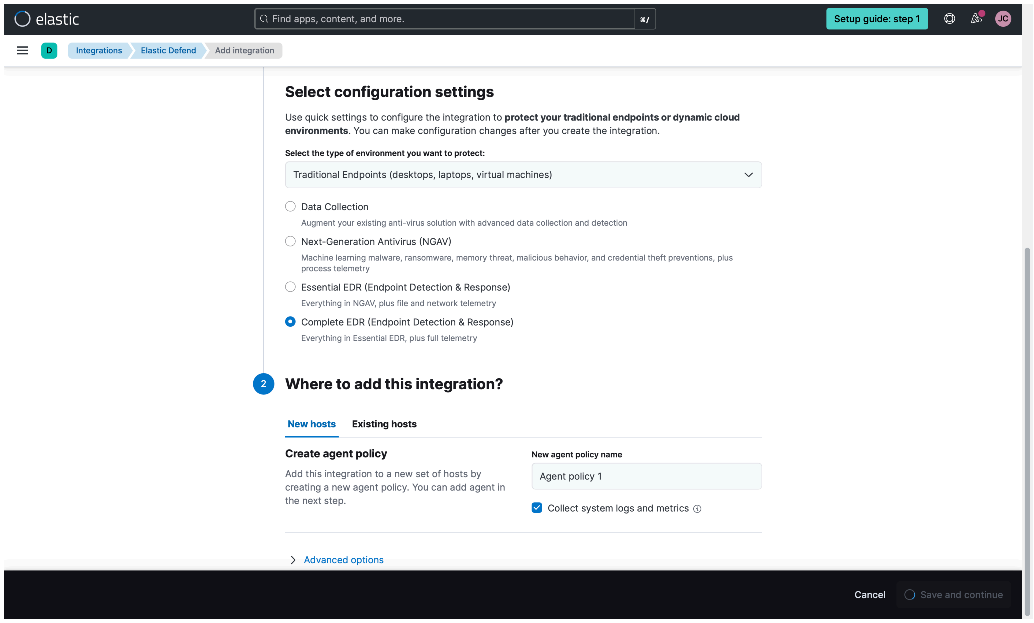
Elastic Stack Configuration and Monitoring

Introduction

In this project, I havr implemented and configured the Elastic Stack (Elasticsearch, Kibana, and Elastic Agent) to establish a centralized logging and security monitoring system. The primary objectives were to ensure seamless log collection, enhance log analysis capabilities, and enable proactive security incident response.

Steps Performed

1. Configuration and Installation



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)

○ Data Collection
Augment your existing anti-virus solution with advanced data collection and detection

○ Next-Generation Antivirus (NGAV)
Machine learning malware, ransomware, memory threat, malicious behavior, and credential theft preventions, plus process telemetry

○ Essential EDR (Endpoint Detection & Response)
Everything in NGAV, plus file and network telemetry

● Complete EDR (Endpoint Detection & Response)
Everything in Essential EDR, plus full telemetry

2 Where to add this integration?

New hosts | Existing hosts

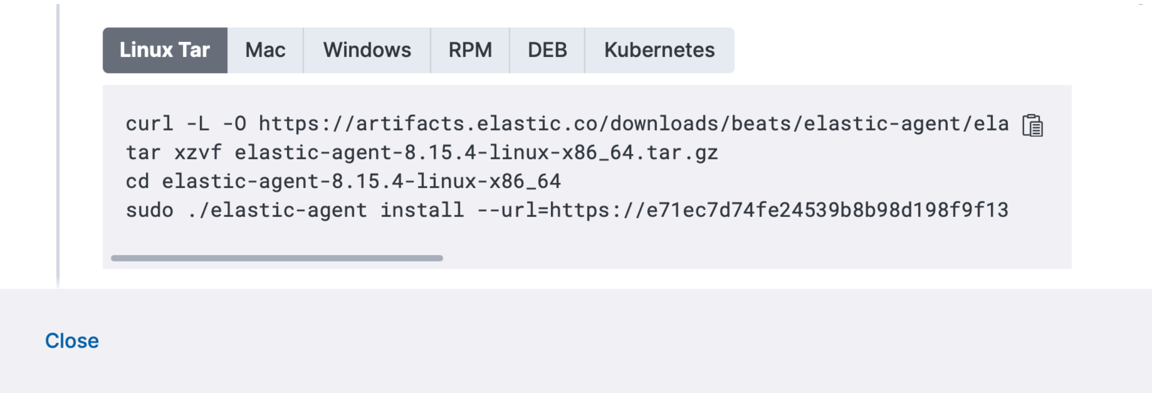
Create agent policy

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

We began by installing the Elastic Stack components on a dedicated Kali Linux (VM) system using the commands in the screenshot below. Elasticsearch was configured as the centralized log storage and indexing engine, while Kibana provided the interface for log visualization and analysis.



