

Building a Home SIEM Lab

Abstract

In this report, I outline the setup of a home lab using Elastic SIEM and a Kali VM. I configured data forwarding from the Kali VM to the SIEM via the Elastic Beats agent, generated security events on the Kali VM using Nmap, and analyzed the logs in the SIEM through the Elastic web interface. Additionally, I created a dashboard to visualize security events and created an alert to detect them.

This home lab provides a practical environment for learning and applying key skills in security monitoring and incident response using Elastic SIEM. I gained hands-on experience with SIEM tools, improving my proficiency in security monitoring and analysis.

The description provides an overview of the functionalities of Elastic Defend.

Elastic Defend Integration

Elastic Defend provides organizations with prevention, detection, and response capabilities with deep visibility for EPP, EDR, SIEM, and Security Analytics use cases across Windows, macOS, and Linux operating systems running on both traditional endpoints and public cloud environments. Use Elastic Defend to:

- **Prevent complex attacks** - Prevent malware (Windows, macOS, Linux) and ransomware (Windows) from executing, and stop advanced threats with malicious behavior (Windows, macOS, Linux), memory threat (Windows, macOS, Linux), and credential hardening (Windows) protections. All powered by [Elastic Labs](#) and our global community.
- **Alert in high fidelity** - Bolster team efficacy by detecting threats centrally and minimizing false positives via extensive corroboration.
- **Detect threats in high fidelity** - Elastic Defend facilitates deep visibility by instrumenting the process, file, and network data in your environments with minimal data collection overhead.
- **Triage and respond rapidly** - Quickly analyze detailed data from across your hosts. Examine host-based activity with interactive visualizations. Invoke remote response actions across distributed endpoints. Extend investigation capabilities even further with the Osquery integration, fully integrated into Elastic Security workflows.
- **Secure your cloud workloads** - Stop threats targeting cloud workloads and cloud-native applications. Gain real-time visibility and control with a lightweight user-space agent, powered by eBPF. Automate the identification of cloud threats with detection rules and machine learning (ML). Achieve rapid time-to-value with MITRE ATT&CK-aligned detections honed by Elastic Security Labs.
- **View terminal sessions** - Give your security team a unique and powerful investigative tool for digital forensics and incident response (DFIR), reducing the mean time to respond (MTTR). Session view provides a time-ordered series of process executions in your Linux workloads in the form of a terminal shell, as well as the ability to replay the terminal session.

I started by creating a free Elastic Cloud account and setting up an Elasticsearch deployment. Next, I configured a Linux VM using Kali Linux with Oracle VirtualBox. I then installed and set up an agent on the Kali VM to collect logs. The agent was configured to forward these logs to my Elastic

SIEM instance. This process allowed me to effectively collect and analyze security-related events.

tier. [Learn more](#) about changing the data retention policy for this integration.

Select configuration settings

Use quick settings to configure the integration to **protect your traditional endpoints or dynamic cloud environments**. You can make configuration changes after you create the integration.

Select the type of environment you want to protect:

Traditional Endpoints (desktops, laptops, virtual machines) ▼

Elastic Defend integration added

To complete this integration, add **Elastic Agent** to your hosts to collect data and send it to Elastic Stack.

[Add Elastic Agent later](#) [Add Elastic Agent to your hosts](#)

2 Where to add this integration?

[New hosts](#) [Existing hosts](#)

Create agent policy

Add this integration to a new set of hosts by creating a new agent policy. You can add agent in the next step.

New agent policy name

Agent policy 1

☒ Collect system logs and metrics ⓘ

Add agent

Add Elastic Agents to your hosts to collect data and send it to the Elastic Stack.

[Enroll in Fleet](#) [Run standalone](#)

Enroll an Elastic Agent in Fleet to automatically deploy updates and centrally manage the agent.

1 Select enrollment token

Agent policy 1 has been selected. Select which enrollment token to use when enrolling agents.

▼ [Authentication settings](#)

Enrollment token Default (7b0cde25-6a42-4749-b501-97231e9359bf) ▼

2 Install Elastic Agent on your host

Select the appropriate platform and run commands to install, enroll, and start Elastic Agent. Reuse commands to set up agents on more than one host. For aarch64, see our [downloads page](#). This guidance is for AMD but you can adapt it to your device architecture. For additional guidance, see our [installation docs](#).

