ELK SIEM Project Documentation

Overview

This project documents the process of setting up an ELK (Elasticsearch, Logstash, Kibana) stack with Beats for centralized log collection and visualization. The goal was to simulate a basic SIEM setup that could be used to monitor Linux and Windows systems for security events in a lab environment.

Tools & Stack

- **Elasticsearch** (Search/Analytics Engine)
- **Logstash** (Used to parse and forward logs before they hit Elasticsearch)
- **Kibana** (Dashboard/Visualization)
- **Filebeat** (Linux log shipper)
- **Winlogbeat** (Windows log shipper)

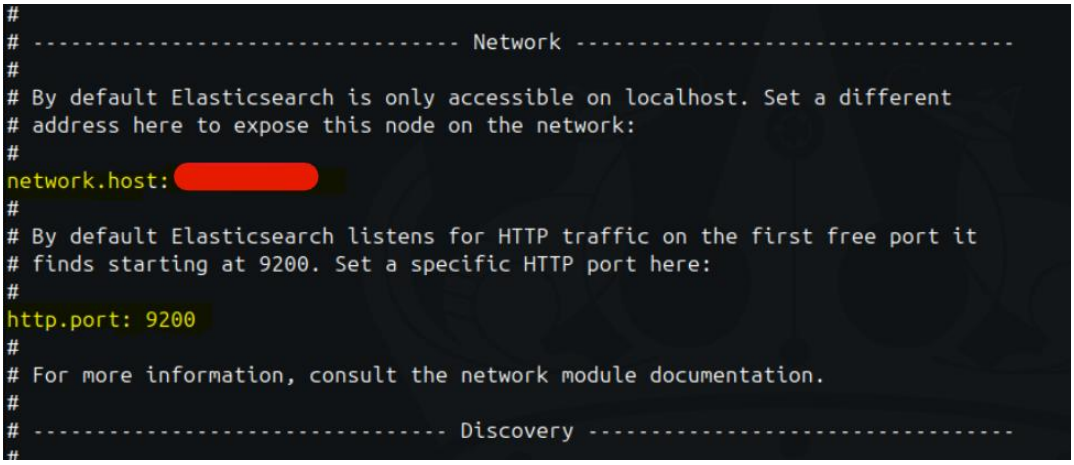
All ELK services were deployed on a Linux (Ubuntu) virtual machine named arzsec-cyber, which also served as the Linux (filebeat) endpoint. My host machine served as the Windows (winlogbeat) endpoint.

Part 1: Setting Up the ELK Stack

1. Install Elasticsearch

sudo apt update

sudo apt install elasticsearch

- Config file: /etc/elasticsearch/elasticsearch.yml
- Made the following changes using sudo nano /file-path/config-file:
    - Uncomment: network.host: <vm-ip-address>
        - Found using the command: ip a | grep inet
    - Uncomment: http.port: 9200
    - 

