Lab 2 - **Offensive Security Operation Lab Setup**

Version 1.0

Table of Content

1. Executive Summary

This lab focused on building a secure, isolated malware analysis environment using FLARE-VM—a popular Windows-based reverse engineering and malware analysis suite developed by Mandiant. The objective was to configure a Windows 10 virtual machine (VM) within VirtualBox, harden the environment by disabling built-in security features, and deploy FLARE-VM via PowerShell for future offensive cyber operations labs. The hands-on process involved setting up system prerequisites, resolving EFI firmware errors, disabling Tamper Protection and Microsoft Defender, and managing VM snapshots for recovery and persistence.

1.1. Lab Objectives

This lab was designed to simulate a real-world malware analyst's preparation workflow. The primary objectives were:

- To prepare a fully functioning isolated and controlled virtual environment for malware analysis and offensive operations.
- To install and configure FLARE-VM on a Windows 10 machine for use in reverse engineering, threat hunting, and binary analysis tasks.
- To practice disabling Microsoft Defender and Windows security protections via Group Policy and manual GUI configuration.

- To take and manage VM snapshots to support safe experimentation and quick recovery during volatile installations.

2. Methodology

The following step-by-step process was used to configure a safe and functional malware analysis environment using FLARE-VM within a VirtualBox-hosted Windows 10 virtual machine (VM). The methodology strictly followed best practices in malware lab isolation and operational readiness for offensive cyber operations.

2.1. Tools & Resources

The following tools, software packages, and resources were used in the lab setup:
Virtualization & Operating Systems

- VirtualBox — Primary hypervisor for hosting the Windows VM
- Windows 10 ISO — Official Microsoft evaluation version used for guest OS
- Guest Additions for VirtualBox — For enabling clipboard, drag-and-drop, and screen resizing
- VM Snapshots — Used throughout to ensure recoverability during risky installations

FLARE-VM & Supporting Toolkits

- FLARE-VM (FireEye Labs Advanced Reverse Engineering Virtual Machine)
- PowerShell (Administrator mode) — Used to install FLARE-VM via Boxstarter and Chocolatey
- Chocolatey Package Manager — Required to automate tool installation
- Boxstarter — Used to script the FLARE-VM environment bootstrap

System Configuration & Security Tools

- Group Policy Editor (gpedit.msc) — Used to disable Defender and Windows Update (partially)
- Windows Security GUI — Manually used to disable Real-time Protection and Tamper Protection
- Task Manager and msconfig — Used to monitor startup apps and confirm system state

Network Resources

- Wired/Wi-Fi connection — Required for downloading FLARE-VM packages (may exceed 3–5 GB)
- Snapshots taken before and after major operations to ensure rollback capability

2.2. Install VirtualBox

- The VirtualBox installer was downloaded from the official VirtualBox website (https://www.virtualbox.org) by selecting "Windows hosts."
- The installation file was located in the "Downloads" folder and executed to initiate the setup process.
- The installation wizard was launched, and the "Next" option was selected on the welcome screen.
- The default installation options and paths were reviewed, and the "Next" option was chosen to proceed.
- Additional features, such as creating Start Menu entries, were selected before continuing with the setup.
- The installation settings were confirmed, and the "Install" button was clicked to begin the process.
- User Account Control (UAC) prompts were addressed by selecting "Yes" to authorize the installation.
- The installation process was completed successfully, and the "Finish" button was clicked to exit the installer.
- VirtualBox was launched from the Start menu to confirm successful installation.

