - Working with Remote Targets | |
| I copied PsInfo.exe from the PsTools folder to the desktop on PLABWIN801 and ran Psinfo.exe \192.168.0.1 -h -d in Command Prompt to get system details from PLABDC01, showing installed hotfixes and disk volume information. PsInfo then displayed PLABDC01's system details. | Fig 4. Psinfo displaying the remote system information |
| Exercise 4 - Working with Finger Command | |
| In this exercise, I utilized the finger command in Linux to gather details about system users. Following the task instructions, I executed the command "finger -s root" at the prompt, which provided information about all the users present on the system. | Fig 5. Using the finger command |
| Exercise 5 - Social Engineering Reconnaissance | |
| In this exercise, I did social engineering recon on an individual called Philip Nomad using an online site “mybook.com”. In this site, he has shared many personal pictures and key information which an attacker could use to build a profile on him. All in all, it was a good demonstration on how profiling is done based on social media information. | Fig 6. Social Engineering profiling |
| Exercise 6 - Use the Social Engineering Toolkit (SET) in Kali Linux | |
| This exercise was very interesting as I implemented SET to attempt gaining access to a victim system using a reverse shell. Firstly, I downloaded the payload onto the victim system and initiated the FTP connection from a remote computer to execute the payload. Once the payload was executed, I utilized msfconsole to gain reverse shell access to the victim's system. | Fig 7. Revershell    Fig 8. Interacting with compromised system |
| Learning Reflection | |
| In this lab, I engaged in network reconnaissance exercises, including identifying live hosts, performing zone transfers, and working with remote targets. Additionally, I explored social engineering techniques through online profiling. These activities provided practical insights into network security assessment and highlighted the importance of understanding vulnerabilities and potential attack vectors. The most interesting part of learning was, how to remotely gain access to the compromised target. | |

# Conclusion

Overall, I found the series of lab exercises to be highly impressive. The gradual increase in complexity challenged me to not only grasp the concepts but also understand the workflow effectively. While the instructions provided by Precipio were generally good, I encountered some labs where additional background information would have been beneficial. Often, I had to refer to external resources to fully understand the topic, its purpose, and its security implications. Nevertheless, navigating through the advanced-level course was a rewarding experience, and I am pleased with the new concepts and skills I acquired. These learnings will undoubtedly help me in executing security-related tasks in the future.

I extend my gratitude to the instructors for offering these valuable learning materials. Despite occasional technical difficulties, which I have grown accustomed to over my two-year experience with Precipio, I eagerly anticipate the next set of labs.