CONTINUOUS ASSESSMENT TASK 2

Integration of ELK stack with Endlessh Honeypot for Enhanced Security Monitoring:

1. ElasticSearch, Logstash and Kibana Stack (SIEM Integration)
2. Endlessh Honeypot (SSH Honeypot Deployment)

Module Leader: Kingsley Ibomo

Submitted by: Nikant Sharma. Msc in Cyber Security - 20005553

DECLARATION

I hereby certify that the work I completed for the Networking and Communications module (B9CY103) at Dublin Business School, titled "Integration of ELK Stack with Endlessh Honeypot for Enhanced Security Monitoring," is original to me. This work, which is being overseen by Mr. Kingsley Ibomo, is turned in for Assignment 2.

I declare that neither this institution nor any other has received submissions of any kind, in whole or in part, for consideration for any other degree or qualification. Except in cases when proper credit is given to the work of others, all of the work I have presented is original to me.

Nikant Sharma (Nick)

30th July 2024

ACKNOWLEDGEMENT

I would like to extend my sincere gratitude to Dublin Business School for providing a supportive and stimulating academic environment that has greatly facilitated my learning and research. I am particularly grateful to Mr. Kingsley Ibomo, whose teaching methodology, which emphasizes practical, hands-on experience and a fundamental understanding of networking concepts, has been a significant source of inspiration. His guidance and dedication have been invaluable in deepening my knowledge and practical skills in the field of cybersecurity.

This project was conducted independently, and no external contributions were made. The work presented here reflects my own efforts and learning, underpinned by the excellent educational resources and mentorship provided by the faculty.

Nikant Sharma (Nick)

30th July 2024

ABSTRACT

Using an Ubuntu 22.04 LTS Jammy virtual machine coupled with the ELK stack and the Endlessh honeypot, we particularly analyze and deploy honeypots in a cybersecurity environment in this assignment. Decoy systems called "honeypots," which are intended to draw in and examine hostile activity, are useful instruments for comprehending and reducing cybersecurity risks. There are several varieties of these, such as low-interaction honeypots that mimic restricted services and high-interaction honeypots that give attackers more extensive interaction capabilities.

In order to conduct thorough log aggregation and analysis, the Endlessh honeypot has to be configured on an Ubuntu machine and integrated with the ELK stack. A Kali Linux system was employed as the assaulting machine in this investigation, and it tried to connect to the honeypot over an unauthorized SSH connection. Real-time monitoring and in-depth analysis of attack patterns were made possible by the ELK stack, which also revealed potential attackers' strategies.

This exercise emphasizes how important honeypots are to cybersecurity because they can be used as a research tool to better understand attacker behavior as well as a detection technique. My Personal take caters to the fact that a honeypot like Endlessh can help in wasting time of a threat actor and inturn allow more time for the SOC and Blue teams to gather IOC from logs and escalate the reports for necessary actions and decision making. Organizations can improve their defensive strategies and proactively detect weaknesses by deploying honeypots. The results highlight how crucial it is to incorporate honeypots with SIEM systems such as ELK in order to facilitate effective data analysis and incident response, which in turn strengthens a company's cybersecurity posture.

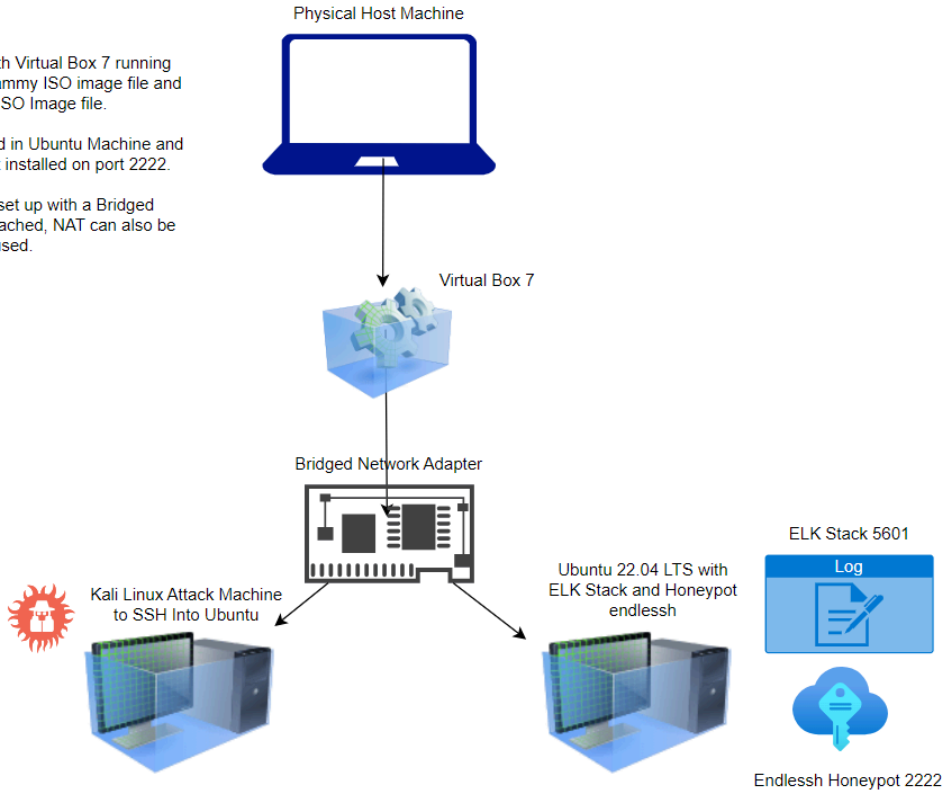
TABLE OF CONTENT

LAB ENVIRONMENT DIAGRAM

1. Host machine with Virtual Box 7 running Ubuntu 22.04 LTS Jammy ISO image file and Kali Linux ISO Image file.

2. ELK Stack installed in Ubuntu Machine and Endlessh Honeypot installed on port 2222.

3. Both VMs are set up with a Bridged Network Adapter attached, NAT can also be used.



INTRODUCTION

In the ever-changing and intricate realm of cybersecurity, the utilization of Security Information and Event Management (SIEM) systems and honeypots is vital for the identification and comprehension of cyber threats. Honeypots are specialized systems made to mimic assets that are weak points in order to lure attackers in. They are used as a lure as well as a research tool, offering thorough insights into the tricks, strategies, and processes (TTPs) that malevolent or threat actors use. There are numerous different kinds of honeypots: low-interaction honeypots, which mimic restricted services; and high-interaction honeypots, which provide a more intricate environment in which attackers can engage in extended engagement. Furthermore, there are specialized honeypots, such as the Endlessh honeypot employed in this investigation, which focuses only on SSH-based attacks.

The integration of a honeypot with a powerful SIEM tool utilizing an Ubuntu 22.04 LTS virtual machine is the main goal of this project. Endlessh was selected as the honeypot for this configuration because of its reputation for being able to imitate an SSH service and record in-depth logs of unwanted access attempts. The ELK stack, an open-source SIEM system that combines Elasticsearch, Logstash, and Kibana to offer robust data aggregation, real-time monitoring, and analytical capabilities, is the system that is used.

In this case, the ELK stack is crucial because it makes it possible to gather, examine, and display the massive amounts of log data that the honeypot generates. This integration makes it easier to comprehend the many forms and frequency of attacks, the strategies employed, and the potential weak points that attackers may try to exploit. For cybersecurity experts looking to improve the security posture of their company, these insights are priceless.

This assignment's practical component entails setting up a controlled lab environment in which an attacker Kali Linux virtual machine interacts with an Ubuntu virtual machine that is setup with the ELK stack and Endlessh honeypot. This setup makes it possible to simulate actual hacking situations and offers a secure setting in which to research and examine attack tactics.