Installing Elastic Agent on my host:

```
secretsika@kali: ~/elastic-agent-8.15.1-linux-x86_64/elastic-agent-8.15.1-linux-x86_64
```

File Actions Edit View Help

```
(secretsika@kali)-[~/elastic-agent-8.15.1-linux-x86_64/elastic-agent-8.15.1-linux-x86_64]
$ sudo ./elastic-agent install --url=https://56c16dd5fe2142669469b66d86183689.fleet.us-cent
rall1.gcp.cloud.es.io:443 --enrollment-token=QkhFRDk1RUJlQmZ2encxSmVwa046Uhlb195NUhRM2lwSmpfw
C1Cmld0UQ=
```

Elastic Agent will be installed at /opt/Elastic/Agent and will run as a service. Do you want to continue? [Y/n]:y

```
[ ==] Service Started [22s] Elastic Agent successfully installed, starting enrollment.
[ ==] Waiting For Enroll ... [27s] {"log.level":"info","@timestamp":"2024-09-15T14:58:25.412
-0400","log.origin":{"function":"github.com/elastic/elastic-agent/internal/pkg/agent/cmd.(*en
rollCmd).enrollWithBackoff","file.name":"cmd/enroll_cmd.go","file.line":518},"message":"Start
ing enrollment to URL: https://56c16dd5fe2142669469b66d86183689.fleet.us-central1.gcp.cloud.e
s.io:443/","ecs.version":"1.6.0"}
[ ==] Waiting For Enroll ... [30s] {"log.level":"info","@timestamp":"2024-09-15T14:58:28.553
-0400","log.origin":{"function":"github.com/elastic/elastic-agent/internal/pkg/agent/cmd.(*en
rollCmd).daemonReloadWithBackoff","file.name":"cmd/enroll_cmd.go","file.line":481},"message":
"Restarting agent daemon, attempt 0","ecs.version":"1.6.0"}
[ ==] Waiting For Enroll ... [30s] {"log.level":"info","@timestamp":"2024-09-15T14:58:28.665
-0400","log.origin":{"function":"github.com/elastic/elastic-agent/internal/pkg/agent/cmd.(*en
rollCmd).Execute","file.name":"cmd/enroll_cmd.go","file.line":299},"message":"Successfully tr
iggered restart on running Elastic Agent.","ecs.version":"1.6.0"}
Successfully enrolled the Elastic Agent.
[ ==] Done [30s]
Elastic Agent has been successfully installed.
```

```
(secretsika@kali)-[~/elastic-agent-8.15.1-linux-x86_64/elastic-agent-8.15.1-linux-x86_64]
$
```

After the installation, I made sure that the agent has been correctly installed on Kali Linux by running the following command:

[illegible]

We then run a nmap scan to detect open ports and identify services running on those ports:

```
secretsika@kali: ~/elastic-agent-8.15.1-linux-x86_64
File Actions Edit View Help
(secretsika@kali)-[~/elastic-agent-8.15.1-linux-x86_64]
$ nmap -p- localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-15 15:21 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000065s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 65533 closed tcp ports (conn-refused)
PORT      STATE SERVICE
6789/tcp  open  ibm-db2-admin
6791/tcp  open  hnm

Nmap done: 1 IP address (1 host up) scanned in 8.89 seconds

(secretsika@kali)-[~/elastic-agent-8.15.1-linux-x86_64]
$ sudo nmap -sS localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-15 15:22 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000020s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
6789/tcp  open  ibm-db2-admin

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds

(secretsika@kali)-[~/elastic-agent-8.15.1-linux-x86_64]
$
```

Querying for Security Events in the Elastic SIEM

With the data successfully forwarded from the Kali VM to the SIEM, I can now start querying and analyzing the logs. To do this, I accessed my Elastic Deployment and clicked on the menu icon with the three horizontal lines at the top-left. From there, I selected the “Logs” tab under “Observability” to view and analyze the logs generated from the Kali VM. This allows me to monitor and investigate the security events collected by the SIEM.

