EncryptEdge Labs

Cybersecurity Analyst Internship

Task Report

atalmamun@gmail.com

Task No: 15

Credit: Offensive Security

EncryptEdge Labs

Table of Contents

1.0 EncryptEdge Labs Internship Task Report

1.1 Introduction

In the ever-evolving field of cybersecurity, log analysis plays a critical role in monitoring system activities, detecting unauthorized access, and identifying potential security incidents. Every action taken within an operating system, application, or network component leaves a trace in the form of logs. By analyzing these logs, cybersecurity professionals can uncover suspicious patterns, respond to threats, and ensure overall system integrity. This task focuses on equipping interns with hands-on experience in collecting, analyzing, and interpreting log data using tools such as the ELK Stack (Elasticsearch, Logstash, Kibana) or Splunk. Through this process, interns develop essential skills that form the foundation of effective security monitoring and incident response.

1.2 Objective

The primary objectives of this task are:

- To understand the importance of log analysis in cybersecurity operations.
- To learn how to collect and centralize logs from various sources such as operating systems, applications, and network devices.
- To gain hands-on experience using log analysis tools (e.g., ELK Stack or Splunk).
- To analyze log data in order to identify anomalies and potential security incidents.
- To correlate logs from different sources to detect and interpret security events.
- To complete practical labs that reinforce theoretical concepts and simulate real-world scenarios.

1.3 Requirements

To complete this task, the following tools and resources are required:

- **Virtual Environment**: VirtualBox or VMware for setting up the log analysis environment.
- **Log Analysis Tools**: ELK Stack (Elasticsearch, Logstash, Kibana) or Splunk.
- **Log Forwarders**: Filebeat (for Linux logs), Winlogbeat (for Windows logs).
- **Sample Log Files**: Provided log files or publicly available repositories.
- **TryHackMe Labs**:
    - *Intro to Logs*
    - *Intro to Log Analysis*
    - *Servidae: Log Analysis in ELK*

2.0 Importance of Log Analysis

Log analysis is a foundational aspect of cybersecurity operations. It enables security teams to detect suspicious behavior, trace incidents, and maintain visibility into system and network activities. Without proper log analysis, organizations may miss early warning signs of breaches or system failures, making them vulnerable to security threats.

2.1 Role of Log Analysis in Cybersecurity

Log analysis is essential for detecting, investigating, and responding to security incidents. Logs provide detailed records of events occurring across different systems, including user activity, system processes, application behavior, and network traffic. By continuously monitoring and analyzing logs, cybersecurity professionals can:

- **Detect unauthorized access attempts** such as brute-force login attacks.
- **Monitor system integrity** by identifying unauthorized changes to files or configurations.
- **Identify malware infections** through abnormal behavior like unexpected process executions or outbound connections.
- **Understand the scope of an incident**, including the timeline and affected systems.
- **Troubleshoot performance issues** by analyzing patterns in error logs or system resource utilization.

In short, log analysis provides critical visibility that enables proactive threat detection and swift incident response.

2.2 Key Information Found in Logs

Different types of logs contain specific data points that can be used to detect anomalies and security incidents. Some common and critical pieces of information found in logs include:

- **Timestamps**: Indicate when an event occurred, helping to build a timeline of incidents.
- **IP Addresses**: Identify the source and destination of network traffic. Unusual or foreign IP addresses can indicate external threats.
- **Usernames and Login Attempts**: Help detect brute-force attacks or account misuse.
- **Event Types and Status Codes**: Provide details on actions such as login success/failure, file access, or application crashes.
- **System and Application Messages**: Give insight into configuration issues, errors, or malicious behavior.

- **Process Activity**: Can reveal suspicious processes or unauthorized software running on a system.

Each of these log elements plays a role in identifying patterns or anomalies that signal security threats. For example, repeated failed login attempts from a single IP within a short time frame could indicate a brute-force attack, while a sudden spike in outbound traffic might point to data exfiltration.

3.0 Log Collection Setup

3.1 Installation of Log Analysis Tool

For this task, I have chosen to use the **ELK Stack** (Elasticsearch, Logstash, Kibana) as my log analysis tool. The **ELK Stack** provides a comprehensive and scalable solution for collecting, processing, storing, and analyzing log data. Additionally, I have configured **Filebeat** to collect and forward logs to Logstash and Elasticsearch.

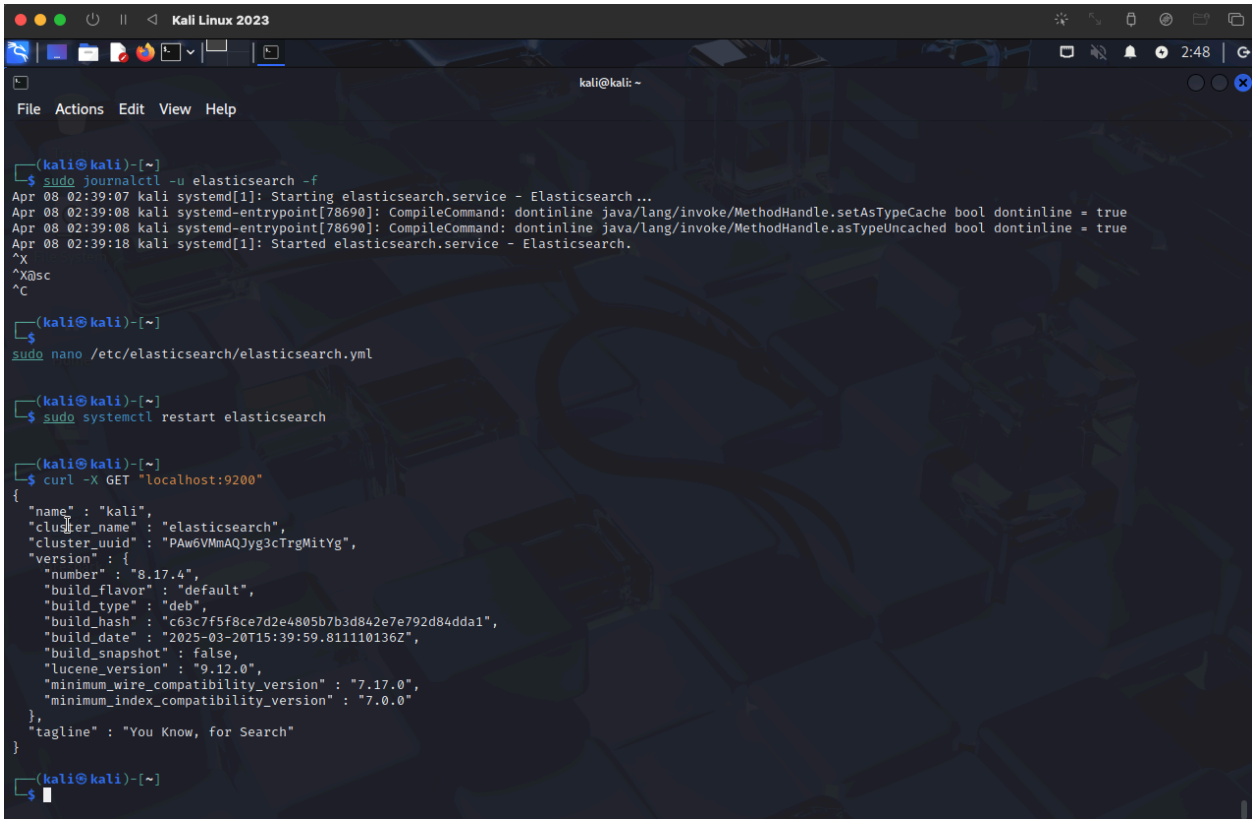
3.2 Setting Up Elasticsearch

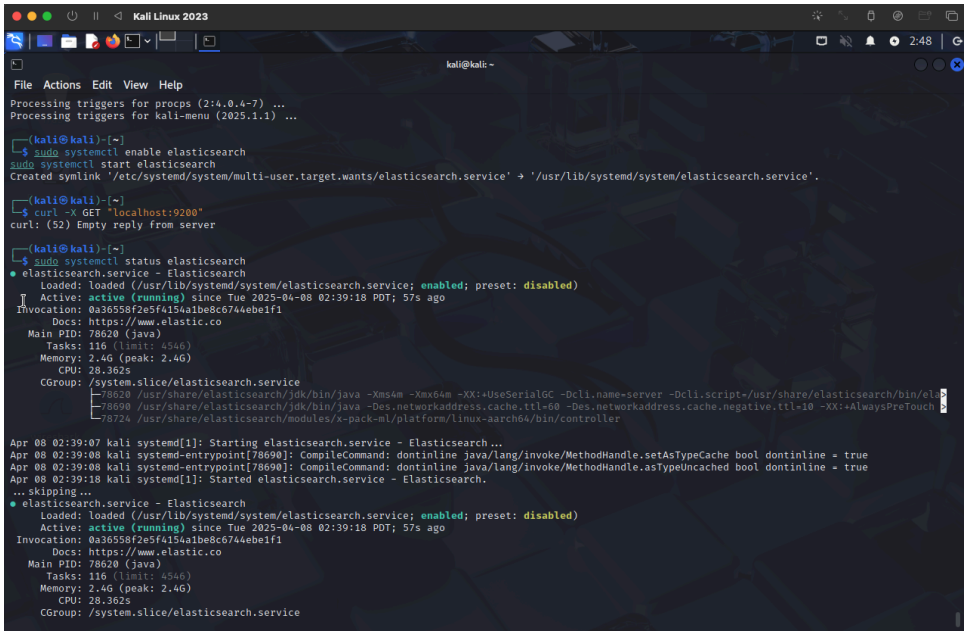
To begin the setup, I installed **Elasticsearch** on my Kali Linux instance. The installation was performed using the `apt` package manager and involved configuring the necessary repositories and security settings. Once installed, I configured the system to allow access on port **9200** for HTTP communication.

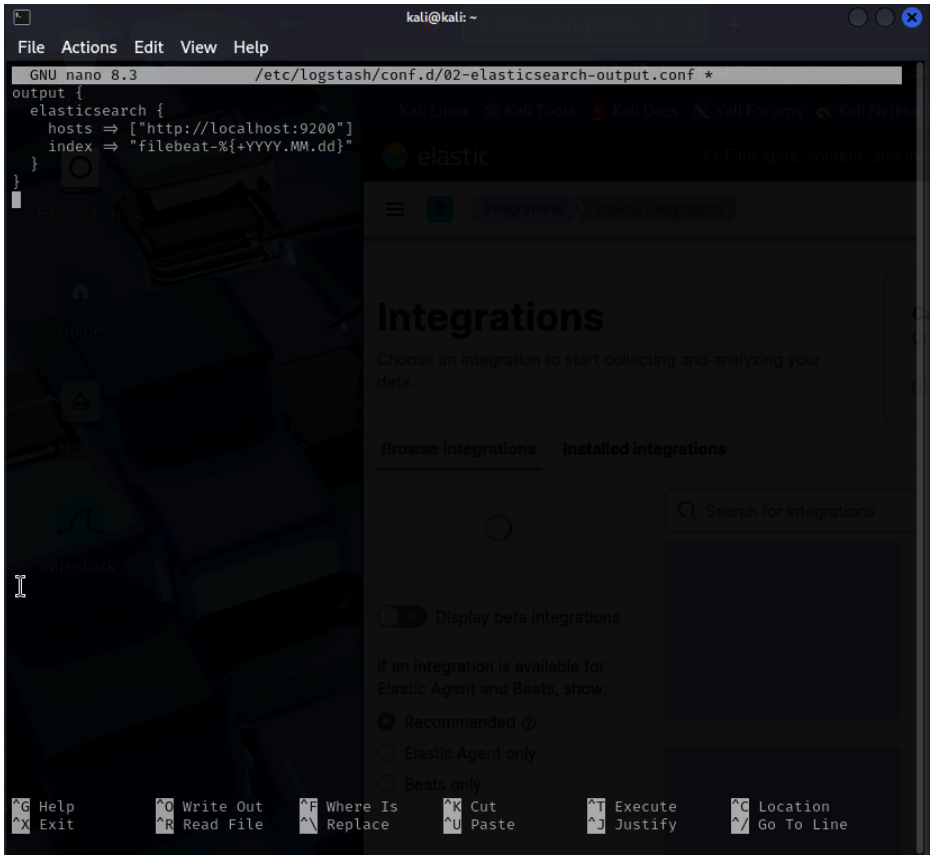
Steps Taken:

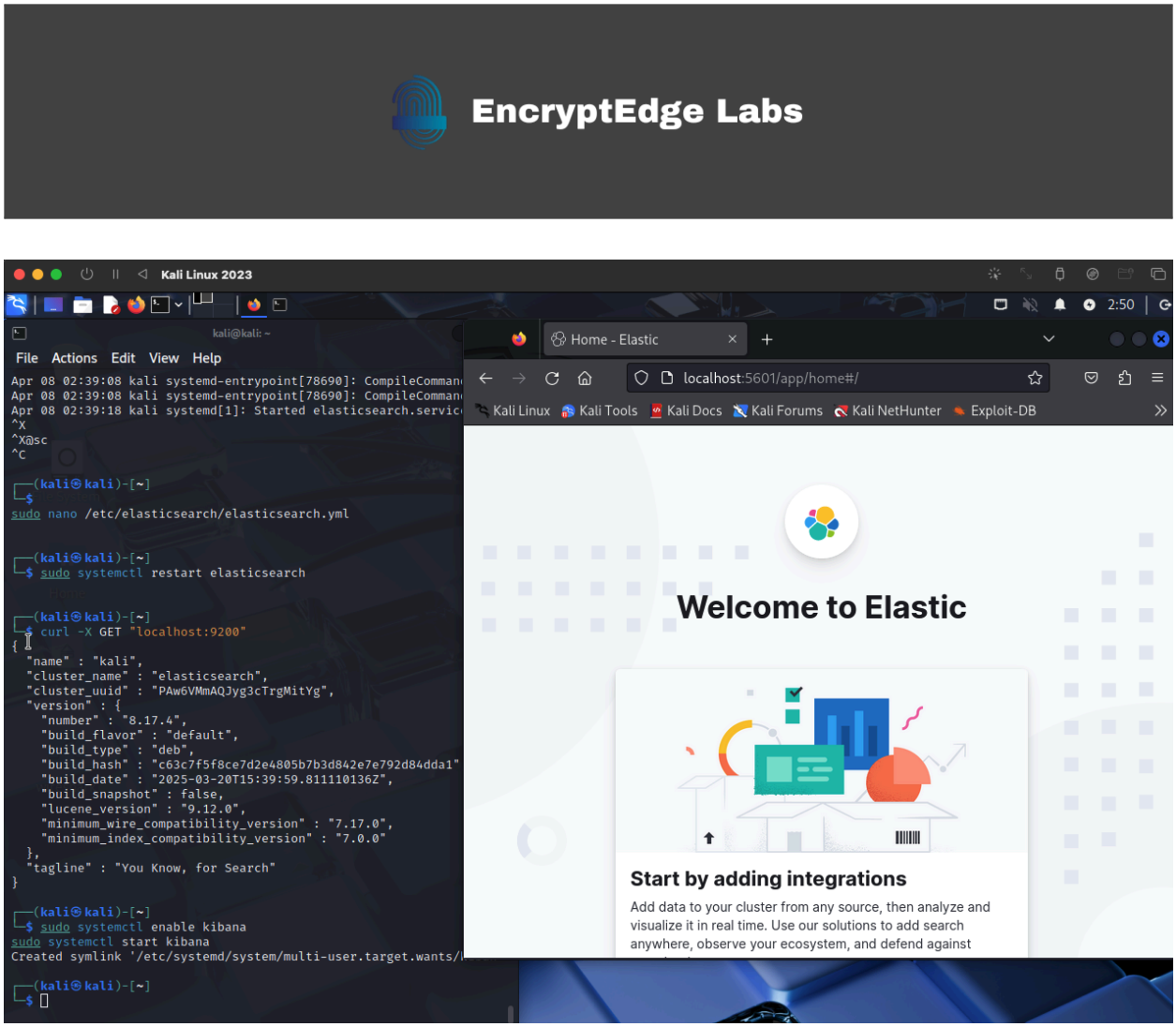
1. Elasticsearch was installed from the Elastic repository.

