Mini Security Operations Center (SOC) Project Using Elastic Stack

---

BY:

Blue Shield Syndicate

Supervisor:

Eng: Wessam Elkhaligy

Digital Egypt Pioneers

Team Members

Ministry of Communications and Information Technology

Mohamed Mahmoud Abdelaleem

Abdallah Mohamed Khaled

Mostafa Mohamed Zanon

Mina Osama Naseem

Hesham Othman Emam

Contents

1. Introduction

➢ Importance of SOCs

- General idea about the importance of SOCs in organizations.
- Why we build a Mini SOC in a lab?
- Hands-on training.
- Understanding SIEM operations.
- Simulating real-world attacks.

➢ Tools Used

- Elastic Stack
- Suricata IDS
- Windows & Linux logs

2. SOC Design Overview

Data Source Layer

Collecting data from Windows, Linux, Firewall, IDS.

Collection Layer

Data ingestion using Elastic Agent.

Processing Layer

Normalization and parsing of logs.

Storage Layer

Storing data inside Elasticsearch indices.

Analysis Layer

Kibana dashboards + Detection rules.

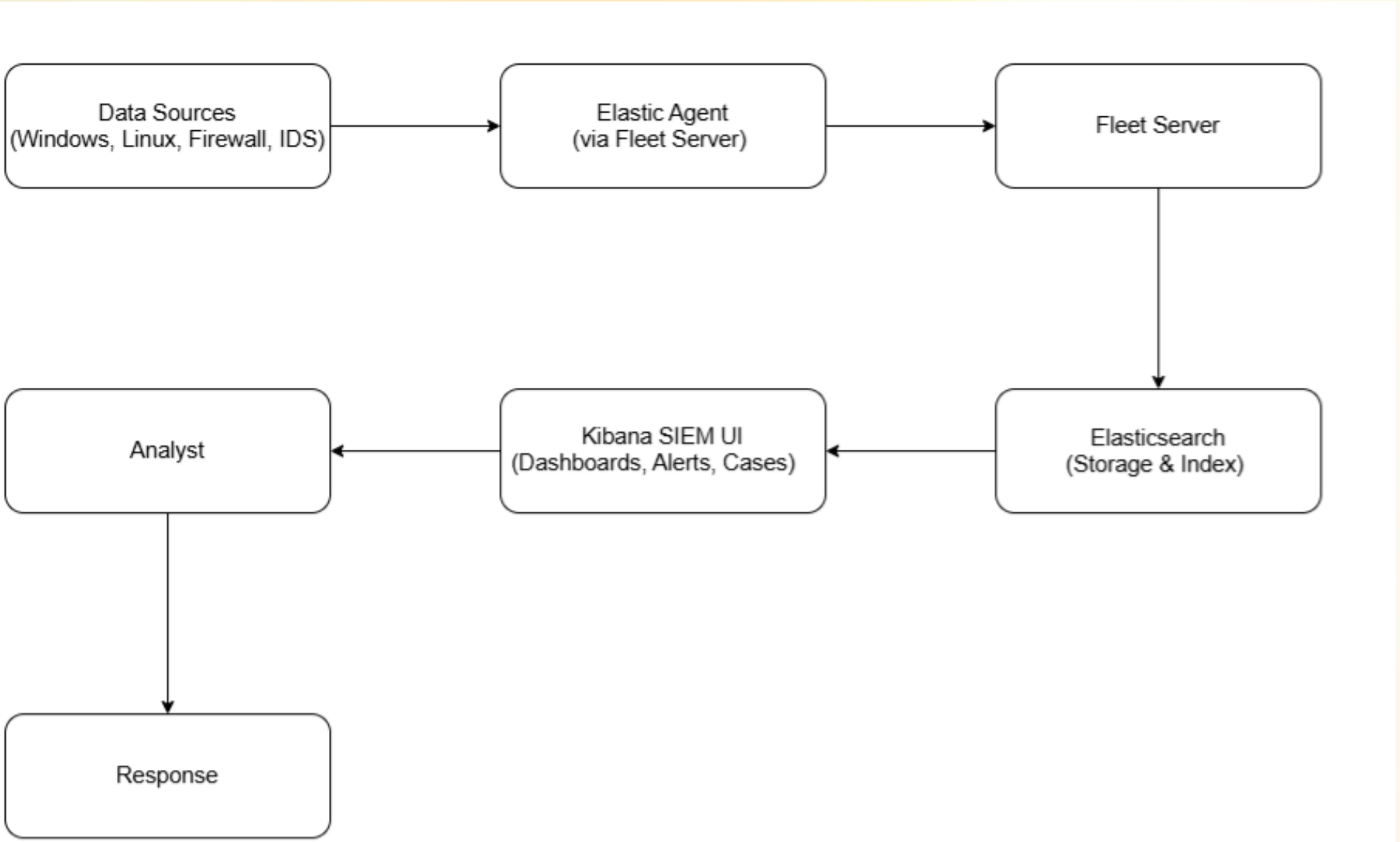
Response Layer

Incident investigation and response.

3. Architecture Components

1. Elasticsearch
2. Kibana
3. Fleet Server
4. Elastic Agent
5. Suricata IDS
6. Windows & Linux endpoints

4. Data Flow


Data Sources
(Windows, Linux, Firewall, IDS)
Elastic Agent
(via Fleet Server)
Fleet Server
Elasticsearch
(Storage & Index)
Kibana SIEM UI
(Dashboards, Alerts, Cases)
Analyst
Response