Through this activity, the assignment highlights the need for practical experience in recognising and mitigating cyber threats for education, as well as how cybersecurity tools may be used in practice. Through investigating various varieties of honeypots and SIEM systems, students acquire a more profound understanding of the complexities involved in network security and the significance of preemptive defence strategies in preserving digital assets. By taking a comprehensive approach, instructors can give students the knowledge and abilities they need to successfully traverse the complicated world of cybersecurity, better preparing them for the obstacles they will encounter in the workplace.

STATE OF THE ART

Honeypots - Honeypots are specialized security devices used in controlled environments to draw attention to, identify, and examine hostile activity or unauthorized access. They act as ruses to divert attackers from vital systems and acquire information about potential security risks. The complexity, degree of interaction, and particular use cases of each sort of honeypot determine its classification. After extensive research, This report summarizes some of the Types followed by their purpose, functionality and benefits.

Types of Honeypots	Purpose	Functionality	Benefits
Production In 2015, a financial institution used production honeypots to identify and analyse a sophisticated phishing campaign targeting its employees, leading to improved phishing detection and response mechanisms (Smith et al., 2016).	These are set up alongside real servers and services in a business's production network. Serving as an early warning system and reducing the possibility of future assaults is their main goal.	By simulating actual systems, they tempt attackers to engage with them rather than actual resources. Security teams can identify intrusions and take action before real systems are affected by keeping an eye on these exchanges.	They can deflect attacks from important assets and aid in recognising and comprehending the tactics, methods, and procedures (TTPs) employed by attackers.
Research The Honeynet Project, an open-source security initiative, utilizes research honeypots to study global cyber threats, providing insights into the evolution of malware and attack techniques (Spitzner, 2003).	The primary uses of these honeypots are investigation and analysis. Their purpose is to obtain information about attack techniques, instruments, and patterns.	To draw in skilled attackers, they are usually more intricate and interactive, mimicking a wide range of systems and services.	Sharing insights with the larger cybersecurity community, enhancing detection and prevention methods, and creating better security solutions are all made possible by the information gathered.
Machinery (SCADA) In 2017, researchers deployed SCADA honeypots to understand attack strategies on power grids, leading to enhanced protective measures (Jones & Clark, 2017).	The purpose of these devices, sometimes referred to as Industrial Control Systems (ICS) honeypots, is to mimic industrial settings and machinery, including SCADA systems.	They imitate the actions of vital industrial systems in order to draw in attackers who may be looking to compromise industrial processes or obtain unauthorized access to private information.	They aid in finding weaknesses in the administration and functioning of industrial machinery, strengthening industrial system defenses, and comprehending threats to key infrastructure.

Networking A university implemented networking honeypots to study the prevalence of DDoS attacks on its network, resulting in improved network traffic filtering and mitigation strategies (Liu et al., 2018).	The goal of these honeypots is to draw in hostile activity that targets network infrastructure with an emphasis on network-level attacks.	By simulating networking equipment such as switches, routers, and whole networks, they can shed light on attack vectors unique to networks, including man-in-the-middle attacks and distributed denial-of-service assaults.	By assisting network managers in comprehending the risks to their networks, they improve the security of network architecture and protocols.
Email "Detection and Analysis of Spam and Phishing Using Email Honeypots," Journal of Information Security Research, 2015.	Email honeypots are essential for comprehending and reducing email-based risks because they are made to draw in and examine spam and phishing attempts.	These honeypots make use of fictitious identities and email addresses that are purposefully made public to spammers and phishers. They gather information about the types of email attacks, such as those that involve malicious attachments, links, and social engineering techniques.	By identifying phishing campaigns, enhancing email filtering and security measures, and creating more effective user education programmes to help people identify and steer clear of email risks, new insights are gathered.
Database "Using Database Honeypots for SQL Injection Analysis in Financial Systems," International Journal of Cybersecurity, 2018.	The purpose of these honeypots is to draw in assaults directed towards database systems. They are engineered to identify and examine SQL injection, unapproved access attempts, and data exfiltration.	They create the impression that an attacker has gained access to important data by simulating a database environment with fictitious or irrelevant data. The honeypot records and keeps an eye on everything.	By using this data, organizations may improve database security, comprehend popular attack pathways, and hone their access control and monitoring protocols.
Malware (Honeynets) "Analyzing Malware Propagation with Honeynets: A Case Study," Proceedings of the ACM Conference on Computer and Communications Security, 2017.	The purpose of these honeypots is to gather and examine harmful software and the methods employed in its distribution. They are specifically made to draw in and examine malware.	They frequently consist of networks of linked honeypots, or honeynets, which mimic actual settings where malware could proliferate. They seize malware samples and observe how they behave in a regulated setting.	Understanding the lifespan and spread of malware, creating antivirus and anti-malware programmes, and improving threat intelligence databases all depend on the information gathered from malware honeypots.

Security Information and Event Management Tools - Modern cybersecurity infrastructure cannot function without Security Information and Event Management (SIEM) technologies, which offer extensive monitoring, analysis, and response capabilities for security incidents. Logs and events from several sources are combined and analyzed by SIEM systems in order to identify suspicious activity and possible security breaches. For the purpose of this assignment, We will be drawing comparisons between Splunk, IBM QRadar and ELK Stack.

Features	**Splunk**	**IBM QRadar**	**ELK Stack**
Purpose	Real-time insights from machine data are the main focus of data monitoring and analysis.	In big businesses, threat detection, incident response, and compliance management are all important	Open-source data analytics, visualisation, and log management for a range of use cases.
Data Handling	Provides a single interface for all log data by indexing and searching data from several sources.	Connects events from many sources, spotting irregularities and possible security issues.	Utilises Kibana for visualisation, Elasticsearch for analysis, and Logstash for log collection.
Security Analytics	Enhanced analytics capabilities, such as machine learning and predictive analytics.	Correlates and analyses security events using artificial intelligence to help with precise threat identification.	Offers analytics and dashboards that can be customised, making anomaly detection and in-depth security insights possible.
Compliance Reporting	Offers customisable and pre-built compliance reports that are compatible with a number of regulatory frameworks.	Robust features for compliance management, such as audit trail documentation and automatic reporting.	It can be set up for compliance-related data analysis even while compliance is not its main focus.
Cost	Usually more expensive, particularly for large-scale installations, but has a wealth of features.	Because of its extensive functionality and enterprise-grade solutions, it is frequently more expensive.	Free and open-source, with the main expenses coming from possible third-party services and hardware.
Case Studies	Used by companies like Nasdaq for monitoring and analysing financial transactions (Johnson & Wu, 2019).	Deployed by financial institutions to enhance threat detection and compliance (Garcia et al., 2017).	Utilised by various organisations for cost-effective log management and data analytics (Smith & Brown, 2020).

PROBLEM STATEMENT

Organizations are facing more dangers from skilled attackers in the quickly developing sector of cybersecurity. There is an urgent need for sophisticated detection and analysis technologies to reduce these dangers. In this environment, security information and event management (SIEM) systems and honeypots have become essential elements. Honeypots, like the Endlessh honeypot, are made to entice and examine attempts at unauthorized access, giving important information about the methods and behavior of attackers.

Although they work well, combining honeypots with SIEM technologies such as the ELK Stack presents a number of difficulties. Elasticsearch, Logstash, and Kibana, together known as the ELK Stack, provide a robust open-source log management and analysis solution. But in order to execute it effectively, real-time analysis capabilities, scalability, and data management must all be carefully considered.

