Complete Wazuh + Filebeat + Elasticsearch OSS + Kibana OSS Setup with Cisco Switch Log Collection

This document outlines a **step-by-step guide** to set up Wazuh (with Filebeat), Elasticsearch OSS, and Kibana OSS to collect and analyze logs from network devices like **Cisco Catalyst 1000**. It includes all successful commands, their outputs, and detailed descriptions.

Why This Stack? (Theory &

Purpose) Wazuh

Wazuh is a **Security Information and Event Management (SIEM)** solution used for:

- Log analysis and correlation
- Threat detection
- Compliance monitoring (e.g., PCI-DSS, GDPR)

It reads logs from various sources and matches them against detection rules to generate alerts.

Filebeat

Filebeat is a lightweight log shipper:

- Sends Wazuh alerts or other logs to Elasticsearch
- Acts as a forwarder so Wazuh doesn't directly write into Elasticsearch

Elasticsearch OSS

Elasticsearch is a **distributed search and analytics engine**:

- Stores and indexes logs from Wazuh via Filebeat
- Provides real-time querying capabilities

Kibana OSS

Kibana is a web-based visualization tool:

- Connects to Elasticsearch
- Displays dashboards, search results, and log alerts interactively

Why rsyslog?

rsyslog is used to collect syslog data (e.g., from Cisco switches) and write it to a file Wazuh can read.

Flow Summary

Cisco Switch → rsyslog → /var/log/cisco_switch.log → Wazuh → Filebeat → Elasticsearch →
Kibana

Install Elasticsearch OSS 7.10.2

Elasticsearch is the core data store used by Wazuh and visualized via Kibana.

wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-oss-7.10.2-amd64.deb sudo dpkg -i elasticsearch-oss-7.10.2-amd64.deb

sudo systemctl daemon-reexec

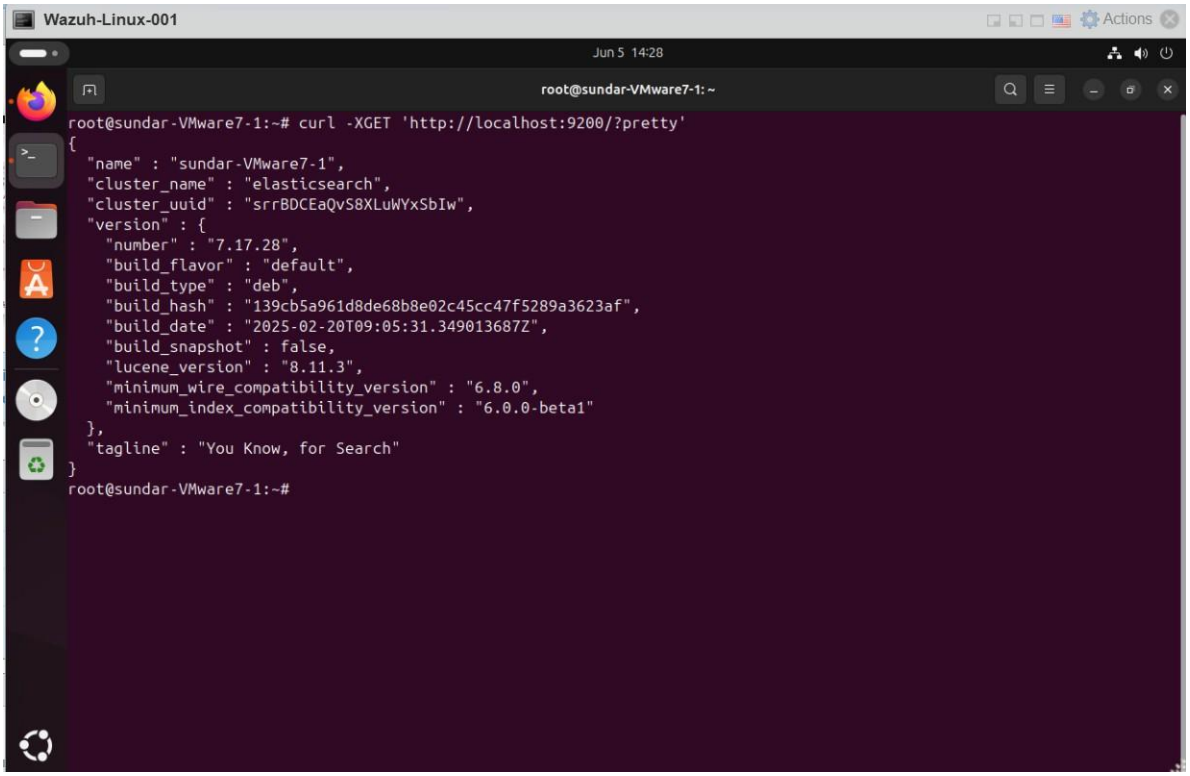
sudo systemctl enable elasticsearch --now

🟩 Verify Elasticsearch is Running

curl -XGET 'http://localhost:9200/?pretty'

