Complete SOC Project Report

Project Title

Centralized Windows Endpoint Monitoring using Splunk (Security, Sysmon & Application Logs)

Author

Gourab Das

Target Role

SOC Analyst – Tier 1 / Junior SOC Analyst

---

1. Executive Summary

This project demonstrates a complete SOC-style implementation of centralized Windows endpoint monitoring using Splunk Enterprise. The project focuses on collecting, indexing, and analyzing Windows Security logs, Sysmon telemetry, and Application logs to detect authentication attacks, suspicious process execution, network activity, and application errors.

The lab simulates real-world SOC onboarding challenges such as audit configuration, permission issues, log ingestion failures, and troubleshooting. The project is fully documented with architecture, configuration files, detections, troubleshooting notes, and proof artifacts.

---

2. Business & SOC Context

Security Operations Centers rely heavily on endpoint telemetry to detect threats such as brute-force attacks, malware execution, lateral movement, and system abuse. Windows endpoints generate critical security signals, but without centralized logging and correlation, incident detection becomes difficult.

This project mirrors a real SOC use case where endpoint logs are forwarded to a SIEM for monitoring, investigation, and alerting.

---

3. Project Objectives

- Centralize Windows endpoint logs using Splunk
- Enable and validate Windows audit policies
- Collect Windows Security, Sysmon, and Application logs
- Separate logs into dedicated indexes
- Build SOC-relevant detection use cases
- Troubleshoot real-world ingestion and permission issues
- Document the project to enterprise SOC standards

---

4. Environment & Tools

Operating Systems

- Windows Endpoint (Log Source VM)
- Windows System running Splunk Enterprise (SOC Server)

Tools & Technologies

- Splunk Enterprise (Free Trial)
- Splunk Universal Forwarder
- Windows Event Logging
- Sysmon (Sysinternals)
- Windows NT Security Architecture

Network

- Local lab environment
- Secure log forwarding over TCP 9997

---

5. Technical Architecture

Architecture Overview

- Windows Endpoint generates logs
- Splunk Universal Forwarder collects logs
- Logs forwarded securely to Splunk Enterprise
- Splunk indexes, searches, and analyzes data

Data Flow

Windows Endpoint → Splunk Universal Forwarder → TCP 9997 → Splunk Enterprise → Detection & Analysis

---

6. Windows NT Architecture & Security Context

Windows NT (New Technology) is the core architecture behind modern Windows operating systems. It provides:

- Secure kernel-based design
- Security Identifiers (SIDs)
- Access tokens
- Privileges vs permissions
- Auditing and event logging

All Windows Security Event IDs used in this project are generated by the NT security subsystem.

---

7. Windows Audit Configuration

Audit Enablement

Windows Security logs are generated only when audit policies are enabled.

Audit configuration was performed using the `auditpol` utility due to Windows Home limitations.

```
auditpol /set /category:"Logon/Logoff" /success:enable /failure:enable
```

Verification

- Generated failed login attempts
- Confirmed EventCode 4625 generation

---

8. Splunk Universal Forwarder Configuration

Forwarder Role

- Lightweight agent for log collection
- Forwards logs without indexing locally

Service Account

- **Lab Setup:** NT AUTHORITY\SYSTEM
- **Reason:** Required to read protected Security Event Logs

Production Recommendation

- Use dedicated service account or gMSA

- Apply least privilege
- No interactive login rights

---

9. Log Sources & Inputs Configuration

Configuration File

- File: `inputs.conf`
- Location: `SplunkUniversalForwarder/etc/system/local/`

Windows Security Logs

```
[WinEventLog://Security]
disabled = 0
index = wineventlog
sourcetype = WinEventLog:Security
start_from = oldest
```

Windows Application Logs

```
[WinEventLog://Application]
disabled = 0
index = application
sourcetype = WinEventLog:Application
start_from = oldest
```

Sysmon Logs

```
[WinEventLog://Microsoft-Windows-Sysmon/Operational]
disabled = 0
index = sysmon
sourcetype = xmlwineventlog:sysmon
start_from = oldest
```

---

10. Indexing Strategy

Separate indexes were created to improve clarity, performance, and investigation efficiency:

Index	Log Type
wineventlog	Windows Security
application	Windows Application
sysmon	Sysmon Operational

---

11. Sysmon Integration

Sysmon was deployed to extend endpoint visibility beyond native Windows Security logs.

Key Sysmon Event IDs Used

- Event ID 1 – Process Creation
- Event ID 3 – Network Connection
- Event ID 11 – File Creation

Sysmon provides high-fidelity telemetry useful for malware and post-exploitation detection.

---

12. Log Ingestion Verification

Verification Searches

```
index=wineventlog
index=application
index=sysmon
```

Key Events Observed

- 4624 – Successful logon
- 4625 – Failed logon
- Sysmon EventCode 1 – Process creation
- Sysmon EventCode 3 – Network connections

---

13. Detection Use Cases

13.1 Failed Login / Brute-Force Detection

```
index=wineventlog EventCode=4625
| stats count by Account_Name, host
| where count > 5
```

13.2 Suspicious Process Execution

```
index=sysmon EventCode=1
| stats count by Image, CommandLine
```

13.3 Suspicious Network Connections

```
index=sysmon EventCode=3
| stats count by DestinationIp, Image
```

13.4 Application Error Monitoring

```
index=application Level=2
```

---

14. Challenges Faced & Troubleshooting

Multiple real-world issues were encountered and resolved, including:

- Missing audit policies
- Permission issues accessing Security logs
- Incorrect input configuration
- Windows Home OS limitations
- Sysmon ingestion issues

Detailed resolution steps are documented in the Troubleshooting folder.

---

15. Security Considerations

- Local System used only for lab simplicity
- Production environments require least privilege
- Logs forwarded securely over TCP
- Configuration files protected from unauthorized access

---

16. Learning Outcomes

- Hands-on SIEM log onboarding experience
- Deep understanding of Windows NT security
- SOC-style troubleshooting methodology
- SPL-based detection development
- Endpoint telemetry analysis

17. Future Enhancements

- Convert detections into alerts
- Map detections to MITRE ATT&CK
- Add Active Directory authentication logs
- Implement Splunk deployment server
- Tune Sysmon configuration

---

18. Conclusion

This project reflects a realistic SOC environment where endpoint logs are onboarded, validated, troubleshot, and analyzed using Splunk. The inclusion of Security, Sysmon, and Application logs provides strong visibility into authentication activity, endpoint behavior, and application health.
