TASK 2
SECURITY ALERT MONITORING & INCIDENT RESPONSE

INTERNSHIP BATCH: Future Interns
REPORT BY: Singari Keerthi

TABLE OF CONTENTS

Introduction

In this activity, we were tasked with monitoring and analyzing security logs with Splunk, an open-source SIEM (Security Information and Event Management) software. The main objective was to create, gather, and log system and authentication logs so that we could identify events like failed login, brute force attempts, unauthorized access, and web application errors.

Splunk allows security experts to consume huge amounts of raw log data and convert it into useful information through queries, dashboards, and visualizations. Splunk makes it possible to identify abnormal patterns that would otherwise remain undetected by indexing and correlating disparate log sources.

- In a typical enterprise environment, SIEM solutions such as Splunk are important in:
- Threat Detection: Alerting for malicious activities such as brute force attacks or unusual user logins.
- Incident Response: Delighting investigators with actionable intelligence to back-trace the attack timeline.
- Compliance Monitoring: Assuring compliance with rules and regulations by keeping and processing logs.
- Operational Visibility: Providing administrators with a single pane of glass to observe system health and security stance.
- Proactive Defense: Allowing for alerts and automated responses to be configured to block threats before they grow.

This practice exposed us to how SIEMs are set up, how log data is consumed, and how various types of security incidents can be simulated and monitored. By actually completing each step by hand, not only did we get practice with Splunk as a tool, but we also solidified our knowledge of how attackers leave a footprint in logs and how defenders can use them for defense.

Environment Setup

1. **Operating System:** Kali Linux
2. **Tool Installed:** Splunk Enterprise (10.0.0, Linux 64-bit)
3. **Logs Collected:**
    a. Authentication logs (SSH login attempts)
    b. System logs (journal logs)
4. **Browser Used for Splunk Web:** Firefox (http://127.0.0.1:8000)

Installation of Splunk

Step 1: Download Splunk

1. Go to Splunk Downloads
2. Sign up / log in with a free Splunk account.
3. Download the Linux `.deb` package (choose the 64-bit `.deb` version).

Step 2: Install Splunk (Terminal)