2. I modified the configuration file (`elasticsearch.yml`) to enable security features and set up the cluster.
3. Elasticsearch was started and verified to be running by checking the status using the `systemctl` command.
4. I tested the connection by running a `curl` command to ensure the service was responsive.




Kali Linux 2023
kali@kali: ~
File Actions Edit View Help
Processing triggers for procps (2:4.0.4-7) ...
Processing triggers for kali-menu (2025.1.1) ...
(kali㉿kali)-[~]
$ sudo systemctl enable elasticsearch
sudo systemctl start elasticsearch
Created symlink '/etc/systemd/system/multi-user.target.wants/elasticsearch.service' → '/usr/lib/systemd/system/elasticsearch.service'.
(kali㉿kali)-[~]
$ curl -X GET "localhost:9200"
curl: (52) Empty reply from server
(kali㉿kali)-[~]
$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; preset: disabled)
Active: active (running) since Tue 2025-04-08 02:39:18 PDT; 57s ago
Invocation: 0a36558f2e5f4154a1be8c6744ebe1f1
Docs: https://www.elastic.co
Main PID: 78620 (java)
Tasks: 116 (limit: 4546)
Memory: 2.4G (peak: 2.4G)
CPU: 28.362s
CGroup: /system.slice/elasticsearch.service
78620 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=server -Dcli.script=/usr/share/elasticsearch/bin/el
78690 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch
78724 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-aarch64/bin/controller
Apr 08 02:39:07 kali systemd[1]: Starting elasticsearch.service - Elasticsearch...
Apr 08 02:39:08 kali systemd-entrypoint[78690]: CompileCommand: dontinline java/lang/invoke/MethodHandle.setAsTypeCache bool dontinline = true
Apr 08 02:39:08 kali systemd-entrypoint[78690]: CompileCommand: dontinline java/lang/invoke/MethodHandle.asTypeUncached bool dontinline = true
Apr 08 02:39:18 kali systemd[1]: Started elasticsearch.service - Elasticsearch.
... skipping ...
● elasticsearch.service - Elasticsearch
Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; preset: disabled)
Active: active (running) since Tue 2025-04-08 02:39:18 PDT; 57s ago
Invocation: 0a36558f2e5f4154a1be8c6744ebe1f1
Docs: https://www.elastic.co
Main PID: 78620 (java)
Tasks: 116 (limit: 4546)
Memory: 2.4G (peak: 2.4G)
CPU: 28.362s
CGroup: /system.slice/elasticsearch.service


kali@kali: ~
File Actions Edit View Help
GNU nano 8.3 /etc/logstash/conf.d/02-elasticsearch-output.conf *
output {
elasticsearch {
hosts ⇒ ["http://localhost:9200"]
index ⇒ "filebeat-%{+YYYY.MM.dd}"
}
}
^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line



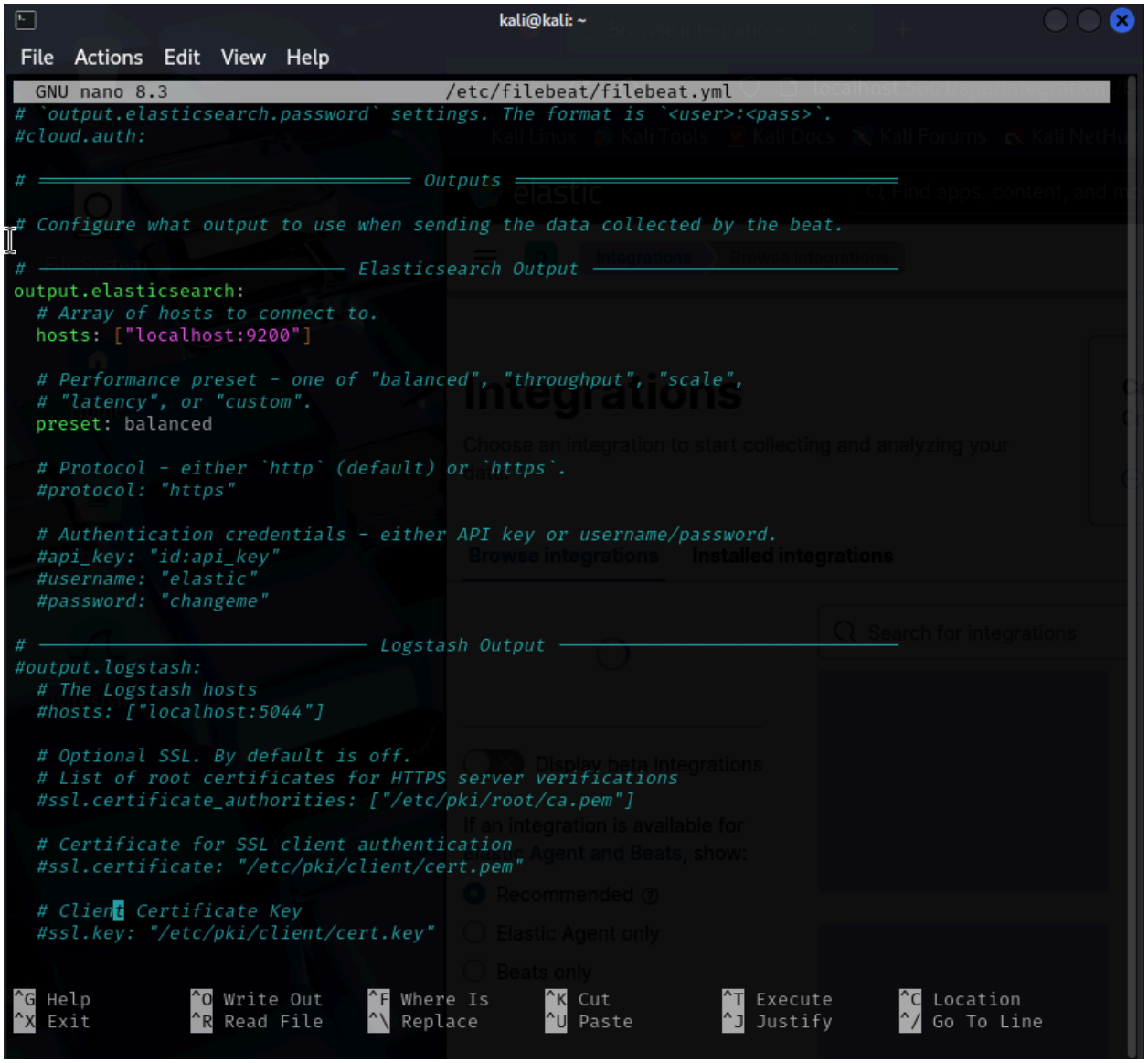
3.3 Setting Up Filebeat for Log Collection

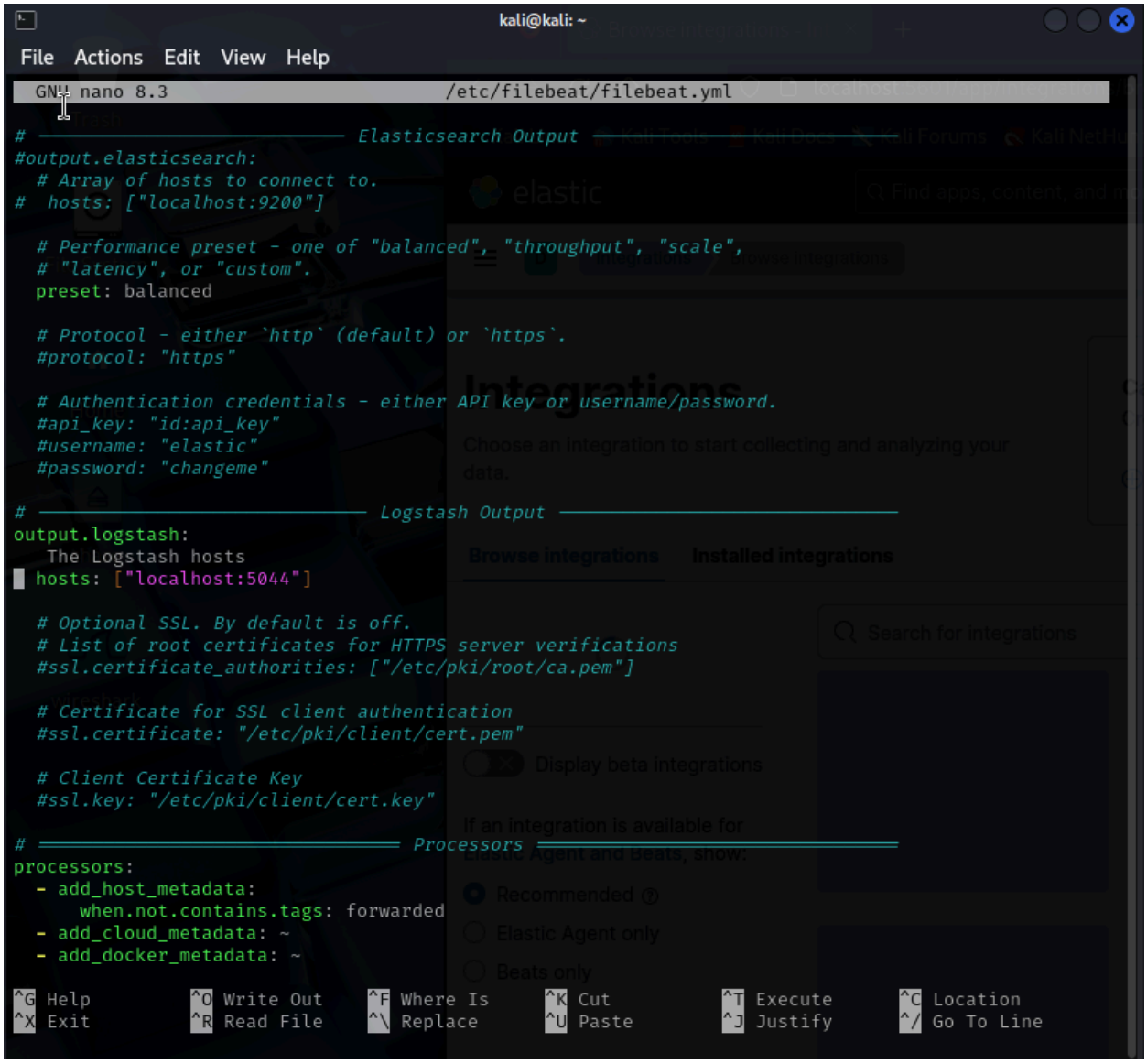
Filebeat was used to collect logs from my Kali Linux system and send them to Elasticsearch and Logstash. Filebeat is lightweight and ideal for forwarding log data to central systems for further analysis. I configured **Filebeat** to read logs from various sources on the system and send them to Elasticsearch for indexing and storage.

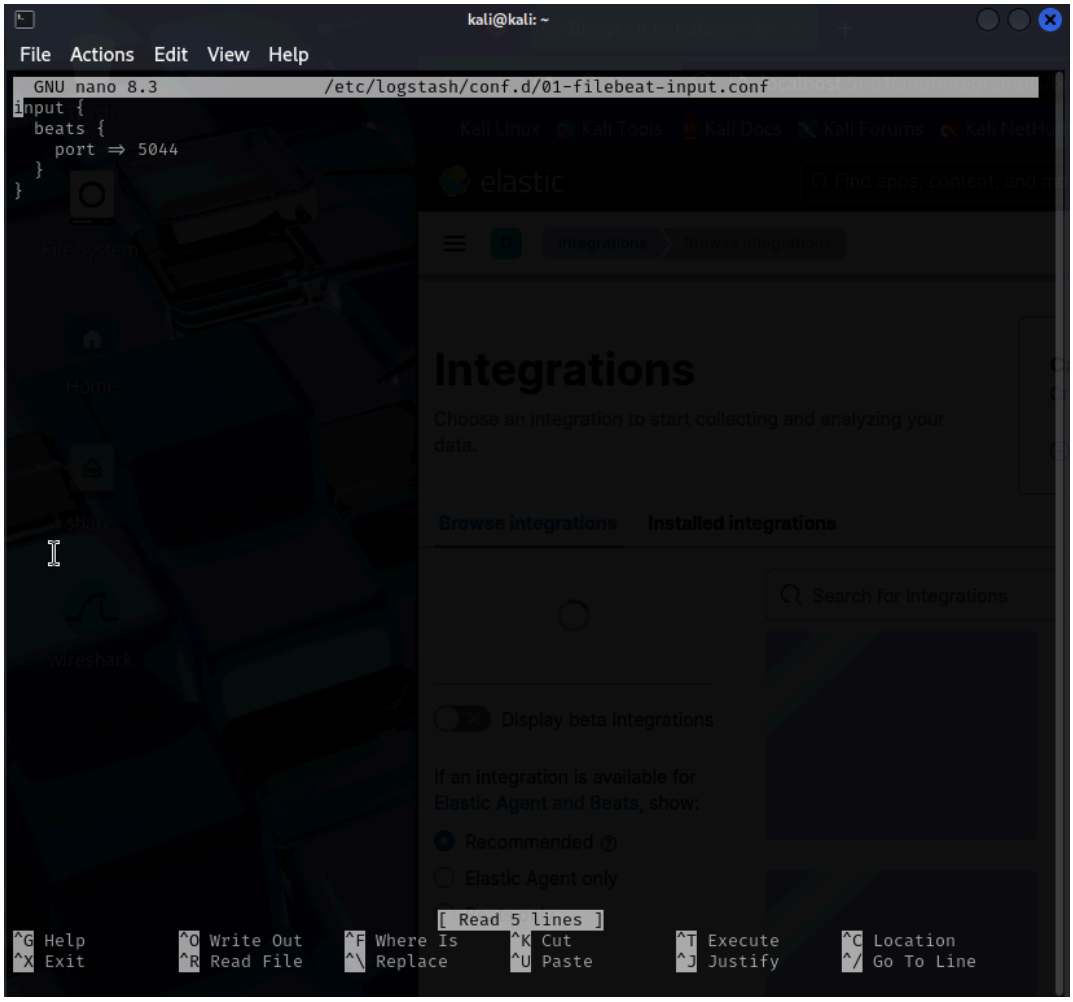
Steps Taken:

1. I installed **Filebeat** on the Kali Linux system using the `apt` package manager.
2. I edited the `filebeat.yml` configuration file to specify the output destination for the logs (either to **Elasticsearch** directly or via **Logstash**).

