Building and Operating a Mini Security Operations Center (SOC)

1. Project Planning

The project aims to build and simulate a functional mini–Security Operations Center (SOC) environment. This SOC will centralize the collection, processing, analysis, and response to security events using open-source and Elastic Stack technologies such as Elasticsearch, Kibana, Fleet Server, and Elastic Agent. The mini-SOC helps students and practitioners understand how real SOC environments operate by handling event logs, alerts, and security incidents.

1.1 Scope

This project covers the design and deployment of a small-scale SOC setup using Elastic Agent and Fleet Server for unified log collection. It includes connecting multiple log sources, establishing log ingestion and analysis workflows, developing detection rules, and producing dashboards and reports for visualization.

1.2 Objectives

- Design a mini-SOC architecture with multiple log sources.
- Deploy the Elastic Stack (Elasticsearch, Kibana, Fleet Server, Elastic Agent) as a centralized SIEM platform.
- Ingest logs from Windows, Linux, Firewall, and IDS systems via Elastic Agent integrations.
- Analyze and visualize security data using Kibana dashboards and Security Analytics.
- Implement alerting and triage processes using Elastic Security features.

2. Tools and Technologies

Category	Tool / Technology
Operating System	Ubuntu Server, Kali Linux, Windows
SIEM Platform	Elasticsearch, Kibana, Fleet Server, Elastic Agent
Log Collection	Elastic Agent via Fleet integrations (Windows, Linux, Firewall, IDS)
IDS	Suricata
EDR	Elastic Defend
Firewall	pfSense (or simulated logs)
Ticketing & Reporting	PDF

3. Stakeholder Analysis

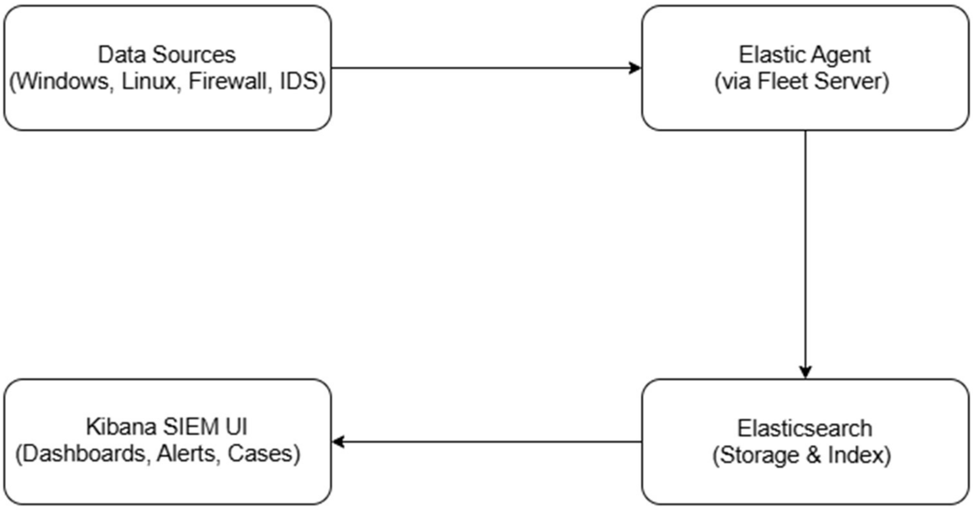
1. ***Project Supervisor/Instructor:*** Provides guidance, reviews deliverables, and ensures alignment with project objectives.
2. ***SOC Analyst:*** Implements SOC setup, configures tools, monitors data, and analyzes incidents.
3. ***System Administrator:*** Provides log data from Windows, Linux, and Firewall systems.

4. SOC Architecture Overview

The SOC architecture is designed around the Elastic Stack, using Fleet Server and Elastic Agent for centralized log collection and management. The architecture defines how data is generated, collected, processed, stored, and analyzed.