1. cd ~/Downloads
2. sudo dpkg -i splunk-10.0.0-e8eb0c4654f8-linux-amd64.deb

Step 3: Start Splunk

1. cd /opt/splunk/bin
2. sudo ./splunk start --accept-license

Step 4: Access Splunk

1. http://127.0.0.1:8000

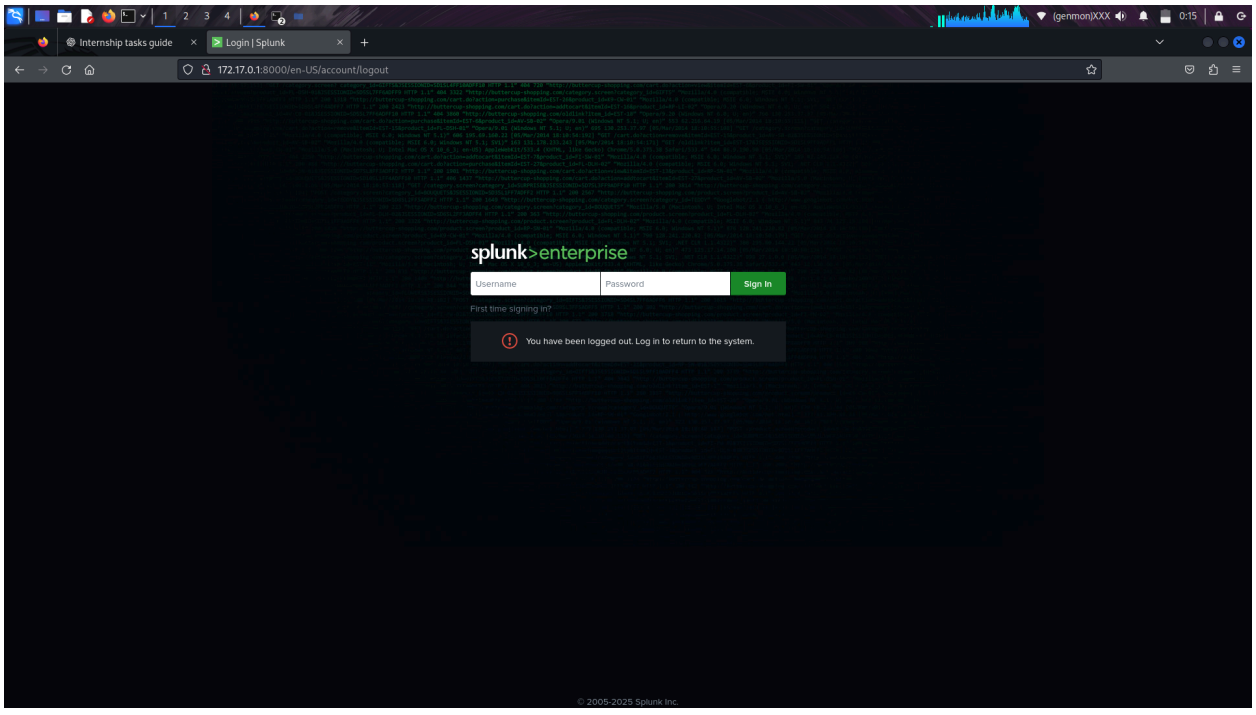

Figure 1: Splunk Login Page

Generating Logs

Authentication Logs (SSH)

1. **Installed SSH server:**

   sudo apt install -y openssh-server
   sudo systemctl enable --now ssh

2. **Created failed login attempts** (simulating brute force):

   for i in {1..5}; do ssh wronguser@127.0.0.1 -o ConnectTimeout=2 || true; done

3. **Created a successful login**:

   ssh kali@127.0.0.1
   exit

4. **Exported SSH-related logs:**

   sudo journalctl -u ssh --no-pager > ~/future_interns/logs/auth.log
   sudo journalctl -b --no-pager > ~/future_interns/logs/syslog.log