2.3. Download and Install Windows 10 ISO on VirtualBox

- The Windows 10 ISO file was downloaded from the official Microsoft website to ensure authenticity and reliability. VirtualBox was launched, and a new virtual machine named "Win10-FLARE" was created.
    - The operating system type was set as "Microsoft Windows" with "Windows 10 (64-bit)" as the version.
    - Memory allocation was configured to 4096 MB (4 GB) for optimal performance.
    - A virtual hard disk was created in the VDI format with dynamically allocated storage and set to 80 GB.
- Initially, the virtual machine failed to boot from the ISO file, displaying an "EFI Network" error. To resolve this, the `Windows10.vmx` file was manually edited by changing the line: firmware = "efi" to firmware = "bios". The virtual machine was restarted, and the Windows 10 installer loaded successfully.
- The Windows 10 operating system installation was completed, and initial setup tasks such as creating a user account and configuring basic preferences were performed.
- For initial system optimization, VirtualBox Guest Additions were installed. This step enhanced performance and enabled better integration, including features like seamless mouse movement, screen resolution adjustment, and clipboard sharing.
- Shared folders were configured to enable data exchange between the host and guest systems. This was achieved by:
    - Navigating to the VirtualBox settings for the "Win10-FLARE" virtual machine.
    - Selecting the "Shared Folders" option and adding a shared folder from the host system.
    - Enabling "Auto-mount" and setting the folder as "Read/Write" for seamless access.
- Final tests were performed to verify the functionality of the virtual machine, ensuring that the guest operating system operated correctly and shared folders functioned as intended.
- A snapshot of the VM was taken at this stage