- Data Source Layer: Log generation from Windows, Linux, Firewall, and IDS systems.
- Collection Layer: Elastic Agents installed on endpoints collect system, application, and security logs, managed centrally by Fleet Server.
- Processing Layer: Data streams are normalized automatically by Elastic integrations before indexing into Elasticsearch.
- Storage Layer: Elasticsearch indexes and stores logs for search, correlation, and long-term analysis.
- Analysis Layer: Kibana provides dashboards, threat detection rules, and visualization.
- Response Layer: Security analysts investigate, triage, and respond to alerts in Kibana's Security app.

Data Flow Diagram:



5. Database Design (Log Storage & Flow)

Elasticsearch serves as the central database for log storage. Each integration creates its own data stream and index pattern, allowing for efficient querying and filtering. Logs are stored in JSON format and Bytes with structured fields for analytics and correlation.

Examples of index patterns:

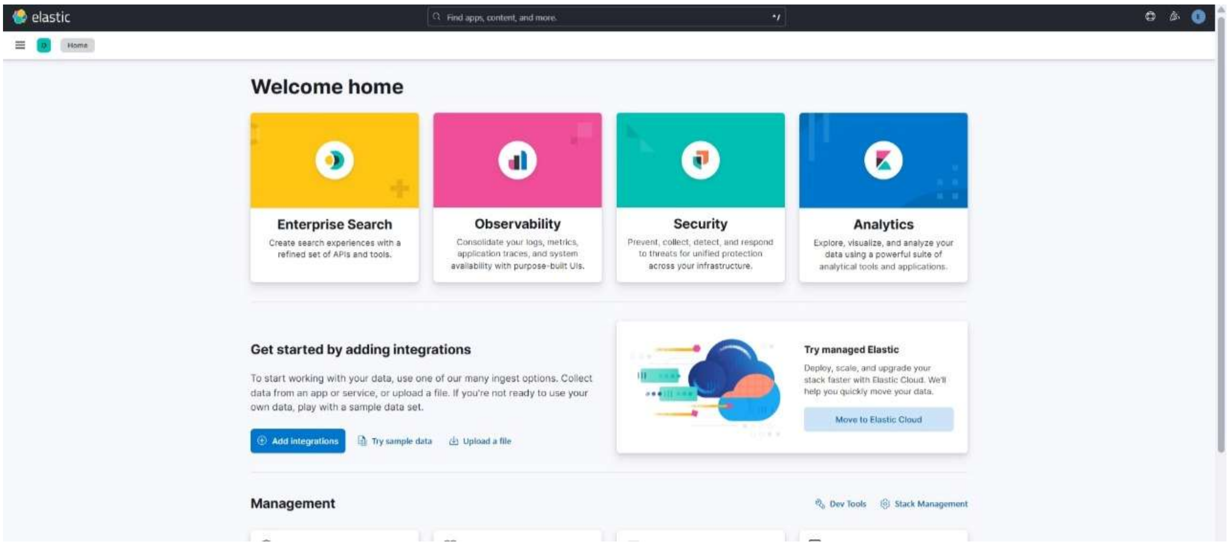
(Windows Event Logs, Linux System Logs, Firewall Logs, IDS Alerts) → logs-*

Each document in Elasticsearch includes fields such as @timestamp, source.ip, destination.ip, event.action, event.category, event.severity, host.name, and agent.name.

6. Security Controls

- Access Control: Role-based access management (RBAC) in Kibana for SOC roles.
- Integrity: Elastic Agents verify event delivery using ACK mechanisms and digital signatures.
- Authentication: API key and Fleet enrollment tokens for agent registration.
- Backup: Automated Elasticsearch snapshots stored on secure external storage.

7. UI/UX Design



elastic — Find apps, content, and more.

Home

Welcome home

Enterprise Search
Create search experiences with a refined set of APIs and tools.

Observability
Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.

Security
Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.

Analytics
Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.