Uploading Logs to Splunk

1. Opened Splunk Web → Settings → Add Data → Upload.
2. Uploaded auth.log, syslog.log.

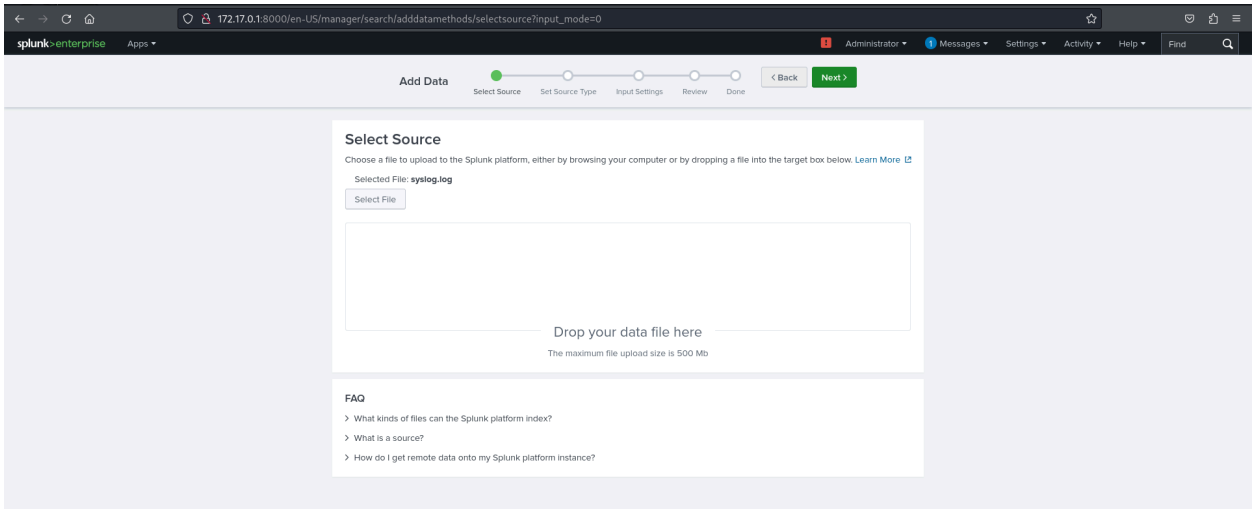

Figure 2: Uploading Logs

3. Assigned Source Types:
    a. syslog → auth.log and syslog.log

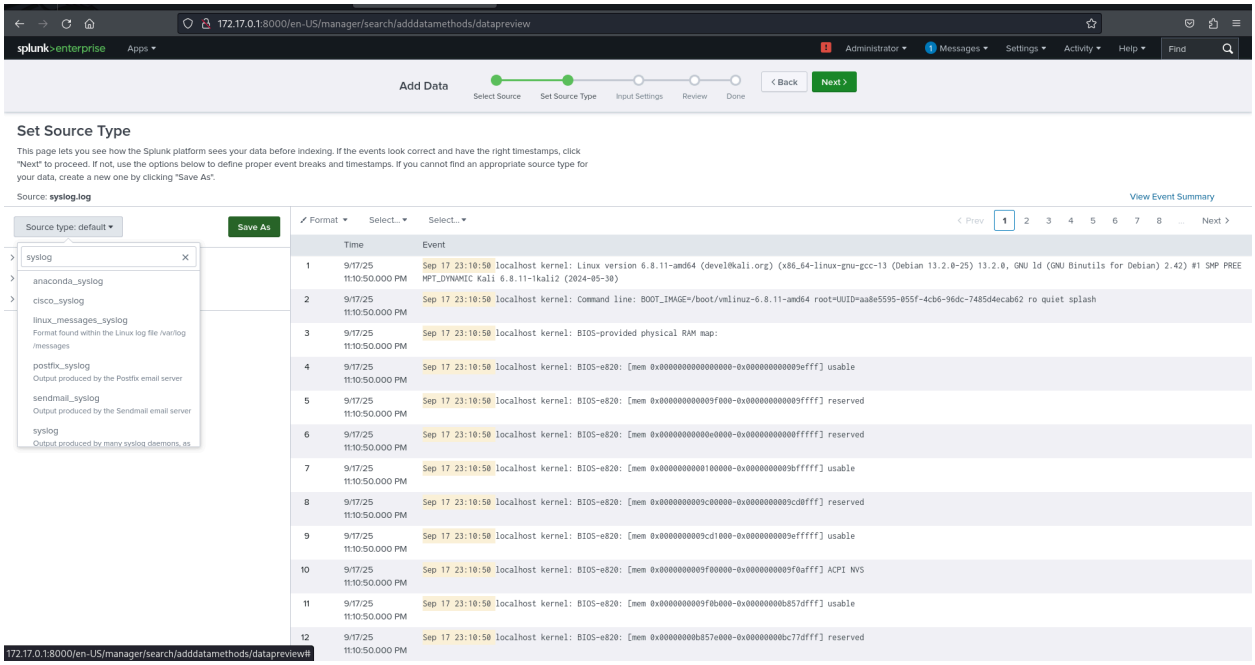

Figure 3: Setting Source Type