3. The configuration was set to forward logs to **localhost:9200** for Elasticsearch, and **localhost:5044** for Logstash.
4. I started **Filebeat** and verified that it was running successfully.




kali@kali: ~
File Actions Edit View Help
GNU nano 8.3 /etc/filebeat/filebeat.yml
---------------------------- Elasticsearch Output ----------------------------
#output.elasticsearch:
Array of hosts to connect to.
hosts: ["localhost:9200"]
Performance preset - one of "balanced", "throughput", "scale",
"latency", or "custom".
preset: balanced
Protocol - either `http` (default) or `https`.
#protocol: "https"
Authentication credentials - either API key or username/password.
#api_key: "id:api_key"
#username: "elastic"
#password: "changeme"
------------------------------ Logstash Output -------------------------------
output.logstash:
The Logstash hosts
hosts: ["localhost:5044"]
Optional SSL. By default is off.
List of root certificates for HTTPS server verifications
#ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]
Certificate for SSL client authentication
#ssl.certificate: "/etc/pki/client/cert.pem"
Client Certificate Key
#ssl.key: "/etc/pki/client/cert.key"
================================ Processors ==================================
processors:
- add_host_metadata:
when.not.contains.tags: forwarded
- add_cloud_metadata: ~
- add_docker_metadata: ~
^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

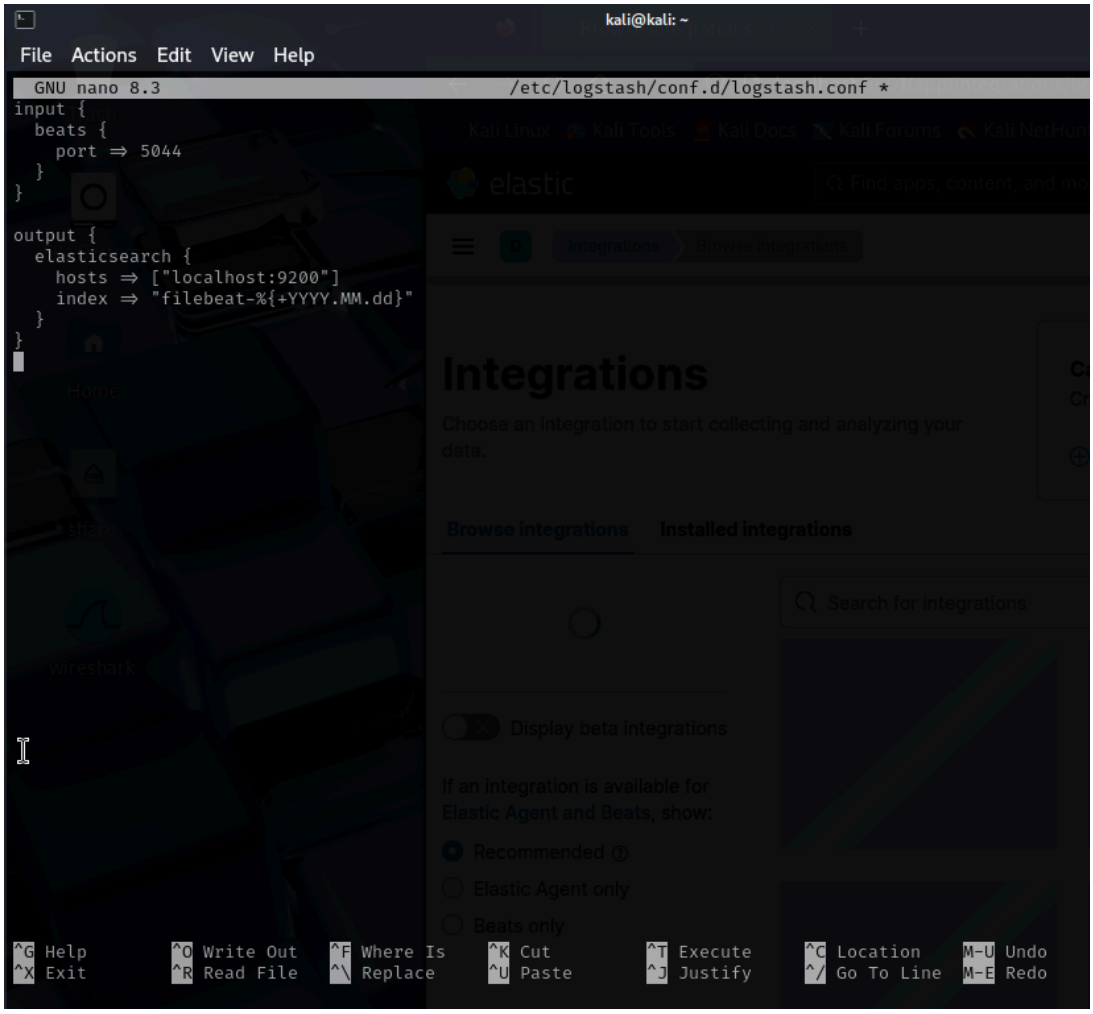


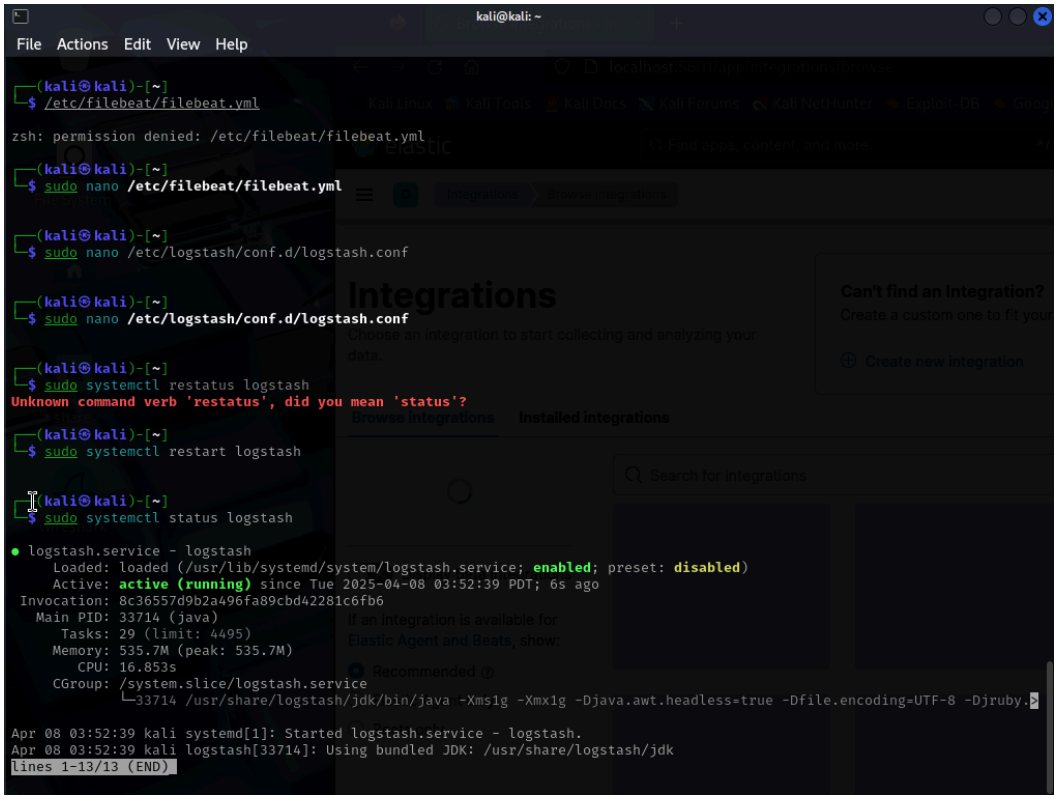
3.4 Setting Up Logstash

Logstash was configured as an intermediate step to process logs from Filebeat before forwarding them to Elasticsearch. I created a Logstash pipeline configuration (`logstash.conf`) to handle the logs from Filebeat. The pipeline is responsible for parsing and transforming logs as needed before sending them to Elasticsearch for storage and analysis.

Steps Taken:

1. Installed **Logstash** using the package manager.
2. Created the `logstash.conf` configuration file to specify input (from Filebeat), output (to Elasticsearch), and filters to process the logs.
3. I ensured that **Logstash** was configured to accept logs on port **5044** (default for Filebeat).
4. **Logstash** was started, and I verified that logs were being processed and forwarded to Elasticsearch correctly.