Get started by adding integrations

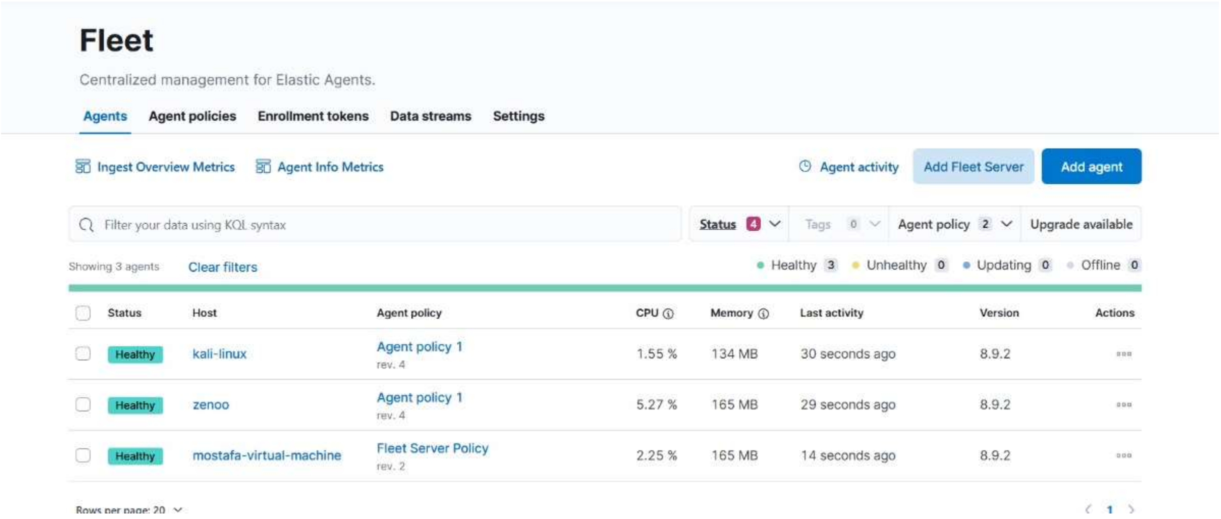
To start working with your data, use one of our many ingest options. Collect data from an app or service, or upload a file. If you're not ready to use your own data, play with a sample data set.

Add integrations | Try sample data | Upload a file

Try managed Elastic
Deploy, scale, and upgrade your stack faster with Elastic Cloud. We'll help you quickly move your data.

Move to Elastic Cloud

Management — Dev Tools | Stack Management



Fleet

Centralized management for Elastic Agents.

Agents | Agent policies | Enrollment tokens | Data streams | Settings

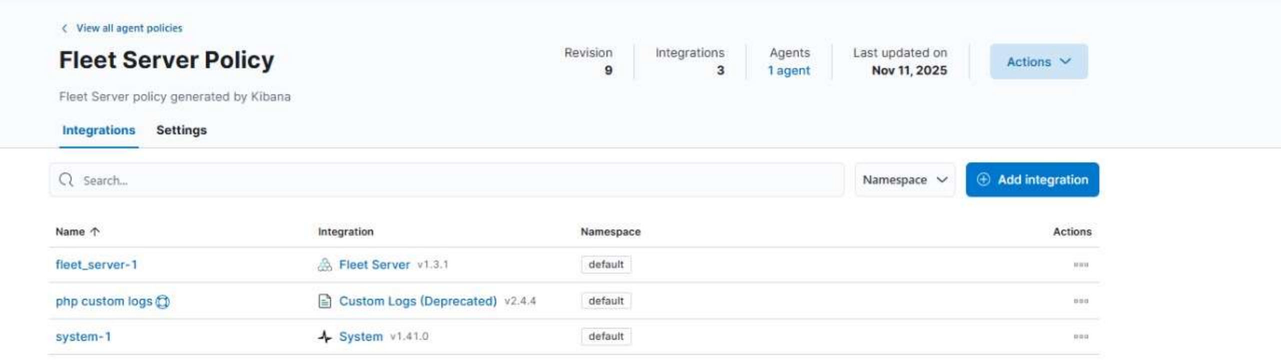
Ingest Overview Metrics | Agent Info Metrics | Agent activity | Add Fleet Server | Add agent