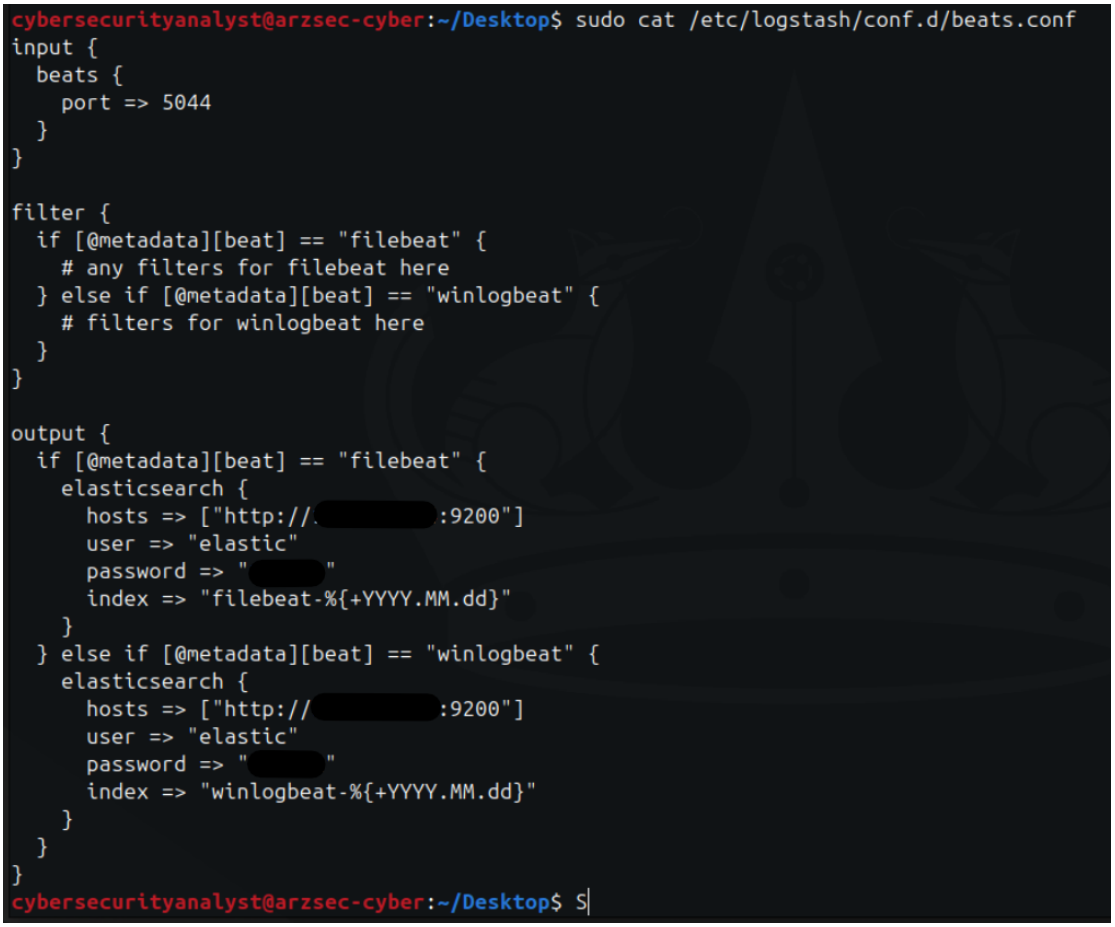
sudo systemctl enable elasticsearch

sudo systemctl start elasticsearch

---

2. Install Logstash

sudo apt install logstash

- Config file: /etc/logstash/logstash.yml
    - Uncomment: path.data: /var/lib/logstash
    - Uncomment: path.logs: /var/log/logstash
- Config directory: /etc/logstash/conf.d/
    - Created a file 'beats.conf' to tell logstash where to ingest data from and how to output it
- 

```
cybersecurityanalyst@arzsec-cyber:~/Desktop$ sudo cat /etc/logstash/conf.d/beats.conf
input {
  beats {
    port => 5044
  }
}

filter {
  if [@metadata][beat] == "filebeat" {
    # any filters for filebeat here
  } else if [@metadata][beat] == "winlogbeat" {
    # filters for winlogbeat here
  }
}

output {
  if [@metadata][beat] == "filebeat" {
    elasticsearch {
      hosts => ["http://[redacted]:9200"]
      user => "elastic"
      password => "[redacted]"
      index => "filebeat-%{+YYYY.MM.dd}"
    }
  } else if [@metadata][beat] == "winlogbeat" {
    elasticsearch {
      hosts => ["http://[redacted]:9200"]
      user => "elastic"
      password => "[redacted]"
      index => "winlogbeat-%{+YYYY.MM.dd}"
    }
  }
}
cybersecurityanalyst@arzsec-cyber:~/Desktop$ S
```

sudo systemctl enable logstash

sudo systemctl start logstash

---

2. Install Kibana

sudo apt install kibana

- Config file: /etc/kibana/kibana.yml

- Common tweaks:
    - server.host: <vm-ip-address>
    - elasticsearch.hosts: ["http:// <vm-ip-address>:9200"]

sudo systemctl enable kibana

sudo systemctl start kibana

---

3. Enable Firewall Ports (if UFW is enabled)

sudo ufw allow 9200/tcp # Elasticsearch

sudo ufw allow 5601/tcp # Kibana

Part 2: Installing and Configuring Beats

1. Filebeat (Linux Logging)

sudo apt install filebeat

Edit /etc/filebeat/filebeat.yml to make sure the following was uncommented:

```
filebeat.inputs:
- type: filestream
  id: my-filestream-id
  enabled: true
  paths:
    - /var/log/*.log
    - /var/log/*/*.log
    - /var/log/messages
    - /var/log/secure
    - /var/log/audit/audit.log
    - /var/log/nginx/*.log
output.logstash:
  hosts: ["http://<vm-ip-address>:9200"]
```