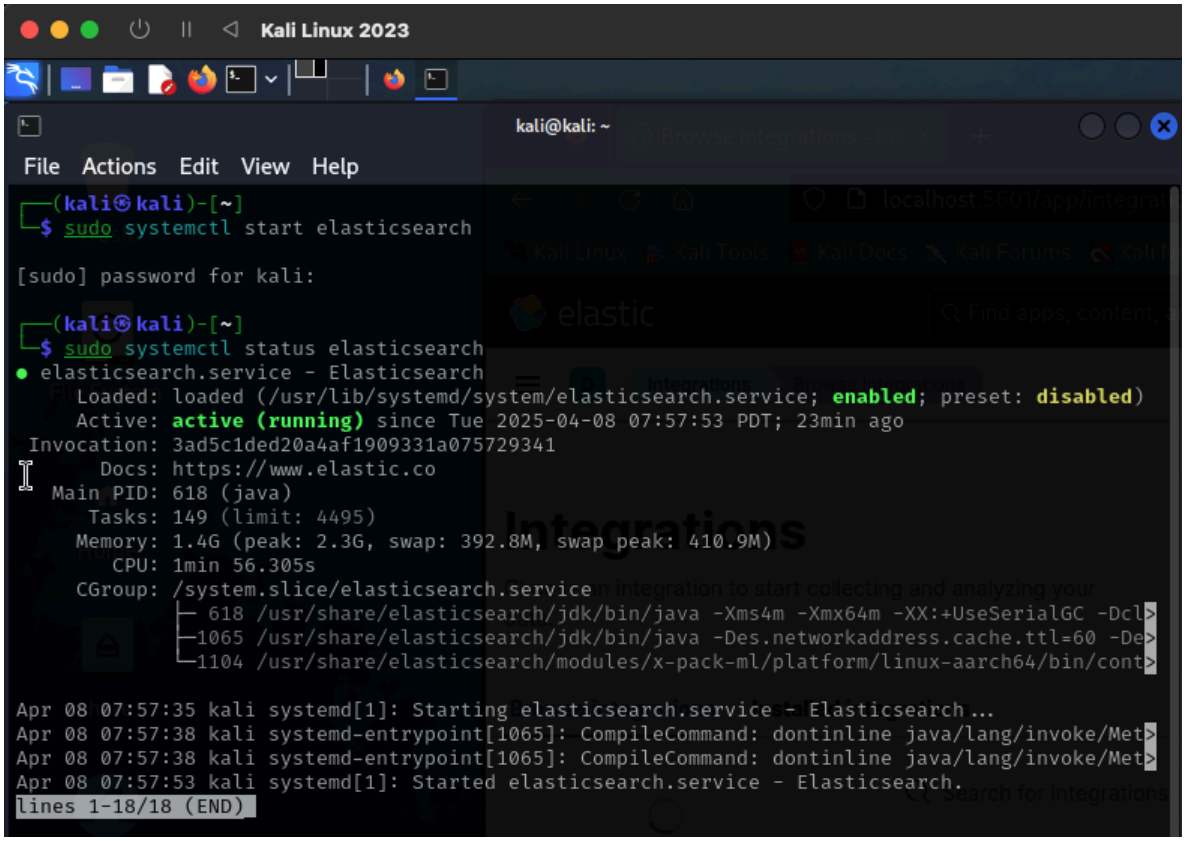


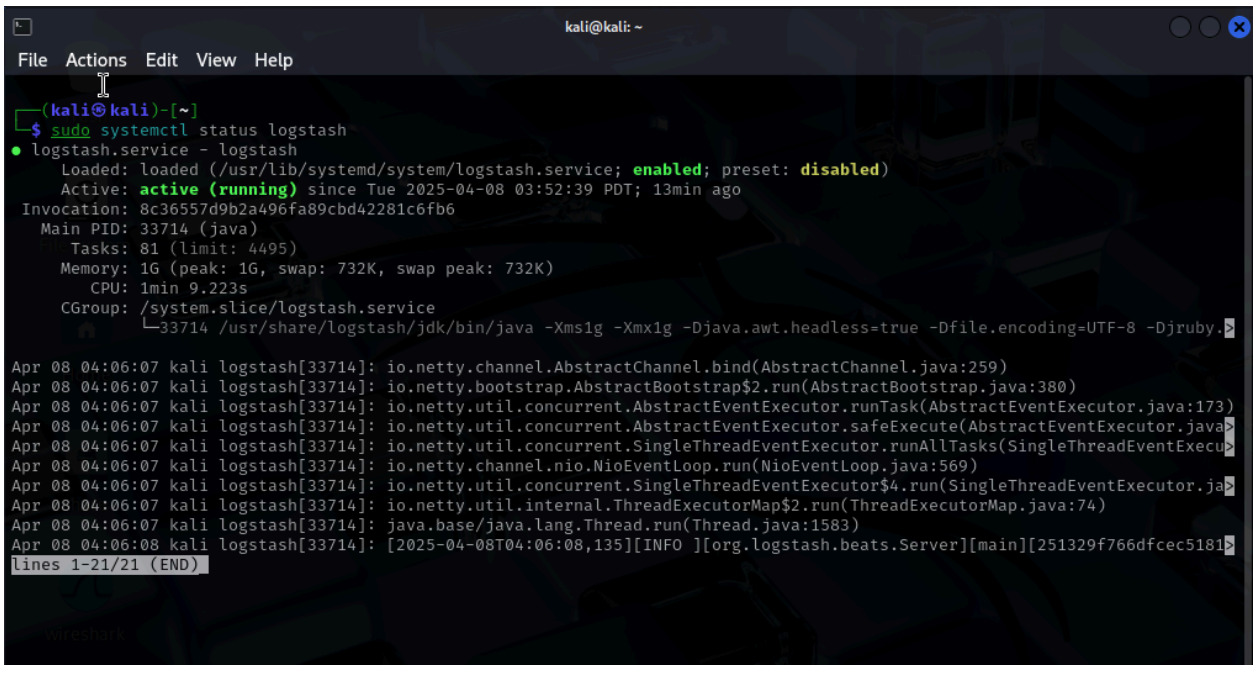


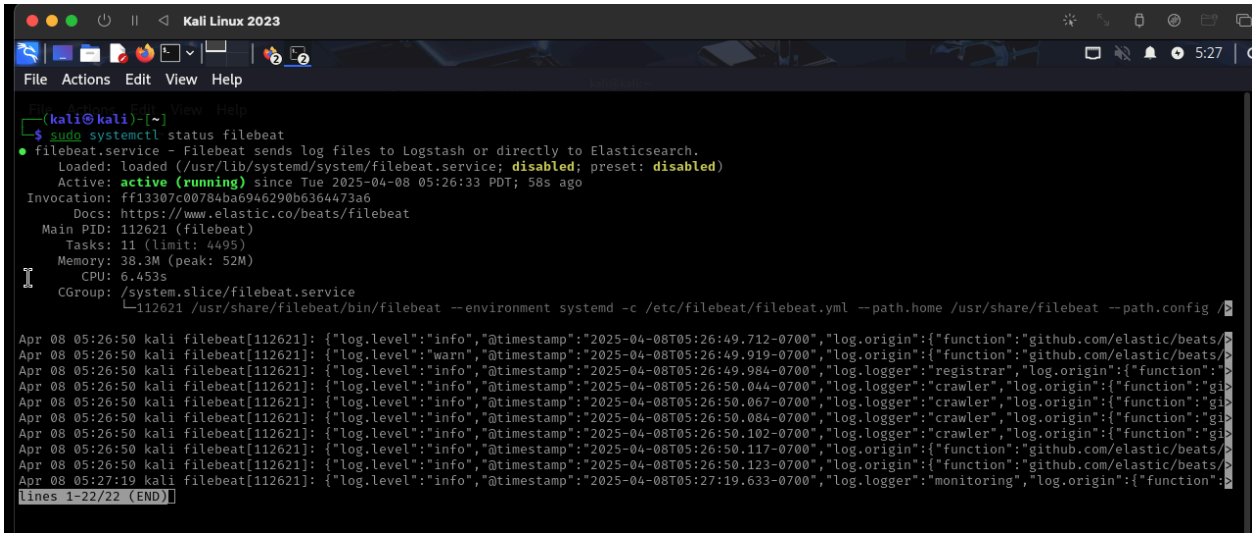
3.5 Verification of Log Collection

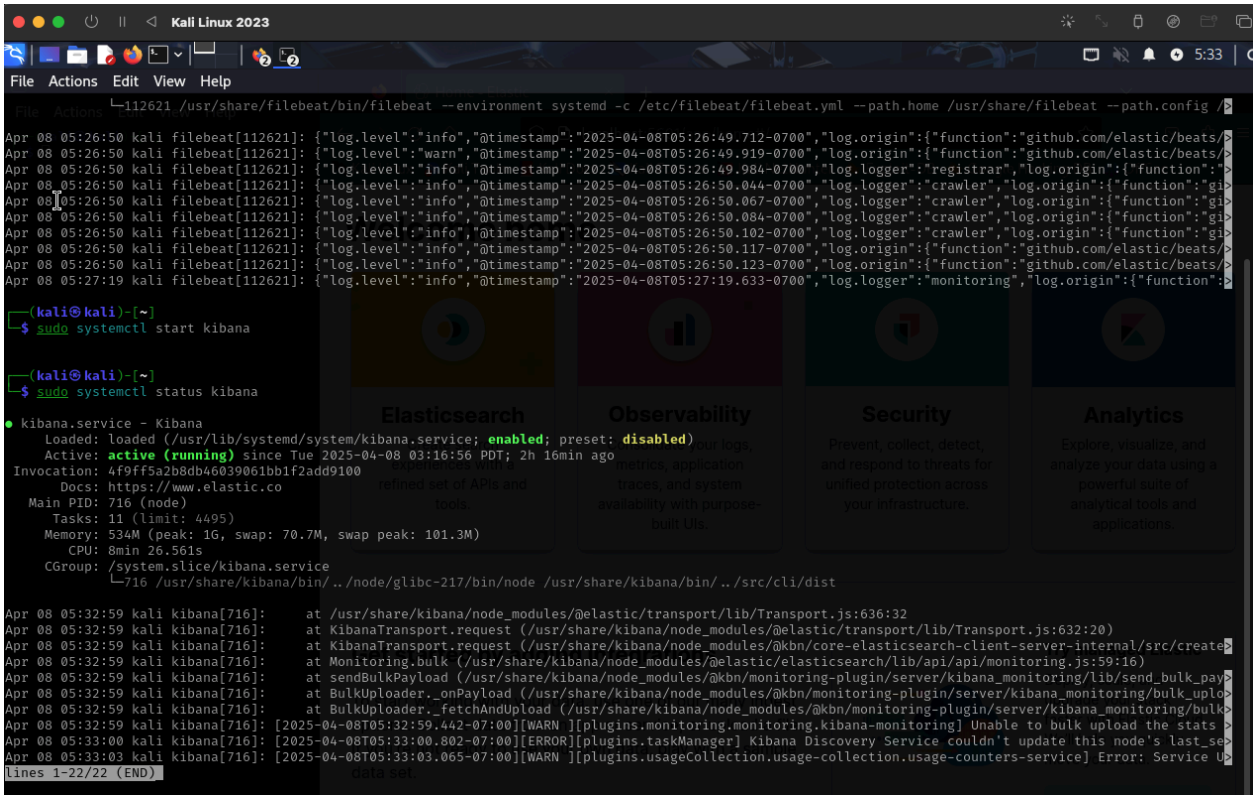
To ensure that logs were being successfully collected and processed, I performed the following checks:

1. Checked the **Filebeat** and **Logstash** logs for any errors or issues with the log forwarding process.
2. Verified that logs were being indexed in **Elasticsearch** by querying the index through the Kibana interface.
3. Checked the **Kibana Discover** page to view the incoming logs and confirm that they were properly parsed and stored.


Kali Linux 2023
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ sudo systemctl start elasticsearch
[sudo] password for kali:
(kali㉿kali)-[~]
$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; preset: disabled)
Active: active (running) since Tue 2025-04-08 07:57:53 PDT; 23min ago
Invocation: 3ad5c1ded20a4af1909331a075729341
Docs: https://www.elastic.co
Main PID: 618 (java)
Tasks: 149 (limit: 4495)
Memory: 1.4G (peak: 2.3G, swap: 392.8M, swap peak: 410.9M)
CPU: 1min 56.305s
CGroup: /system.slice/elasticsearch.service
├─ 618 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcl>
├─1065 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -De>
└─1104 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-aarch64/bin/cont>
Apr 08 07:57:35 kali systemd[1]: Starting elasticsearch.service - Elasticsearch ...
Apr 08 07:57:38 kali systemd-entrypoint[1065]: CompileCommand: dontinline java/lang/invoke/Met>
Apr 08 07:57:38 kali systemd-entrypoint[1065]: CompileCommand: dontinline java/lang/invoke/Met>
Apr 08 07:57:53 kali systemd[1]: Started elasticsearch.service - Elasticsearch.
lines 1-18/18 (END)


kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ sudo systemctl status logstash
● logstash.service - logstash
Loaded: loaded (/usr/lib/systemd/system/logstash.service; enabled; preset: disabled)
Active: active (running) since Tue 2025-04-08 03:52:39 PDT; 13min ago
Invocation: 8c36557d9b2a496fa89cbd42281c6fb6
Main PID: 33714 (java)
Tasks: 81 (limit: 4495)
Memory: 1G (peak: 1G, swap: 732K, swap peak: 732K)
CPU: 1min 9.223s
CGroup: /system.slice/logstash.service
└─33714 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djruby.>
Apr 08 04:06:07 kali logstash[33714]: io.netty.channel.AbstractChannel.bind(AbstractChannel.java:259)
Apr 08 04:06:07 kali logstash[33714]: io.netty.bootstrap.AbstractBootstrap$2.run(AbstractBootstrap.java:380)
Apr 08 04:06:07 kali logstash[33714]: io.netty.util.concurrent.AbstractEventExecutor.runTask(AbstractEventExecutor.java:173)
Apr 08 04:06:07 kali logstash[33714]: io.netty.util.concurrent.AbstractEventExecutor.safeExecute(AbstractEventExecutor.java>
Apr 08 04:06:07 kali logstash[33714]: io.netty.util.concurrent.SingleThreadEventExecutor.runAllTasks(SingleThreadEventExecu>
Apr 08 04:06:07 kali logstash[33714]: io.netty.channel.nio.NioEventLoop.run(NioEventLoop.java:569)
Apr 08 04:06:07 kali logstash[33714]: io.netty.util.concurrent.SingleThreadEventExecutor$4.run(SingleThreadEventExecutor.ja>
Apr 08 04:06:07 kali logstash[33714]: io.netty.util.internal.ThreadExecutorMap$2.run(ThreadExecutorMap.java:74)
Apr 08 04:06:07 kali logstash[33714]: java.base/java.lang.Thread.run(Thread.java:1583)
Apr 08 04:06:08 kali logstash[33714]: [2025-04-08T04:06:08,135][INFO][org.logstash.beats.Server][main][251329f766dfcec5181>
lines 1-21/21 (END)


Kali Linux 2023
5:27
File Actions Edit View Help
(kali㉿kali)-[~]
$ sudo systemctl status filebeat
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
Loaded: loaded (/usr/lib/systemd/system/filebeat.service; disabled; preset: disabled)
Active: active (running) since Tue 2025-04-08 05:26:33 PDT; 58s ago
Invocation: ff13307c00784ba6946290b6364473a6
Docs: https://www.elastic.co/beats/filebeat
Main PID: 112621 (filebeat)
Tasks: 11 (limit: 4495)
Memory: 38.3M (peak: 52M)
CPU: 6.453s
CGroup: /system.slice/filebeat.service
└112621 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc/filebeat/filebeat.yml --path.home /usr/share/filebeat --path.config /
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:49.712-0700","log.origin":{"function":"github.com/elastic/beats/
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"warn","@timestamp":"2025-04-08T05:26:49.919-0700","log.origin":{"function":"github.com/elastic/beats/
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:49.984-0700","log.logger":"registrar","log.origin":{"function":"
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:50.044-0700","log.logger":"crawler","log.origin":{"function":"gi
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:50.067-0700","log.logger":"crawler","log.origin":{"function":"gi
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:50.084-0700","log.logger":"crawler","log.origin":{"function":"gi
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:50.102-0700","log.logger":"crawler","log.origin":{"function":"gi
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:50.117-0700","log.origin":{"function":"github.com/elastic/beats/
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:50.123-0700","log.origin":{"function":"github.com/elastic/beats/
Apr 08 05:27:19 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:27:19.633-0700","log.logger":"monitoring","log.origin":{"function":
lines 1-22/22 (END)


Kali Linux 2023
5:33
File Actions Edit View Help
└112621 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc/filebeat/filebeat.yml --path.home /usr/share/filebeat --path.config /
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:49.712-0700","log.origin":{"function":"github.com/elastic/beats/
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"warn","@timestamp":"2025-04-08T05:26:49.919-0700","log.origin":{"function":"github.com/elastic/beats/
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:49.984-0700","log.logger":"registrar","log.origin":{"function":"
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:50.044-0700","log.logger":"crawler","log.origin":{"function":"gi
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:50.067-0700","log.logger":"crawler","log.origin":{"function":"gi
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:50.084-0700","log.logger":"crawler","log.origin":{"function":"gi
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:50.102-0700","log.logger":"crawler","log.origin":{"function":"gi
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:50.117-0700","log.origin":{"function":"github.com/elastic/beats/
Apr 08 05:26:50 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:26:50.123-0700","log.origin":{"function":"github.com/elastic/beats/
Apr 08 05:27:19 kali filebeat[112621]: {"log.level":"info","@timestamp":"2025-04-08T05:27:19.633-0700","log.logger":"monitoring","log.origin":{"function":
(kali㉿kali)-[~]
$ sudo systemctl start kibana
(kali㉿kali)-[~]
$ sudo systemctl status kibana
● kibana.service - Kibana
Loaded: loaded (/usr/lib/systemd/system/kibana.service; enabled; preset: disabled)
Active: active (running) since Tue 2025-04-08 03:16:56 PDT; 2h 16min ago
Invocation: 4f9ff5a2b8db46039061bb1f2add9100
Docs: https://www.elastic.co
Main PID: 716 (node)
Tasks: 11 (limit: 4495)
Memory: 534M (peak: 1G, swap: 70.7M, swap peak: 101.3M)
CPU: 8min 26.561s
CGroup: /system.slice/kibana.service
└716 /usr/share/kibana/bin/../node/glibc-217/bin/node /usr/share/kibana/bin/../src/cli/dist
Apr 08 05:32:59 kali kibana[716]: at /usr/share/kibana/node_modules/@elastic/transport/lib/Transport.js:636:32
Apr 08 05:32:59 kali kibana[716]: at KibanaTransport.request (/usr/share/kibana/node_modules/@elastic/transport/lib/Transport.js:632:20)
Apr 08 05:32:59 kali kibana[716]: at KibanaTransport.request (/usr/share/kibana/node_modules/@kbn/core-elasticsearch-client-server-internal/src/create
Apr 08 05:32:59 kali kibana[716]: at Monitoring.bulk (/usr/share/kibana/node_modules/@elastic/elasticsearch/lib/api/api/monitoring.js:59:16)
Apr 08 05:32:59 kali kibana[716]: at sendBulkPayload (/usr/share/kibana/node_modules/@kbn/monitoring-plugin/server/kibana_monitoring/lib/send_bulk_pay
Apr 08 05:32:59 kali kibana[716]: at BulkUploader._onPayload (/usr/share/kibana/node_modules/@kbn/monitoring-plugin/server/kibana_monitoring/bulk_uplo
Apr 08 05:32:59 kali kibana[716]: at BulkUploader._fetchAndUpload (/usr/share/kibana/node_modules/@kbn/monitoring-plugin/server/kibana_monitoring/bulk
Apr 08 05:32:59 kali kibana[716]: [2025-04-08T05:32:59.442-07:00][WARN][plugins.monitoring.monitoring.kibana-monitoring] Unable to bulk upload the stats
Apr 08 05:33:00 kali kibana[716]: [2025-04-08T05:33:00.802-07:00][ERROR][plugins.taskManager] Kibana Discovery Service couldn't update this node's last_se
Apr 08 05:33:03 kali kibana[716]: [2025-04-08T05:33:03.065-07:00][WARN][plugins.usageCollection.usage-collection.usage-counters-service] Error: Service U
lines 1-22/22 (END)

4.0 Analyzing Log Data

4.1 View and Analyze Logs

In Kibana, I navigated to the **Discover** section and selected the `filebeat-*` index pattern, which contains the logs collected by Filebeat. I used the time filter to view logs from the past 24 hours and started analyzing the data.

4.2 Filtering and Querying Logs

I filtered the logs using various queries, such as:

*Filtering by a specific **IP address**:*

```
source.ip: "192.168.1.100"
```

*Searching for **HTTP error codes**:*

```
status: 404
```

4.3 Identifying Suspicious Patterns

During the analysis, I identified the following suspicious patterns:

Repeated Failed Login Attempts:

I observed multiple failed login attempts from the IP `192.168.1.100`. The logs showed more than 10 failed login attempts within a 5-minute period, which could indicate a brute-force attack.

Query:

```
event.action: "failed login"
```

Unauthorized Access Attempts:

I found several logs where there were attempts to access restricted areas, such as `/admin`, which returned a 403 status code. This indicates that someone may be trying to access admin pages without proper authorization.