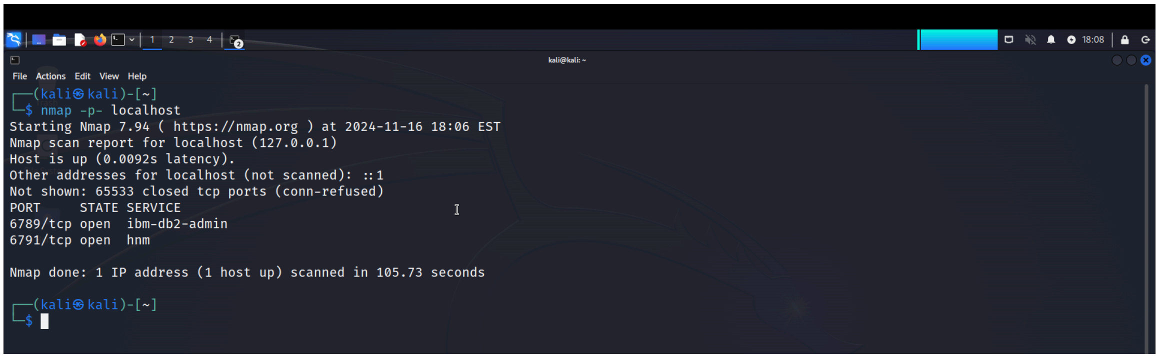
```
curl -L -O https://artifacts.elastic.co/downloads/beats/elastic-agent/ela
tar xzvf elastic-agent-8.15.4-linux-x86_64.tar.gz
cd elastic-agent-8.15.4-linux-x86_64
sudo ./elastic-agent install --url=https://e71ec7d74fe24539b8b98d198f9f13
```

Next, we configured the Elastic Agent on various endpoints to forward logs to Elasticsearch. The agent was set up to capture system logs, network activity, and security events.

Challenges: During the setup, connectivity issues between the Elastic Agent and Elasticsearch were resolved by modifying firewall rules and ensuring proper authentication settings.

2. Simulating Activity to Generate Logs

To validate the Elastic Stack's configuration, I used the ***nmap -p- localhost*** command to simulate network activity. This full port scan of the localhost generated significant traffic and security events, mimicking real-world scenarios of system probing and reconnaissance.