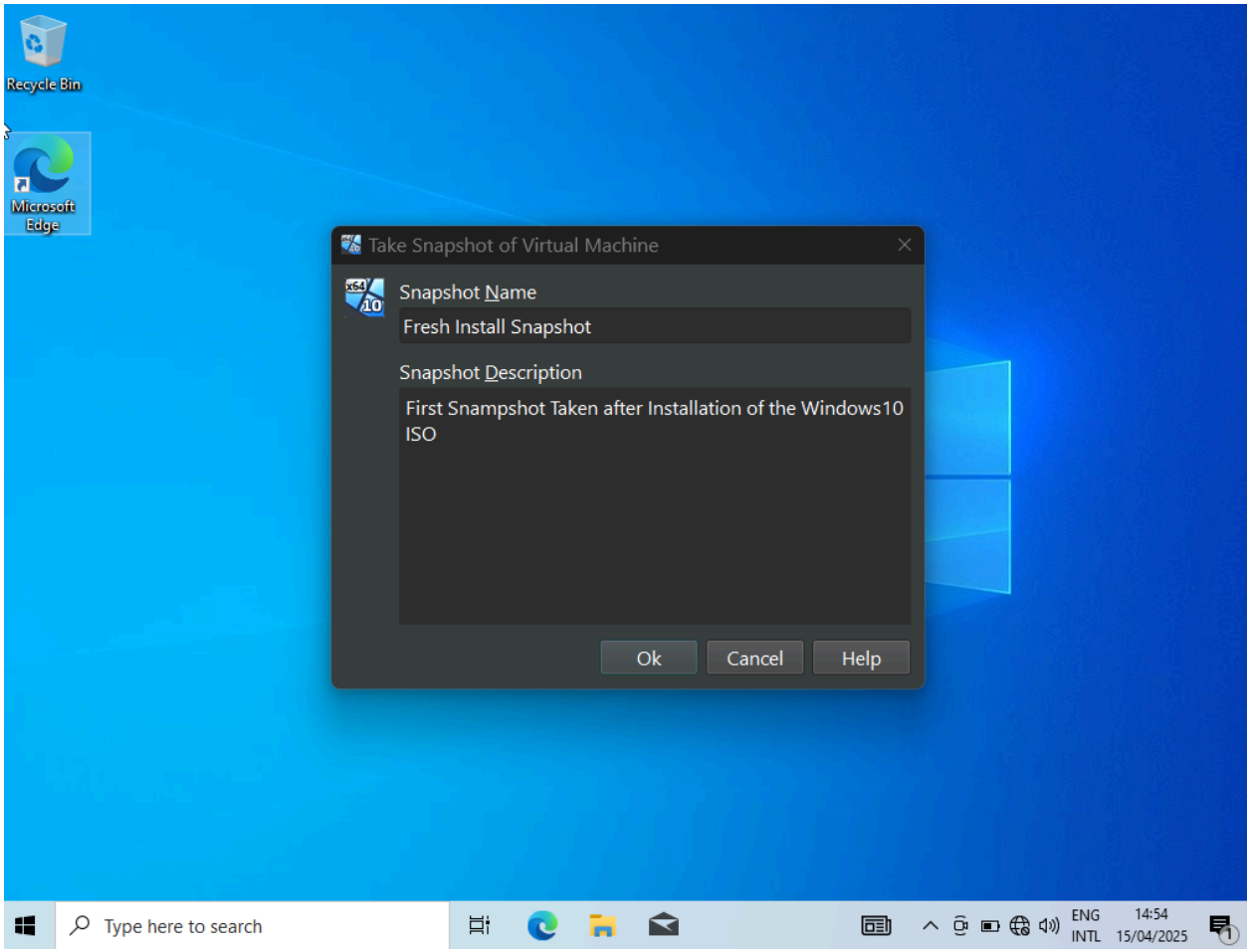

Figure 1: Snapshot Created – “Post-Security-Config”

2.4. Disable Security Protections

- The Group Policy Editor (gpedit.msc) was launched to initiate the process of disabling Windows Defender and associated features. Resource Monitor (resmon.exe) was opened to further disable security processes. Steps taken:
    - Navigated to "Overview" in Resource Monitor.
    - Located "MsMpEng.exe" in the process list.
    - Right-clicked the process and selected "Suspend Process" to halt its operation.
- Real-Time Protection and Tamper Protection were manually disabled through the following path: Settings → Windows Security → Virus & Threat Protection → Manage Settings

- Automatic Windows Updates were disabled via Group Policy settings: Windows Update → Configure Automatic Updates → set to "Disabled"
- A virtual machine snapshot titled "Post-Security-Config" was created. This snapshot preserved the VM's clean state and offered the capability to roll back in case of failed or incomplete installations during the FLARE-VM setup.