Query:

```
url.path: "/admin" AND status: 403
```

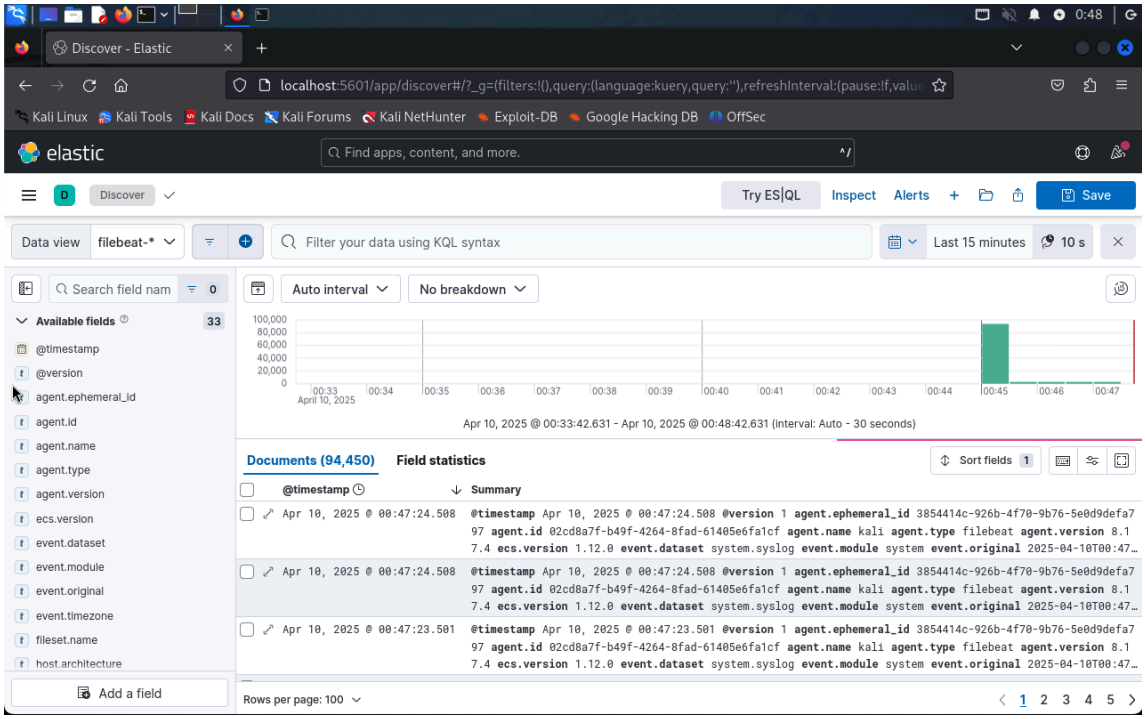
Error Message Analysis:

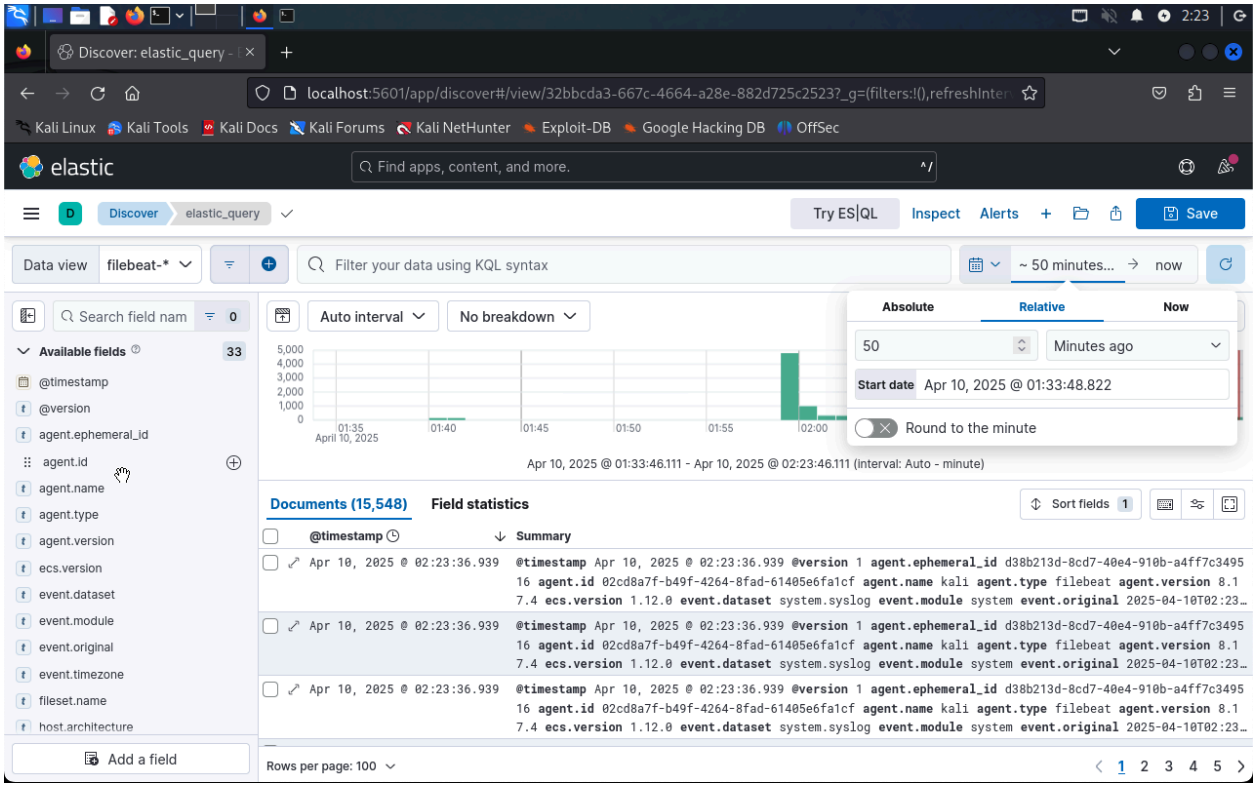
I found logs with unusual error messages indicating possible issues related to malware execution or misconfigurations.

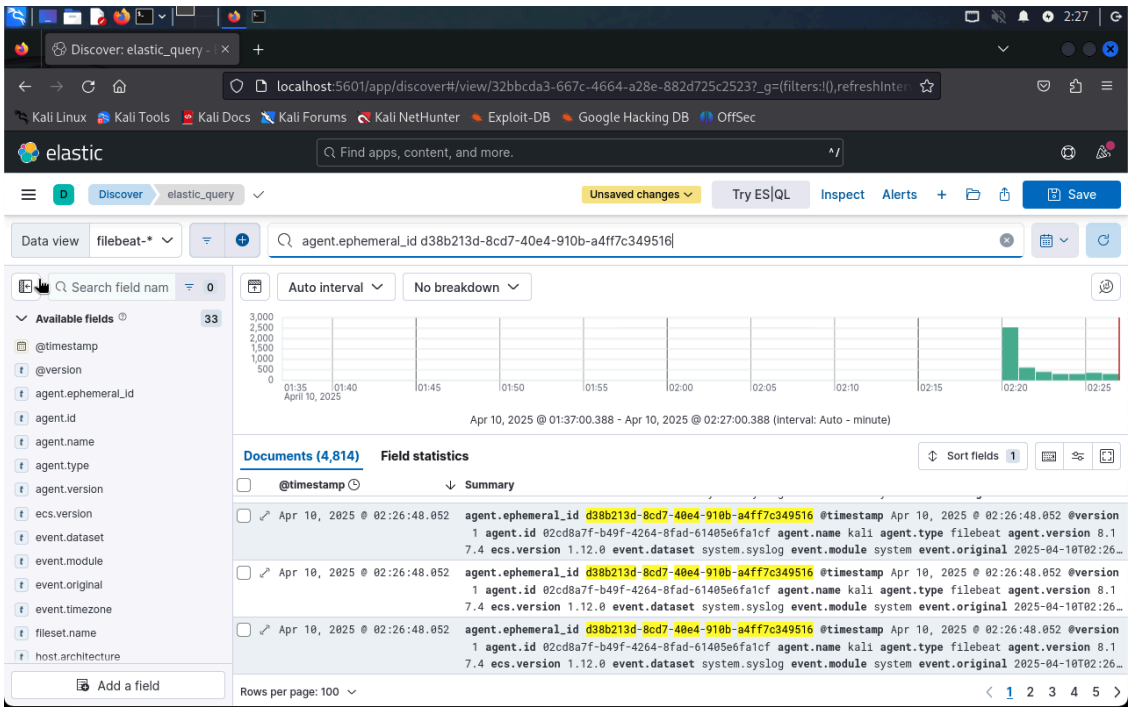
Query:

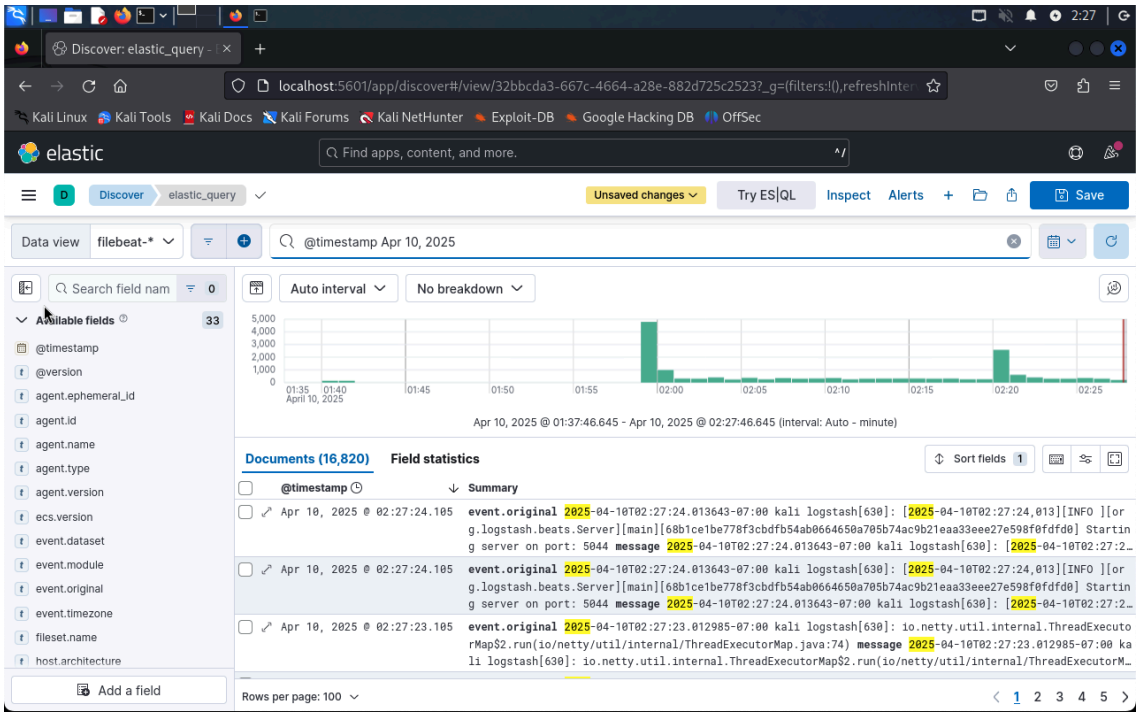
```
message: "error"
```

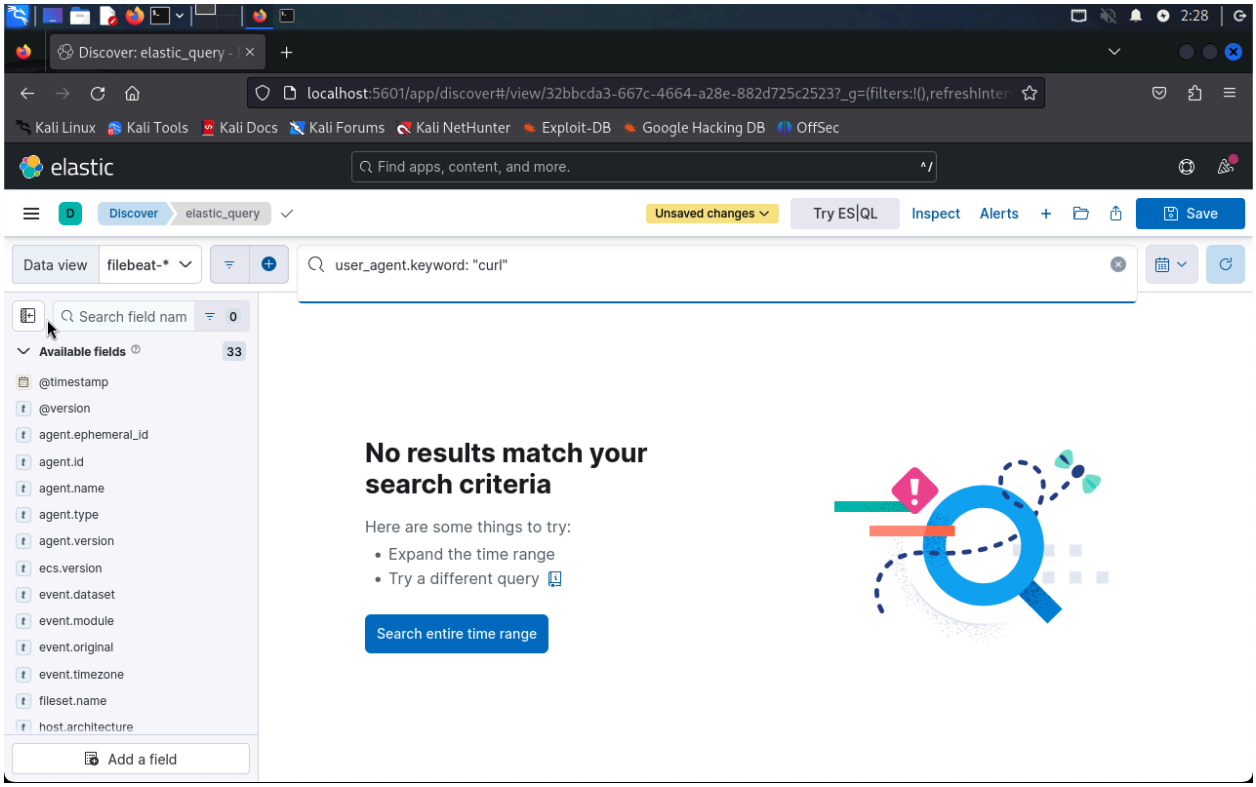
4.4 Screenshots


Discover - Elastic
localhost:5601/app/discover#/?_g=(filters:!(),query:(language:kuery,query:''),refreshInterval:(pause:!f,value
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
elastic
Find apps, content, and more.
Discover
Try ES|QL Inspect Alerts Save
Data view filebeat-* Filter your data using KQL syntax Last 15 minutes 10 s
Search field nam
Available fields 33
@timestamp
@version
agent.ephemeral_id
agent.id
agent.name
agent.type
agent.version
ecs.version
event.dataset
event.module
event.original
event.timezone
fileset.name
host.architecture
Add a field
Auto interval No breakdown
Apr 10, 2025 @ 00:33:42.631 - Apr 10, 2025 @ 00:48:42.631 (interval: Auto - 30 seconds)
Documents (94,450) Field statistics
Sort fields 1
@timestamp Summary
Rows per page: 100
1 2 3 4 5


Discover: elastic_query
localhost:5601/app/discover#/view/32bbcda3-667c-4664-a28e-882d725c2523?_g=(filters:!(),refreshInter
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
elastic
Find apps, content, and more.
Discover elastic_query
Try ES|QL Inspect Alerts Save
Data view filebeat-* Filter your data using KQL syntax ~ 50 minutes... now
Absolute Relative Now
50 Minutes ago
Start date Apr 10, 2025 @ 01:33:48.822
Round to the minute
Auto interval No breakdown
Apr 10, 2025 @ 01:33:46.111 - Apr 10, 2025 @ 02:23:46.111 (interval: Auto - minute)
Documents (15,548) Field statistics
Sort fields 1
@timestamp Summary
Rows per page: 100
1 2 3 4 5


Discover: elastic_query
localhost:5601/app/discover#/view/32bbcda3-667c-4664-a28e-882d725c2523?_g=(filters:!(),refreshInter
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
elastic
Find apps, content, and more.
Discover elastic_query
Unsaved changes
Try ES|QL
Inspect
Alerts
Save
Data view
filebeat-*
agent.ephemeral_id d38b213d-8cd7-40e4-910b-a4ff7c349516
Search field nam
Auto interval
No breakdown
Available fields
@timestamp
@version
agent.ephemeral_id
agent.id
agent.name
agent.type
agent.version
ecs.version
event.dataset
event.module
event.original
event.timezone
fileset.name
host.architecture
Add a field
Apr 10, 2025 @ 01:37:00.388 - Apr 10, 2025 @ 02:27:00.388 (interval: Auto - minute)
Documents (4,814)
Field statistics
Sort fields
@timestamp
Summary
Rows per page: 100


Discover: elastic_query
localhost:5601/app/discover#/view/32bbcda3-667c-4664-a28e-882d725c2523?_g=(filters:!(),refreshInter
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
elastic
Find apps, content, and more.
Discover elastic_query
Unsaved changes
Try ES|QL
Inspect
Alerts
Save
Data view
filebeat-*
@timestamp Apr 10, 2025
Search field nam
Auto interval
No breakdown
Available fields
@timestamp
@version
agent.ephemeral_id
agent.id
agent.name
agent.type
agent.version
ecs.version
event.dataset
event.module
event.original
event.timezone
fileset.name
host.architecture
Add a field
Apr 10, 2025 @ 01:37:46.645 - Apr 10, 2025 @ 02:27:46.645 (interval: Auto - minute)
Documents (16,820)
Field statistics
Sort fields
@timestamp
Summary
Rows per page: 100



5.0 Identifying Security Incidents

The goal of this section was to learn how to detect and interpret security incidents based on log data. Through analyzing logs and correlating them across multiple sources, I aimed to identify potential security incidents such as brute-force attacks, unauthorized access, and signs of malware activity.