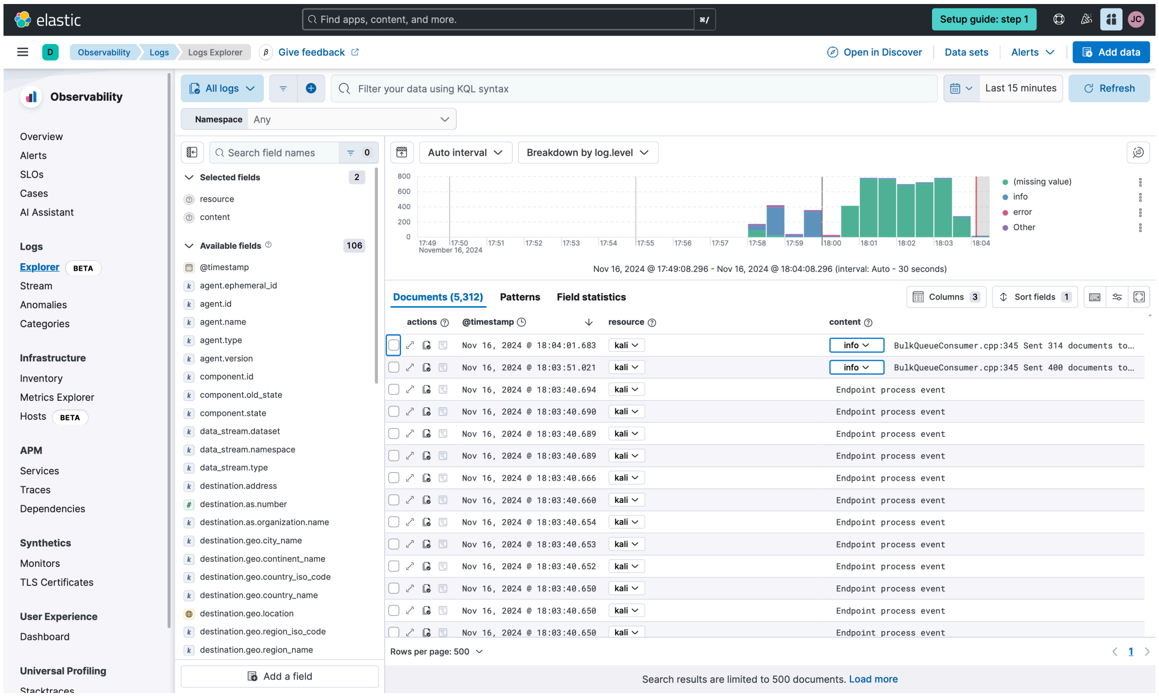


```
┌──(kali㉿kali)-[~]
└─$ nmap -p- localhost
Starting Nmap 7.94 ( https://nmap.org ) at 2024-11-16 18:06 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0092s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 65533 closed tcp ports (conn-refused)
PORT     STATE SERVICE
6789/tcp open  ibm-db2-admin
6791/tcp open  hnm

Nmap done: 1 IP address (1 host up) scanned in 105.73 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

The activity was instrumental in verifying that the Elastic Agent successfully captured and forwarded log data to Elasticsearch. The logs were then visualized in Kibana, showcasing the seamless integration and functionality of the setup. By utilizing Kibana Query Language (KQL), I could efficiently filter logs to analyze specific events or activities. For instance, querying with ***process.args: "<process_name>"*** allowed me to pinpoint logs associated with a particular process, streamlining the investigation process and enhancing log analysis precision

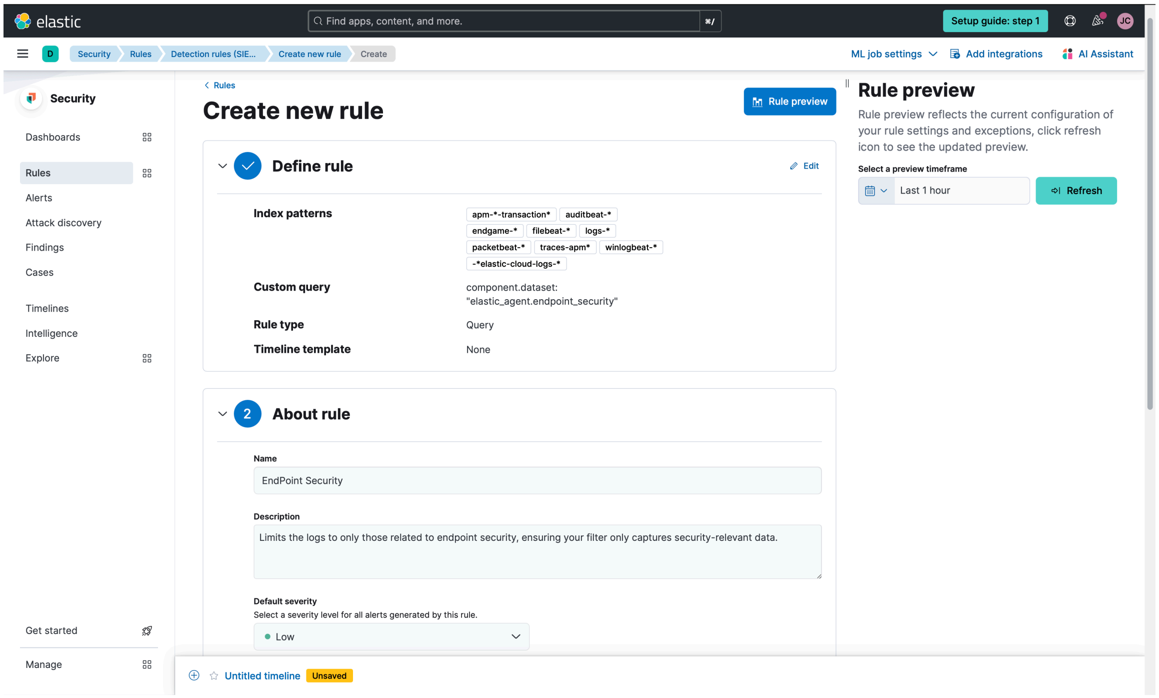


actions	@timestamp	resource	content
□	Nov 16, 2024 @ 18:04:01.683	kali	info BulkQueueConsumer.cpp:345 Sent 314 documents to...
□	Nov 16, 2024 @ 18:03:51.021	kali	info BulkQueueConsumer.cpp:345 Sent 400 documents to...
□	Nov 16, 2024 @ 18:03:40.694	kali	Endpoint process event
□	Nov 16, 2024 @ 18:03:40.690	kali	Endpoint process event
□	Nov 16, 2024 @ 18:03:40.689	kali	Endpoint process event
□	Nov 16, 2024 @ 18:03:40.689	kali	Endpoint process event
□	Nov 16, 2024 @ 18:03:40.666	kali	Endpoint process event
□	Nov 16, 2024 @ 18:03:40.660	kali	Endpoint process event
□	Nov 16, 2024 @ 18:03:40.654	kali	Endpoint process event
□	Nov 16, 2024 @ 18:03:40.653	kali	Endpoint process event
□	Nov 16, 2024 @ 18:03:40.652	kali	Endpoint process event
□	Nov 16, 2024 @ 18:03:40.650	kali	Endpoint process event
□	Nov 16, 2024 @ 18:03:40.650	kali	Endpoint process event
□	Nov 16, 2024 @ 18:03:40.650	kali	Endpoint process event

This ability to filter and search logs using relevant KQL syntax ensures the system's utility in monitoring and responding to targeted activities, significantly improving the effectiveness of threat detection and response workflows.

