Foundational SOC Lab: Windows, Kali Linux, Sysmon, and Splunk Integration

Aryan Vij

Date Started: 11/18/2024

Date Completed: 11/18/2024

Table of Contents

1. Objective

The goal of this lab was to create a basic Security Operations Center (SOC) environment using VirtualBox, setting up a Windows 10 VM and a Kali Linux VM. The lab also involved configuring **Sysmon** on Windows for enhanced logging, setting up **Splunk** for log aggregation, and simulating network activity using **Nmap** from Kali Linux to test the detection capabilities.

2. Virtual Machine Setup

2.1 Windows 10 VM Installation and Configuration

- A **Windows 10 VM** was installed in **VirtualBox** with the following specifications:
    - **Allocated Memory**: 6741 MB
    - **CPU**: 1 Core
    - **Disk**: 100.94 GB VDI (VirtualBox Disk Image)
    - **OS**: Windows 10 (64-bit)

2.2 Kali Linux VM Installation and Configuration

- A **Kali Linux VM** was also installed in **VirtualBox** with similar specifications:
    - **Allocated Memory**: 2048 MB
    - **CPU**: 1 Core
    - **Disk**: 80.09 GB VDI (VirtualBox Disk Image)
    - **OS**: Kali Linux (64-bit)

2.3 VirtualBox Network Configuration

- Both VMs were placed in an **Internal Network** within VirtualBox.
- **Network Configuration**:
    - **Windows 10 VM**: IP Address 192.168.1.20
    - **Kali Linux VM**: IP Address 192.168.1.10
- **IPv4 Settings (Windows 10)**:
    - Subnet Mask: 255.255.255.0
    - Default Gateway: None (no external network connectivity for this lab)
    - DNS: None (internal communication only)
- **Kali Linux Wired Network Settings**:
    - Set up with a static IP address using the **Network Manager** GUI, with the same subnet and configuration (192.168.1.10).
    - Confirmed network connectivity by pinging between both VMs:
        - **Ping from Windows 10 to Kali**: Successful
        - **Ping from Kali to Windows 10**: Successful

2.4 Ping Test Verification

- Both VMs were successfully able to **ping** each other, confirming that the network configuration was correct and both systems were on the same subnet.

3. Sysmon Configuration on Windows

3.1 Installing Sysmon on Windows

- **Sysmon** was installed on the **Windows 10 VM** using the following steps:
    1. **Download Sysmon** from Microsoft Sysinternals:
        - Navigate to the official Sysmon download page and download the tool.
        - Extract `Sysmon.zip` to a directory (e.g., `C:\Sysmon`).
    2. **Download Sysmon Configuration**:
        - The raw **Sysmon configuration file** was downloaded from GitHub:
            - GitHub URL: https://github.com/olafhartong/sysmon-modular/blob/master/sysmonconfig.xml
    3. **Deploy Sysmon with Custom Config**:
        - The **sysmonconfig.xml** file was moved to the **Sysmon directory** after extraction.
        - Sysmon was installed with the following command in **PowerShell**

          ```
          .\Sysmon64.exe -i .\sysmonconfig.xml
          ```

        - Sysmon was successfully installed, and logging for various events (e.g., process creation, network connections) was activated.

3.2 Configuring Sysmon with Modular Configuration

- The **sysmonconfig.xml** file was tailored to provide detailed logging for:
    - Process creation (Event ID 1)
    - Network connections (Event ID 3)
    - File creation time (Event ID 11)

- After installation, Sysmon began generating logs that were visible in the **Windows Event Viewer**, under the **Applications and Services Logs** section, specifically in **Microsoft > Windows > Sysmon**.

3.3 Verification of Sysmon Logs

- In the **Windows Event Viewer**, the following log categories were confirmed:
  - **Process Creation Logs**: Event ID 1
  - **Network Connection Logs**: Event ID 3
- These logs were verified to be working and were confirmed to be collected in the **Event Viewer** as intended.

4. Splunk Configuration

4.1 Installing Splunk

- **Splunk Enterprise** was installed on the **Windows 10 VM** by downloading the installer from the official Splunk website.
- After installation, **Splunk** was configured to run on **localhost** (`127.0.0.1:8000`), and the web interface was accessible for log viewing and analysis.