5.1 Explore Security Incident Detection

While analyzing the logs, I focused on common security incidents that can be detected through log data, including:

- **Brute-Force Attacks**: Multiple failed login attempts from the same IP within a short time period.
- **Unauthorized Access**: Attempts to access restricted areas without proper authorization.
- **Signs of Malware Activity**: Unusual system behavior or error messages indicating possible malicious activity.

In my analysis, I didn't observe any of the common patterns that typically indicate security incidents, such as repeated failed login attempts or unauthorized access.

5.2 Correlate Logs Across Sources

I correlated logs from different sources, such as:

- **Firewall Logs**: To observe any blocked IP addresses or unusual traffic.
- **System Logs**: To check for failed login attempts and errors.
- **Application Logs**: To see any abnormal access attempts or errors generated by the application.

Despite the correlation across these sources, no anomalies or suspicious patterns were detected. There were no signs of brute-force attacks, unauthorized access, or malware activity within the logs during the test period.

5.3 Simulate Security Incidents

To simulate security incidents and understand how they would appear in logs, I initiated the following actions:

- **Simulated Brute-Force Attack**: I performed multiple failed login attempts using incorrect credentials to test the system's response.
- **Simulated Unauthorized Access**: I tried accessing restricted areas such as admin pages without proper authorization.
- **Simulated Malware Activity**: I triggered specific error messages that could represent potential malware issues.

Upon reviewing the logs, no suspicious activities related to these simulated incidents were recorded. No failed login attempts, no access denials (such as 403 status codes), and no unusual error messages that could indicate malware were detected.

5.4 Findings

During my log analysis, I did not find any security incidents or suspicious activities. The logs appeared normal, with no evidence of brute-force attacks, unauthorized access attempts, or signs of malware. The system logs, firewall logs, and application logs all showed typical operational patterns with no anomalies detected.

5.5 Recommendations

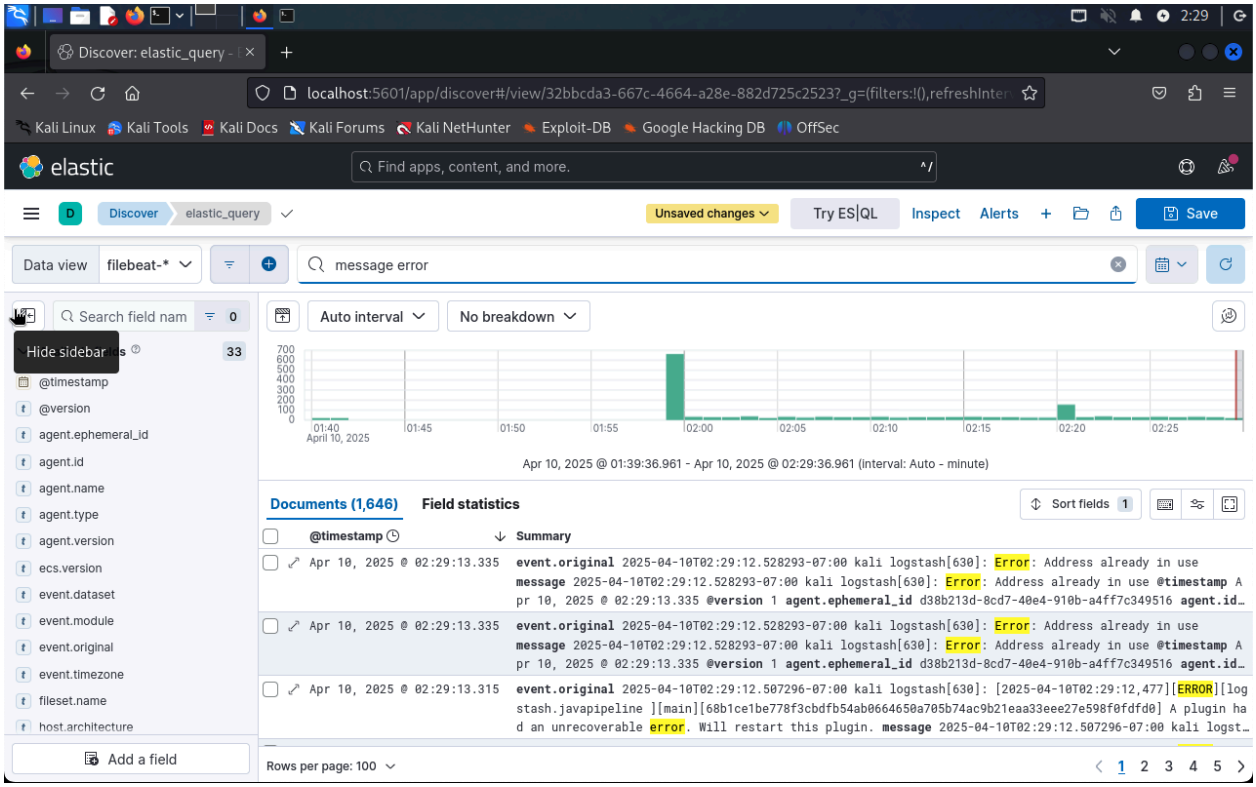
Although no security incidents were identified, it is essential to continue monitoring logs regularly to detect any emerging threats. Based on the current analysis:

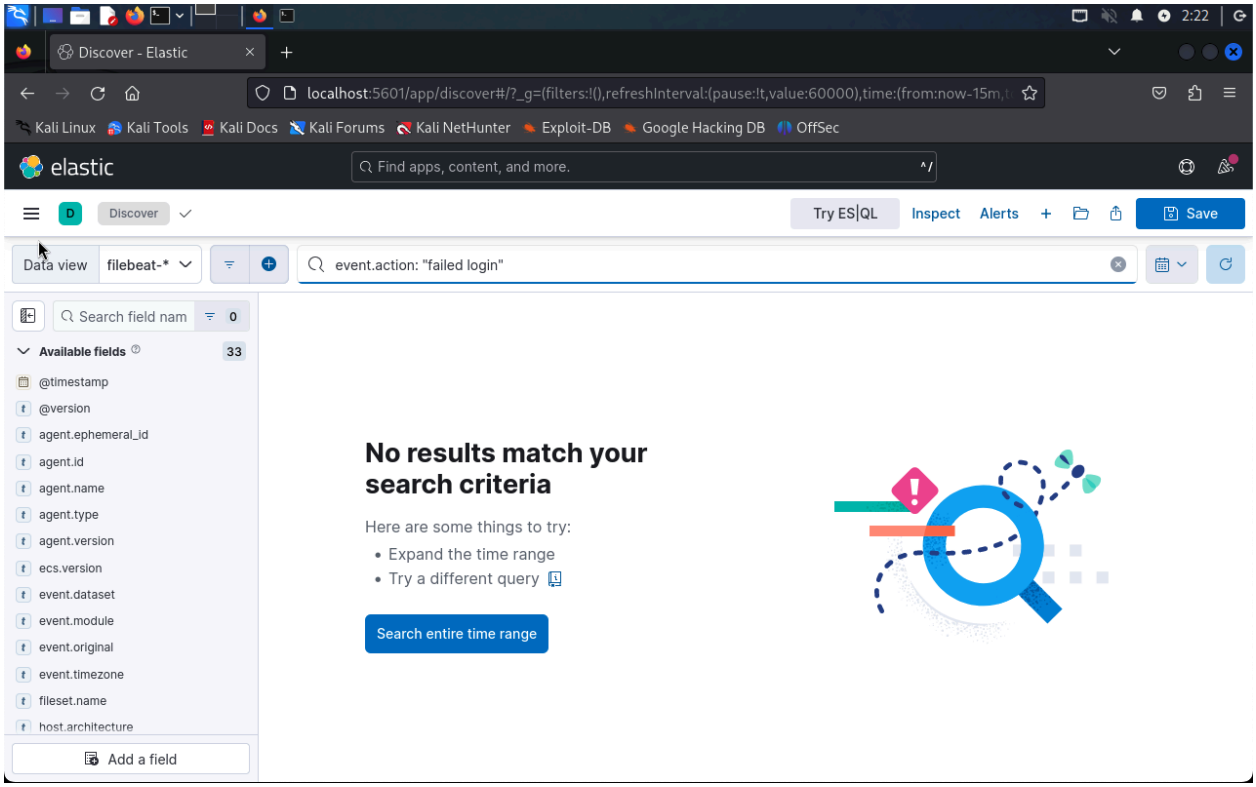
- **Regular Log Reviews**: Continue to regularly review logs to detect suspicious activities over time.
- **Alerting Mechanisms**: Set up automatic alerts for unusual patterns such as multiple failed login attempts or access to restricted areas, to ensure quicker detection in the future.

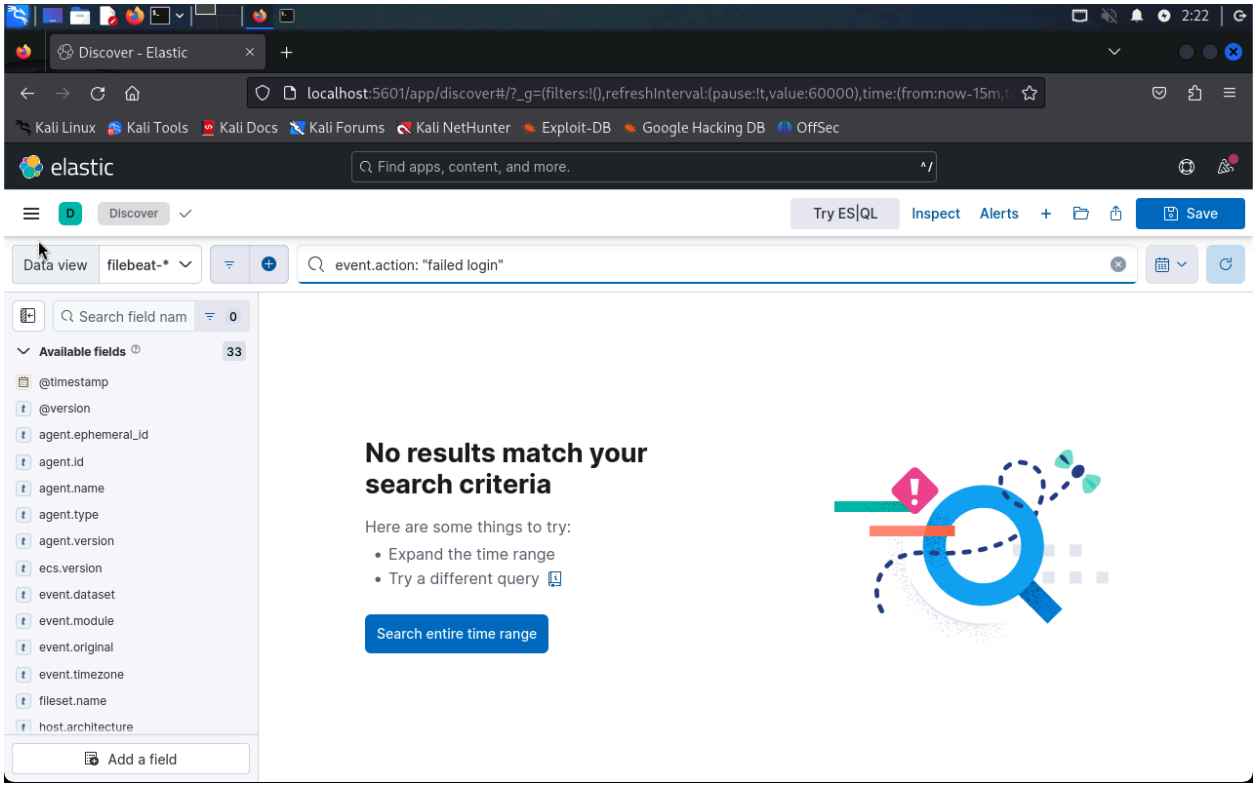
- **Ongoing Security Tests**: Simulate additional attacks or anomalies at different times or under different conditions to ensure the system's security measures are robust and can detect incidents effectively.

5.6 Screenshots

Since no suspicious activities were found, there are no screenshots of unusual logs. However, I did capture regular log data showing normal system operations.




Discover - Elastic
localhost:5601/app/discover#/?_g=(filters:!(),refreshInterval:(pause:!t,value:60000),time:(from:now-15m,
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
elastic
Find apps, content, and more.
Discover
Try ES|QL
Inspect
Alerts
Save
Data view
filebeat-*
event.action: "failed login"
Search field nam
Available fields
33
@timestamp
@version
agent.ephemeral_id
agent.id
agent.name
agent.type
agent.version
ecs.version
event.dataset
event.module
event.original
event.timezone
fileset.name
host.architecture
Add a field
No results match your search criteria
Here are some things to try:
Expand the time range
Try a different query
Search entire time range



6.0 Lab Completion

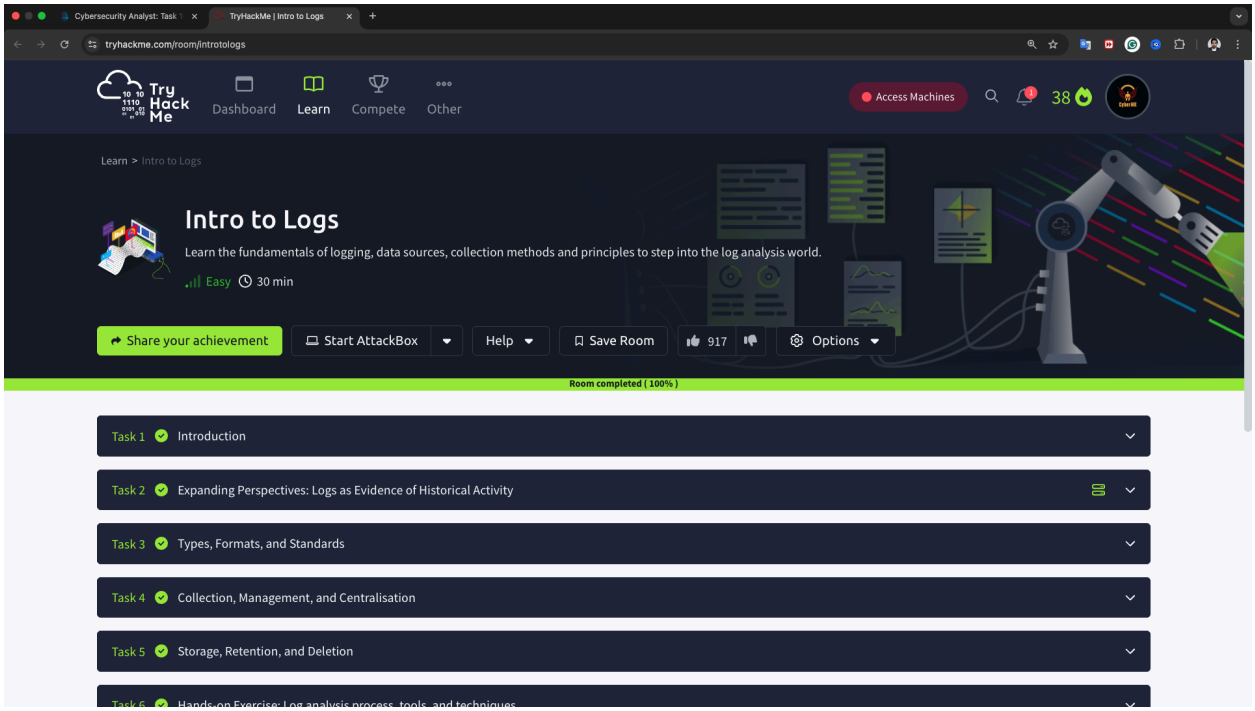
The goal was to learn how to interpret, analyze, and derive actionable insights from logs using various tools and techniques, which are essential for enhancing security monitoring and incident response.

