CYART SOC Internship – Week 2

Security Operations & Incident Response Documentation

Prepared by: **Kalash Mahajan**

Date: **2026-02-11**

Executive Summary

This document presents the activities completed during Week 2 of Security Operations and Incident Response practice. The objective was to gain structured exposure to SOC workflows, including alert prioritization, incident classification, triage validation, forensic evidence preservation, and full attack-to-response simulation.

A controlled lab environment was used to simulate real-world attack scenarios. A vulnerable system (Metasploitable2) was exploited using Metasploit, monitored by Wazuh SIEM, and contained using CrowdSec. Alerts were analyzed, mapped to MITRE ATT&CK techniques, and validated through external threat intelligence platforms such as AlienVault OTX and VirusTotal.

Additionally, forensic procedures including volatile data collection and memory acquisition were practiced with integrity verification using SHA256 hashing. This project demonstrates structured incident response methodology aligned with industry best practices including NIST SP 800-61 and MITRE ATT&CK.

1. Introduction

1.1 Objective of the Week

The objective of this week was to develop foundational and practical skills in Security Operations Center (SOC) activities, including alert prioritization, incident classification, alert triage, evidence preservation, and end-to-end incident response simulation.

1.2 Scope of Work

This project involved both theoretical study and hands-on implementation. The scope included alert management, incident response documentation, triage validation using threat intelligence platforms, forensic evidence handling, and a capstone full attack-to-response simulation.

1.3 Tools and Technologies Used

- Wazuh SIEM
- Metasploit Framework
- CrowdSec
- Velociraptor
- Windows Server 2022 VM
- AlienVault OTX
- VirusTotal
- Google Sheets & Google Docs
- VMware Virtual Environment

1.4 Lab Architecture Overview

The practical activities were performed in an isolated virtual lab environment using VMware. The architecture consisted of:

- **Kali Linux** – Attacker machine (Metasploit, Nmap)
- **Metasploitable2** – Vulnerable Linux target
- **Windows Server 2022** – Monitored endpoint
- **Wazuh Manager & Indexer** – Centralized logging and alerting
- **CrowdSec** – IP-based blocking and containment
- **Threat Intelligence Platforms** – AlienVault OTX and VirusTotal

All systems operated within a segmented virtual network to simulate enterprise SOC monitoring conditions while maintaining isolation and control.

2. Theoretical Foundations

2.1 Alert Priority Levels

Alert severity was classified into Critical, High, Medium, and Low based on impact and urgency. Risk scoring concepts such as CVSS were studied to understand quantitative vulnerability assessment. Alert prioritization considered exploit likelihood, asset criticality, and potential business impact.

2.2 Incident Classification

Incidents were categorized based on type (e.g., malware, phishing, brute force). MITRE ATT&CK techniques were referenced for standardized mapping. Incidents were enriched using contextual metadata such as source IP, timestamps, hostnames, and file hashes.

2.3 Basic Incident Response Lifecycle

The incident response lifecycle followed standard phases: Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned. This structured approach ensured consistent and effective response.

2.4 Risk Assessment Approach

Alert prioritization was performed using a structured risk-based methodology. Severity levels were determined by evaluating:

- Exploit likelihood (public exploit availability)
- Asset criticality (server vs workstation impact)
- Potential business impact (service disruption, privilege escalation)
- Observed attacker behavior

CVSS scoring concepts were referenced for vulnerability severity interpretation. Alerts involving exploitation attempts or elevated privileges were classified as High or Critical, while reconnaissance or benign anomalies were categorized as Medium or Low after validation.

This approach ensured consistent prioritization aligned with SOC operational standards.

3. Practical Implementation

3.1 Alert Management Practice

An alert classification matrix was created using Google Sheets, mapping alert types to MITRE tactics. Risk-based prioritization was implemented, and sample alerts were scored. Incident tickets were drafted with structured fields including title, description, indicators, priority, and assignee. Escalation communication was simulated for critical alerts.

3.1.1 Workflow Summary

1. Created an alert classification matrix in Google Sheets to standardize alert handling.
2. Mapped alert types to relevant MITRE ATT&CK tactics and techniques.
3. Applied risk-based prioritization considering impact, asset criticality, and exploit likelihood.
4. Assigned severity levels (Critical, High, Medium, Low) based on evaluation criteria.
5. Drafted structured incident tickets including title, description, indicators, priority, and assignee.
6. Simulated escalation procedure for critical alerts through formal communication format.
7. Documented all outputs (Excel sheet, ticket format, screenshots) in the Week 2 repository folder.

3.2 Response Documentation

A structured incident response template was developed following SANS-style formatting, including Executive Summary, Timeline, Impact Analysis, Remediation Steps, and Lessons Learned. Investigation steps and a phishing response checklist were documented.

3.2.1 Workflow Summary

1. Designed an incident response report template using structured SANS-style formatting.
2. Included key sections: Executive Summary, Timeline, Impact Analysis, Remediation Steps, and Lessons Learned.
3. Documented investigation steps chronologically to maintain audit clarity.
4. Created a phishing response checklist to standardize validation procedures (header analysis, link reputation, affected user identification).
5. Ensured documentation clarity for both technical and management-level audiences.
6. Saved finalized documentation and supporting screenshots within the Week 2 repository folder.