The overwhelming amount of data produced by honeypots is one of the main problems, as it can cause storage and processing concerns for the SIEM system. Furthermore, organizational resources may be strained due to the complexity of setting up and maintaining such an integrated system, especially in smaller businesses without specialized cybersecurity knowledge.

Furthermore, even if the ELK Stack offers a great deal of customisation, deployment and management are made more difficult by the need for extra levels of security due to the lack of built-in security capabilities. Another difficulty is separating real threats from false positives, which can need a lot of resources and call for advanced analytics and machine learning methods.

In light of these difficulties, the purpose of this study is to investigate the efficient use of honeypots in conjunction with the ELK Stack, with an emphasis on boosting system security, optimizing data handling, and increasing threat detection accuracy. Creating a thorough framework that can be implemented in many organizational contexts and strengthen overall security posture against a dynamic threat landscape is the aim. Furthermore, in order to offer useful insights into the application and administration of these technologies, this research will assess case studies.

IMPLEMENTATION, METHODOLOGY & DATA ANALYSIS

ELK Stack Installation Phase

Figure 1.1 - The commands apt-get update && apt upgrade -y was run to upgrade the system repositories as best practice.

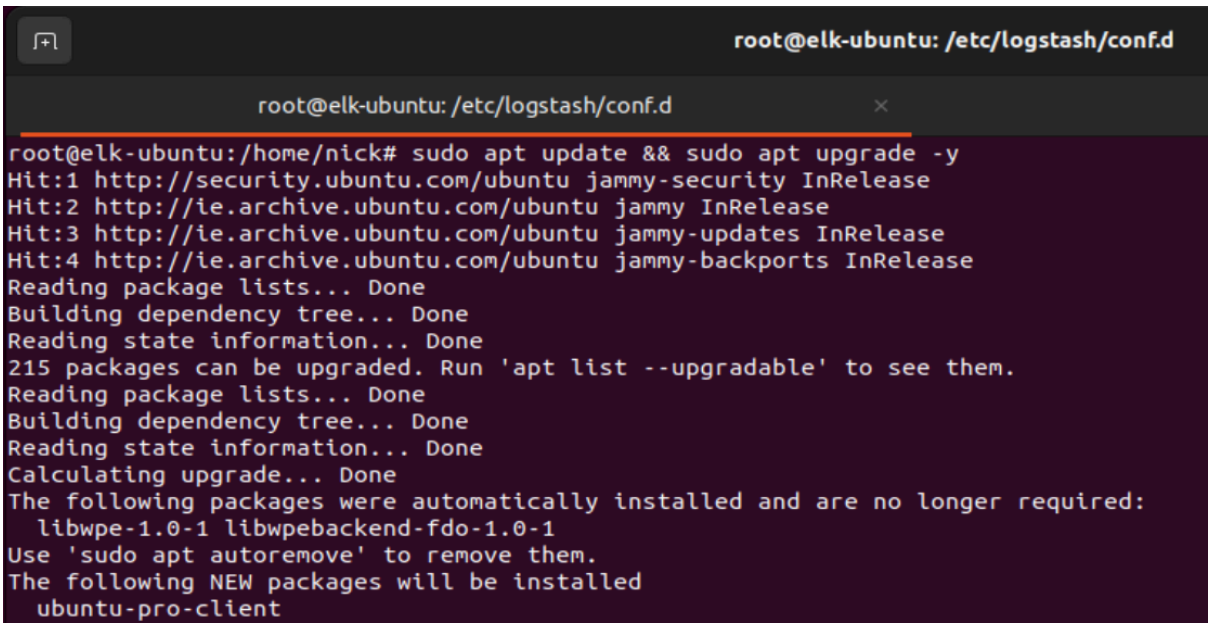


Figure 1.2 - wget followed by artifact link was used to fetch the files with a | - pipe command followed by the gpg key for installation of elasticsearch. Finally, apt-get install elasticsearch command was run to install the application

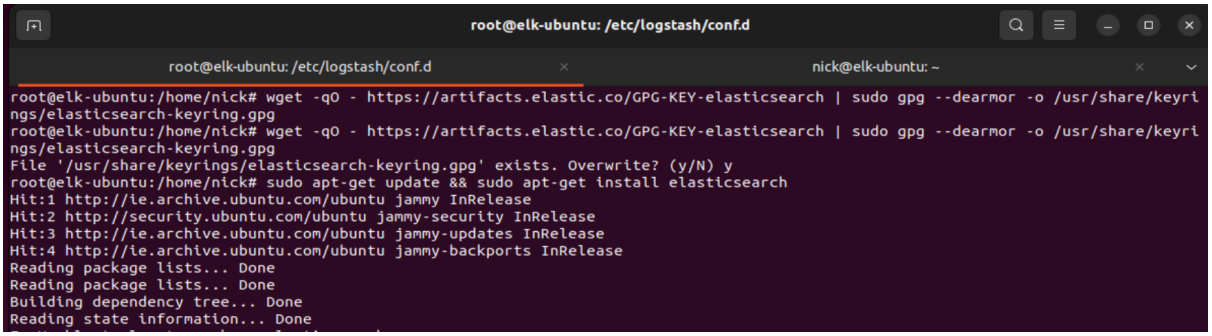


Figure 1.3 - Elasticsearch status is running. Note - daemon-reload, enable and status are some systemctl commands that need to be run first in order to see the running status.

