🛡️Security Alert Monitoring &

Incident Response

REPORT BY: Dharini S INTERNSHIP AT: Future interns

📑 Table of Contents

1. Cover Page

2. Table of Contents

3. Task Overview

4. Tools & Setup

5. Workflow Summary

6. Admin Probe Detection

7. Payload Spike Analysis   
8.Previlage Escation Review

9. SOC Dashboard Summary

10. Analyst Reflection

11. Conclusion

Task Overview

🎯Objective:

To monitor simulated security alerts using Elastic Stack (ELK), identify suspicious activities in log data, classify incidents, and draft a structured incident response report.

📚 Skills Developed:

- Log analysis and alert triage using Discover and Lens panels

- SOC fundamentals including incident classification and timeline correlation

- Building visual dashboards to support forensic summaries

- Writing professional-grade reports to communicate threat behavior

🔹 Tools & Setup:  
  
🛠️ ELK Stack Overview

The ELK Stack (Elasticsearch, Logstash, Kibana) was used as the primary SIEM tool to ingest, analyze, and visualize sample security logs. Together, these components offer powerful filtering, search, and dashboarding capabilities suitable for threat hunting and incident response.

- \*\*Elasticsearch\*\*: Stores and indexes structured log data for fast search and retrieval

- \*\*Logstash/Filebeat\*\*: Handles log ingestion and pipeline formatting

- \*\*Kibana\*\*: Visual exploration and dashboard creation tool for analyzing logs through Lens and Discover

⚙️ Environment Setup Steps

1. Indexed log files into \*\*Elasticsearch\*\* via Filebeat

2. Created index pattern in \*\*Kibana\*\* to access security-related fields

3. Verified incoming data via \*\*Discover\*\* and checked field mappings

4. Used \*\*Lens\*\* to build stacked bar charts, pie charts, and time-series visualizations

5. Enabled filters for fields such as:

- `url.original`

- `source.address`

- `http.response.status\_code`

- `@timestamp`

- `user\_agent.origina  
  
  
 ⚙ Data View Configuration

Before building visualizations, a Data View was configured in \*Stack Management → Data Views\*. This setup ensured that all relevant fields (like url.original, source.address, @timestamp, and http.response.status\_code) were properly recognized and accessible for visual analysis.

Key benefits:

- Enabled granular filtering by endpoint and IP address

- Supported time-series charts and status code breakdowns

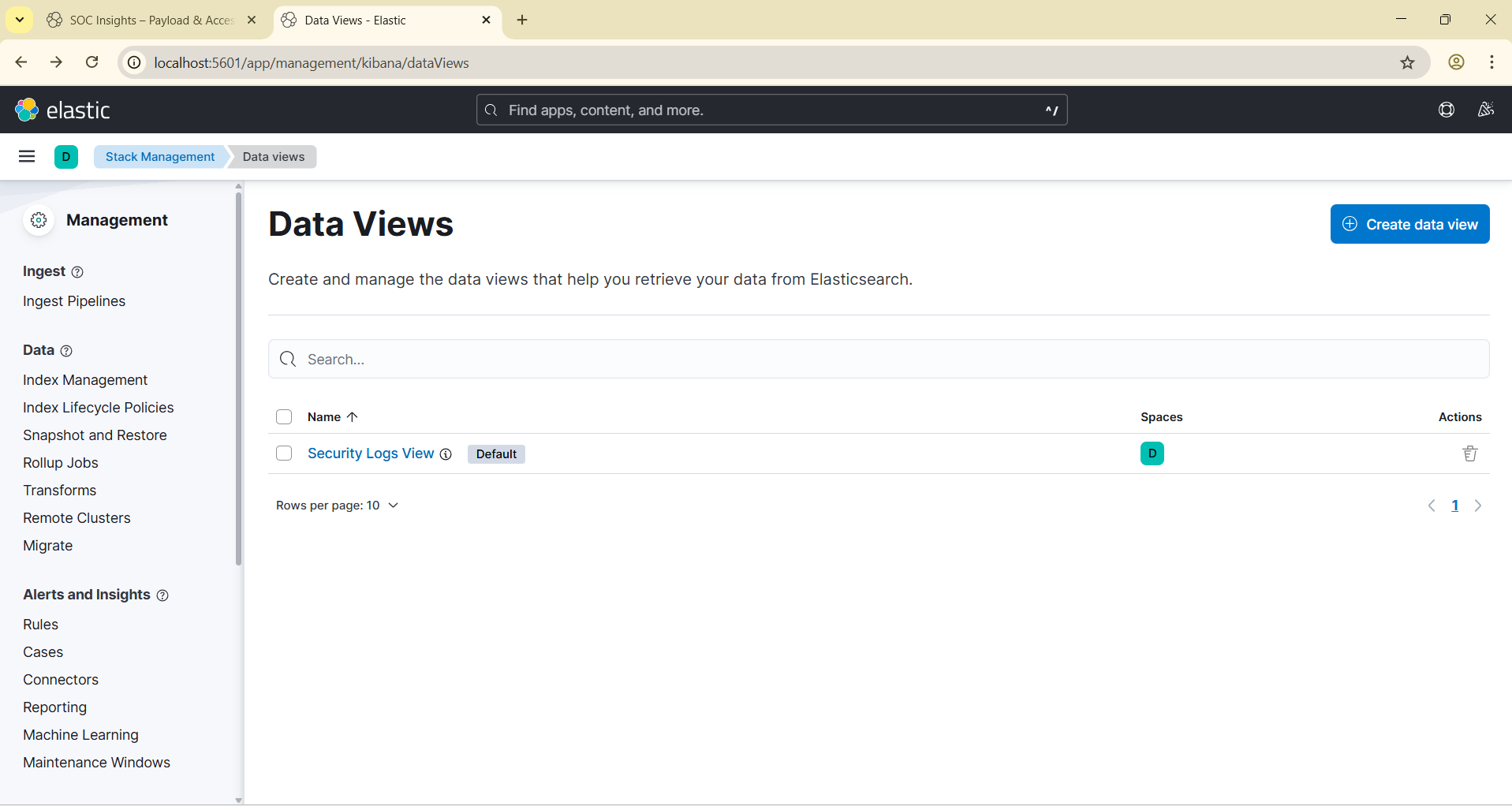
- Ensured logs were correctly mapped and searchable  
  


Figure 1: Kibana Data View Setup confirming availability of log fields for SOC analysis and visualization

🔹 Workflow Summary

🧭 Investigation Workflow

The alert monitoring process followed a structured SOC workflow using Kibana, consisting of the following key stages:

1️⃣ Log Ingestion & Discovery

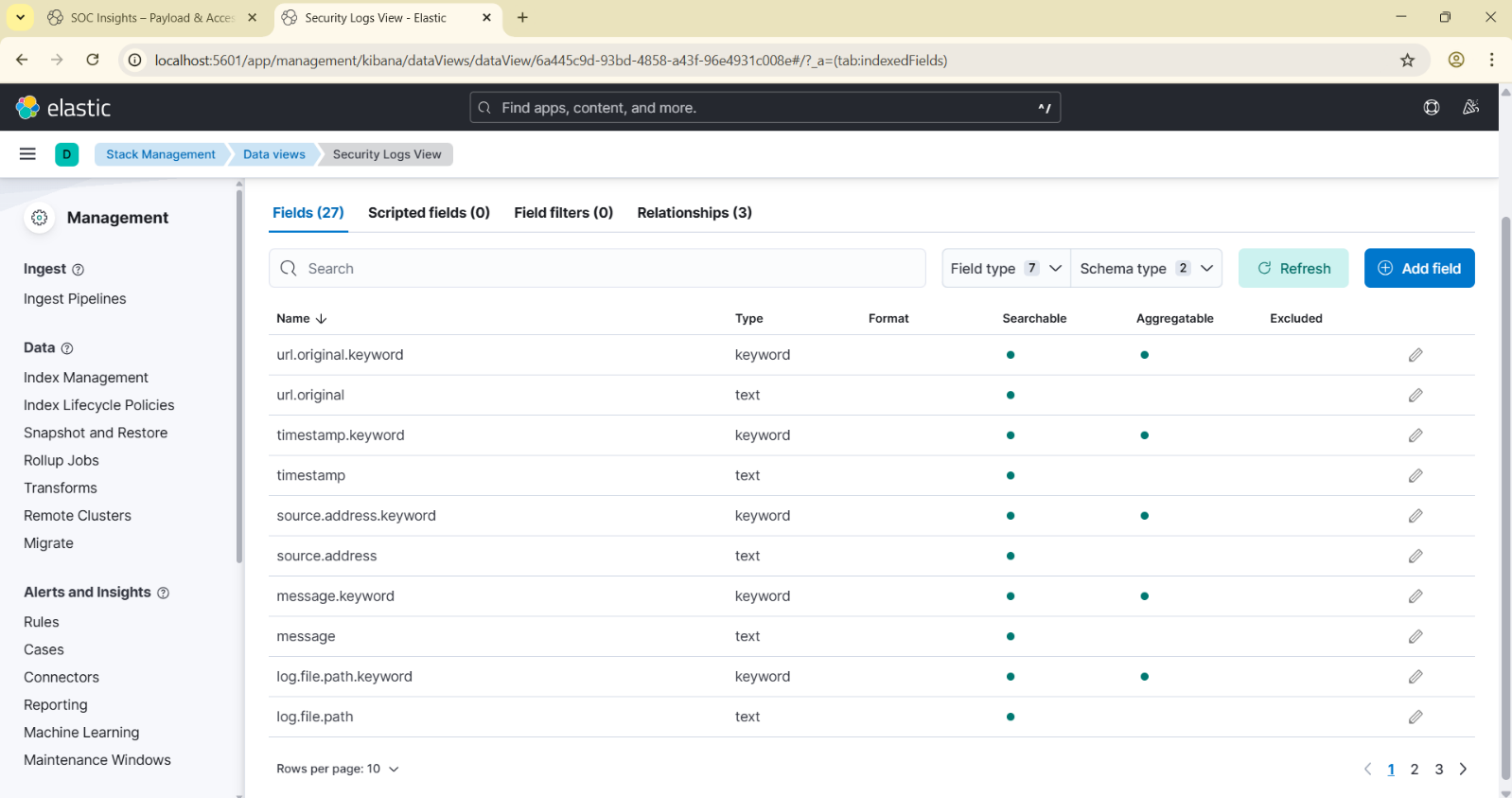
- Security log files were ingested into \*\*Elasticsearch\*\* via Filebeat.

- Using \*\*Kibana Discover\*\*, logs were verified to contain key fields:  
  
 - `url.original` for endpoint detection

- `source.address` for IP correlation

- `http.response.status\_code` for access behavior

- `@timestamp` for timeline correlation

  
  
Figure 2: Kibana Discover tab showing indexed security logs with structured fields used for triage