2. Log Indexing and Custom Filters

To improve log analysis, we developed and deployed custom log filters tailored to specific use cases. The filters enable precise categorization and tagging of logs, facilitating easier search and analysis. This page can be found by navigating the menu under Security **>** Rules.



< Rules

Create new rule

Define rule

Index patterns	apm-*-transaction* auditbeat-* endgame-* filebeat-* logs-* packetbeat-* traces-apm* winlogbeat-* -*elastic-cloud-logs-*
Custom query	component.dataset: "elastic_agent.endpoint_security"
Rule type	Query
Timeline template	None

2 About rule

Name: EndPoint Security

Description: Limits the logs to only those related to endpoint security, ensuring your filter only captures security-relevant data.

Default severity: Select a severity level for all alerts generated by this rule. ● Low

Rule preview

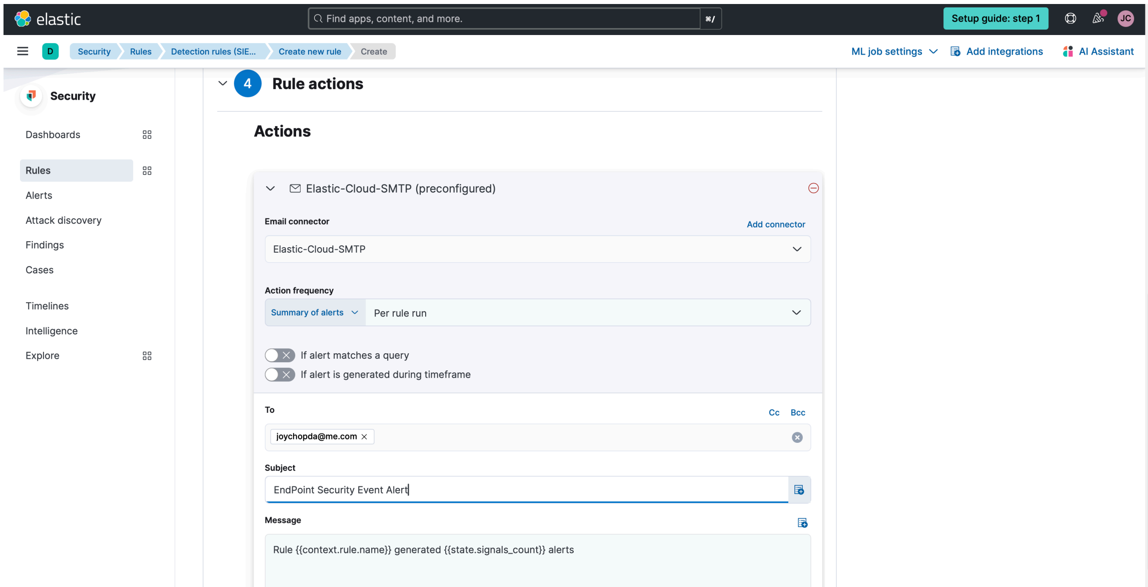
Rule preview reflects the current configuration of your rule settings and exceptions, click refresh icon to see the updated preview.

Select a preview timeframe: Last 1 hour

Additionally, indices were optimized to handle high volumes of data efficiently. This included setting up log rotation to prevent storage overload.

3. Alerting Mechanism

Automated email alerts were configured for critical security events. For instance, alerts were triggered for unauthorized access attempts, unusual network traffic patterns, or high CPU usage. This ensured timely detection and response to potential incidents.



4 Rule actions

Actions

Elastic-Cloud-SMTP (preconfigured)

Email connector: Elastic-Cloud-SMTP

Action frequency: Summary of alerts | Per rule run

○ If alert matches a query
○ If alert is generated during timeframe

To: joychopda@me.com

Subject: EndPoint Security Event Alert

Message: Rule {{context.rule.name}} generated {{state.signals_count}} alerts

Conclusion

The Elastic Stack project successfully demonstrated our ability to set up a robust log collection and monitoring system. Through this experience, we gained practical knowledge of Elastic Stack configuration, troubleshooting, and advanced log analysis.