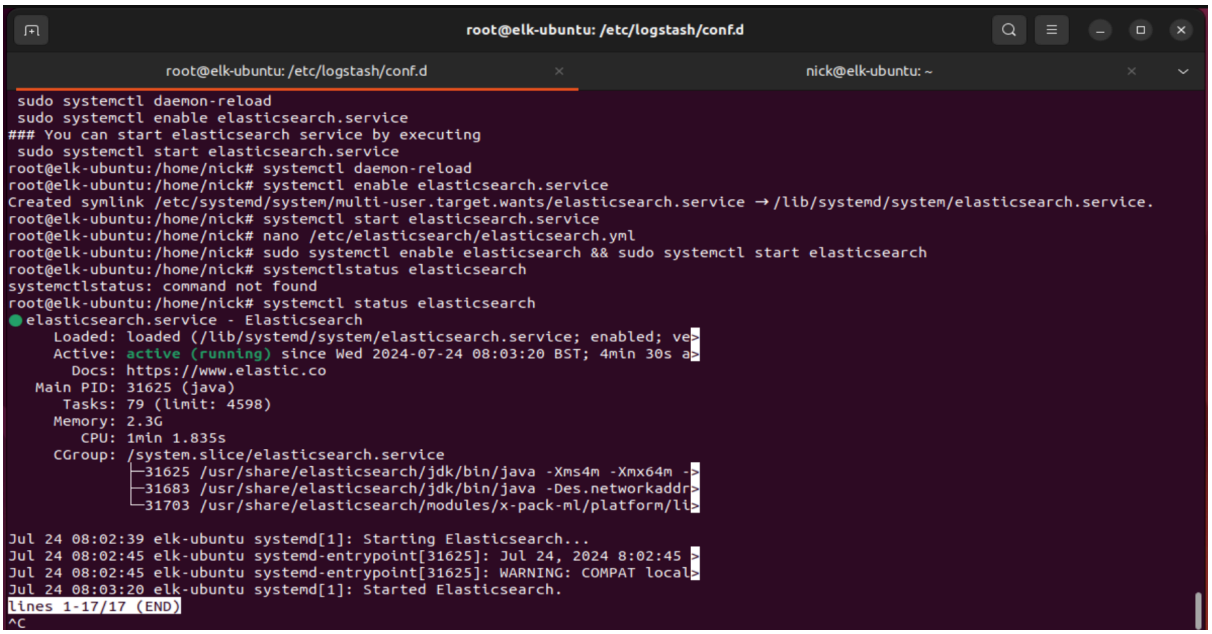


Figure 1.4 - Curl based validation of Elasticsearch running in browser mode

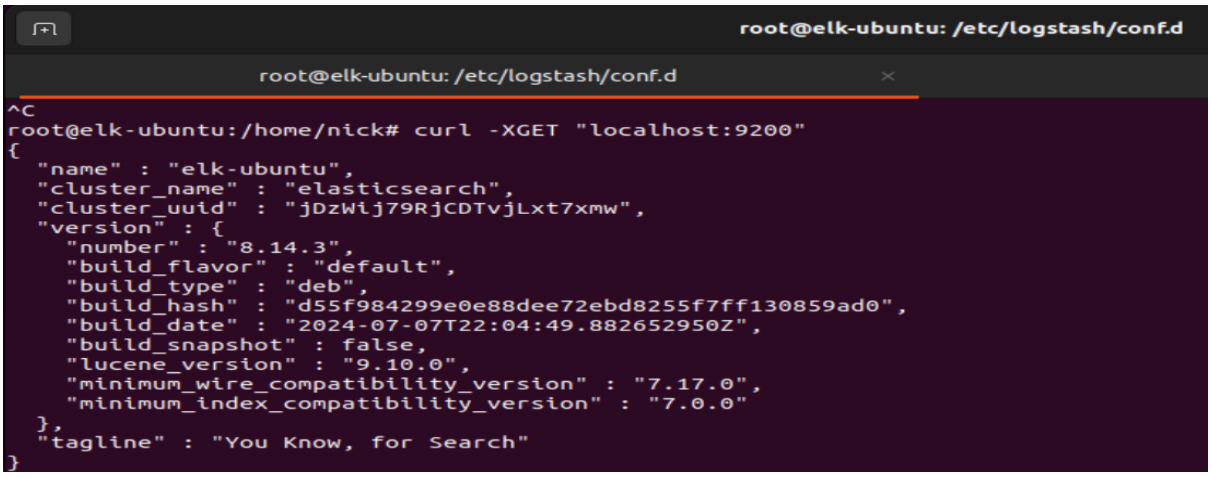


Figure 1.5 - Logstash Installation with - apt install logstash

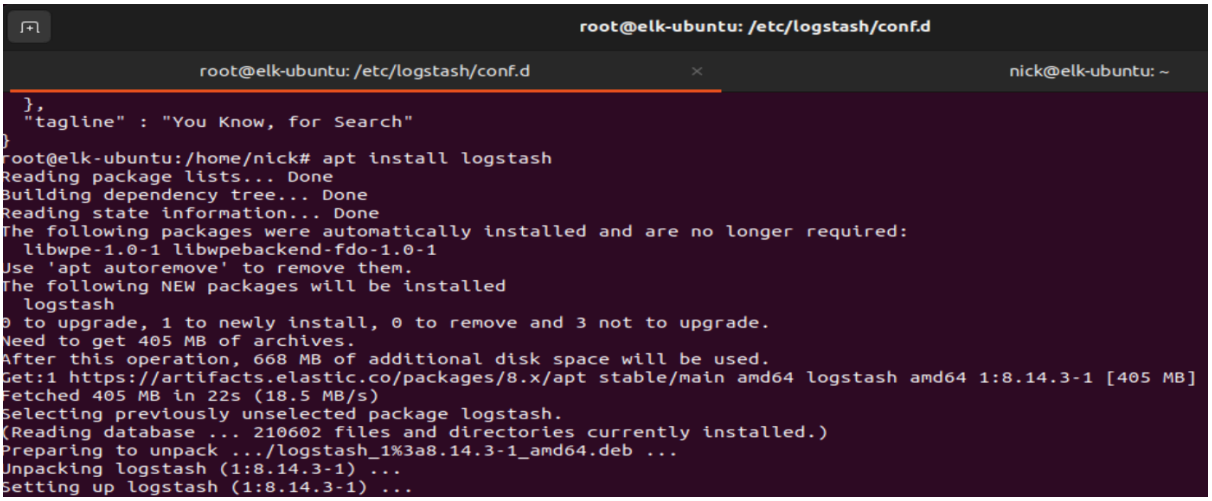


Figure 1.6 - Logstash filter scripts for beats ingestion in /etc/logstash/conf.d file

