SOC Project: Complete Implementation Guide

Automated Threat Detection & Incident Response Framework

Comprehensive Implementation Guide with Splunk, Wazuh, MITRE ATT&CK;, and Jira Integration

Generated on July 29, 2025

Table of Contents

Chapter 1: Introduction & Overview

1.1 Project Objectives

The SOC (Security Operations Center) Project represents a comprehensive security monitoring and incident response framework designed for modern cloud and hybrid environments. This enterprise-grade solution implements automated threat detection, incident management, and response capabilities using industry-standard tools and frameworks.

- Establish real-time threat detection across cloud and on-premises environments
- Implement automated incident response and ticket management
- Provide comprehensive security monitoring with MITRE ATT&CK; integration
- Create a scalable and maintainable security operations platform
- Enable compliance with industry standards and regulations

1.2 Technology Stack Overview

The SOC project integrates four core technologies to provide comprehensive security monitoring and incident response capabilities:

Tool	Purpose	Key Features
Splunk Enterprise	SIEM Platform	Log aggregation, real-time monitoring, advanced search
Wazuh	EDR Solution	Endpoint detection, file integrity monitoring, active response
MITRE ATT&CK	Threat Intelligence	Attack technique mapping, threat categorization
Jira	Incident Management	Ticket creation, workflow automation, team collaboration

1.3 System Requirements

The SOC project requires specific hardware and software configurations to ensure optimal performance and reliability. The following requirements are minimum specifications for a production environment.

Component	Minimum Specs	Recommended Specs
Splunk Server	8 CPU cores, 16GB RAM, 500GB storage	16 CPU cores, 32GB RAM, 1TB SSD
Wazuh Manager	4 CPU cores, 8GB RAM, 100GB storage	8 CPU cores, 16GB RAM, 200GB SSD
Jira Server	4 CPU cores, 8GB RAM, 100GB storage	8 CPU cores, 16GB RAM, 200GB SSD
Network	1Gbps connectivity	10Gbps backbone, redundant paths

1.4 Project Timeline

The SOC project implementation is divided into five phases, each building upon the previous phase to create a comprehensive security operations center.

Phase	Duration	Key Activities
Phase 1: Foundation	Week 1-2	Environment setup, tool installation, basic configuration
Phase 2: Core Implementation	Week 3-4	Splunk/Wazuh config, Jira integration, cloud logs
Phase 3: Detection & Monitoring	Week 5-6	Detection rules, alerts, dashboards, MITRE integration
Phase 4: Automation & Response	Week 7-8	Automated responses, incident workflows, testing
Phase 5: Testing & Optimization	Week 9-10	Comprehensive testing, optimization, documentation

Chapter 4: Tool Installation

4.1 Splunk Enterprise Installation

Splunk Enterprise is the primary SIEM platform for the SOC project. Follow these step-by-step instructions to install and configure Splunk Enterprise.

1. Download Splunk Enterprise from the official website
2. Extract the installation package to /opt/splunk
3. Run the Splunk installation script
4. Configure Splunk admin password
5. Start Splunk services
6. Access Splunk web interface on port 8000

Splunk Installation Commands:

```
# Download and extract Splunk

wget -O splunk.tgz 'https://download.splunk.com/products/splunk/releases/9.0.0/linux/splunk-9.0.0-17e00c557dc1-Linux-x86_64.tgz'

tar -xzf splunk.tgz -C /opt

# Start Splunk for the first time

cd /opt/splunk

./bin/splunk start --accept-license

# Set admin password

./bin/splunk edit user admin -password 'YourSecurePassword' -role admin -auth admin:changeme
```

4.2 Wazuh Installation

Wazuh provides endpoint detection and response capabilities. Install Wazuh manager and agents according to these instructions.

```
# Install Wazuh repository

curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | sudo apt-key add -

echo 'deb https://packages.wazuh.com/4.x/apt/ stable main' | sudo tee /etc/apt/sources.list.d/wazuh.list

# Install Wazuh manager

sudo apt-get update

sudo apt-get install wazuh-manager
```

```
# Start Wazuh manager
sudo systemctl daemon-reload
sudo systemctl enable wazuh-manager
sudo systemctl start wazuh-manager
```

4.3 Jira Setup

Jira provides incident management and ticket tracking capabilities. Set up Jira Cloud or Server according to your organization's requirements.

1. Create Jira Cloud account or install Jira Server
2. Create a new project called 'Security Incidents'
3. Configure issue types: Security Incident, Security Alert, Threat Intelligence
4. Set up custom fields for MITRE ATT&CK; techniques
5. Configure user permissions and access controls
6. Generate API token for integration

Chapter 5: Splunk Configuration

5.1 Initial Configuration

Configure Splunk Enterprise for optimal performance and security monitoring. This section covers essential configuration steps.