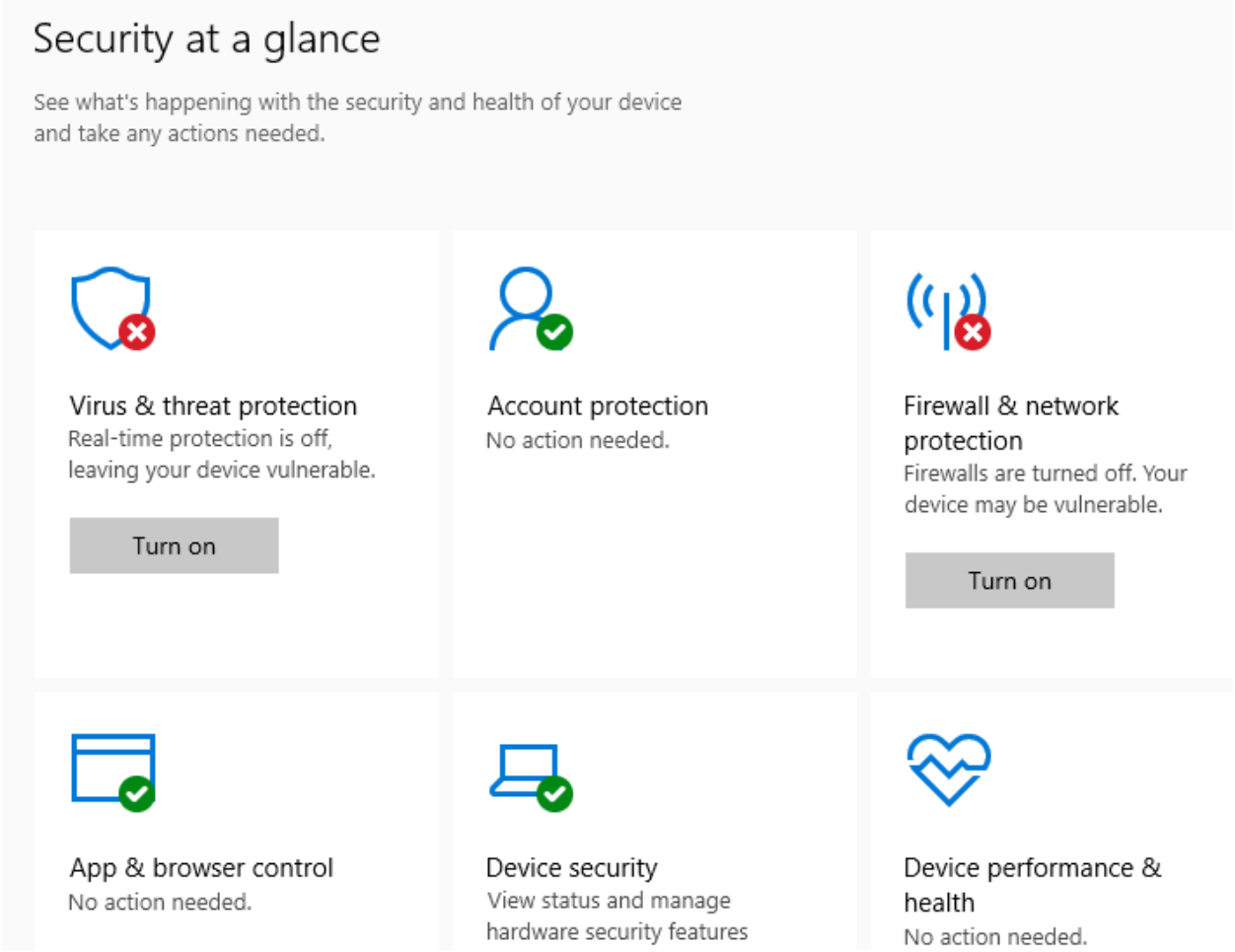

Figure 2: Windows-Defenders-Disabled

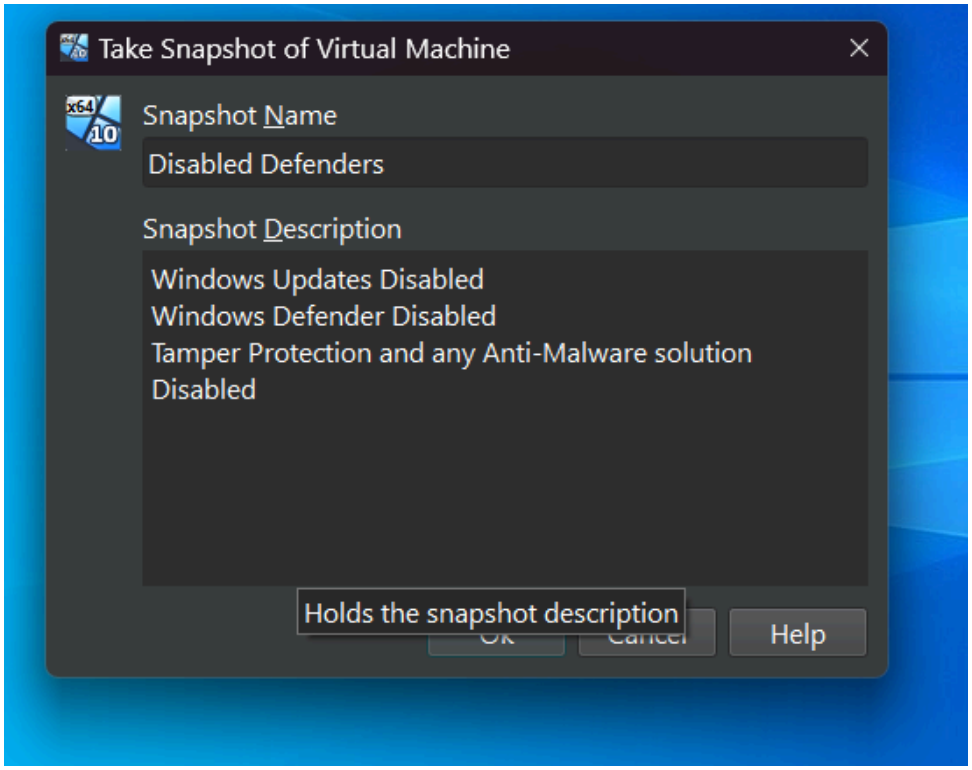

Figure 3: Snapshot Created – “Post-Disabled-Defenders”

2.5. Disable Security Protections

- A folder named "FlareVM" was created in the Desktop directory to organize the installation process.
- PowerShell was launched with administrative privileges from within the "FlareVM" folder.
- The installation script `installer.ps1` was downloaded to the Desktop using the command: (New-Object net.webclient).DownloadFile('https://raw.githubusercontent.com/mandiant/flare-vm/main/install.ps1',"$([Environment]::GetFolderPath('Desktop'))\install.ps1")
- The script was unblocked to allow execution by running: Unblock-File .\install.ps1
- The execution policy was set to unrestricted to enable the script to run successfully using: Set-ExecutionPolicy Unrestricted -Force
- The installation script was executed in CLI-only mode with minimal user interaction using the command: .\install.ps1 -password <password> -noWait -noGui
- Upon completing the installation, the network configuration was switched from “Bridged Network” to “host-only” mode to enhance security and isolate the virtual machine.
- A VM snapshot was taken, titled "Post-FlareVM-Config," to capture the system state and allow rollback in case of any errors or incomplete installations.