Figure 1.7 - Logstash status running

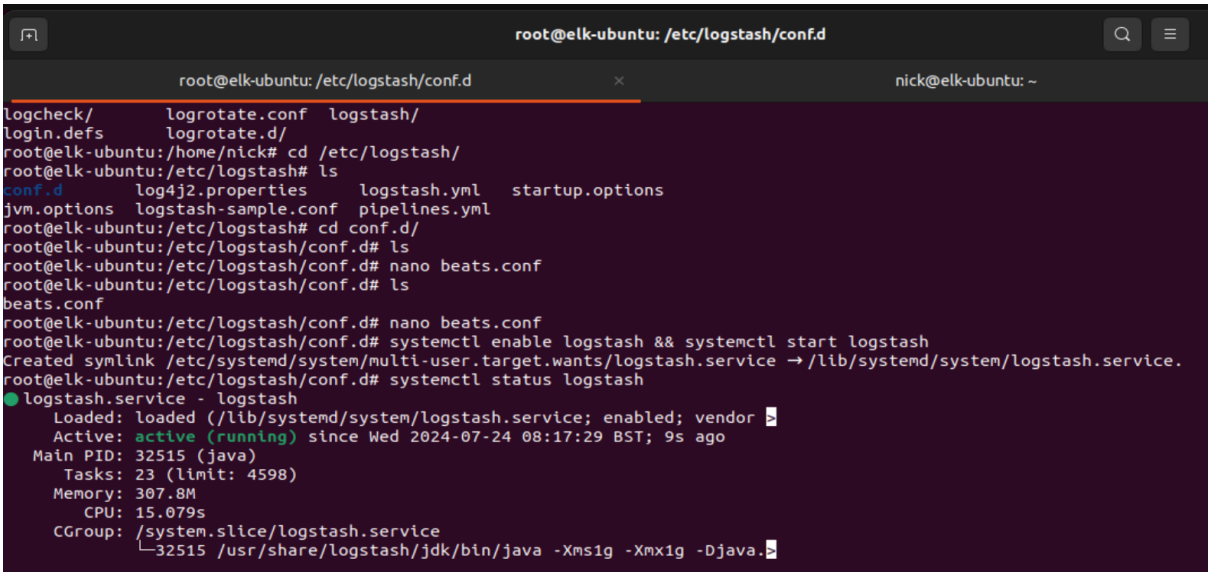


Figure 1.8 - Kibana installation and running status

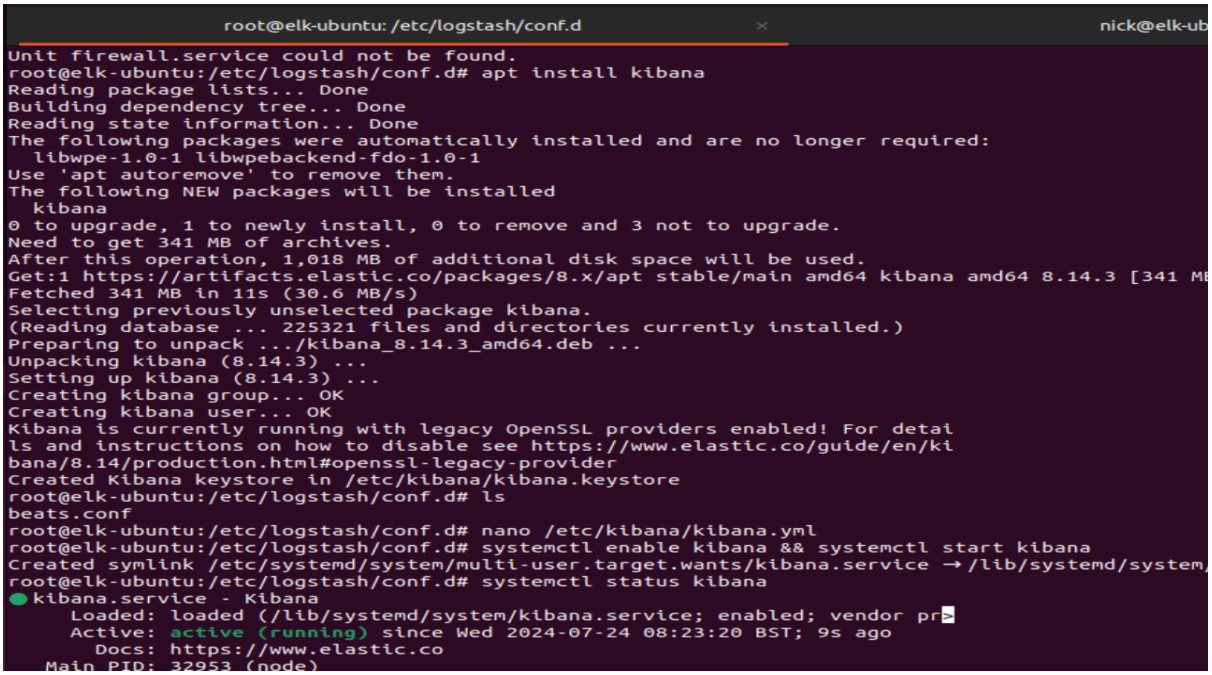


Figure 1.9 - Kibana.yml file host and port configuration

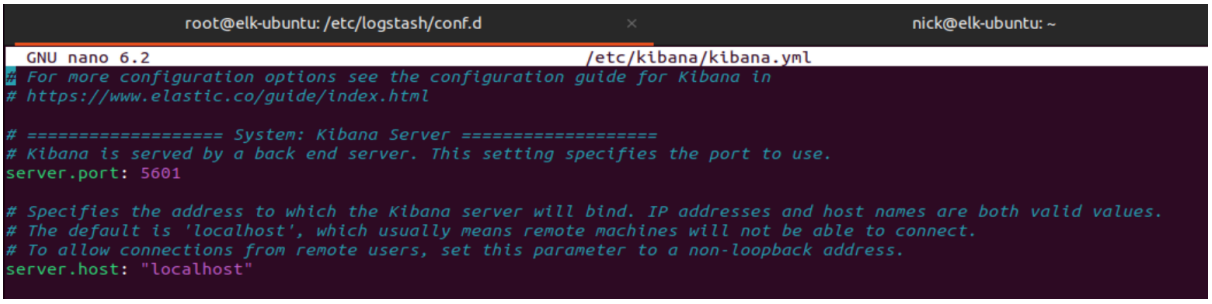


Figure 1.10 and 1.11 - Elasticsearch.yml file configuration

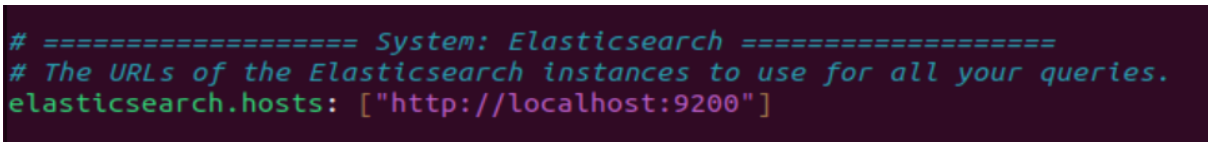


Figure 1.11 - xpack security all set to false for demo purpose only. **Not advisable for production environments.**

```
  GNU nano 6.2                                  /etc/elasticsearch/elasticsearch.yml *
#----------------------- BEGIN SECURITY AUTO CONFIGURATION -----------------------
#
# The following settings, TLS certificates, and keys have been automatically
# generated to configure Elasticsearch security features on 24-07-2024 07:01:09
#
# --------------------------------------------------------------------------------

# Enable security features
xpack.security.enabled: false

xpack.security.enrollment.enabled: false

# Enable encryption for HTTP API client connections, such as Kibana, Logstash, and Agents
xpack.security.http.ssl:
  enabled: false
  keystore.path: certs/http.p12

# Enable encryption and mutual authentication between cluster nodes
xpack.security.transport.ssl:
  enabled: false
  verification_mode: certificate
  keystore.path: certs/transport.p12
  truststore.path: certs/transport.p12
# Create a new cluster with the current node only
# Additional nodes can still join the cluster later
cluster.initial_master_nodes: ["elk-ubuntu"]

# Allow HTTP API connections from anywhere
# Connections are encrypted and require user authentication
http.host: 0.0.0.0
```

Figure 1.12 - Filebeat MQ installation for log ingestion into ELK Stack. Simple curl command followed by the artifact link. Depackage -i command for depackaging and installation. Changes to the filebeat.yml file using nano

```
root@elk-ubuntu: /etc/filebeat                    nick@elk-ubuntu: /var/log
 dpkg-deb --control subprocess returned error exit status 2
Errors were encountered while processing:
 filebeat-8.9.2-amd64.deb
root@elk-ubuntu:/home/nick# curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-8.9.2-amd64.deb sudo dpkg -i
 filebeat-8.9.2-amd64.deb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 46.8M  100 46.8M    0     0  7044k      0  0:00:06  0:00:06 --:--:-- 8871k
curl: (6) Could not resolve host: sudo
curl: (6) Could not resolve host: dpkg
curl: (6) Could not resolve host: filebeat-8.9.2-amd64.deb
root@elk-ubuntu:/home/nick# ls
Desktop    Downloads                 filebeat-8.9.2-amd64.deb.1  Pictures  snap       Videos
Documents  filebeat-8.9.2-amd64.deb  Music                       Public    Templates
root@elk-ubuntu:/home/nick# dpkg -i filebeat-8.9.2-amd64.deb.1
Selecting previously unselected package filebeat.
(Reading database ... 320381 files and directories currently installed.)
Preparing to unpack filebeat-8.9.2-amd64.deb.1 ...
Unpacking filebeat (8.9.2) ...
Setting up filebeat (8.9.2) ...
root@elk-ubuntu:/home/nick# ls
Desktop    Downloads                 filebeat-8.9.2-amd64.deb.1  Pictures  snap       Videos
Documents  filebeat-8.9.2-amd64.deb  Music                       Public    Templates
root@elk-ubuntu:/home/nick# cd /etc/filebeat/
root@elk-ubuntu:/etc/filebeat# ls
fields.yml  filebeat.reference.yml  filebeat.yml  modules.d
root@elk-ubuntu:/etc/filebeat# nano filebeat.yml
root@elk-ubuntu:/etc/filebeat# systemctl enable filebeat
Synchronizing state of filebeat.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable filebeat
Created symlink /etc/systemd/system/multi-user.target.wants/filebeat.service → /lib/systemd/system/filebeat.service.
root@elk-ubuntu:/etc/filebeat# systemctl start filebeat
root@elk-ubuntu:/etc/filebeat# systemctl status filebeat
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
     Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2024-07-24 08:57:18 BST; 7s ago
       Docs: https://www.elastic.co/beats/filebeat
```