In the search bar, I entered a query to filter the logs. For instance, to find all logs related to Nmap scans, I used the query: event.action: "nmap_scan" or process.args: "sudo". This query helps narrow down the logs to those specifically associated with Nmap scans or commands run with sudo privileges.

process.args:nmap

Customize

Highlights

Sep 15, 2024	event.dataset	Message
15:22:42.911	endpoint.events.process	Endpoint process event
15:22:42.912	endpoint.events.process	Endpoint process event
15:22:42.923	endpoint.events.process	Endpoint process event
15:22:42.961	endpoint.events.process	Endpoint process event
15:22:42.961	endpoint.events.process	Endpoint process event
15:22:42.976	endpoint.events.process	Endpoint process event
15:22:42.977	endpoint.events.process	Endpoint process event
15:22:42.987	endpoint.events.process	Endpoint process event
15:22:42.987	endpoint.events.process	Endpoint process event
15:22:43.000	endpoint.events.process	Endpoint process event
15:22:43.001	endpoint.events.process	Endpoint process event
15:22:43.001	endpoint.events.process	Endpoint process event
15:22:43.081	endpoint.events.process	Endpoint process event
15:22:43.082	endpoint.events.process	Endpoint process event
15:22:43.083	endpoint.events.process	Endpoint process event
15:22:43.115	endpoint.events.process	Endpoint process event
15:22:43.117	endpoint.events.process	Endpoint process event

The screenshot below shows the details of one of the logs I selected. It also shows the exact command that was entered and its respective arguments.

Details for log entry Dmok95EBfxKYC4sFSMDk


Investigate

From index .ds-logs-endpoint.events.process-default-2024.09.15-000001






host.os.version	Unknown version
message	Endpoint process event
process.Ext.ancestry	MWNjZjFiZTEtMTNhMS00ZGFILWEwMWMtNjE4ZWlzZmM0OGEzLTE4NzI1MS0xNzI2NDI2MDE1, MWNjZjFiZTEtMTNhMS00ZGFILWEwMWMtNjE4ZWlzZmM0OGEzLTE4NzI1MS0xNzI2NDI2MDE0, MWNjZjFiZTEtMTNhMS00ZGFILWEwMWMtNjE4ZWlzZmM0OGEzLTEtMTcyNjQwMzlyNg==
process.args	sudo, nmap, -sS, localhost
process.args_count	4
process.command_line	sudo nmap -sS localhost
process.command_line.caseless	sudo nmap -ss localhost
process.command_line.text	sudo nmap -sS localhost


Creating a Dashboard to Visualize the Events


To analyze the logs and identify patterns or anomalies, I created a dashboard in the SIEM app. For instance, I set up a simple dashboard to display a count of security events over time. This visualization helps in monitoring trends and detecting any unusual activity in the data.

Edit visualization  [Edit in Lens](#)