Enable Filebeat modules (optional but helpful):

sudo filebeat modules enable system

Start the service:

sudo systemctl enable filebeat

sudo systemctl start filebeat

2. Winlogbeat (Windows Event Logging)

- Download Winlogbeat[.]zip from Elastic's website
- Create a folder: C:/Program Files/Winlogbeat/ and extract the zip file to this new folder
- configure winlogbeat.yml:

winlogbeat.event_logs:

- name: Security

level: information

- name: System

- name: Application

```
winlogbeat.event_logs:
  - name: Application
    ignore_older: 72h

  - name: System

  - name: Security

  - name: Microsoft-Windows-Sysmon/Operational

  - name: Windows PowerShell
    event_id: 400, 403, 600, 800

  - name: Microsoft-Windows-PowerShell/Operational
    event_id: 4103, 4104, 4105, 4106

  - name: ForwardedEvents
    tags: [forwarded]
```

output.logstash:

hosts: ["http://<vm-ip-address>:9200"]

```
# ------------------------------ Logstash Output -------------------------------
output.logstash:
  # The Logstash hosts
  hosts: ["10.0.0.214:5044"]

  # 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"
```

Install and run as service via powershell (administrator):

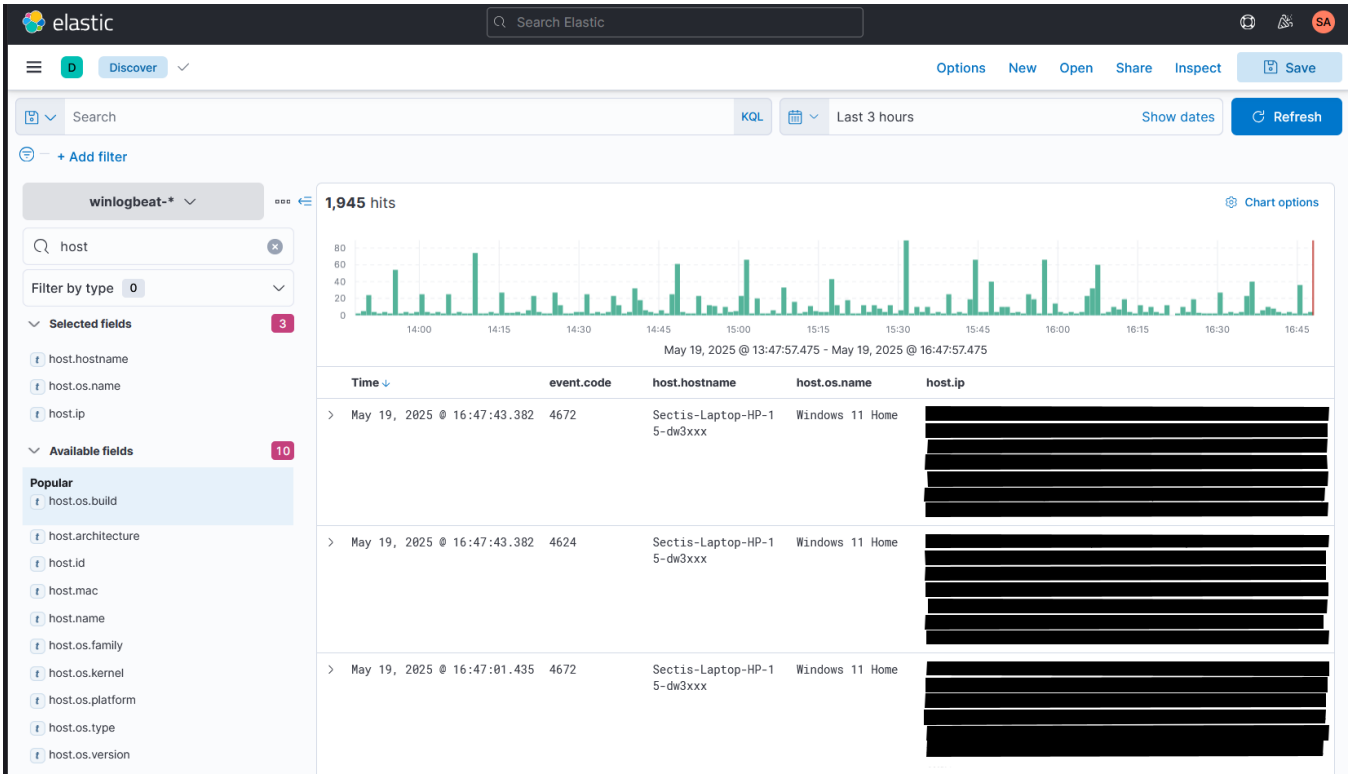
```
.\install-service-winlogbeat.ps1

Start-Service winlogbeat
```