Figure 1.13 - filebeat.yml configurations for log ingestion via mentioned paths.

```
# filestream is an input for collecting log messages from files.
- type: filestream

  # Unique ID among all inputs, an ID is required.
  id: my-filestream-id

  # Change to true to enable this input configuration.
  enabled: true

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/*.log
    - /var/log/apache2/*.log
    #- c:\programdata\elasticsearch\logs\*
```

Figure 1.14 & 15 - Using Logstash and not Elasticsearch in Filebeat.yml. Commenting out the latter and Uncommenting the Former for seamless ingestion and avoid conflicts in log management

Figure 1.15

```
# ================================== Outputs ===================================

# Configure what output to use when sending the data collected by the beat.

# ---------------------------- Elasticsearch Output ----------------------------
#output.elasticsearch:
  # Array of hosts to connect to.
#  hosts: ["localhost:9200"]

  # 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: ["10.0.2.15:5044"]

  # Optional SSL. By default is off.
  # List of root certificates for HTTPS server verifications
  #ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]
```

Figure 1.16 & 17 - editing /etc/filebeat/modules.d/apache.yml and system.yml for enabling log ingestion by setting enabled: true and providing log paths at var.paths: both access and error log.

```
root@elk-ubuntu: /etc                                                    nick
  GNU nano 6.2                          /etc/filebeat/modules.d/apache.yml
# Module: apache
# Docs: https://www.elastic.co/guide/en/beats/filebeat/8.9/filebeat-module-apache.html

- module: apache
  # Access logs
  access:
    enabled: true

    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths: ["/var/log/apache2/access.log*"]

  # Error logs
  error:
    enabled: true

    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths: ["/var/log/apache2/error.log*"]
```

Figure 1.17

```
root@elk-ubuntu: /etc                          nick@elk-ubuntu: /var/log
  GNU nano 6.2                          /etc/filebeat/modules.d/system.yml
# Module: system
# Docs: https://www.elastic.co/guide/en/beats/filebeat/8.9/filebeat-module-system.html

- module: system
  # Syslog
  syslog:
    enabled: true

    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths:

  # Authorization logs
  auth:
    enabled: true

    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths:
```

Figure 1.18 - enabling system and apache modules for filebeat log ingestion from CLI.

```
root@elk-ubuntu:/etc/filebeat# nano filebeat.yml
root@elk-ubuntu:/etc/filebeat# filebeat modules enable system
Enabled system
root@elk-ubuntu:/etc/filebeat# filebeat modules enable apache
Enabled apache
```

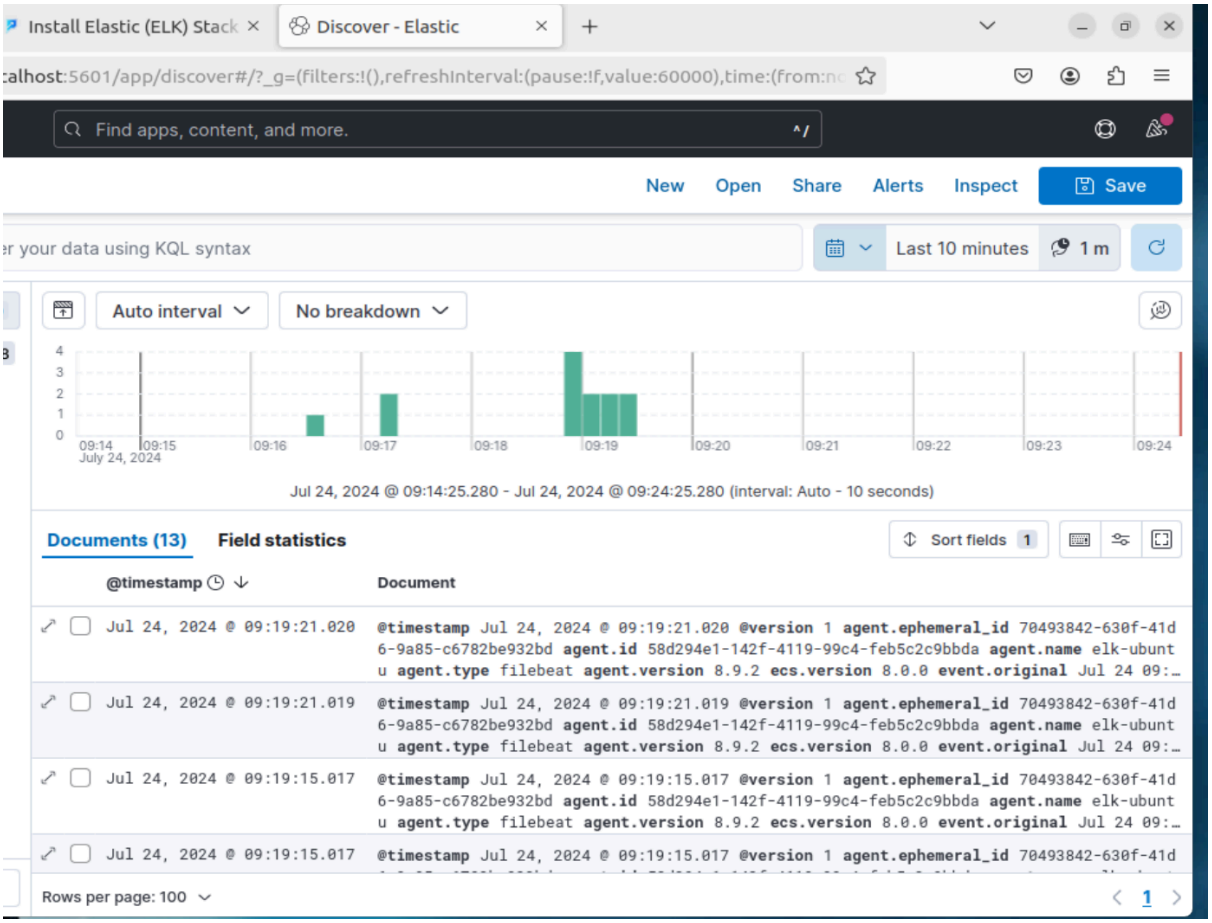
Figure 1.19 - Creating kibana dashboards from pre existing templates via CLI

```
root@elk-ubuntu:/etc# sudo filebeat setup --index-management -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["loc
alhost:9200"]'
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite: true` for enabling.

Index setup finished.
root@elk-ubuntu:/etc# sudo filebeat setup -E output.logstash.enabled=false -E output.elasticsearch.hosts=['localhost:9200'] -E set
up.kibana.host=localhost:5601
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite: true` for enabling.

Index setup finished.
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
Loaded Ingest pipelines
```

Figure 1.20 - Visualization dashboard created and can be seen after clicking the “Discover” option from the hamburger menu. localhost:5601 - kibana interface



Endlessh Installation Phase

Figure 2.1 - copy of the Endlessh honeypot was created by cloning and compiling its GitHub repository. After that, it was installed and set up based on the ReadMe file instructions.