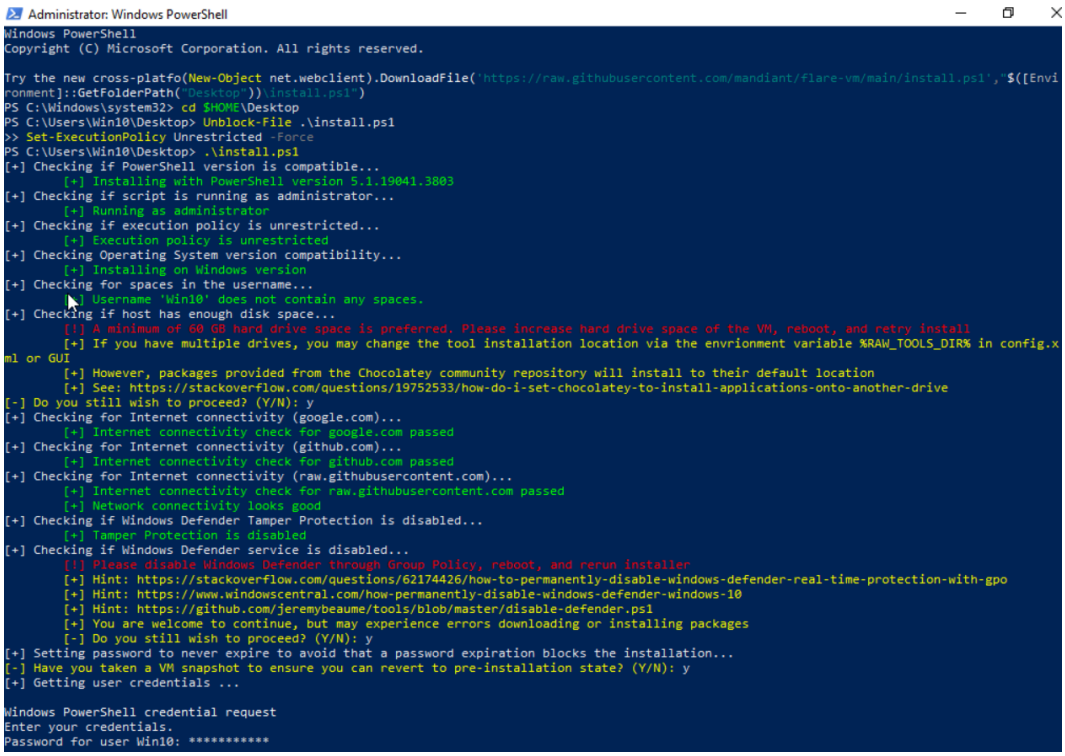

Figure 4: PowerShell terminal during automated package installation

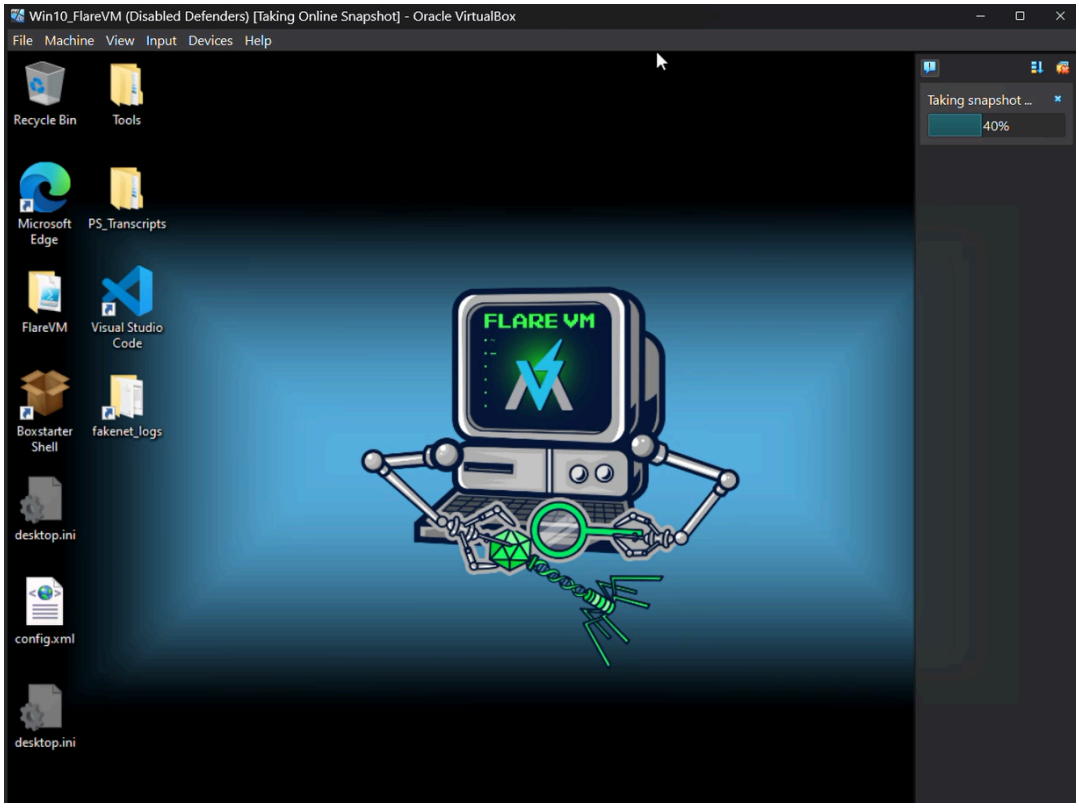

Figure 5: Post FLARE-VM Installation Snapshot

2.6. Challenges

2.6.1: EFI Network Boot Error

During the initial boot of the virtual machine, an “EFI Network” boot error prevented the system from recognizing the Windows 10 ISO. This error occurred due to an incompatible firmware setting in the virtual machine configuration. The issue was resolved by manually editing the .vmx file of the virtual machine. The line: ‣ firmware = "efi" was changed to: ‣ firmware = "bios". After saving the changes and restarting the virtual machine, the ISO booted successfully, allowing the Windows installation to proceed.