5. Use Cases Overview

- Brute Force Attack
- Command Injection Attack
- Malware Attack

5.1. Brute Force Attack

1

Repeated SSH or Windows login attempts.

2

Windows Event ID 4625 / Linux auth.log.

3

SIEM detects high failed-login patterns.

4

Key information: IP, username, attempts count, geo location.

BRUTE FORCE ATTACK

5.1. Brute Force Attack

IOC + MITRE – Brute Force

- IOCs: IP address, username, Event 4625.
- MITRE Technique: T1110 – Brute Force.

Detection Rule – Brute Force

- Threshold: 5+ failed login attempts within 2–5 minutes.
- Data sources: Windows logs, Linux auth.log.
- Goal: Early detection of brute force attempts.

5.2. Command Injection Attack

1	2
Vulnerable PHP page using shell_exec().	Attacker executes commands via ?cmd=whoami.
3	**4**
Logs show abnormal command execution.	SIEM detects suspicious execution patterns.

5.2. Command Injection Attack

IOC + MITRE – Command Injection

- IOC: URL payloads, attacker IP, executed commands, User-Agent.
- MITRE Technique: T1059.003.

Detection Rule – Command Injection

- Detecting suspicious URL patterns.
- Detecting php/nginx/apache spawning bash or cmd.
- Goal: Prevent unauthorized command execution on the server.

5.3. Malware Attack

1. A malicious attachment is executed.
2. PowerShell encoded commands appear.
3. Antivirus & Sysmon logs show malicious behavior.
4. SIEM generates alert on malware activity.

