ELK Stack Overview

ELK stands for **Elasticsearch**, **Logstash**, and **Kibana** — the **three core tools** that make up the **Elastic Stack**. Although ELK is **not a traditional SIEM,** many SOC teams use it like one because of its **strong searching and visualization capabilities.**

Elastic Stack was originally developed to **store, search, and visualize large amounts** of data. Over time, its features made it popular in **security operations**.

Components of the ELK Stack

1. Elasticsearch

1. A **search and analytics engine.**
2. Stores and organizes data **(usually in JSON).**
3. Allows fast searching, analysis, and correlation.
4. Accessed through a RESTful API.

2. Logstash

A **data processing tool** used for **collecting and transforming data.**

A Logstash configuration has **three parts:**

- **Input:** Where the **data comes from.**
- **Filter:** How the data is **processed or normalized.**
- **Output:** Where the processed data is sent.

Logstash supports many plugins for input, output, and filters.

3. Beats

Lightweight agents installed on **endpoints.**

Beats can send data:

- **Directly to Elasticsearch** (for **storage and indexing**), or
- **Via Logstash** (if filtering or transformation is needed).

Sending via Logstash is optional.

4. Kibana

- A **web-based visualization tool.**
- Connects to **Elasticsearch to display dashboards, charts, and reports.**
- Provides front-end interaction for analysis.

Kibana Discover Tab

The **Discover tab is where SOC analysts spend most of their time.**

Key Sections in Discover:

- **Logs:** Each row is a single log with fields and values.
- **Fields Pane:** Shows parsed fields; clicking a field adds/removes filters.
- **Index Pattern:** Determines which data Elasticsearch should display.
- **Search Bar:** Enter queries using text or KQL.
- **Time Filter:** Limits logs to a specific time range.

- **Time Interval Chart:** Shows event counts over time.
- **Top Bar:** Save, open, share, or manage searches.
- **Add Filter:** Create filters without typing queries.

Index Pattern

- Tells **Kibana which data to explore.**
- Maps to **specific indices and fields.**
- One index pattern can **cover multiple indices.**
- Logs are normalized into fields and assigned a pattern.

Example: `vpn_connections` index pattern contains VPN logs.

Fields Pane

- Shows all normalized fields on the left.
- Clicking a field shows top 5 values and their percentage.
- **Clicking + adds filter showing logs with that value.**
- **Clicking – adds filter excluding that value.**
- You can also manually add filters using **Add filter**.

KQL (Kibana Query Language)

Used in the search bar to query logs.

KQL supports:

1. **Free text search**
2. **Field-based search**

---

1. Free Text Search

- Searches for terms across **all fields**.
- Example: Searching for **`security`** **returns all logs containing that exact term.**
- KQL does **not** match partial words unless you use `*` wildcard.

Example:

- **`Secur`** → **No results.**
- `United*` → Matches United, United States, etc.

Boolean Operators

- **AND** returns logs containing both terms.
  - `"United States" AND "Virginia"`
- **OR** returns logs containing either term.
  - `"United States" OR "England"`

- **NOT** excludes terms.
    - `"United States" AND NOT "Florida"`

Parentheses Usage

- Required when grouping multiple conditions.
    - `"United States" AND NOT ("Florida" OR "Georgia")`

Quotes vs Parentheses

- Quotes `" "` → exact phrase.
- **Parentheses `( )` → group logical conditions.**

---

2. Field-Based Search

Uses the **syntax: Field : Value**

You provide both:

- The field name.
- The value you want to match.

Example format: `source.ip: "10.1.1.1"`

SIEM = Detects

EDR = Detects + Protects

SOAR = Automates Response

SOAR (Security Orchestration, Automation, and Response)

Purpose

SOAR automates **SOC tasks and coordinates tools together.** As threats grow more complex and advanced, SOC teams face **challenges like alert fatigue, manual processes, too many disconnected tools,** and difficulties in communication across teams.

Challenges Faced by SOC Teams

Alert Fatigue

SOC tools generate a **massive number of alerts,** many of which are f**alse positives.** Analysts become overwhelmed and may miss real threats.

Too Many Disconnected Tools

Security tools often **don't integrate with each other**. Analysts must **switch** between multiple independent systems (firewalls, EDR, logs), **causing inefficiency and tool overload.**

Manual, Undocumented Processes

Investigation steps are often **based on tribal knowledge** instead of documented **playbooks**. This slows investigations and increases response time.

Talent Shortage

Finding skilled SOC analysts is difficult. Combined with **high alert volume, existing analysts become overworked**, reducing efficiency and increasing the time adversaries have to operate.

SOAR as a Unified SOC Tool

SOAR **unifies all security tools** used in a SOC. With SOAR, analysts **do not need to switch** between SIEM, EDR, Firewall, and other security tools. They can operate all these tools within a single SOAR interface.

SOAR **also provides ticketing and case management features so analysts can document, track, and resolve incidents** in a structured way.

The core strength of a SOAR tool comes from the following three main capabilities:

Orchestration

SOC analysts traditionally switch between multiple tools (SIEM, TI platforms, IAM systems, ticketing tools) when investigating alerts, **which slows down response time.**

Orchestration solves this by connecting these tools inside a SOAR platform and coordinating them through predefined workflows called **playbooks**.

How Orchestration Helps

- Integrates tools from **different vendors into one interface.**
- Automates the steps analysts **normally perform manually.**
- Ensures consistent and efficient investigations.

Example Playbook for VPN Brute Force

1. Receive alert from SIEM.
2. Query SIEM to check if the user normally uses that IP.
3. Check threat intelligence sources for IP reputation.
4. Query SIEM for any successful logins.
5. Escalate to containment (e.g., disable user).

Playbooks are dynamic: the next step depends on previous results. If activity looks normal, the playbook may stop early.

Automation

The art of coordinating with **multiple tools through predefined actions** (Playbooks), which we studied in Orchestration, can be automated.

Automation means **no more manual clicks needed from SOC analysts.** SOAR will itself follow the playbooks.

Automated VPN Brute Force Playbook

- SOAR **receives the alert from SIEM.**
- It automatically queries the SIEM for the user's historical logins.
- It automatically **verifies the IP's reputation** through TI platforms.
- If the IP is malicious, it automatically **disables the user from the IAM.**
- Lastly, it automatically opens a ticket in the ticketing system with all the details to initiate an investigation.

This saves a tremendous amount of time for SOC analysts. They can handle hundreds of alerts without burning out.

Response

SOAR also **automates the response,** as seen in its Automation capability.

For example, SOAR can follow the playbook of VPN Brute Force and:

- Block the IP on the firewall.
- Disable the user in the IAM.
- Open a ticket with all details.

However, **SOAR does not replace SOC analysts**. **Complex investigations still require an analyst.**

SOAR cannot make **judgment calls at critical points,** but analysts can. Analysts understand **threats in the broader business context.**

Playbooks for different alert types are also created by SOC analysts.

SOAR Playbooks are predefined workflows that tell the SOAR tool what actions to take during a specific investigation.