3.3 Alert Triage Practice

A high-severity Wazuh alert involving PowerShell script execution in the Windows Temp directory was analyzed. The activity was reviewed under the SYSTEM account context. Threat intelligence validation was conducted using AlienVault OTX and VirusTotal. The file hash returned no malicious reputation and no associated threat campaigns. The alert was classified as a false positive after technical validation.

3.3.1 Workflow Summary

1. Reviewed the high-severity alert generated in Wazuh and extracted key details (Alert ID, host, source IP, severity level).
2. Analyzed event logs to understand execution context (PowerShell running under SYSTEM account).
3. Identified and extracted the file hash associated with the script execution.
4. Performed threat intelligence validation using AlienVault OTX and VirusTotal.
5. Confirmed no malicious reputation, threat pulses, or vendor detections.
6. Documented findings and determined the alert to be a false positive based on technical evidence.
7. Updated alert status and recorded triage outcome in documentation.

3.4 Evidence Preservation

Volatile data was collected using Velociraptor (Windows.Network.Netstat artifact). A memory acquisition was performed for forensic practice. Evidence integrity was validated using SHA256 hashing.

SHA256 Hash:
20492a4d0d84f8beb1767f6616229f85d44c2827b64bdbfb260ee12fa1109e0e

A chain-of-custody record was documented to ensure integrity and accountability.

3.4.1 Workflow Summary

1. Deployed Velociraptor agent on the Windows Server 2022 virtual machine.
2. Executed the Windows.Network.Netstat artifact to collect live network connection data.
3. Exported the collected volatile data in CSV format for preservation and later analysis.
4. Performed memory acquisition for forensic practice and securely stored the memory image.
5. Generated a SHA256 hash of the memory dump using system hashing utilities to verify integrity.
6. Recorded the hash value and collection details in a structured chain-of-custody table.
7. Ensured secure storage of evidence files to maintain integrity and accountability.

4. Capstone Project – Full Alert-to-Response Cycle

4.1 Attack Simulation

A vulnerability on a Metasploitable2 machine (vsftpd 2.3.4 backdoor) was exploited using Metasploit to simulate a real-world attack scenario.

4.2 Detection and Alert Generation

Wazuh monitored system activity and generated alerts corresponding to suspicious behavior and authentication attempts.

4.3 Triage and Analysis

Alerts were analyzed, correlated with logs, and mapped to MITRE techniques such as Initial Access (T1190). Source IP identification and behavior review were conducted.

4.3.1 MITRE ATT&CK Mapping

Tactic	Technique ID	Description
Initial Access	T1190	Exploit Public-Facing Application
Credential Access	T1110	Brute Force Attempts
Privilege Escalation	T1548	Abuse of Elevation Control Mechanism

4.4 Containment and Response Actions

Containment measures included isolating the affected system and implementing IP blocking mechanisms using CrowdSec.

4.5 Incident Timeline

Timestamp	Event	Source	Action Taken
Exploit Execution	Metasploit triggered	Kali Linux	Shell obtained
Alert Generated	Wazuh	Target System	Alert logged
Triage Initiated	SOC Analyst	Wazuh Dashboard	Log correlation performed
Threat Intel Check	OTX / VirusTotal	File Hash	No malicious reputation
Containment	CrowdSec	Source IP	IP blocked
Verification	Network Test	SOC	Access restricted

4.6 Reporting and Documentation

A structured incident report was drafted including executive summary, timeline of events, analysis findings, and recommended security improvements. A stakeholder briefing was prepared for non-technical management.

4.7 Capstone Workflow Summary

1. Configured attacker (Kali Linux) and target (Metasploitable2) within the virtual lab environment.
2. Executed exploit module targeting vsftpd 2.3.4 vulnerability to simulate initial compromise.
3. Confirmed successful shell access on the target system.
4. Monitored generated logs and alerts within Wazuh SIEM.
5. Correlated alert data with source IP and attack behavior.
6. Mapped observed activity to relevant MITRE ATT&CK techniques.
7. Validated indicators using threat intelligence platforms where applicable.
8. Implemented containment by isolating the affected system and blocking the attacker IP using CrowdSec.
9. Verified containment effectiveness through network testing.
10. Documented findings in formal incident report format and prepared stakeholder briefing.

5. Key Learnings and Skills Developed

• Practical SOC workflow exposure
• Alert triage and threat intelligence validation
• Incident documentation and reporting
• Evidence handling and forensic integrity verification
• End-to-end attack simulation and response coordination

6. Limitations and Challenges

During the practical implementation, certain environmental and configuration challenges were encountered:

- Legacy service compatibility issues during SSH validation
- Temporary dashboard instability during SIEM configuration
- Environment constraints affecting full memory acquisition tooling
- Lab-based simulation limitations compared to production-scale infrastructure

These challenges were mitigated through manual validation and structured troubleshooting.

7. Recommendations

Based on the observations and simulated incident handling, the following improvements are recommended:

- Enable automated alert correlation for faster triage
- Strengthen SSH configurations to prevent brute-force attempts
- Integrate SOAR-based automation for containment workflows
- Enhance logging verbosity for improved forensic visibility
- Conduct periodic red-team simulations to test detection readiness

These measures would further strengthen proactive detection and response capabilities.

8. Conclusion

This week’s activities provided comprehensive exposure to SOC operations, including detection, analysis, response, documentation, and forensic handling. The structured approach strengthened both technical and analytical skills required for real-world security operations.