4.2 Configuring Splunk to Monitor Local Event Logs

- After logging into **Splunk** with the **admin** account, the following steps were followed to set up log monitoring:
  - **Add Data** > **Monitor** > **Local Event Logs**.
  - Selected **Application**, **Security**, and **System** logs for monitoring.
  - Created a new **index** called `main` to store the logs.
- **Splunk Indexing**:
  - Logs from **Sysmon** and **Windows Event Viewer** were successfully indexed into Splunk's `main` index.

    - Events from the **Application**, **Security**, and **System** logs were being ingested and displayed in the Splunk web interface.

4.3 Verification of Data Ingestion

- After configuring the data inputs, events from Windows logs were visible in **Splunk**.
    - **Search Query**: index=main confirmed the presence of events such as **process creation**, **network connections**, and **system errors**.
- Splunk dashboards were configured to visualize and track incoming event data.

5. Nmap Scanning from Kali Linux

5.1 Performing Nmap Scan on Windows 10 VM

- From the **Kali Linux VM**, I initiated a **network scan** of the **Windows 10 VM** using **Nmap** to simulate an attack and test the detection capabilities of **Splunk** and **Sysmon**.

Nmap Command: nmap -sS -p 3389 192.168.1.20

- This scan was designed to check if port 3389 (Remote Desktop) was open on the **Windows 10 VM**.

5.2 Event Generation

- The **Sysmon** configuration generated logs related to **network activity** (e.g., connections to open ports) and **process creation** (if the `nmap` process was logged).

5.3 Observing Logs in Splunk

- I checked **Splunk** for any event logs related to the Nmap scan, specifically looking for logs associated with:

- **Network Connections**: Event ID 3 (Sysmon)
- **Process Creation**: Event ID 1

6. Conclusion

6.1 Summary of Lab Setup

In this lab, I successfully set up a foundational **Security Operations Center (SOC)** environment using **VirtualBox**, consisting of two virtual machines (VMs): a **Windows 10 VM** and a **Kali Linux VM**. The network configuration was designed to ensure internal communication between the two systems, with each machine assigned static IPs in the **192.168.1.x** subnet. I also configured **Sysmon** on the Windows 10 VM to enhance event logging, specifically targeting key security events such as process creation and network connections.

Next, I installed **Splunk** on the Windows 10 VM, which was used to collect and index logs from the local Windows Event Viewer and **Sysmon**. Through Splunk's powerful search capabilities, I was able to monitor and visualize the event data, which is essential for security monitoring in a SOC environment. Finally, I simulated a **network scan** using **Nmap** from the Kali Linux VM to test the detection and logging capabilities of both **Sysmon** and **Splunk**.

6.2 Key Takeaways

- **Virtualization & Networking**: Setting up an isolated internal network in **VirtualBox** allowed the two VMs (Windows and Kali Linux) to communicate securely without external network access. This approach is useful for sandboxing and controlled testing.
- **Sysmon for Windows**: Configuring **Sysmon** for enhanced Windows event logging significantly improved visibility into system-level activities such as process creation, network connections, and file events. This is critical for detecting suspicious behavior in a SOC environment.

- **Splunk for Log Aggregation**: **Splunk** proved to be an excellent tool for aggregating and analyzing log data from both **Windows Event Viewer** and **Sysmon**, providing insights into the security posture of the networked systems.
- **Network Scanning for Detection**: The Nmap scan on the Windows 10 machine helped test the detection capabilities of **Sysmon** and **Splunk**, validating the SOC setup and providing valuable learning on event correlation.

6.3 Future Improvements

While this lab provides a solid foundational SOC environment, there are several areas where the setup can be expanded to create a more realistic and robust system:

- **Adding Windows Server 2019**: Integrating a **Windows Server 2019** VM as the **Domain Controller** will allow me to implement **Active Directory** (AD) for centralized user management, authentication, and more granular logging.
- **Connecting Windows 10 to the Domain**: Once the Domain Controller is set up, I plan to join the **Windows 10 machine** to the domain, enabling centralized user authentication and group policy enforcement.
- **Implementing Password Policies**: As part of the AD setup, I will configure **password policies** (e.g., complexity requirements, expiration, and lockout settings) to ensure compliance with security best practices.
- **Centralized PowerShell Logging**: I plan to enable **centralized PowerShell logging** to track administrative actions and detect any potentially malicious use of PowerShell in the environment.
- **User and Role Creation**: With **Active Directory**, I'll create roles and users with appropriate permissions to manage access control, ensuring a more secure and manageable system environment.