⿢ Log Filtering & Pattern Isolation

- Applied KQL filters to narrow down unauthorized access attempts:

```kql

url.original: "/admin" AND http.response.status\_code: 403

🔹Admin Probe Detection

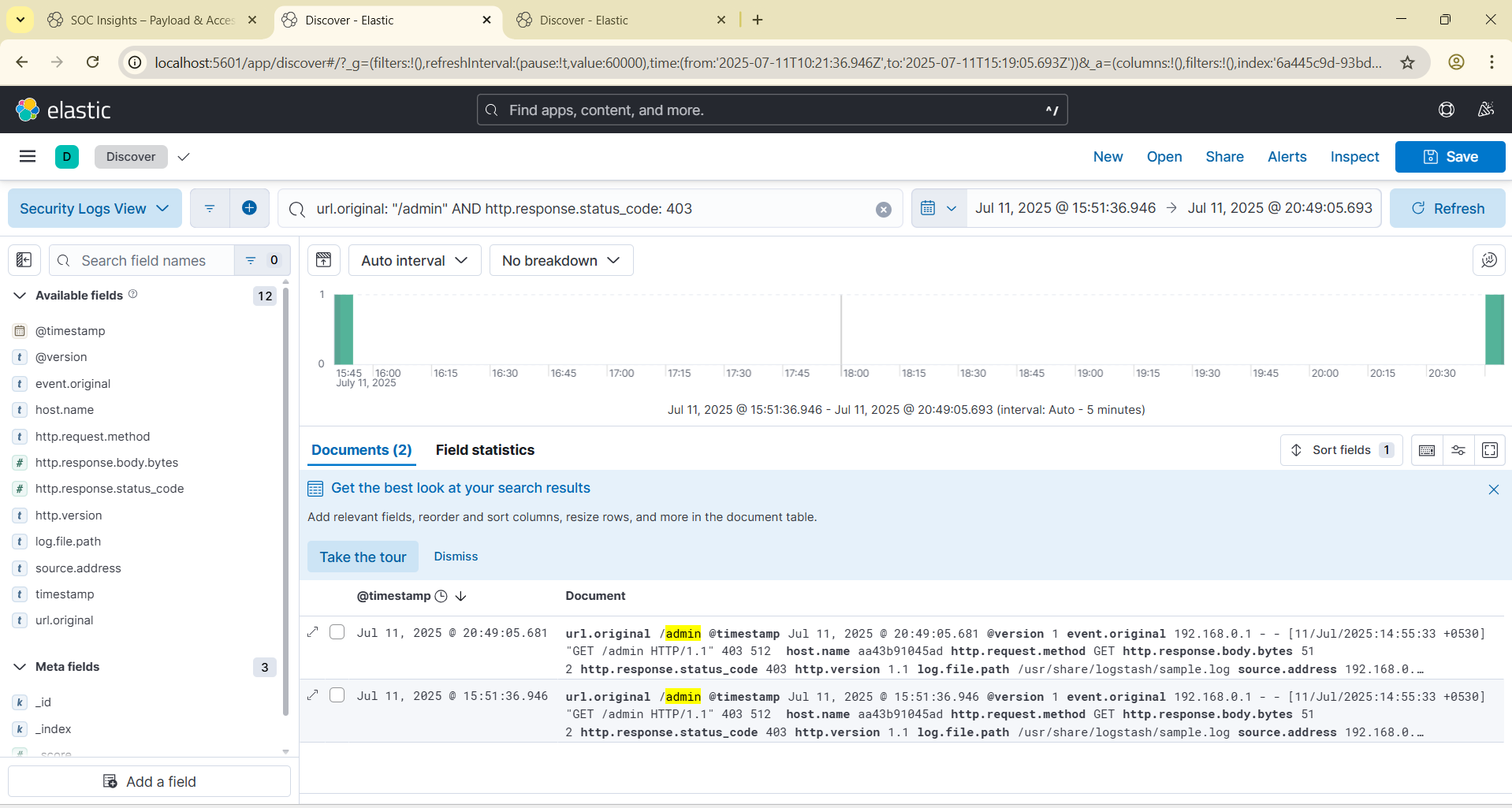
🛑 Unauthorized Endpoint Probing

During log analysis, a recurring pattern of failed access attempts