4. Indexed all logs under `main`.

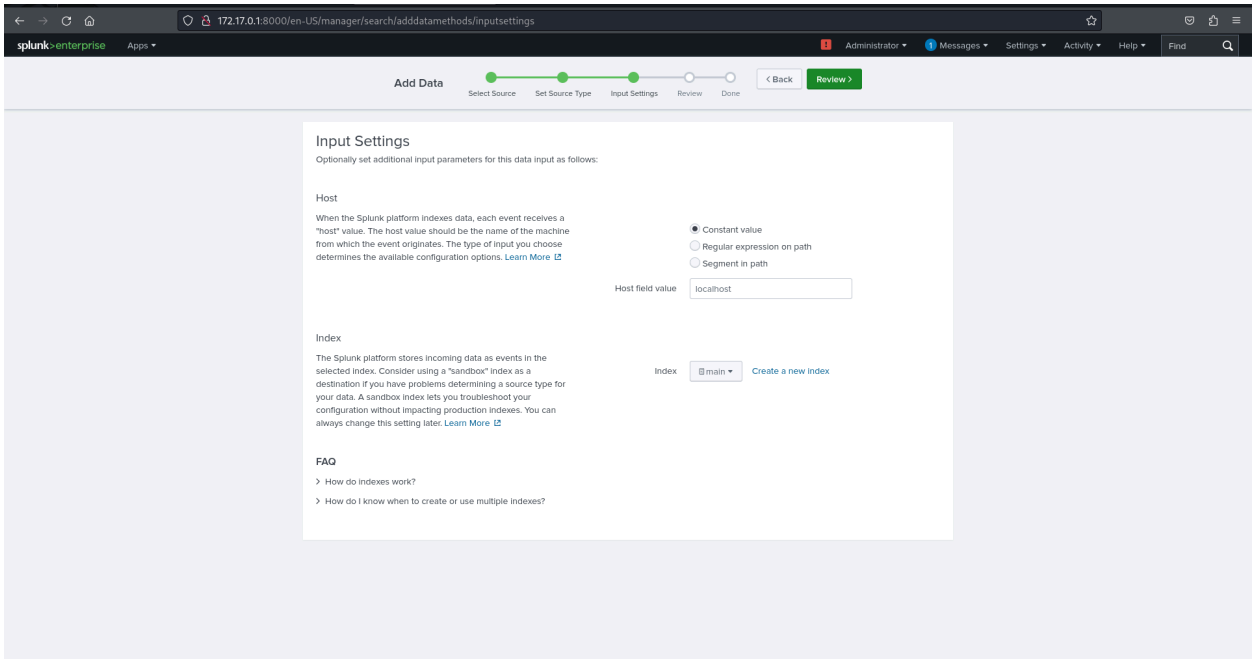

Figure 4: Indexing Logs

Analyzing Logs in Splunk

1. Show all events

index=main | head 20

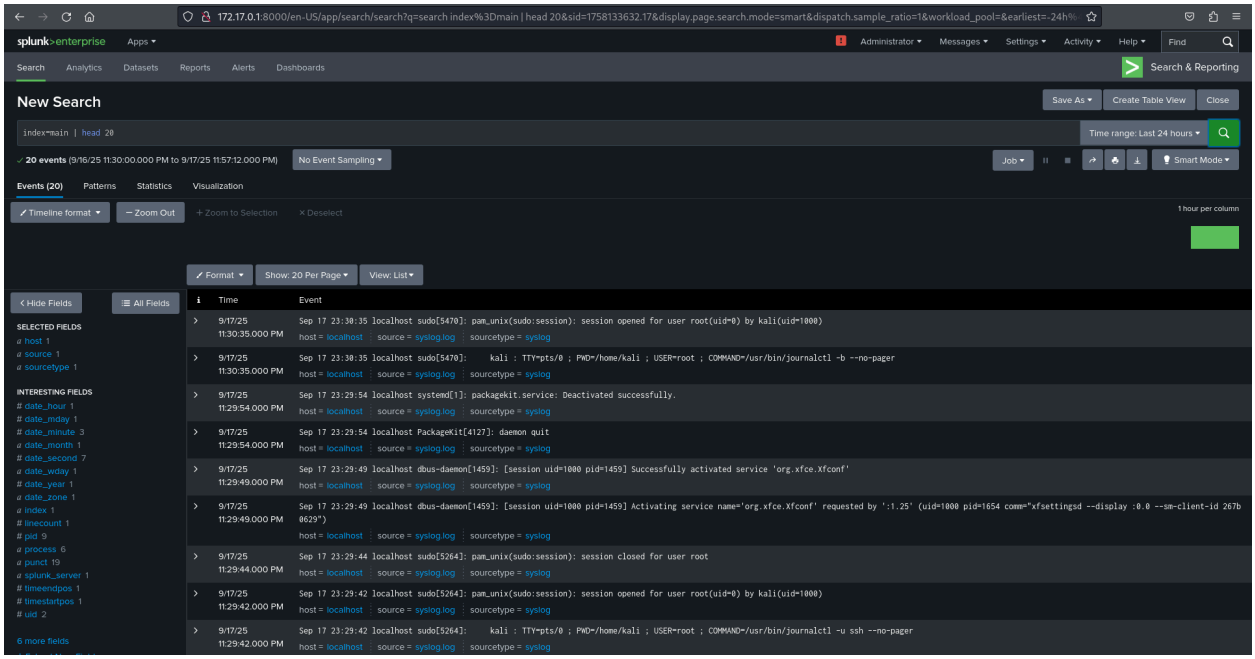

Figure 5: Top 20 Logs

2. Detect Failed SSH Logins

index=main "Failed password" OR "authentication failure"