These improvements will allow me to simulate a more comprehensive corporate IT infrastructure and better monitor and secure user activity, making the lab environment closer to a production-ready SOC setup.

References

1. **Splunk Official Website**

   Splunk Enterprise and associated tools can be accessed from the official Splunk website.

   URL: https://www.splunk.com/
2. **Sysmon Download from Microsoft Sysinternals**

   The Sysmon tool, available for download via Microsoft's Sysinternals suite.

   URL: https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon
3. **Splunk Getting Started Tutorial**

   A tutorial on setting up and using Splunk, providing a foundation for data monitoring and analysis.

   URL: https://dev.splunk.com/enterprise/tutorials/module_getstarted/envsetup/
4. **What is a SOC?**

   An article from Spiceworks explaining the concept of a Security Operations Center (SOC) and its role in IT security.

   URL:

   https://www.spiceworks.com/it-security/vulnerability-management/articles/what-is-soc/

Appendix

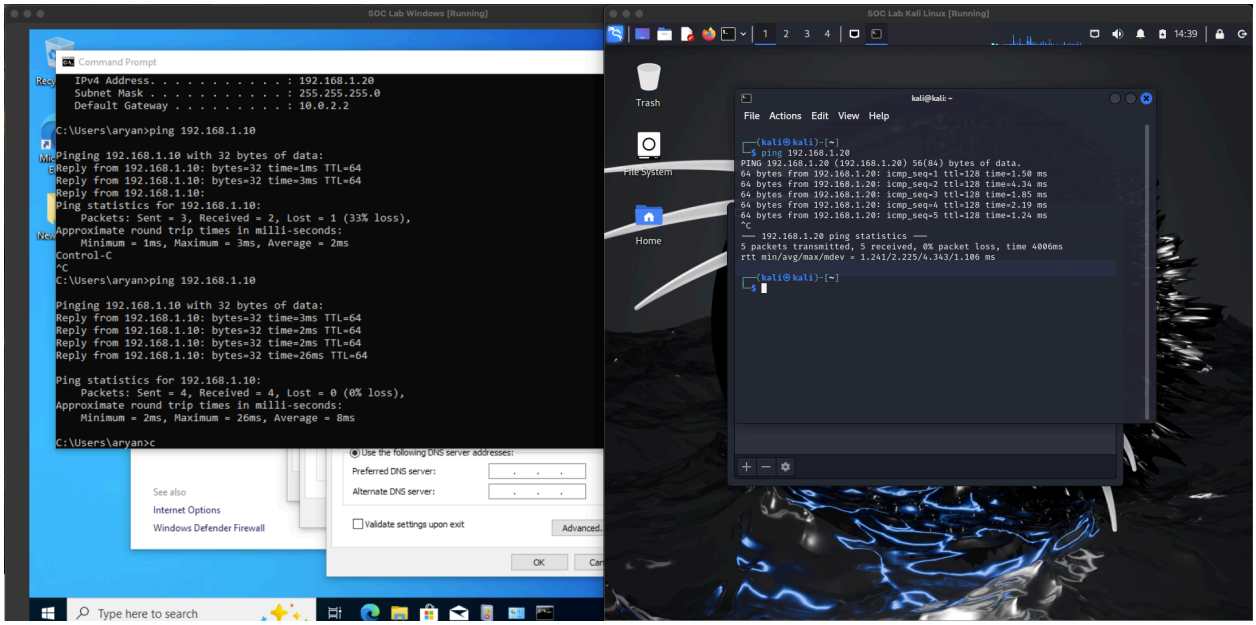


Image 1: Successful Pings

- This image shows the successful ping tests between the **Windows 10 VM** and **Kali Linux VM**, confirming network connectivity between the two systems in the internal network.