Part 3: Ingesting Logs into Kibana

1. Create Index Patterns

- Go to **Kibana → Stack Management → Index Patterns**
- Add:
  - filebeat-*
  - winlogbeat-*
- Use @timestamp as the time filter field

2. Use Discover Tab

- Navigate to **Discover**
- Explore logs live from both sources
- 

Included below is a cheat sheet for relevant winlogbeat search filters:

Event Code	What It Means	Why It Matters
4624	Successful login	Establishes a timeline of logins
4625	Failed login	Detect brute force attempts
4634	Logoff	Completes login session context
4672	Admin privilege logon	Track when privileged accounts log in
4688	Process creation	Key for detecting malicious script execution
4697	New service installed	Attackers may install persistence
7045	A service was installed	Indicates potential backdoor setup
4720	User account created	Suspicious in production environments
4722	User account enabled	Reactivating old/stale accounts
4723/4724	Password reset/changed	Important in account compromise investigation
1102	Audit log cleared	Often seen in cover-up behavior
5156	Network connection allowed	Used for basic network monitoring

Included below is a cheat sheet for relevant filebeat search filters:

Filter	What It Means	Why It Matters
message: "sshd"	SSH-related activity	Track logins, brute force
message: "Failed password"	Failed SSH logins	Brute force, password guessing
message: "Accepted password"	Successful SSH login	Confirm compromise or access
message: "useradd" or "adduser"	New user added	Watch for unauthorized access
message: "sudo"	Privileged command used	Abuse of elevated access
message: "su: authentication failure"	Failed su command	Attempted privilege escalation
message: "systemd"	Service changes	Similar to 7040 on Windows
message: "iptables"	Firewall changes	Potential bypass of restrictions