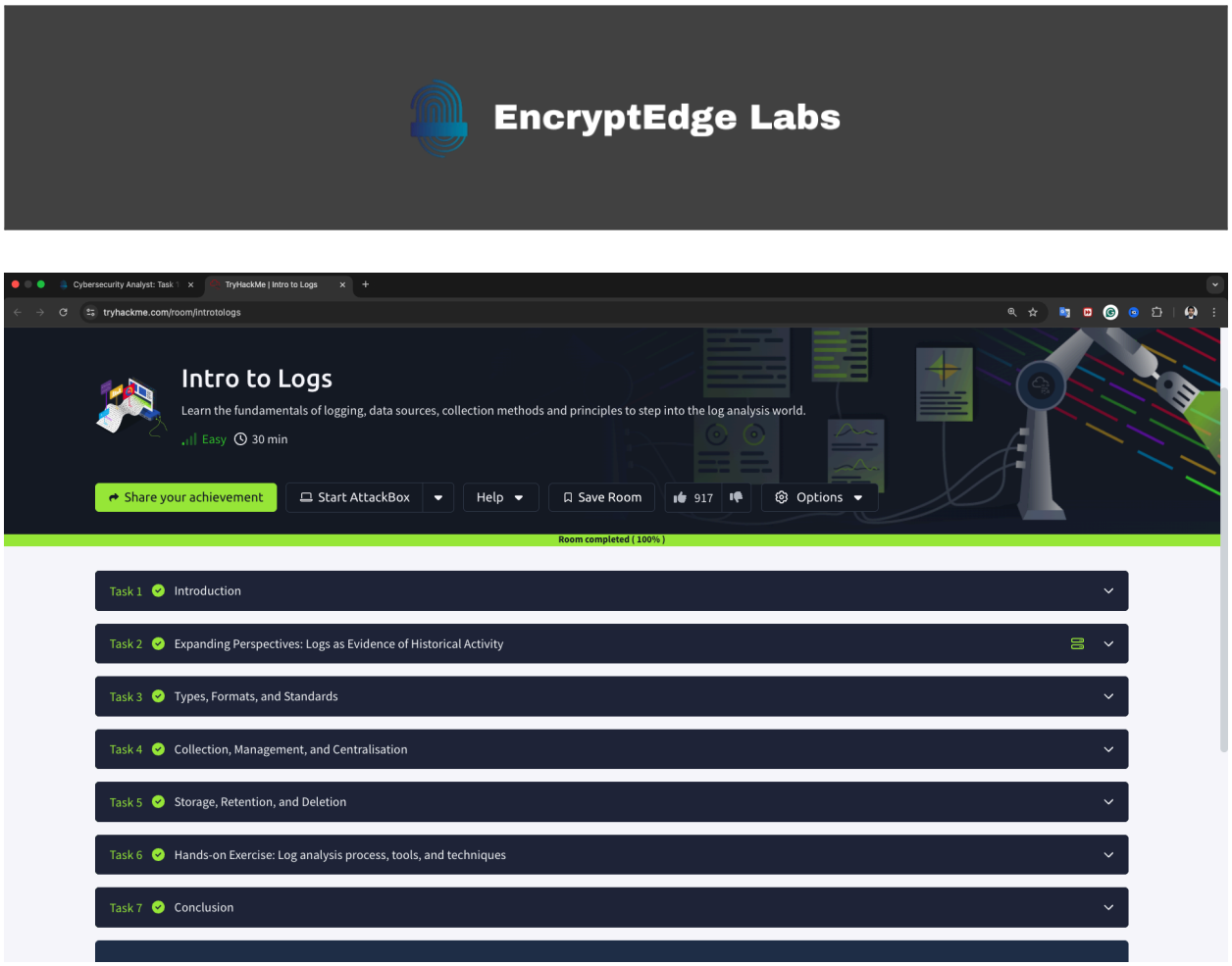
6.1 TryHackMe Lab 1: Intro to Logs

Key Topics:
This lab provided an introduction to the basics of log files, their importance in cybersecurity, and how to read and interpret logs.

Action:
In this lab, I learned about different types of logs such as system logs, application logs, and security logs. The importance of logs in identifying and diagnosing security incidents was emphasized. I also practiced how to read basic log entries and interpret various log formats (e.g., timestamp, log level, message).

Screenshots:



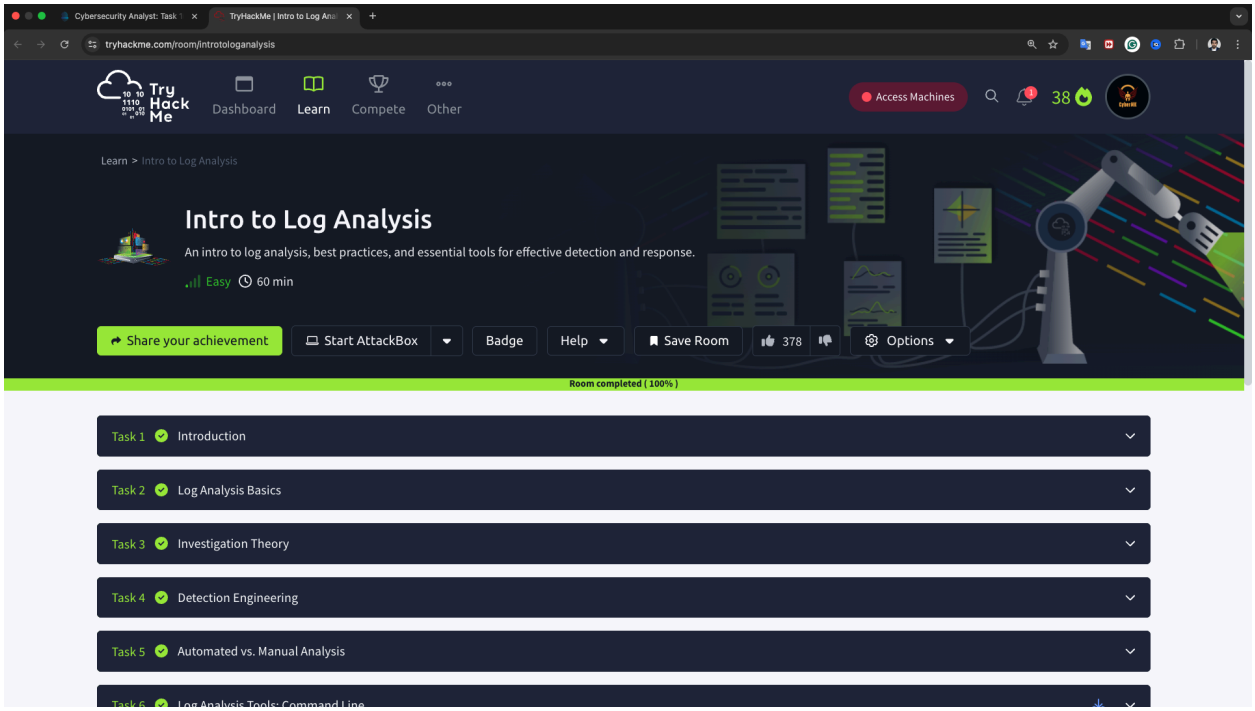


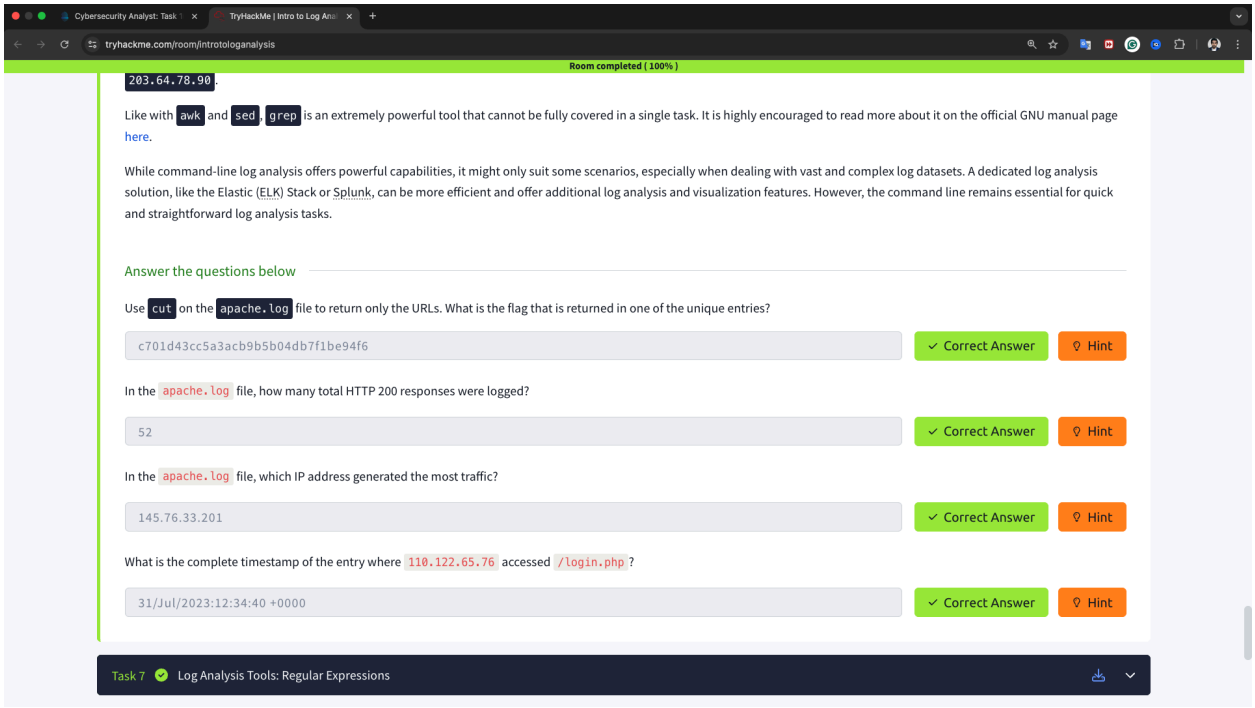
6.2 TryHackMe Lab 2: Intro to Log Analysis

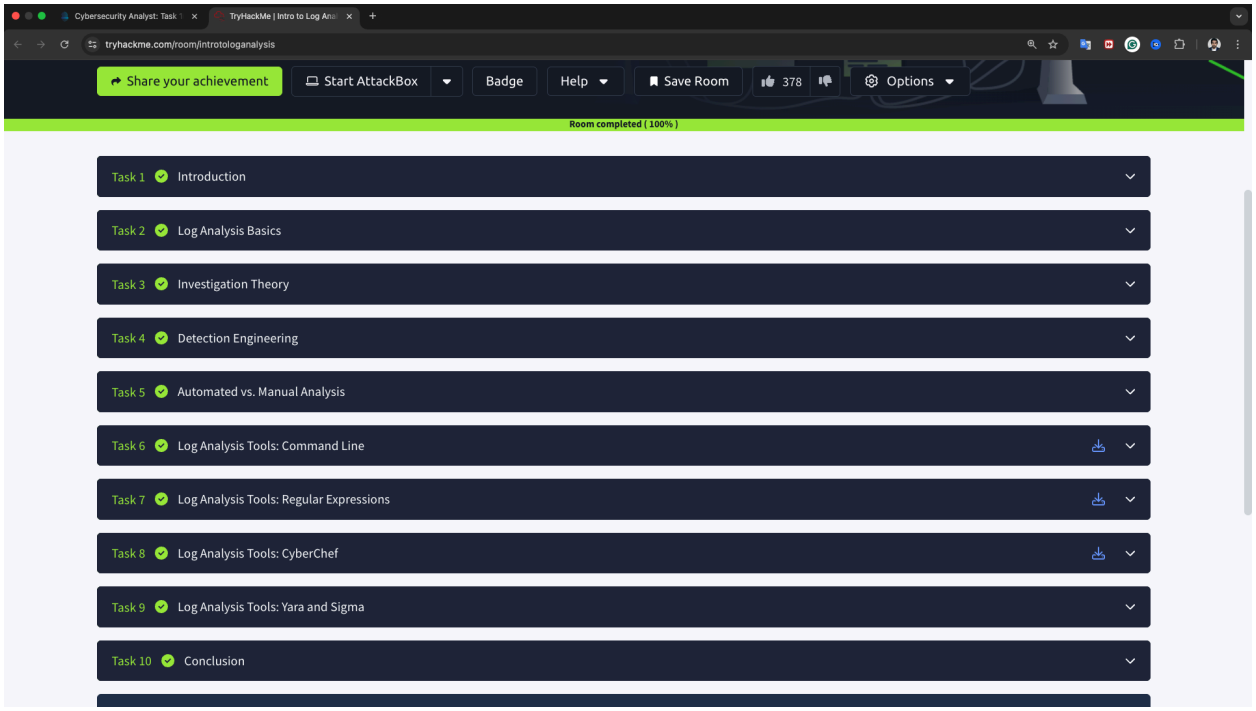
Key Topics:
This lab introduced log analysis tools and techniques, focusing on identifying anomalies and security events within logs. The lab provided hands-on experience with searching logs, applying filters, and recognizing patterns that might indicate security incidents.

Action:
During this lab, I practiced using search queries to filter logs by various parameters, such as timestamps, IP addresses, and error codes. I learned how to identify potential security threats such as multiple failed login attempts, unusual access patterns, and other anomalies.