Filter your data using KQL syntax | Status 4 | Tags 0 | Agent policy 2 | Upgrade available

Showing 3 agents | Clear filters | ● Healthy 3 | ● Unhealthy 0 | ● Updating 0 | ● Offline 0

□	Status	Host	Agent policy	CPU	Memory	Last activity	Version	Actions
□	Healthy	kali-linux	Agent policy 1 rev. 4	1.55 %	134 MB	30 seconds ago	8.9.2	⋯
□	Healthy	zenoo	Agent policy 1 rev. 4	5.27 %	165 MB	29 seconds ago	8.9.2	⋯
□	Healthy	mostafa-virtual-machine	Fleet Server Policy rev. 2	2.25 %	165 MB	14 seconds ago	8.9.2	⋯

Rows per page: 20 | ‹ 1 ›



‹ View all agent policies

Fleet Server Policy

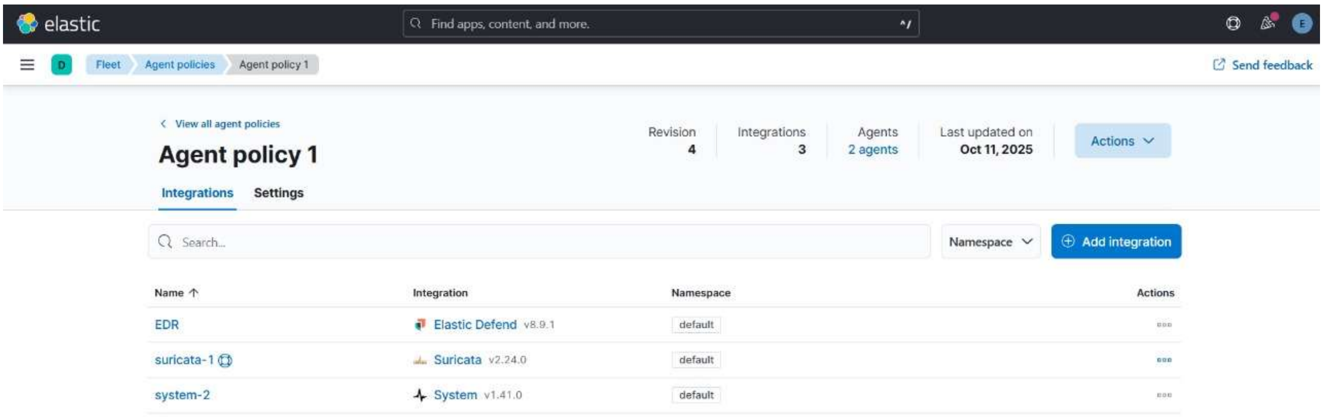
Fleet Server policy generated by Kibana

Revision 9 | Integrations 3 | Agents 1 agent | Last updated on Nov 11, 2025 | Actions

Integrations | Settings

Search... | Namespace | Add integration

Name ↑	Integration	Namespace	Actions
fleet_server-1	Fleet Server v1.3.1	default	⋯
php custom logs	Custom Logs (Deprecated) v2.4.4	default	⋯
system-1	System v1.41.0	default	⋯



elastic — Find apps, content, and more.