Part 4: Building Dashboards

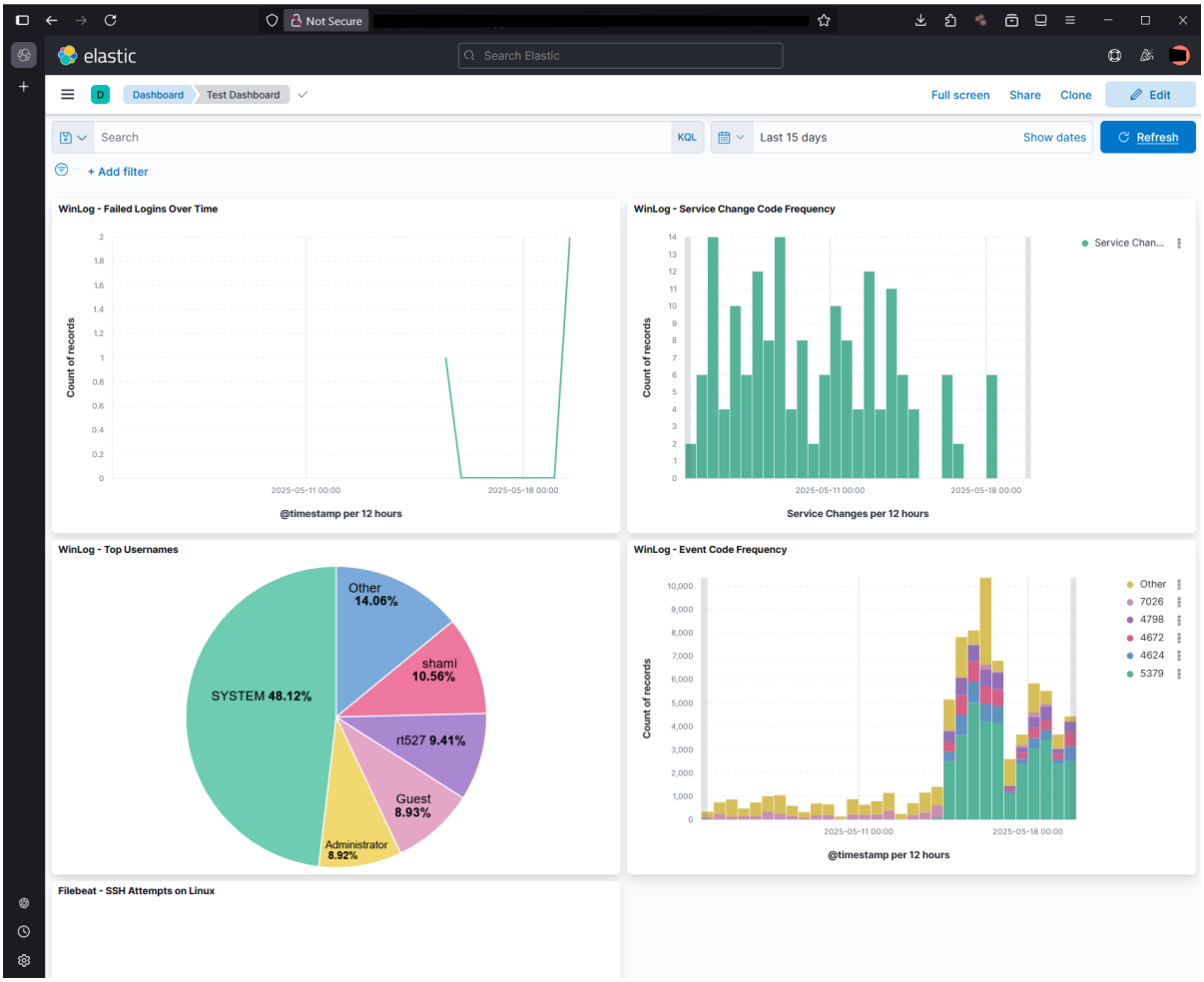
Go to **Kibana → Dashboard → Create New**

Sample Visualizations

- **Failed Logins Over Time**:
    - Index: winlogbeat-*
    - Filter: event.code: 4625
    - Type: Line chart
- **Top Usernames**:
    - Index: winlogbeat-*
    - Field: winlog.event_data.TargetUserName
    - Type: Pie chart
- **SSH Attempts on Linux**:
    - Index: filebeat-*
    - Filter: message: "sshd"
- **Event Code Frequency**:
    - Field: event.code
    - Bar graph of common event IDs

(Find a sample dashboard on the next page)

Sample Dashboard:



Notes

- If Kibana shows no logs, check:
  - Filebeat/Winlogbeat/Logstash service status
  - Elasticsearch/Logstash/Kibana logs
  - Time filter set to "Last 15 minutes"
- Restart Beats after every config change
- Tag your Beats agents with tags: for better filtering

Next Steps

Once you're collecting logs, you can:

- Simulate attacks (failed logins, service stops, malware)
- Monitor live logs
- Set up alerts using Elastic's SIEM app

- Tune dashboards for incident response

Closing Thoughts

Setting up this ELK-based SIEM lab was a valuable learning experience in understanding the flow of logs from endpoint systems to a centralized dashboard. It bridged the gap between raw system events and meaningful security insights. One of the key takeaways was how powerful and flexible the ELK stack can be when properly configured, especially with Beats agents providing targeted log collection.

Throughout the process, I encountered and resolved several challenges—ranging from service startup failures and misconfigured YAML files to gaps in visibility caused by firewall restrictions and time filter mismatches in Kibana. These issues reinforced the importance of careful troubleshooting and log review as part of system monitoring.

Going forward, I plan to build on this foundation by:

i. Enabling and tuning detection rules using Elastic's Security app

ii. Setting up email or webhook-based alerts for high-severity events

iii. Expanding the lab to include Metricbeat and Packetbeat for system and network telemetry

iv. Simulating attack scenarios using tools like Invoke-AtomicRedTeam or Metasploit to generate more complex log data

v. Exploring integration with other platforms like Wazuh or TheHive for a more robust security operations workflow

Overall, this project provided not only technical experience but also a stronger conceptual understanding of how security teams leverage SIEM platforms to detect and investigate threats in real time.