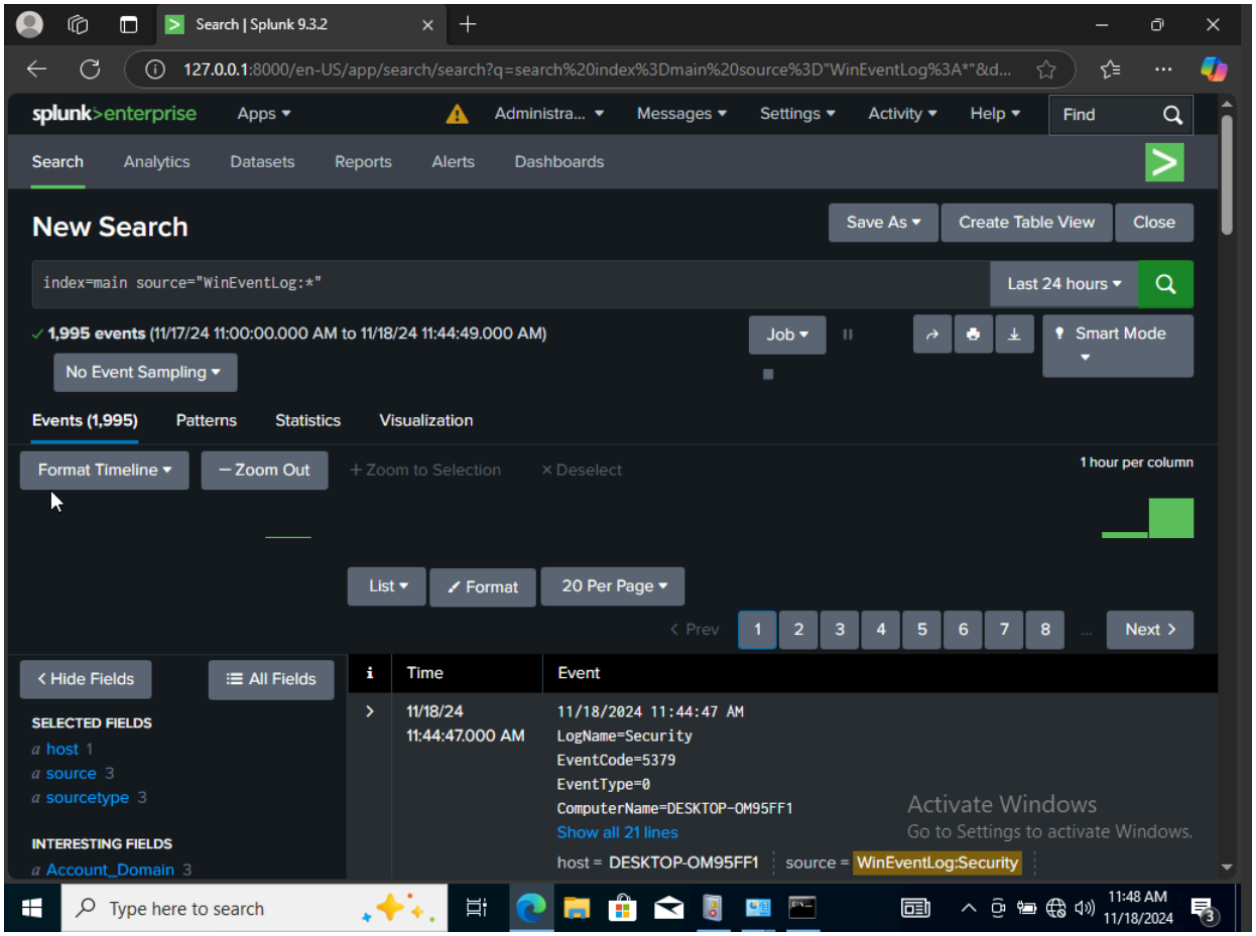


Image 2: Splunk Dashboard

- The Splunk dashboard image displays the ingested log data from **Sysmon** and the **Windows Event Viewer**, with logs from **Application**, **Security**, and **System** being indexed and visualized in Splunk.