was detected targeting the /admin endpoint. These probes consistently triggered HTTP 403 Forbidden responses, suggesting unauthorized actors attempting privilege escalation or recon activity.

🔍 Filter Applied in Kibana

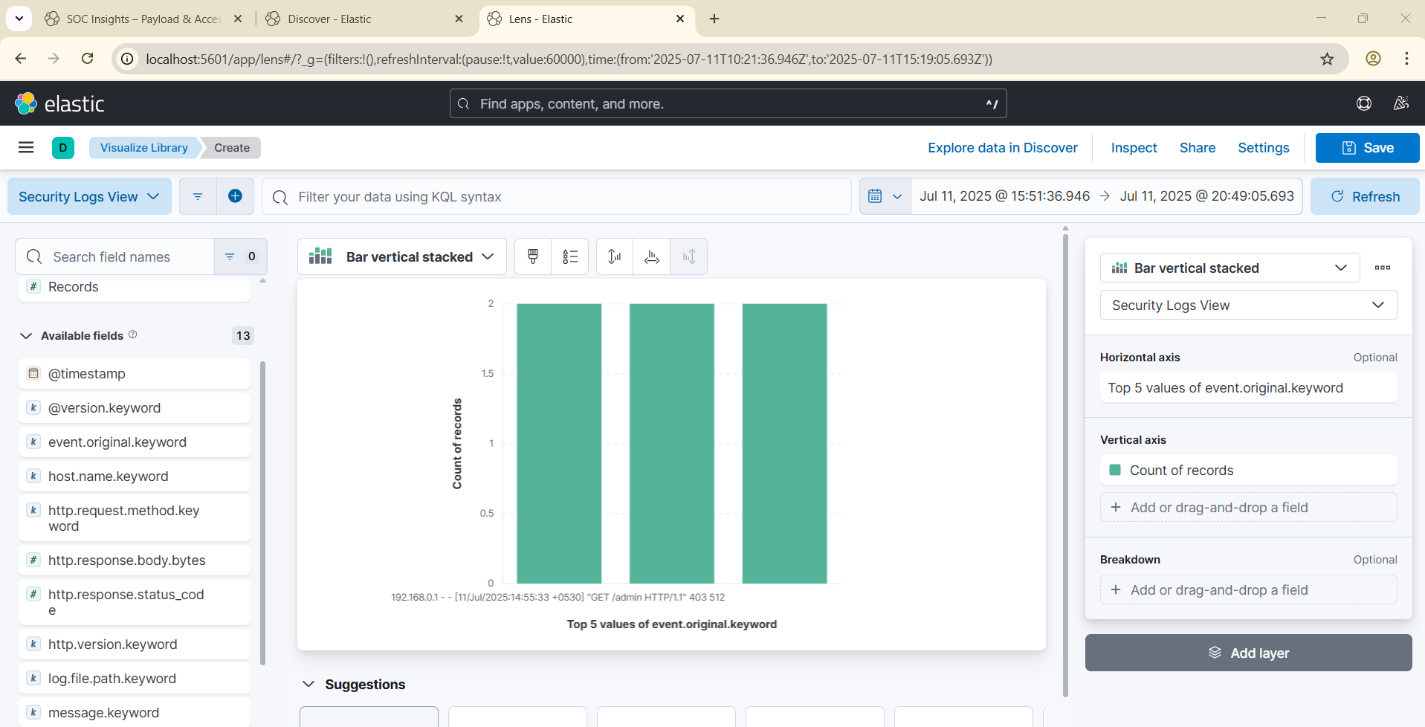
To isolate these events, the following KQL query was used in the Discover tab:

```kql

url.original: "/admin" AND http.response.status\_code: 403  
  
  
  
  
  
  
  
  
Figure 3: Filtered Discover view showing repeated 403 responses for /admin endpoint

📊 Visualization Summary

To better understand probe behavior, a **stacked bar chart** was constructed using Kibana Lens:

* Groups /admin hits by source.address (IP)  
    
  Figure 4: Admin Probe Visualization displaying IP-based frequency of forbidden access attempts
* Highlights high-frequency access attempts from specific IPs
* Time-series data layered to show spikes across analysis window

🔹 Payload Analysis

📦 Investigating Response Size Patterns

Beyond access denial, log entries showed varying response payload sizes for /admin requests. While all attempts returned HTTP 403, the \*payload size\* (http.response.body.bytes) sometimes fluctuated—potentially revealing system behavior nuances or data exposure risk.

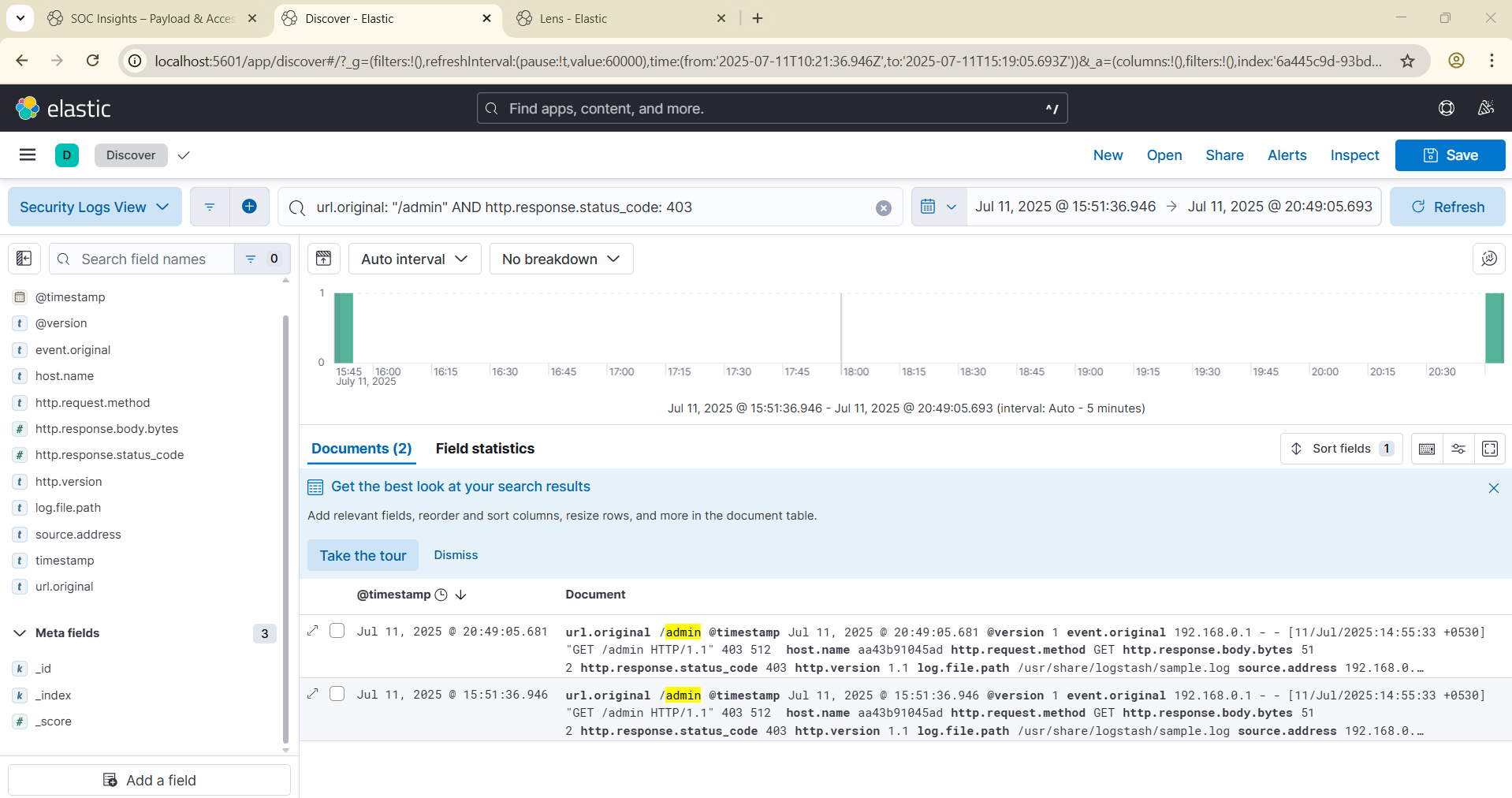
🔍 Filter Configuration

To focus on payloads returned from forbidden /admin hits, this KQL query was applied in Discover:

```kql

url.original: "/admin" AND http.response.status\_code: 403  
Additional field added:

* http.response.body.bytes – to examine the size of each response

   
  
Figure 6: Discover view with response payload sizes highlighted for /admin probes

🧪 Observations

* Majority of forbidden /admin responses returned payload size of **512 bytes**
* Size consistency implies standard error handling behavior
* However, even minor fluctuations could hint at:
* Differentiated error templates
* Conditional messaging based on probe metadata
* Backend process engagement triggered by crafted requests

📊 Visualization Summary

To explore payload distribution over time, a **median visualization** was built using Kibana Lens:

* **Y-axis:** Median of http.response.body.bytes
* **X-axis:** Time intervals (e.g., hourly buckets)
* Optional breakdown: by source.address for attacker profiling

🚨 Interpretation

Stable response sizes are ideal for minimizing reconnaissance value. Any observable variance, even small, could:

* Enable attackers to infer backend logic
* Suggest verbose server errors not properly sanitized
* Indicate internal behavior change during probe attempts

These were flagged in the report as mitigation candidates—suggesting message normalization and response hardening for sensitive paths like /admin.

🔹 Privilege Escalation Review

🚨 Behavioral Indicators of Escalation Attempts

During structured log analysis and payload triage, IP 192.168.0.1 emerged as a consistent actor probing restricted resources such as /admin. While responses returned HTTP 403 Forbidden, repeated access and payload stability hinted at calculated recon attempts.

🧠 Inferred Attack Objectives

From frequency and targeting patterns, this behavior may reflect early stages of:

- \*Privilege escalation testing\*

- \*Endpoint enumeration\* for administrative entry points

- Attempts to \*fingerprint access controls\* based on response timing and size

The attacker’s persistence—using multiple identical probes despite denial—suggests possible scripting or automated tooling.  
  
🛠 Mitigation Strategy

To reduce exposure:

- Harden access controls on /admin with auth tokens or geofencing

- Implement rate limiting and probe detection alerts

- Normalize response sizes across denial templates to reduce leakage

🧪 Closing Analysis

Although the escalation attempt was unsuccessful, this activity showcases how minor patterns in log data—such as repeated 403s and consistent payload sizes—can reveal underlying attacker motives. Preemptive action is recommended to reduce future escalation risks.  
  