2.6.2: Group Policy Failed to Disable

While disabling Tamper Protection and any Anti-Malware solution (e.g., Windows Defender) Windows Defender
disabled, preferably via Group PolicyThe challenge of disabling Defender protections via Group Policy exposed real-world limitations in relying solely on policy-based controls. It demonstrated the necessity of validating configurations via GUI and process monitoring tools such as Resource Monitor and Task Manager.

2.6.3: Interrupted FLARE-VM Installation Due to Network Failure

During the FLARE-VM setup, an unstable network connection caused the installation process to fail midway. This resulted in partially installed components and dependency issues. To recover, the virtual machine was rolled back to a snapshot taken before the installation began (“Post-Security-Config”). Once restored, the FLARE-VM installation was restarted using PowerShell. This time, with a stable connection, the installation completed successfully.

3. Conclusion

The successful deployment of a FLARE-VM environment within a VirtualBox-hosted Windows 10 virtual machine demonstrates the feasibility and effectiveness of establishing a secure, isolated platform for malware analysis and reverse engineering. The process involved meticulous configuration of virtualization settings, strategic disabling of native security features, and the installation of specialized tools tailored for cybersecurity operations.

Key takeaways from this setup include:

- Utilizing VirtualBox ensures that malware samples can be analyzed without risking the host system, providing a controlled environment for testing and observation.
- FLARE-VM offers a comprehensive suite of tools essential for malware analysis, streamlining the setup process and ensuring consistency across analysis environments.

- Disabling Windows Defender and automatic updates is crucial to prevent interference during malware analysis. This requires a combination of Group Policy adjustments and manual configurations.
- Regularly creating snapshots at critical stages allows for quick recovery in case of errors or system instability, enhancing the resilience of the analysis environment.

This lab setup serves as a foundational model for cybersecurity professionals seeking to establish their own malware analysis environments, emphasizing the importance of preparation, configuration, and safety.

3.1. Recommendations

Based on the lab setup experience, the following recommendations are proposed:

1. Implement a routine of taking **snapshots** before major changes or installations to facilitate easy rollback in case of issues.
2. After setting up and updating the environment, **switch the network mode to 'Host-Only'** to prevent the virtual machine from accessing the internet, reducing the risk of malware communicating with external servers.
3. Maintain a detailed **documentation** of configuration records, tool installations, and changes made to the system. This aids in troubleshooting and ensures reproducibility.
4. Ensure that all activities conducted within the analysis environment comply with relevant laws and organizational policies.

4. References

Mandiant. (n.d.). *FLARE-VM*. GitHub. Retrieved from https://github.com/mandiant/flare-vm

Microsoft. (n.d.). *Download Windows 10 Disc Image (ISO File)*. Retrieved from https://www.microsoft.com/en-us/software-download/windows10ISO

Super User. (2023). *How to disable Windows 10 update + defender completely?*. Retrieved from https://superuser.com/questions/1824712/how-to-disable-windows-10-update-defender-completely

Make Tech Easier. (n.d.). *How to Permanently Disable Windows Defender in Windows 10*. Retrieved from https://www.maketecheasier.com/permanently-disable-windows-defender-windows-10/

Windows Central. (n.d.). *How to stop updates installing automatically on Windows 10*. Retrieved from https://www.windowscentral.com/how-stop-updates-installing-automatically-windows-10

Stack Overflow. (n.d.). *How to permanently disable Windows Defender Real-Time Protection with GPO*. Retrieved from https://stackoverflow.com/questions/62174426/how-to-permanently-disable-windows-defender-real-time-protection-with-gpo

YouTube. (n.d.). *How to Install FLARE-VM*. Retrieved from https://www.youtube.com/watch?v=i8dCyy8WMKY

YouTube. (n.d.). *FLARE-VM Setup Tutorial*. Retrieved from https://www.youtube.com/watch?v=BiSdnusy2AQ