5.3. Malware Attack

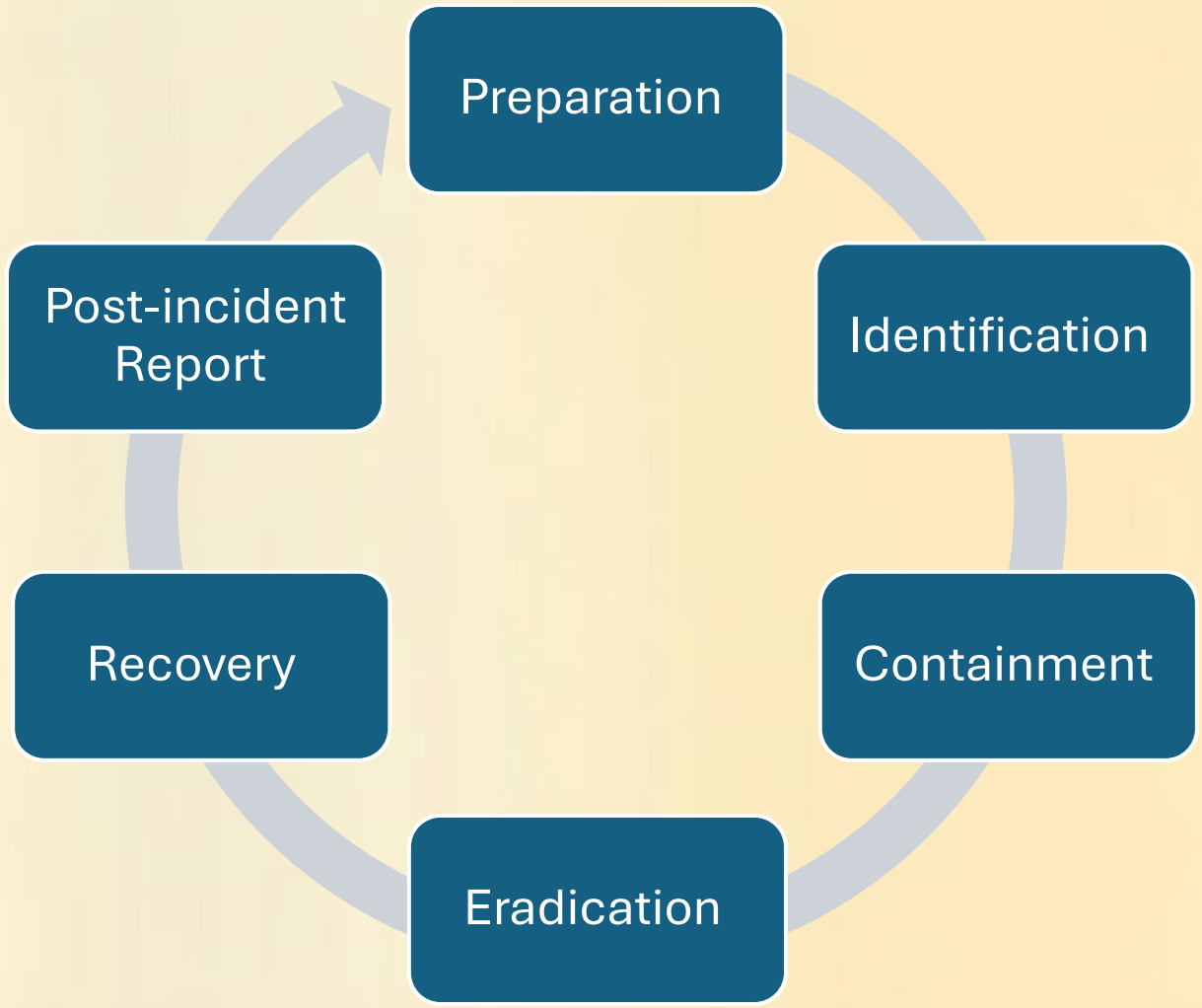
IOC + MITRE – Malware

- IOC: File hash, domain, registry changes, encoded commands.
- MITRE Techniques: T1566, T1059.

Detection Rule – Malware

- AV malware detection.
- Suspicious execution from temp/appdata folders.
- Encoded PowerShell command detection.

6. Incident Response Strategy (General Workflow)



6.1. Brute Force

Reset the compromised password.

Kill active sessions.

Revoke malicious tokens/keys.

Remove persistence mechanisms.

6.2. Malware (Virus)

1. Kill the malicious process.
2. Delete the executable/files.
3. Clean persistence mechanisms.
4. Re-image the host (If necessary).

6.3. Command Injection

- Terminate the reverse shell.
- Delete dropped webshells.
- Patch the Vulnerability.
- Sanitize affected databases/files.

7. Improvement & KPIs

1	2	3
MTTD improved: 5 mins → 20 sec.	MTTR improved: 5 mins → 1 min.	Detection coverage: 20% → 80%.

4	5
False positives: 30% → 10%.	Alerts increased and accuracy improved.

8. Project Summary

Built a complete Mini SOC.

Handled 3 different attack scenarios.

Implemented Detection → Investigation → Containment.

Enhanced performance through rule tuning.

Practical and realistic SOC environment model.

Thank You