Visualization configuration



Area 

metrics-* 

Horizontal axisOptional

@timestamp


Vertical axis

Count of records

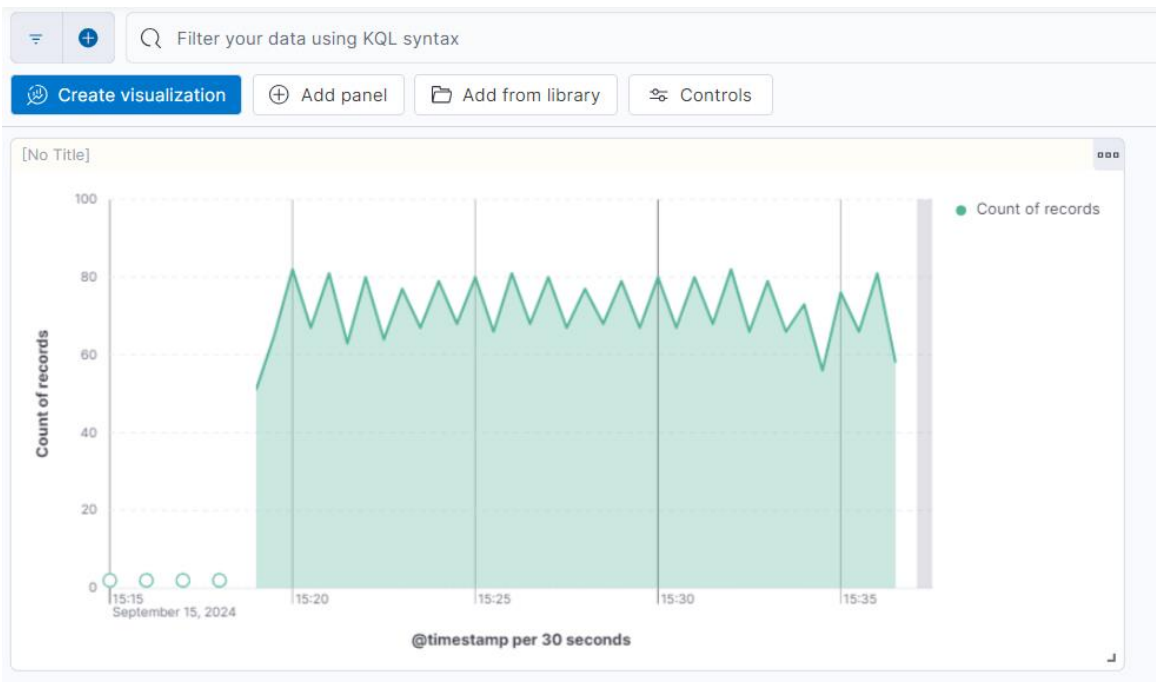
+ Add or drag-and-drop a field

BreakdownOptional

+ Add or drag-and-drop a field

 **Add layer**

As you can see, in the "Metrics" section of the visualization editor, I selected "Count" as the vertical field type and "Timestamp" for the horizontal field. This configuration allows me to display the count of security events over time, providing a clear view of event activity and trends within the SIEM dashboard.



In a SIEM, alerts are crucial for detecting and responding to security incidents in a timely manner. To create an alert in the Elastic SIEM instance to monitor for Nmap scans, I began by clicking on the menu icon at the top-left and selecting “Alerts” under the “Security” section. Next, I accessed “Manage rules” in the top right corner and clicked on the “Create new rule” button.

1 Define rule

Rule type

Custom query

Use KQL or Lucene to detect issues across indices.

Selected

Machine Learning

Select ML job to detect anomalous activity.

Select

Threshold

Aggregate query results to detect when number of matches exceeds threshold.

Select

Event Correlation

Use Event Query Language (EQL) to match events, generate sequences, and stack data

Select

Indicator Match

Use indicators from intelligence sources to detect matching events and alerts.

Select

New Terms

Find documents with values appearing for the first time.

Select

I then specified the conditions using a custom query designed to identify Nmap scan events, such as one that matches events with the action “nmap_scan,” and created/enabled the rule.

Index Patterns

Data View

Index patterns

apm-*transaction* X

auditbeat-* X

endgame-* X

filebeat-* X

logs-* X

packetbeat-* X

traces-apm* X

winlogbeat-* X

-*elastic-cloud-logs-* X

Enter the pattern of Elasticsearch indices where you would like this rule to run. By default, these will include index patterns defined in Security Solution advanced settings.

Custom query

Import query from saved timeline

+

Q event.action: "nmap_scan"

X

This configuration ensures that the SIEM will monitor for Nmap scan events and alert me when such activities are detected. I selected Email so that I can be notified when the rule is triggered.

Select a connector type

D3

D3 Security

Email

IBM

IBM Resilient

Index

Jira

Microsoft Teams

Opsgenie

P

PagerDuty

Server log

now

ServiceNow ITOM

now

ServiceNow ITSM

now

ServiceNow SecOps

Slack

Swimlane

Tines

Torq

Webhook

xMatters

nmap scan

Created by: 2384493002 on Sep 15, 2024 @ 15:55:01:192

Updated by: 2384493002 on Sep 15, 2024 @ 15:55:01:192

Last response:

Notify when alerts generated

Enable

Edit rule settings

Save

About

detects nmap scan

Severity

Low

Risk score

21

Max alerts per run

100

Definition

Index patterns

apm-*transaction* auditbeat-* endgame-*
filebeat-* logs-* packetbeat-* traces-apm*
winlogbeat-* -*elastic-cloud-logs-*

Custom query

event.action: "nmap_scan"