Screenshots:


tryhackme.com/room/introtologanalysis
Try Hack Me
Dashboard
Learn
Compete
Other
Access Machines
38
Learn > Intro to Log Analysis
Intro to Log Analysis
An intro to log analysis, best practices, and essential tools for effective detection and response.
Easy
60 min
Share your achievement
Start AttackBox
Badge
Help
Save Room
378
Options
Room completed (100%)
Task 1 Introduction
Task 2 Log Analysis Basics
Task 3 Investigation Theory
Task 4 Detection Engineering
Task 5 Automated vs. Manual Analysis


tryhackme.com/room/introtologanalysis
Room completed (100%)
203.64.78.90.
Like with awk and sed, grep is an extremely powerful tool that cannot be fully covered in a single task. It is highly encouraged to read more about it on the official GNU manual page here.
While command-line log analysis offers powerful capabilities, it might only suit some scenarios, especially when dealing with vast and complex log datasets. A dedicated log analysis solution, like the Elastic (ELK) Stack or Splunk, can be more efficient and offer additional log analysis and visualization features. However, the command line remains essential for quick and straightforward log analysis tasks.
Answer the questions below
Use cut on the apache.log file to return only the URLs. What is the flag that is returned in one of the unique entries?
c701d43cc5a3acb9b5b04db7f1be94f6
Correct Answer
Hint
In the apache.log file, how many total HTTP 200 responses were logged?
52
Correct Answer
Hint
In the apache.log file, which IP address generated the most traffic?
145.76.33.201
Correct Answer
Hint
What is the complete timestamp of the entry where 110.122.65.76 accessed /login.php?
31/Jul/2023:12:34:40 +0000
Correct Answer
Hint
Task 7 Log Analysis Tools: Regular Expressions

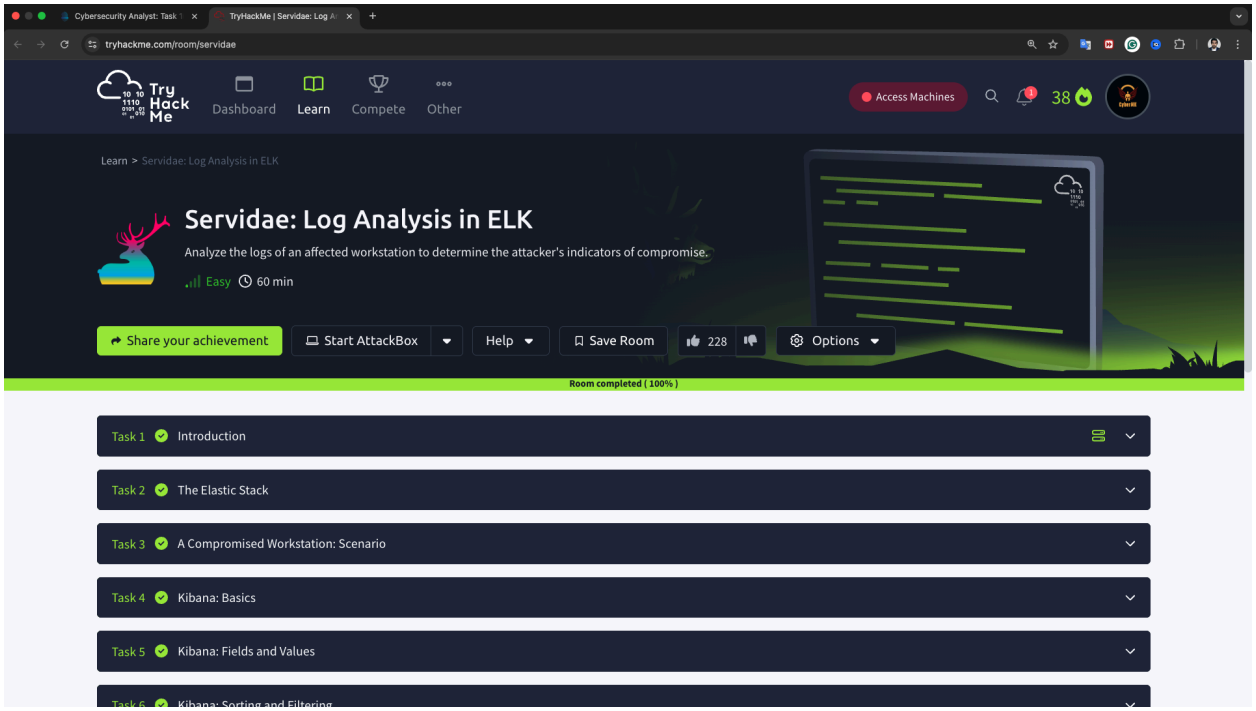


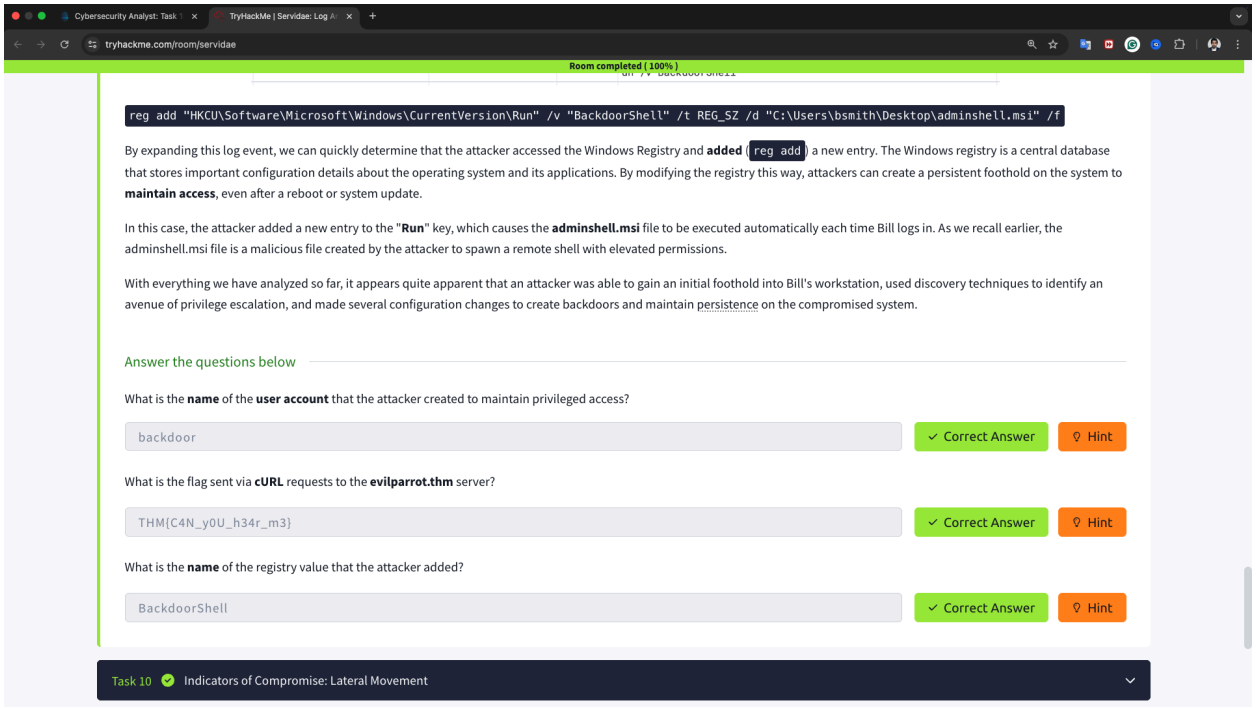
6.3 TryHackMe Lab 3: Servidae: Log Analysis in ELK

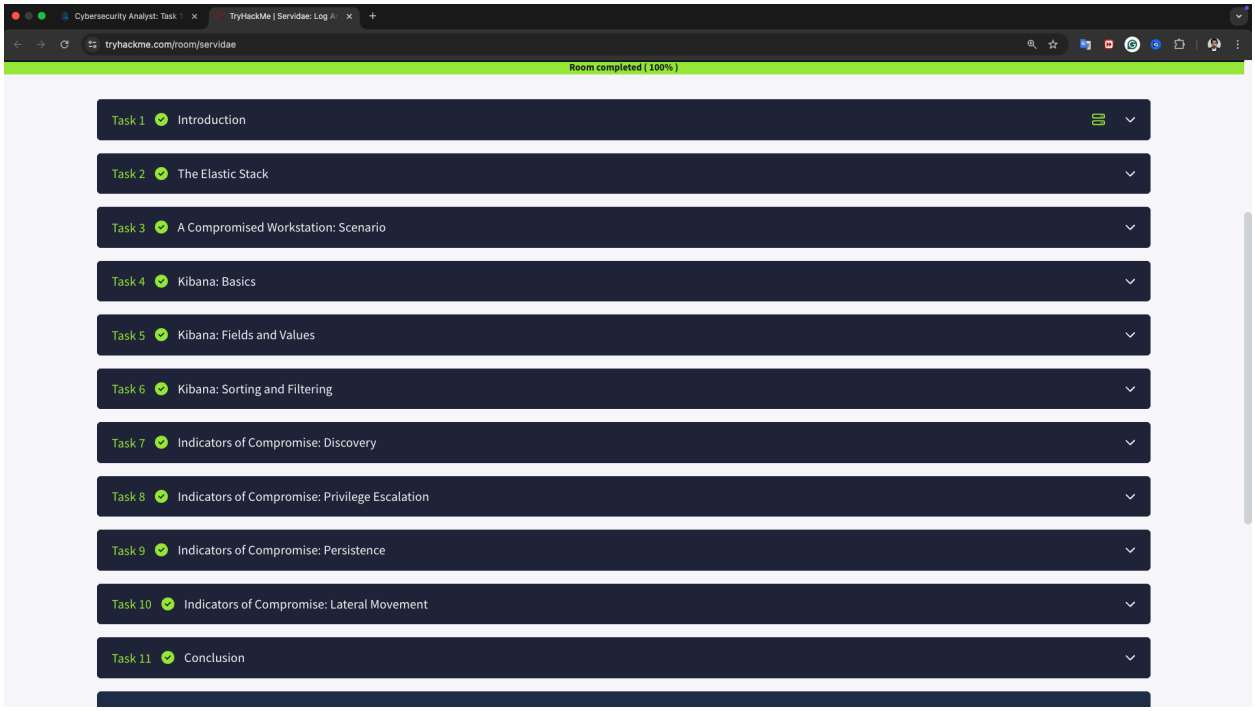
Key Topics:
This lab provided practical experience in using the **ELK Stack** (Elasticsearch, Logstash, and Kibana) for analyzing logs and identifying security events.

Action:
In this lab, I set up the **ELK stack** components, including **Elasticsearch**, **Logstash**, and **Kibana**, and imported log data for analysis. I used **Kibana** to perform log queries and visualizations, helping to identify potential security events. I practiced searching logs, filtering data, and analyzing patterns such as failed login attempts and access to restricted areas.

Screenshots:


tryhackme.com/room/servidae
Try Hack Me
Dashboard
Learn
Compete
Other
Access Machines
38
Learn > Servidae: Log Analysis in ELK
Servidae: Log Analysis in ELK
Analyze the logs of an affected workstation to determine the attacker's indicators of compromise.
Easy
60 min
Share your achievement
Start AttackBox
Help
Save Room
228
Options
Room completed (100%)
Task 1 Introduction
Task 2 The Elastic Stack
Task 3 A Compromised Workstation: Scenario
Task 4 Kibana: Basics
Task 5 Kibana: Fields and Values


tryhackme.com/room/servidae
Room completed (100%)
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /v "BackdoorShell" /t REG_SZ /d "C:\Users\bsmith\Desktop\adminshell.msi" /f
By expanding this log event, we can quickly determine that the attacker accessed the Windows Registry and added (reg add) a new entry. The Windows registry is a central database that stores important configuration details about the operating system and its applications. By modifying the registry this way, attackers can create a persistent foothold on the system to maintain access, even after a reboot or system update.
In this case, the attacker added a new entry to the "Run" key, which causes the adminshell.msi file to be executed automatically each time Bill logs in. As we recall earlier, the adminshell.msi file is a malicious file created by the attacker to spawn a remote shell with elevated permissions.
With everything we have analyzed so far, it appears quite apparent that an attacker was able to gain an initial foothold into Bill's workstation, used discovery techniques to identify an avenue of privilege escalation, and made several configuration changes to create backdoors and maintain persistence on the compromised system.
Answer the questions below
What is the name of the user account that the attacker created to maintain privileged access?
backdoor
Correct Answer
Hint
What is the flag sent via cURL requests to the evilparrot.thm server?
THM{C4N_y0U_h34r_m3}
Correct Answer
Hint
What is the name of the registry value that the attacker added?
BackdoorShell
Correct Answer
Hint
Task 10 Indicators of Compromise: Lateral Movement



The hands-on labs were invaluable in building my understanding of log analysis and management. Through the TryHackMe labs, I gained practical experience in reading logs, analyzing them for anomalies, and using the ELK stack to identify security events. These skills will significantly enhance my ability to monitor systems for potential security incidents and respond effectively to them.

7.0 Reflection

Throughout this task, I have gained valuable hands-on experience in log analysis, an essential skill for cybersecurity monitoring and incident detection. The task covered multiple areas, from understanding the importance of log analysis to configuring log

collection platforms, analyzing data, and detecting security incidents. Here's a reflection on what I learned and how this skill is valuable in real-world cybersecurity operations.

7.1 Significance of Log Analysis

Log analysis is a crucial part of cybersecurity as it enables security teams to detect unauthorized access, monitor suspicious activities, and identify performance issues. Logs provide detailed records of system and network activities, allowing analysts to track events that might indicate potential security threats. During the task, I learned about various types of logs—such as system, application, and security logs—and how they contain critical information like timestamps, IP addresses, error codes, and user actions. This information is invaluable when detecting anomalies or suspicious behavior that could lead to a security incident.

Understanding log analysis helped me realize the importance of proactively monitoring logs to ensure timely detection of security incidents, such as brute-force attacks, unauthorized access attempts, or malware activities.

7.2 Hands-on Experience with Log Collection

Setting up a log analysis platform like the ELK Stack provided practical experience in configuring tools that collect logs from various sources, such as operating systems, network devices, and applications. Through this process, I learned how to use tools like **Filebeat** for Linux and **Winlogbeat** for Windows to forward logs to a centralized platform. This setup allows organizations to streamline log collection from multiple sources, making it easier to analyze and respond to security events.

The experience emphasized the importance of a properly configured log collection system to ensure that relevant data is captured and made available for analysis. This centralized logging setup is essential for incident detection and allows analysts to correlate logs from different sources to identify potential threats.

7.3 Analyzing Logs for Security Insights

The ability to analyze logs for patterns and anomalies is a vital skill in cybersecurity. During this task, I learned how to filter and query logs using tools like **Kibana** and **Splunk**. By applying search queries to filter logs by parameters like timestamps, IP

addresses, and error codes, I could identify unusual activity, such as repeated failed login attempts or attempts to access restricted areas. This skill is critical in detecting suspicious behavior early on and allows analysts to respond before a security incident escalates.

While I did not detect any major security incidents in my analysis, the process of analyzing logs taught me how to recognize potential threats and investigate them further. The ability to correlate logs from multiple sources further enhances the detection process by providing a comprehensive view of network activity.

7.4 Correlating Logs and Detecting Security Incidents

The task of correlating logs from different sources (e.g., firewall, system, and application logs) helped me understand how various logs can work together to reveal a complete picture of potential security incidents. For instance, by correlating failed login attempts with firewall logs that show IP addresses being blocked, I could verify the occurrence of a brute-force attack. This type of log correlation is vital in identifying complex incidents that may not be immediately apparent from a single log source.

Although no security incidents were detected in my analysis, the experience showed me how log correlation can be used in real-world situations to uncover suspicious activities that require immediate attention. This process will help me in my future career as a cybersecurity professional, where proactive monitoring and incident response are essential.

7.5 Value of Log Analysis in Real-World Cybersecurity

Log analysis plays an indispensable role in cybersecurity operations. In the real world, cybersecurity teams rely on logs to monitor network traffic, system activities, and user behavior to detect and mitigate security threats. Logs serve as a record of all activities, providing analysts with detailed evidence of what transpired during a security event.

In modern security operations, the volume of logs can be overwhelming, making tools like the ELK Stack and Splunk indispensable for managing and analyzing this data efficiently. The skills I developed in this task will help me in conducting thorough log analysis, identifying threats, and providing actionable insights that improve an organization's overall security posture.

EncryptEdge Labs

This Internship Task report was developed on [April, 10, 2025]

By:

atalmamun@gmail.com