Fleet › Agent policies › Agent policy 1 | Send feedback

‹ View all agent policies

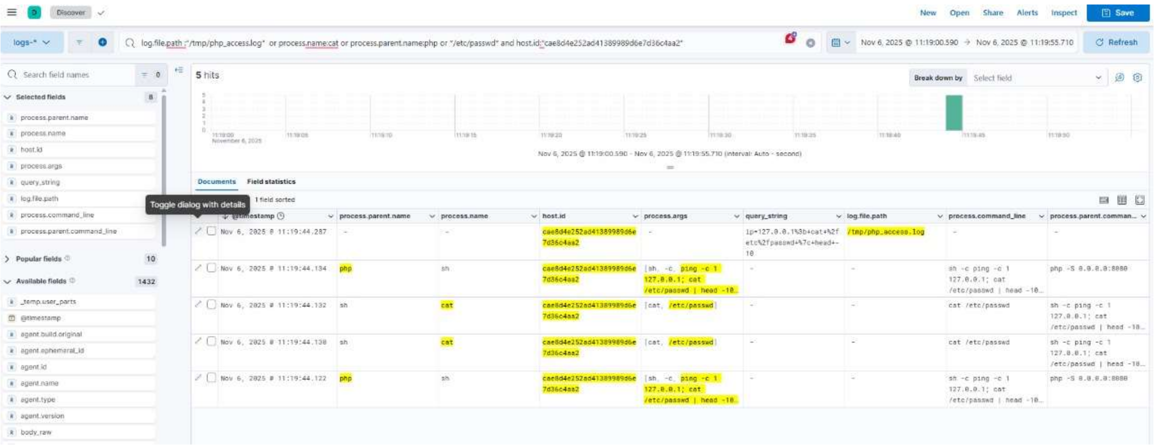
Agent policy 1

Revision 4 | Integrations 3 | Agents 2 agents | Last updated on Oct 11, 2025 | Actions

Integrations | Settings

Search... | Namespace | Add integration

Name ↑	Integration	Namespace	Actions
EDR	Elastic Defend v8.9.1	default	⋯
suricata-1	Suricata v2.24.0	default	⋯
system-2	System v1.41.0	default	⋯



```
log.file.path :"/tmp/php_access.log" or process.name:cat or process.parent.name:php or "/etc/passwd" and host.id:"cae8d4e252ad41389989d6e7d36c4aa2"
```

Nov 6, 2025 @ 11:19:00.590 → Nov 6, 2025 @ 11:19:55.710

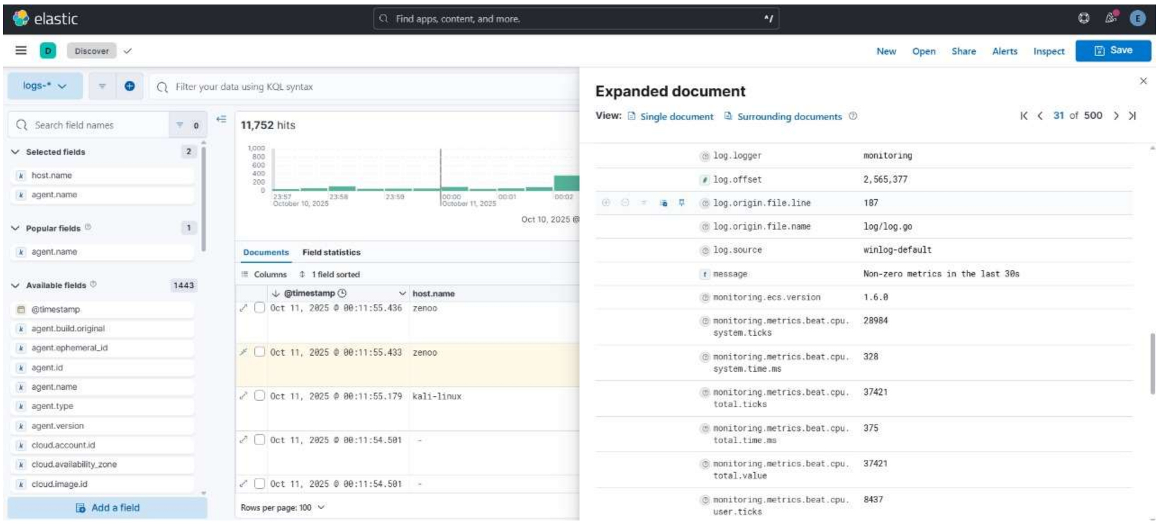
5 hits

Nov 6, 2025 @ 11:19:00.590 - Nov 6, 2025 @ 11:19:55.710 (interval: Auto - second)

Selected fields: process.parent.name, process.name, host.id, process.args, query_string, log.file.path, process.command_line, process.parent.command_line