Rule type

Query

Timeline template

None

I used nmap on Kali Linux to test if the rule that I previously enabled works:

```
secretsika@kali: ~/elastic-agent-8.15.1-linux-x86_64
File Actions Edit View Help

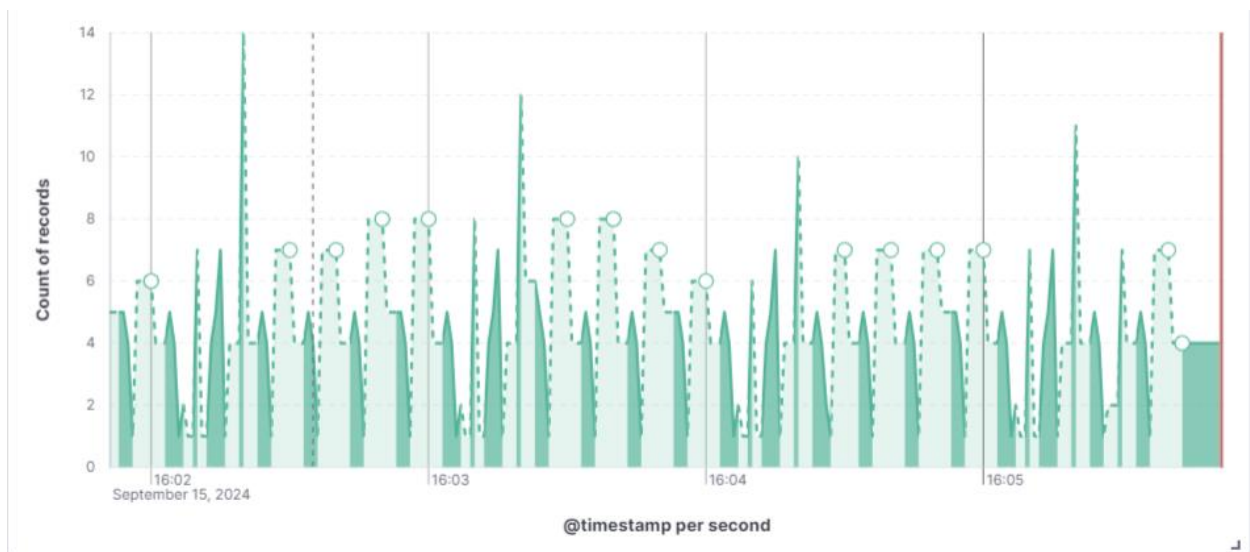
(secretsika@kali)-[~/elastic-agent-8.15.1-linux-x86_64]
$ nmap google.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-15 16:03 EDT
Stats: 0:00:01 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 3.40% done; ETC: 16:04 (0:00:28 remaining)
Stats: 0:00:03 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 18.15% done; ETC: 16:03 (0:00:14 remaining)
Nmap scan report for google.com (172.217.1.14)
Host is up (0.035s latency).
Other addresses for google.com (not scanned): 2607:f8b0:400b:80f::200e
rDNS record for 172.217.1.14: iad23s25-in-f14.1e100.net
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 6.81 seconds

(secretsika@kali)-[~/elastic-agent-8.15.1-linux-x86_64]
$ nmap -sV localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-15 16:03 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000059s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE          VERSION
6789/tcp  open  ssl/ibm-db2-admin?

Service detection performed. Please report any incorrect results at https://nmap.org/su
.
Nmap done: 1 IP address (1 host up) scanned in 12.34 seconds
```

Here is the Dashboard visualization:



What I learned from this project:

Through setting up this home lab with Elastic SIEM and a Kali VM, I gained valuable insights into several key areas:

1. **Elastic SIEM Configuration:** I learned how to configure Elastic SIEM to receive and analyze data from various sources, including setting up a deployment and managing integrations.
2. **Data Forwarding:** I understood how to use the Elastic Beats agent to forward logs from the Kali VM to the SIEM, ensuring seamless data collection.
3. **Log Analysis:** I acquired skills in querying and analyzing logs using the Elastic web interface, which helped in identifying and understanding security events.
4. **Dashboard Creation:** I practiced creating visualizations and dashboards to monitor security events effectively, allowing for better insight into trends and patterns.
5. **Alert Configuration:** I learned how to set up alerts based on custom queries to detect specific security incidents, such as Nmap scans, enabling proactive security monitoring.

Overall, this experience has enhanced my practical skills in using SIEM tools, improving my ability to monitor, analyze, and respond to security events.

Yushika Jhundoo

Home SIEM Lab Project