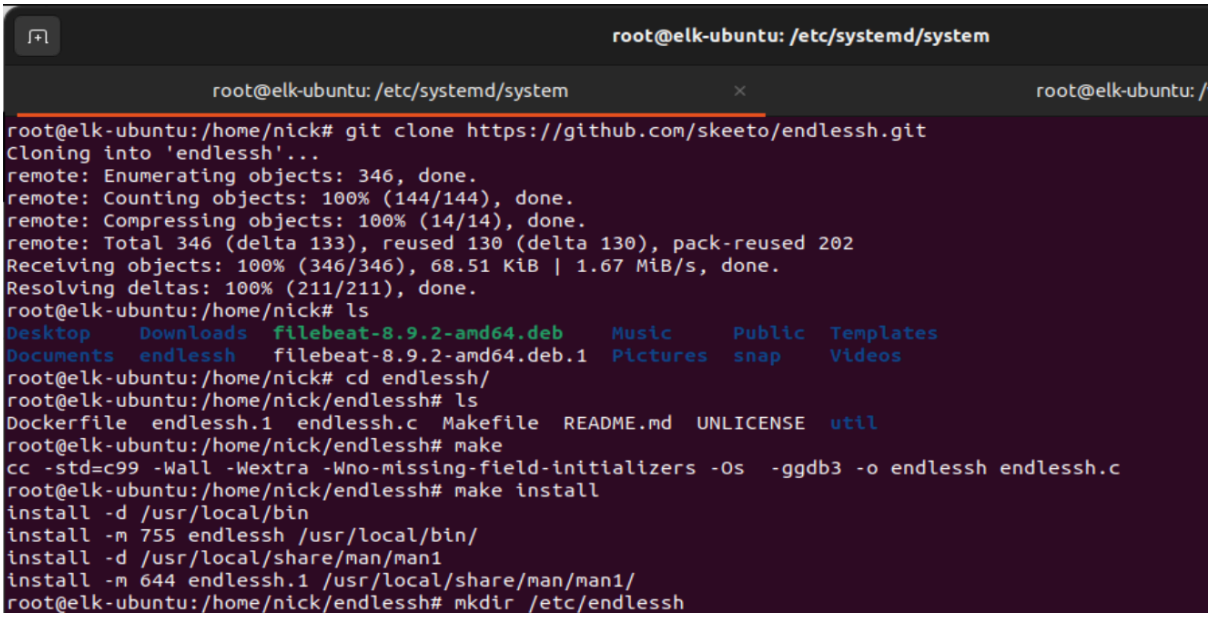


Figure 2.2 and 2.3 - adding “port 2222” to the /etc/endlessh/config file and copying the endless.service file in the etc/systemd/system/ directory to function as a systemd file thus replacing the actual ssh with the honeypot endless-ssh.

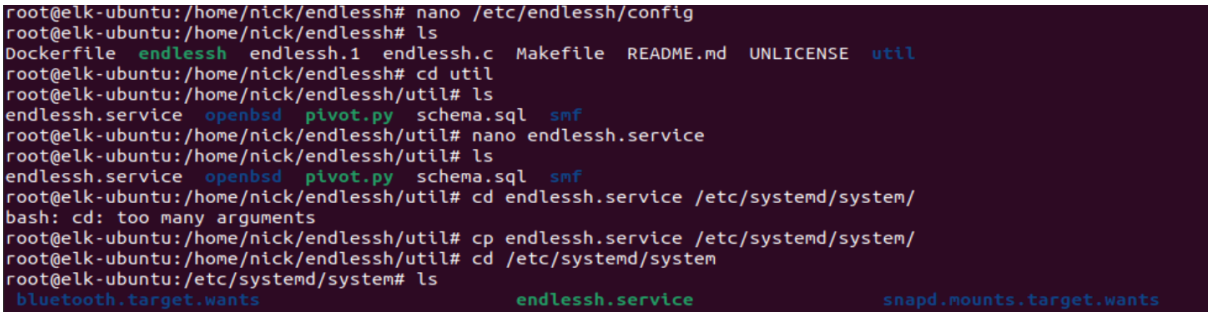


Figure 2.3

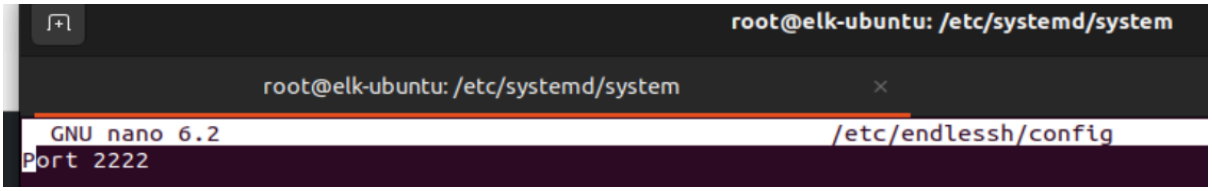


Figure 2.4 - Endless status running on ubuntu machine

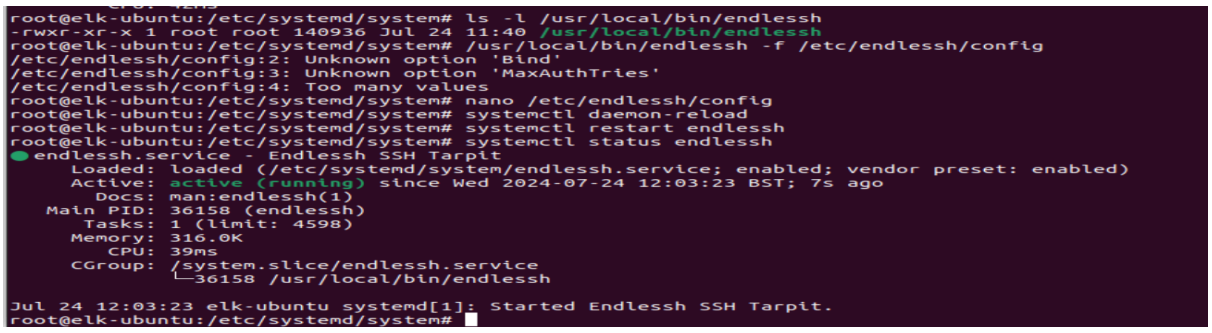


Figure 3.1 - Kali Linux SSH Attempt with no login but endless debugging banner successful.

