SOC Project Report

Project Title

Centralized Log Monitoring of Kali Linux Using Splunk Enterprise (Windows)

Author

Gourab Das

Target Role

SOC Analyst – Tier 1 / Junior SOC Analyst1.

Project Objective

The objective of this project is to demonstrate successful ingestion of Linux system logs from a Kali Linux virtual machine into Splunk Enterprise running on a Windows SOC server. This enables centralized visibility, monitoring, and basic SOC-style analysis of Linux logs.

This lab focuses on **log ingestion and visibility**, not deep threat detection.

---

2. Environment Details

SOC Server

- OS: Windows 10 / 11
- SIEM: Splunk Enterprise
- Role: Indexer & Search Head

Log Source

- OS: Kali Linux
- Deployment: Virtual Machine (VMware / VirtualBox)
- Log Forwarder: Splunk Universal Forwarder (Linux)
- Logging Service: rsyslog (enabled)

Network Mode

- NAT / Bridged (Lab environment)

---

3. Architecture Overview

Log Flow:

Kali Linux (Logs Generated)
→ rsyslog / journald
→ Splunk Universal Forwarder
→ TCP 9997
→ Splunk Enterprise (Windows)
→ Index: `linux`

This architecture represents a standard SOC log collection pipeline for Linux endpoints.

---

4. Logs Collected

The following Linux logs are ingested into Splunk:

- `/var/log/auth.log` – Authentication & sudo activity
- `/var/log/syslog` – System-level events

- `/var/log/kern.log` – Kernel messages

Note: These logs became available after installing and enabling **rsyslog** on Kali Linux.

---

5. Configuration Summary

5.1 Splunk Universal Forwarder (Kali Linux)

- Forwarder installed under `/opt/splunkforwarder`
- Configured `inputs.conf` to monitor Linux log files
- Forwarding enabled to Windows Splunk Enterprise on port `9997`

5.2 Splunk Enterprise (Windows)

- Custom index created: `linux`
- Receiving enabled on TCP port `9997`
- Data successfully indexed and searchable

---

6. Verification & Proof of Ingestion

Log ingestion was verified using the following Splunk searches:

- `index=linux`
- `index=linux | stats count`
- `index=linux | timechart count`

These searches confirm:

- Logs are actively flowing

- Events are being indexed correctly
- Time-based log continuity is present

---

8. Key Learning Outcomes

- Understood Linux logging mechanisms (journald vs rsyslog)
- Configured Splunk Universal Forwarder on Linux
- Created and validated custom indexes in Splunk
- Verified log ingestion using SPL
- Built SOC-style documentation and evidence

---

9. Problems Faced & Resolutions

Problem 1: No Linux Logs Visible in Splunk

Issue: After installing Splunk Universal Forwarder on Kali Linux, no logs were appearing in Splunk.

Root Cause: Kali Linux uses **systemd-journald** by default and traditional `/var/log/syslog` files were not present.

Resolution:

- Verified journald was active using:
  `systemctl status systemd-journald`
- Installed and enabled **rsyslog** to convert journald logs into syslog format.
- Restarted rsyslog and Splunk Forwarder services.

Problem 2: `/var/log/auth.log` and `/var/log/syslog` Missing

Issue: Commands like `ls /var/log/auth.log` returned *No such file or directory.*

Root Cause: rsyslog service was not installed or running.

Resolution:

- Installed rsyslog:
  ```
  sudo apt install rsyslog -y
  ```
- Enabled and started the service:
  ```
  sudo systemctl enable rsyslog --now
  ```
- Verified log file creation in `/var/log/`.

Problem 3: Logs Received but Dropped by Splunk

Issue: Splunk showed messages like `Received event for unconfigured index`.

Root Cause: Index was not created in `indexes.conf`.

Resolution:

- Created a dedicated index (`linux`) in Splunk.
- Restarted Splunk to apply changes.

Problem 4: Confusion Between Journald and Syslog

Issue: Difficulty understanding why logs were not flowing even though journald was active.

Resolution:

- Learned that Splunk UF does not read journald directly by default.

- Used rsyslog as a bridge to forward logs into traditional log files.

Problem 5: Timechart Showing Zero or Unexpected Counts

Issue: `| timechart count` output was confusing.

Resolution:

- Understood that **count** represents the number of events per time bucket.
- Verified correct index and time range selection.

10. Conclusion

This project demonstrates successful ingestion of Kali Linux system logs into Splunk Enterprise using rsyslog and the Splunk Universal Forwarder. Log flow was verified through SPL searches and visualizations, while resolving real-world issues improved understanding of Linux logging and Splunk ingestion—forming a solid foundation for SOC monitoring and future detections.