🔹 SOC Dashboard Summary

🖥 Overview

The SOC dashboard created in Kibana visualized key security events extracted from HTTP logs. By tracking endpoint access patterns, response codes, and source IP behavior, the dashboard enabled real-time alert triage and forensic insight.

📊 Core Visual Panels  
1. Endpoint Access Trends

- Displays hit frequency for sensitive URLs like /admin, /login, etc.

- Time-series bar chart with peak analysis across selected intervals

2. Response Code Breakdown

- Pie chart showing ratio of HTTP status codes

- High count of 403 Forbidden responses flagged as potential threat vector

3. IP-Based Access Frequency

- Bar chart grouping source IPs by hit count

- Revealed IP 192.168.0.1 as a persistent actor targeting restricted paths

4. Payload Size Monitoring

- Median response payload sizes visualized over time   
- Stability confirmed, with few deviations in http.response.body.bytes

🔹 Analyst Reflection

💡 Personal Insights as a Security Analyst

This case study deepened my understanding of log triage, probe detection, and privilege escalation tactics within a SOC context. Each phase—filtering for unauthorized access, visualizing probe frequency, and analyzing response payloads—helped sharpen my investigative approach.

Using tools like \*Burp Suite, \*\*Kibana, and \*\*jwt.io\*, I was able to connect real-world behavior (403 responses to /admin) with underlying reconnaissance intent. Visual dashboards and Discover filters turned raw logs into structured intelligence.  
🛠 Skill Development

- Applied structured KQL filters to isolate high-risk access patterns

- Used Kibana Lens to visualize IP-based attack frequency

- Conducted response payload analysis to detect fingerprinting risks

- Documented findings in a professional report format with evidence-backed screenshots

These experiences refined my abilities in both technical analysis and polished report creation—essential for effective SOC operations.

🔹Conclusion

📘 Summary of Findings

This security case study traced a simulated unauthorized access scenario targeting the /admin endpoint. Using Kibana and Elasticsearch, structured HTTP logs were analyzed to uncover repeated probe attempts returning 403 Forbidden responses, all initiated by IP 192.168.0.1.

Visualization and payload inspection revealed behavioral patterns consistent with reconnaissance and privilege escalation testing.

Key findings:

- Repeated access denial patterns to high-value paths

- Consistent payload sizes signaling system-level response uniformity

- Visualization confirming probe frequency grouped by IP

🧠 Analyst Perspective

Through log triage, response inspection, and dashboard creation, this study demonstrated:

- The power of structured filtering (KQL) to surface attacker intent

- The importance of consistent error handling to reduce reconnaissance value

- The value of visual storytelling for SOC workflows

The process not only confirmed technical access control but also reinforced the need for forensic readiness.

🔐 Mitigation Actions

Recommended steps include:

- Response size normalization to prevent fingerprinting

- Access control reinforcement for sensitive endpoints

- SOC alerting rules to detect repeated probe patterns

- Visual dashboards to support ongoing monitoring

By identifying escalation attempts before they succeed, defenders gain the edge through visibility and clarity.