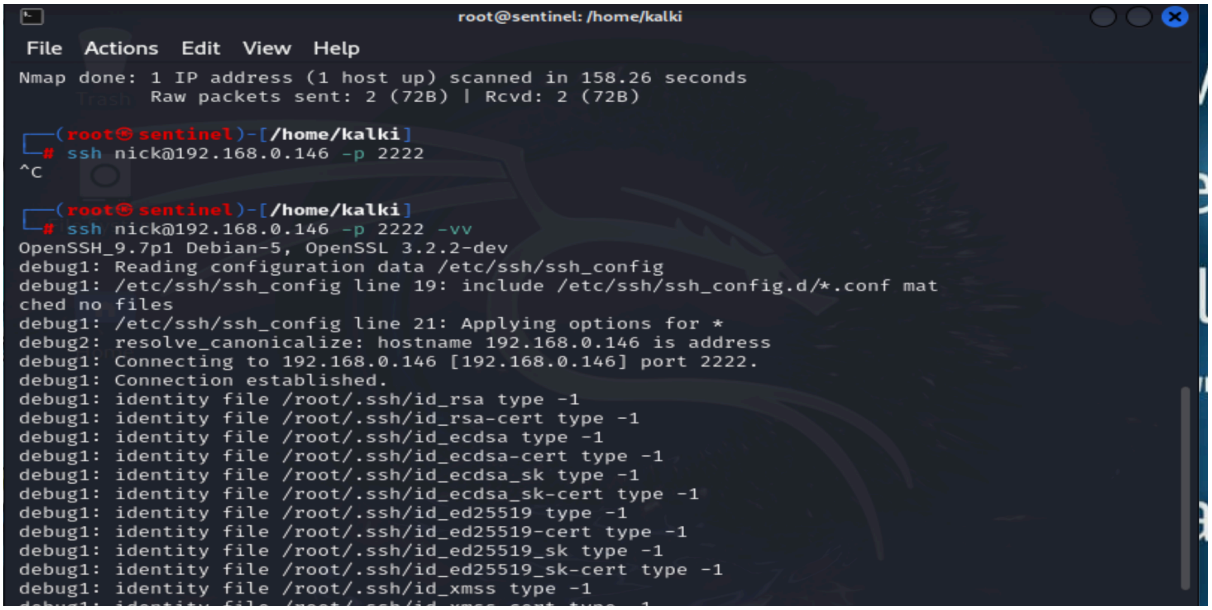
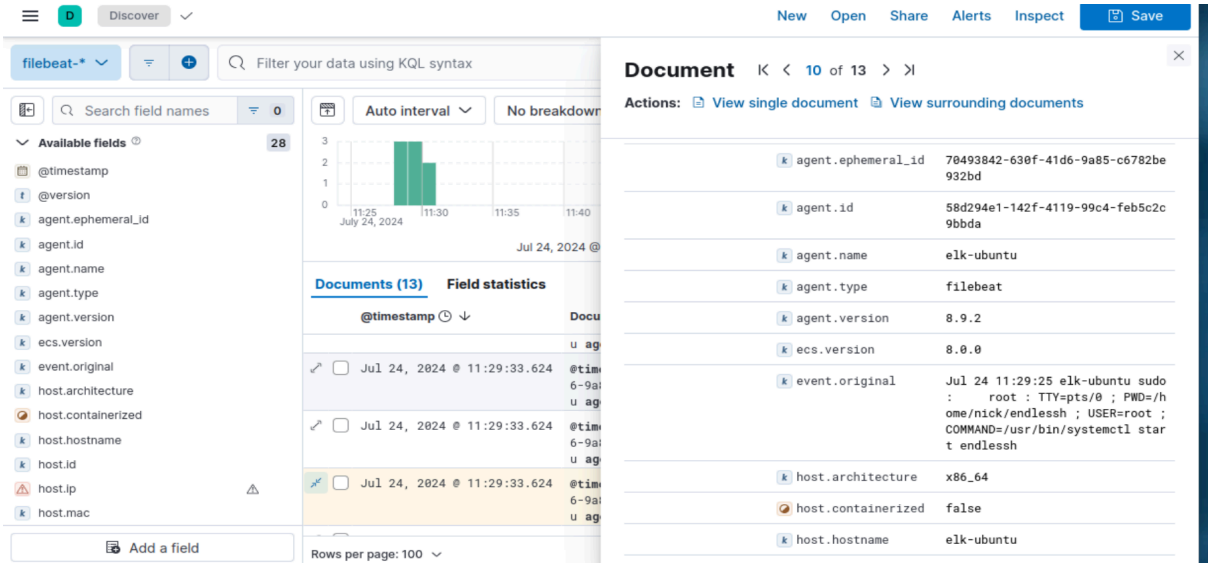


Figure 3.1 - Information of Endlessh in ELK stack for starting endless and various logs from apache2 and system. Showcasing just 1 example due to the length of the assignment and screenshots.



Methodology and Data Analysis -

Data collection: Logs from /var/log/authlog /var/log/apache2 and /var/log/syslog were gathered using Filebeat. To create a complete dataset, logs from web servers, system logs, and the Endlessh honeypot were gathered.

Data Ingestion: After applying filtering and parsing rules based on timezone and hostIP and hostname, Logstash ingested the gathered data and sent the processed logs to Elasticsearch.

Data Storage and Indexing: Elasticsearch efficiently stored and retrieved data by indexing the logs. This configuration is essential for managing substantial log volumes and preserving optimal query performance.

RESULTS AND CONCLUSION

Important insights into cybersecurity issues and attacker behavior were obtained by integrating the Endlessh honeypot with the ELK Stack on an Ubuntu 22.04 LTS virtual machine. The configuration successfully recorded and examined unsanctioned SSH access attempts, proving the value of SIEM and honeypot systems in augmenting security monitoring and analysis. This implementation's 4 main outcomes are as follows:

1. ELK Stack components, namely Elasticsearch, Logstash, and Kibana, facilitated the efficient gathering, archiving, and examination of substantial amounts of log data. The collection of logs from multiple sources, such as the Endlessh honeypot, system logs, and web server logs, was made possible in large part by Filebeat. The thorough data collection made it possible to analyze network activity and any security events in great detail.
2. Recognising Unauthorized Access Requests: Configured to resemble an SSH service, the Endlessh honeypot was able to record several attempts at unauthorized access. Kibana was used to log and analyze these attempts, and the results showed patterns suggestive of brute-force attacks. Real-time visualization and examination of these data gave important information about the types of threats.
3. Enhanced Security Posture: The assignment showed an improved capacity to identify and address security threats by setting up a honeypot and connecting it with a strong SIEM system. Future security plans and tactics will be informed by the attack patterns and possible vulnerabilities found in the log data analyzed using Kibana dashboards.
4. Practical Advice: The thorough log analysis offered practical advice on enhancing network security. SSH security configurations should be tightened, access controls should be strengthened, and additional monitoring and protection should be taken into consideration.

In conclusion, An efficient method for tracking and evaluating cybersecurity risks turned out to be the virtualized environment's combination of the ELK Stack and the Endlessh honeypot. This configuration made it easier to identify attempts at unauthorized access and gave researchers a better knowledge of the attack strategies used. Log management and analysis become affordable for a wide range of organizational sizes and industries with the help of open-source solutions such as the ELK Stack.

The experiment demonstrated the crucial function that honeypots play in cybersecurity—they act as a magnet for intruders and a priceless source of information. Honeypots allow organizations to improve their defenses and response tactics by gathering comprehensive data on attack paths and behaviors.

Overall, this assignment demonstrated the value of a proactive strategy for cybersecurity by combining SIEM and honeypot systems to build an all-encompassing framework for security monitoring. The knowledge acquired from this activity is essential for improving the security posture as a whole and making sure that companies are better able to identify, evaluate, and react to possible threats promptly. This project is the basis of the continuous endeavor to construct strong, durable cybersecurity defenses.

REFERENCES

Installation and Setup References and Resources utilised for ELK stack and Endlessh installation.

1. **ELK Stack Installation**:
    - Elasticsearch Installation Guide
    - Logstash Installation Guide
    - Kibana Installation Guide
2. **Artifact GPG Key Links**:
    - Elasticsearch GPG Key
    - Logstash GPG Key
    - Kibana GPG Key
3. **Endlessh Honeypot**:
    - Endlessh GitHub Repository
4. **Operating System ISO Downloads**:
    - Ubuntu 22.04 LTS ISO
    - Kali Linux ISO

CITATIONS

Honeypots:

1. Malwarebytes Labs. (2021). What is a honeypot? How they are used in cybersecurity. Retrieved from Malwarebytes
2. Spitzner, L. (2003). The Honeynet Project: Trapping the Hackers. Addison-Wesley.
3. DOE. (2018). Cybersecurity for Energy Delivery Systems. Retrieved from Energy.gov
4. Liu, W., Liu, X., & Zhang, Z. (2018). DDoS Attack Detection and Mitigation Using Honeypot and Cloud Computing. IEEE.

SIEM Tools:

1. Johnson, J., & Wu, T. (2019). Enhancing Healthcare Cybersecurity with Splunk. Journal of Healthcare Information Management.
2. Garcia, A., et al. (2017). Compliance and Risk Management with QRadar. IBM Security. From IBM Security
3. Smith, J., & Brown, P. (2020). Implementing ELK for Log Management and Analysis. OpenSource.com. Retrieved from OpenSource.com
4. Smith, R., et al. (2014). Analysis of the Target Data Breach. Journal of Cybersecurity. From Journal of Cybersecurity