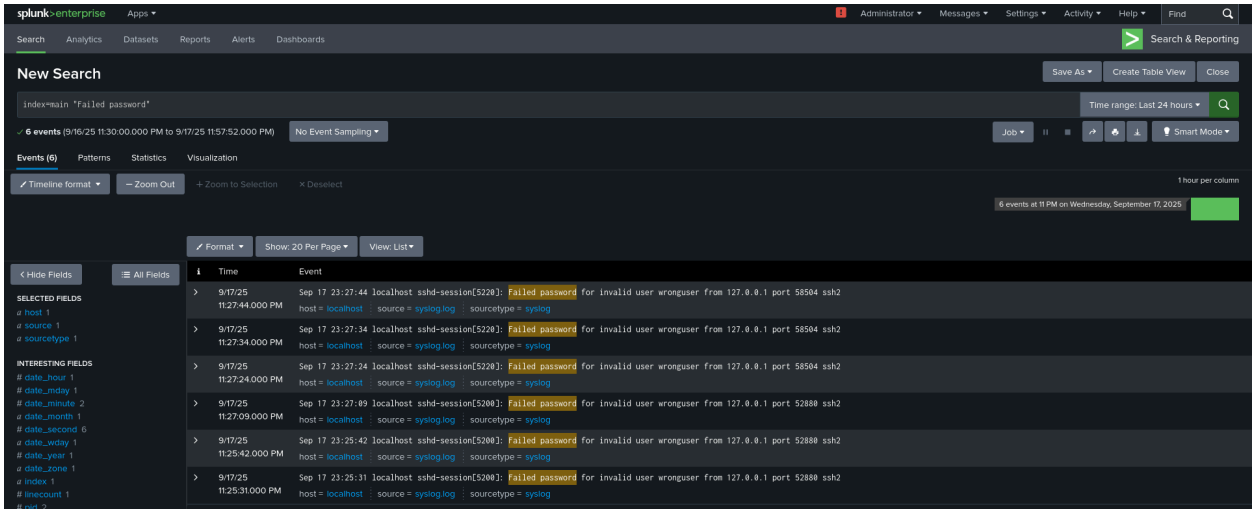

Figure 6: Logs with Failed Passwords

Findings

Event	Splunk Query	Impact	Mitigation
Failed SSH logins (Brute Force)	index=main "Failed password"	Attackers attempting to brute force passwords	Enable account lockouts, use SSH key authentication, rate limiting
Successful SSH login	index=main "Accepted password"	Valid login (verify if authorized)	Monitor authorized logins, use MFA
Unauthorized Web Access	index=main	Attempted access to restricted resources	Proper access control, WAF, strict role-based permissions

Remediation Recommendations

1. Enforce robust passwords & account lockout policies.
2. Implement Multi-Factor Authentication (MFA).
3. Limit SSH to allowed IP ranges.
4. Deploy a Web Application Firewall (WAF) for unauthorized request filtering.
5. Centralize log monitoring & enable automated alerts for repeated failures.

Conclusion

We learned how to do the following through this exercise:

- Install and set up Splunk as a SIEM solution.
- Create and gather authentication, system, and web server logs.
- Employ Splunk SPL queries to identify failed logins, brute force attempts, and unauthorized access.
- Develop alerts and reports to continuously monitor suspicious activity.
- Learn the significance of log analysis for incident detection and response.

This hands-on exercise reinforced how SIEM offerings like Splunk turn raw log data into actionable security intelligence, which is essential for real-world threat prevention and monitoring. Through the simulation of brute force attempts, invalid requests, and legitimate logins, we saw how logs expose patterns that point to threats, and how Splunk facilitates automation.

In an enterprise context, having the capability to centralize and analyze logs enables security teams to:

- Identify and categorize incidents rapidly.
- Promptly take remediation measures prior to damage.
- Enhance compliance posture by maintaining extensive audit trails.
- Enhance the overall security maturity of the organization through incident-based learning.

Overall, this exercise not only showcased Splunk's technical capabilities but reinforced the general significance of log analysis within an effective overall cybersecurity approach. It underscored how visibility, detection, and response are interdependent pillars of a strong defense system.

References

1. https://docs.splunk.com/
2. https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html
3. https://access.redhat.com/solutions/2112