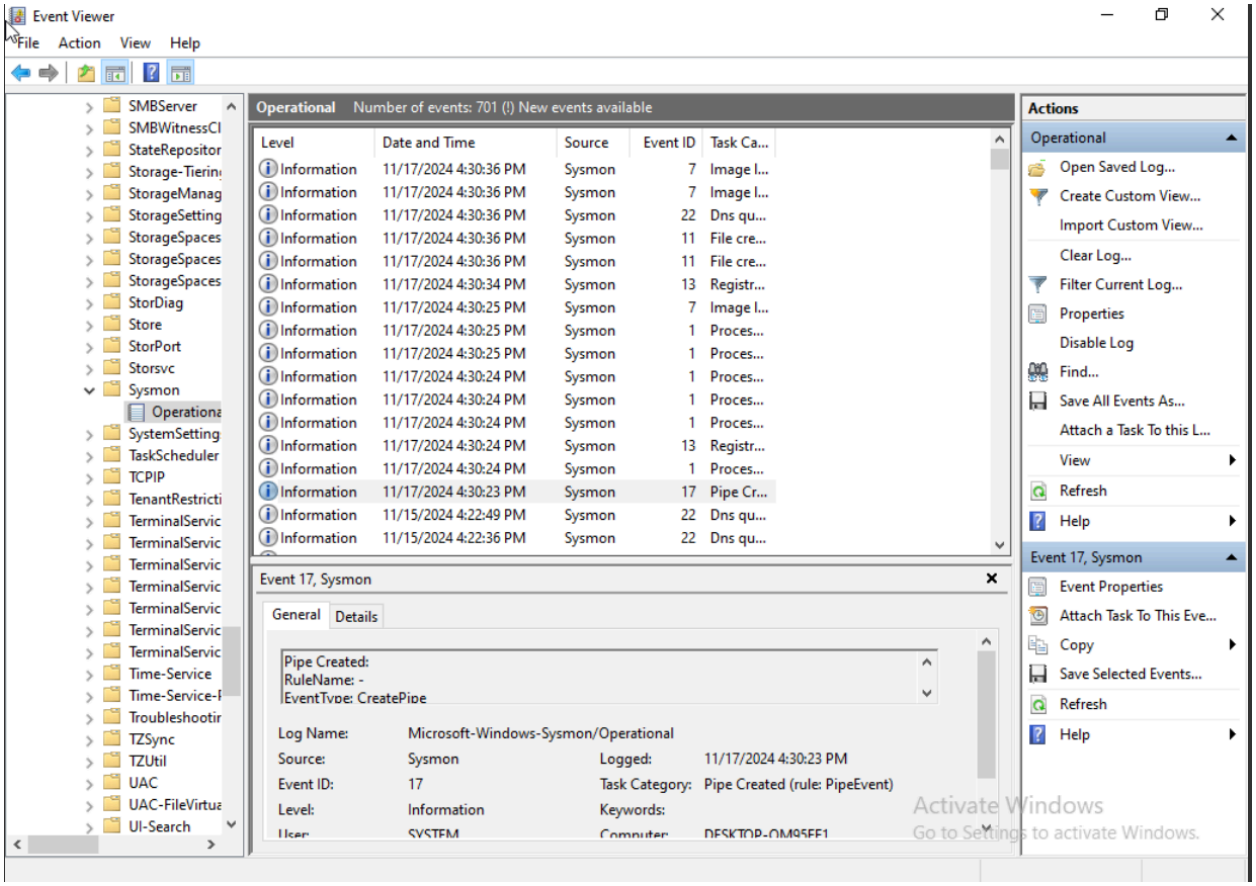


Image 3: Sysmon Dashboard

- This image shows the Sysmon logs in **Windows Event Viewer**, highlighting key security events such as **process creation** (Event ID 1) and **network connections** (Event ID 3).

Oracle VM VirtualBox Manager

Tools

New Add Settings Discard Show

vulnerablemachine
Saved

hackingmachine
Powered Off

SOC Lab Kali Linux
Running

SOC Lab Windows
Running

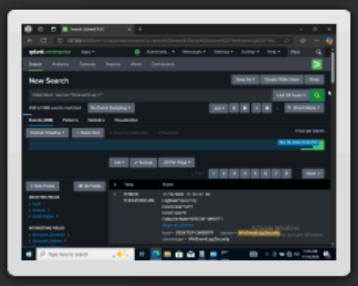
General

Name: SOC Lab Windows
Operating System: Windows 10 (64-bit)

System

Base Memory: 6741 MB
Processors: 2
Boot Order: Floppy,Optical,Hard Disk
Acceleration: Nested Paging,Hyper-V Paravirtualization

Preview



Display

Video Memory: 200 MB
Scale-factor: 1.20
Graphics Controller: VBoxSVGA
Acceleration: 3D
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: SATA
SATA Port 0: SOC Lab Windows.vdi (Normal,100.94 GB)
SATA Port 1: [Optical Drive] Win10_22H2_English_x64v1.iso (5.72 GB)

Audio

Host Driver: Default
Controller: Intel HD Audio

Network

Adapter 1: Intel PRO/1000 MT Desktop (Internal Network,'SOC_LAB')

USB

USB Controller: xHCI
Device Filters: 0 (0 active)

Shared folders

None

Image 4: VirtualBox Settings

- A screenshot of the **VirtualBox network settings**, showing the configuration for both the **Windows 10** and **Kali Linux** VMs.