Available fields: _temp.user_parts, @timestamp, agent.build.original, agent.ephemeral_id, agent.id, agent.name, agent.type, agent.version, body_raw

@timestamp	process.parent.name	process.name	host.id	process.args	query_string	log.file.path	process.command_line	process.parent.comman…
Nov 6, 2025 @ 11:19:44.287	-	-	cae8d4e252ad41389989d6e7d36c4aa2	-	ip=127.0.0.1%3b+cat+%2fetc%2fpasswd+%7c+head+-10	/tmp/php_access.log	-	-
Nov 6, 2025 @ 11:19:44.134	php	sh	cae8d4e252ad41389989d6e7d36c4aa2	[sh, -c, ping -c 1 127.0.0.1; cat /etc/passwd | head -10…	-	-	sh -c ping -c 1 127.0.0.1; cat /etc/passwd | head -10…	php -S 0.0.0.0:8080
Nov 6, 2025 @ 11:19:44.132	sh	cat	cae8d4e252ad41389989d6e7d36c4aa2	[cat, /etc/passwd]	-	-	cat /etc/passwd	sh -c ping -c 1 127.0.0.1; cat /etc/passwd | head -10…
Nov 6, 2025 @ 11:19:44.130	sh	cat	cae8d4e252ad41389989d6e7d36c4aa2	[cat, /etc/passwd]	-	-	cat /etc/passwd	sh -c ping -c 1 127.0.0.1; cat /etc/passwd | head -10…
Nov 6, 2025 @ 11:19:44.122	php	sh	cae8d4e252ad41389989d6e7d36c4aa2	[sh, -c, ping -c 1 127.0.0.1; cat /etc/passwd | head -10…	-	-	sh -c ping -c 1 127.0.0.1; cat /etc/passwd | head -10…	php -S 0.0.0.0:8080



11,752 hits

Selected fields: host.name, agent.name

Popular fields: agent.name

Available fields: @timestamp, agent.build.original, agent.ephemeral_id, agent.id, agent.name, agent.type, agent.version, cloud.account.id, cloud.availability_zone, cloud.image.id

@timestamp	host.name
Oct 11, 2025 @ 00:11:55.436	zenoo
Oct 11, 2025 @ 00:11:55.433	zenoo
Oct 11, 2025 @ 00:11:55.179	kali-linux
Oct 11, 2025 @ 00:11:54.501	-
Oct 11, 2025 @ 00:11:54.501	-

log.logger	monitoring
log.offset	2,565,377
log.origin.file.line	187
log.origin.file.name	log/log.go
log.source	winlog-default
message	Non-zero metrics in the last 30s
monitoring.ecs.version	1.6.0
monitoring.metrics.beat.cpu.system.ticks	28984
monitoring.metrics.beat.cpu.system.time.ms	328
monitoring.metrics.beat.cpu.total.ticks	37421
monitoring.metrics.beat.cpu.total.time.ms	375
monitoring.metrics.beat.cpu.total.value	37421
monitoring.metrics.beat.cpu.user.ticks	8437

8. Attacks Scenarios

1. *Brute Force Attack*

- Multiple SSH brute-force attempts against the VM were detected, generating numerous authentication events with outcome: failure.

- Elastic Security flagged the activity with "Potential Linux Hack Tool Launched" alerts tied to the hydra process, recording process arguments, source IP, and failed login details.

2. *Command Injection Attack*

- A vulnerable PHP endpoint allowed user-supplied input to be executed by the shell, resulting in remote command execution traces (web server spawning shell/processes and unexpected exec-style arguments in logs).

- This activity indicates a command-injection compromise with high impact arbitrary commands ran on the host, showing up in process events and risking data exposure, persistence, or further lateral movement.

3. *Malware Attack (Virus)*

- Attempted Windows persistence via service creation (MITRE T1543) was observed a new service/process was created and writes were made to system locations (\ProgramData/\Windows\System32), producing process-create and file-write events.

- Elastic/endpoint telemetry flagged the activity as suspicious for persistence, recording the service-related process, parent process, and modified system paths for triage.