1. Configure indexes for security data
2. Set up user accounts and roles
3. Configure HTTP Event Collector (HEC)
4. Install security add-ons
5. Configure data retention policies
6. Set up monitoring and alerting

5.2 Index Configuration

Create dedicated indexes for different types of security data to optimize search performance and data management.

```
# Create security indexes
curl -k -u admin:password https://localhost:8089/services/data/indexes \
-d name=security_events \
-d maxTotalDataSizeMB=10000 \
-d frozenTimePeriodInSecs=7776000

# Create cloud logs index
curl -k -u admin:password https://localhost:8089/services/data/indexes \
-d name=cloud_logs \
-d maxTotalDataSizeMB=5000 \
-d frozenTimePeriodInSecs=2592000
```

5.3 HTTP Event Collector Setup

Configure HTTP Event Collector (HEC) to receive logs from external sources including cloud platforms and security tools.

```
# Enable HEC
curl -k -u admin:password https://localhost:8089/services/data/inputs/http \
-d name=hec \
-d index=security_events \
-d token=your-hec-token
```

```
# Configure HEC settings

curl -k -u admin:password
https://localhost:8089/services/data/inputs/http/hec \

-d enableSSL=1 \

-d useDeploymentServer=0
```

Chapter 10: Splunk Detection Rules

10.1 Detection Rule Development

Develop effective detection rules using Splunk Search Processing Language (SPL) to identify security threats and anomalies in real-time.

10.2 Brute Force Detection

Detect brute force attacks by monitoring authentication failures and suspicious login patterns across multiple systems.

```
# Brute Force Detection SPL

index=security_events (authentication_failure OR login_failed OR
failed_login)

| stats count by src_ip, user, _time

| where count > 5

| eval threat_type="Brute Force Attack"

| eval mitre_technique="T1110"

| table _time, src_ip, user, count, threat_type, mitre_technique
```

10.3 Privilege Escalation Detection

Monitor for privilege escalation attempts by tracking user privilege changes and suspicious administrative activities.

```
# Privilege Escalation Detection SPL

index=security_events (useradd OR usermod OR groupadd OR sudo)

| stats count by src_ip, user, command

| where count > 3

| eval threat_type="Privilege Escalation"

| eval mitre_technique="T1068"

| table _time, src_ip, user, command, count, threat_type, mitre_technique
```

Chapter 13: Automated Response Implementation

13.1 Response Automation Overview

Implement automated responses to security threats to reduce response time and minimize the impact of security incidents. Automated responses should be carefully designed to avoid false positives and unintended consequences.

13.2 IP Blocking Automation

Automatically block malicious IP addresses when threats are detected. This response can be implemented through firewall rules or network access controls.

```
#!/bin/bash

# IP Blocking Script

MALICIOUS_IP=$1

FIREWALL_RULE="iptables -A INPUT -s $MALICIOUS_IP -j DROP"


# Add firewall rule

sudo $FIREWALL_RULE


# Log the action

echo "$(date): Blocked IP $MALICIOUS_IP" >> /var/log/soc/ip_blocks.log


# Create Jira ticket

python3 /opt/soc/scripts/jira_integration.py \

--summary "IP Blocked: $MALICIOUS_IP" \

--description "Automatically blocked malicious IP address" \

--severity "Medium"
```

13.3 User Account Management

Automatically disable compromised user accounts to prevent further unauthorized access and privilege escalation attempts.

```
#!/bin/bash

# User Account Management Script

COMPROMISED_USER=$1


# Disable user account

sudo usermod -L $COMPROMISED_USER


# Log the action

echo "$(date): Disabled user $COMPROMISED_USER" >>
/var/log/soc/user_actions.log
```

```
# Create Jira ticket
python3 /opt/soc/scripts/jira_integration.py \
--summary "User Disabled: $COMPROMISED_USER" \
--description "Automatically disabled compromised user account" \
--severity "High"
```

Appendix A: Configuration Files

A.1 Splunk Configuration Files

Essential Splunk configuration files for the SOC project implementation.

```
# /opt/splunk/etc/system/local/indexes.conf
[security_events]
homePath = $SPLUNK_DB/security_events/db
coldPath = $SPLUNK_DB/security_events/colddb
thawedPath = $SPLUNK_DB/security_events/thaweddb
maxTotalDataSizeMB = 10000
frozenTimePeriodInSecs = 7776000

[cloud_logs]
homePath = $SPLUNK_DB/cloud_logs/db
coldPath = $SPLUNK_DB/cloud_logs/colddb
thawedPath = $SPLUNK_DB/cloud_logs/thaweddb
maxTotalDataSizeMB = 5000
frozenTimePeriodInSecs = 2592000
```

A.2 Wazuh Configuration Files

Essential Wazuh configuration files for endpoint detection and response.

```
# /var/ossec/etc/ossec.conf

yes
yes

soc_cluster
soc_manager
master
your_cluster_key
1516

firewall-drop
local
6
600
```