Project Report

Enterprise SOC Simulation & Threat Detection Lab

Author: Sakina Teliya
Date: December 2025
Project Type: Cybersecurity / Blue Team Operations / Infrastructure Implementation

1. Executive Summary

This project involved the end-to-end design, deployment, and configuration of a virtualized Security Operations Center (SOC) home lab. The objective was to simulate a corporate endpoint environment, deploy advanced telemetry sensors (Sysmon), and aggregate logs into a centralized SIEM (Splunk) for threat hunting.

The project required overcoming significant technical challenges, including Windows 11 hardware requirement bypasses, OOBE (Out of Box Experience) network restrictions, and manual configuration of Splunk log inputs due to GUI limitations.

2. Technical Architecture

- **Host System:** Personal Workstation (Windows).
- **Virtualization Hypervisor:** Oracle VirtualBox.
- **Guest Operating System:** Windows 11 Enterprise (Evaluation Edition).
- **Endpoint Detection & Response (EDR):** Microsoft Sysmon (v15.0) configured with the SwiftOnSecurity exclusion ruleset.
- **SIEM (Security Information & Event Management):** Splunk Enterprise (Free Trial Version).
- **Log Ingestion Pipeline:** Local Windows Event Log forwarding to Splunk Indexer.

3. Implementation Log & Troubleshooting

Phase 1: Virtual Infrastructure & OS Installation

Challenge: Installing Windows 11 on a virtual machine triggers compatibility errors   due to the lack of TPM 2.0 and Secure Boot modules in standard VirtualBox configurations.

Implementation Steps:

1. **VM Creation:** Configured a new VirtualBox machine (`SOC-Lab-Windows`) with 4GB RAM and 2 vCPUS.
2. **The Registry Bypass (TPM/SecureBoot):**
   - Upon booting the installer, the "This PC can't run Windows 11" error was encountered.
   - **Solution:** Accessed the Command Prompt via `Shift + F10` during installation.
     Launched `regedit` and manually created the `HKEY_LOCAL_MACHINE\SYSTEM\Setup\LabConfig` key.
   - Added DWORD values `BypassTPMCheck` (1) and `BypassSecureBootCheck` (1) to force the installer to ignore hardware checks.
3. **The OOBE Network Trap:**
   - Windows 11 Setup (OOBE) prevented local account creation, forcing a Microsoft Account sign-in.
   - **Attempt 1:** Used `OOBE\BYPASSNRO` command to enable the "I don't have internet" button. This failed on the specific os version used.
   - **Attempt 2:** Used the "Banned Email" trick (no@thankyou.com) to force an error. This was patched by Microsoft and failed.
   - **Final Solution (Domain Join & Cable Pull):** Successfully bypassed the restriction by selecting "Domain Join instead" and physically disconnecting the virtual network adapter via VirtualBox (**Devices > Network > Uncheck Connect Adapter**). This forced Windows to fallback to a "Limited Setup," allowing the creation of a local Administrator account named `Analyst`.

Phase 2: Post-Deployment Configuration

1. **Guest Additions:** Installed VirtualBox Guest Additions to enable full-screen resolution and mouse pointer integration.
2. **Network Re-establishment:** Reconnected the virtual network adapter to enable internet access for tool retrieval.

Phase 3: Telemetry Deployment (Sysmon)

Objective: Standard Windows logs are insufficient for deep analysis. Sysmon was deployed to capture granular data such as Process Creation (Event ID 1) and Network Connections (Event ID 3).

Implementation Steps:

1. Downloaded **Sysmon** from the Microsoft Sysinternals repository.
2. Retrieved the industry-standard **SwiftOnSecurity** configuration file (`sysmonconfig.xml`) to filter out noise (benign background processes).
3. Installed via Powershell with Administrative privileges:
   ```
   .\Sysmon64.exe -i sysmonconfig.xml
   ```
4. **Verification:** Validated the service status using `Get-Service Sysmon64` and checked Event Viewer to confirm the generation of `Microsoft-Windows-Sysmon/Operational` logs.

Phase 4: SIEM Deployment & Log Ingestion (Splunk)

Challenge: During the “Add Data” configuration in Splunk, the GUI list of available Windows Event Logs was truncated. The interface stopped rendering at the “Media Foundation” logs, making it impossible to select the “Microsoft-Windows-Sysmon” checkbox via the UI.

Solution (Manual Configuration): Instead of relying on the GUI, the connection was hard-coded into Splunk’s backend configuration files.

1. **Accessing Configuration Directory:** Navigated to `C:\Program Files\Splunk\etc\system\local`.
2. **Creating** `inputs.conf` : Created a new configuration file to explicitly define the input source.
3. **Configuration Syntax:**
   ```
   [WinEventLog://Microsoft-Windows-Sysmon/Operational]
   disabled = 0
   renderXml = 1
   checkpointInterval = 5
   current_only = 0
   ```
4. **Service Restart:** Restarted the `Splunkd` service via the Windows Services management console (`services.msc`) to force the reload of the new configuration.
5. **Verification:** Executed a Splunk search `index=* sourcetype="XmlWinEventLog:Microsoft-Windows-Sysmon/Operational"` which confirmed successful log ingestion.

4. Operational Testing & Threat Hunting

To validate the detection pipeline, two distinct simulation scenarios were executed.

Scenario A: Persistence Mechanism (The "Hacker" Test)

Attack vector: Adversaries often create backdoor accounts to maintain access to a compromised host (MITRE ATT&CK T1136.001).

- **Execution:** Opened Command Prompt as Administrator and ran: `net user hacker 12345 /add`
- **Detection:**
  - Splunk successfully indexed the event immediately.
  - **Evidence Captured:**
    - **Process:** `net.exe`
    - **CommandLine:** `net user hacker 12345 /add`
    - **Parent Process:** `cmd.exe`
    - **User:** `Analyst` (The compromised account used to run the command).

Scenario B: Malware Download Test (EICAR)

Attack vector: Simulating the download of a malicious payload.

- **Execution:** Accessed `eicar.org` and downloaded the standard anti-malware test file.
- **Detection:**
  - Windows Defender immediately quarantined the file.
  - Sysmon captured the file creation event (`Event ID 11`) prior to deletion.
  - Splunk logs showed the browser (`msedge.exe`) initiating a file write to the Downloads directory, confirming the source of the "infection."

5. Conclusion & Skills Demonstrated

This project successfully established a functional detection lab capable of monitoring advanced threats. It demonstrated not only the ability to use security tools but also the engineering mindset required to troubleshoot OS-level restrictions and software bugs.

Key Competencies:

- **Virtualization:** Managing resources and guest OS isolation.
- **System Administration:** Registry manipulation, service management, and OOBE bypass techniques.

- **Log Analysis:** Understanding the structure of Windows Event Logs and Sysmon Event IDs.
- **SIEM Engineering:** Configuring `inputs.conf` manually to overcome UI limitations and writing SPL (Search Processing Language) queries for real-time monitoring.