Expected output:

```
{
 "name" : "sundar-
 VMware7-1",
 "cluster_name" :
 "elasticsearch", "version"
 : {
  "number" : "7.10.2",
  ...
 },
 "tagline" : "You Know, for Search"
}
```



Install Kibana OSS 7.10.2

Kibana is the frontend interface to visualize alerts and logs.

wget https://artifacts.elastic.co/downloads/kibana/kibana-oss-

7.10.2-amd64.deb sudo dpkg -i kibana-oss-7.10.2-amd64.deb

Configure Kibana to Connect to Elasticsearch

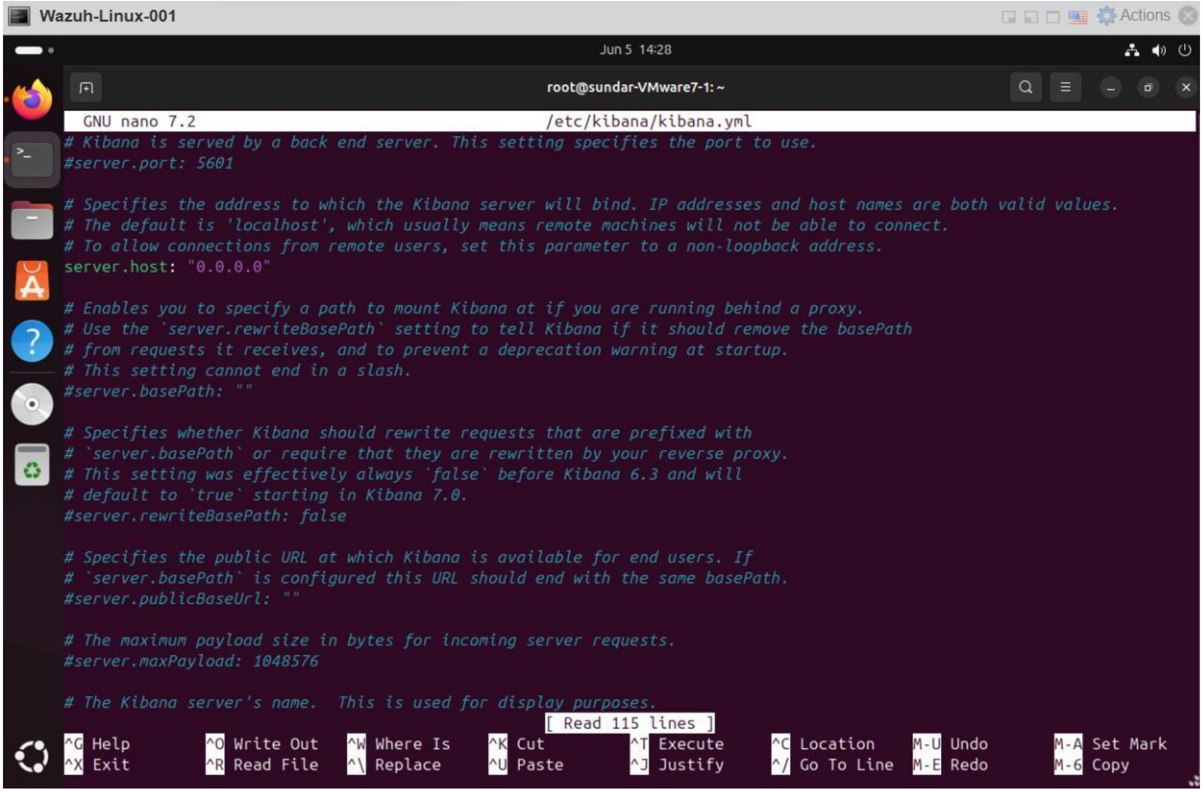
sudo nano

/etc/kibana/kibana.yml

Add the following:

server.host: "0.0.0.0"

elasticsearch.hosts: ["http://localhost:9200"]



Start Kibana:

sudo systemctl enable kibana --now

You can access the Kibana web interface at http://<your-ip>:5601.

Install Wazuh Manager

Wazuh analyzes and correlates logs for security and

compliance monitoring. curl -sO

https://packages.wazuh.com/4.7/wazuh-manager-4.7.3.deb

sudo dpkg -i wazuh-manager-4.7.3.deb

sudo systemctl enable wazuh-manager --now

Configure Wazuh to Parse Cisco Logs

Wazuh needs to know where to read logs from:

sudo nano /var/ossec/etc/ossec.conf

```
<ossec_config>
 <global>
  <jsonout_output>yes</jsonout_output>
  <alerts_log>yes</alerts_log>
  <logall>no</logall>
  <logall_json>no</logall_json>
  <email_notification>no</email_notification>
  <update_check>yes</update_check>
  <agents_disconnection_time>10m</agents_disconnection_time>
  <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
  <white_list>127.0.0.1</white_list>
  <white_list>^localhost.localdomain$</white_list>
  <white_list>127.0.0.53</white_list>
 </global>

 <alerts>
  <log_alert_level>3</log_alert_level>
  <email_alert_level>12</email_alert_level>
  <scan_on_start>yes</scan_on_start>

 </alerts>

 <logging>

  <log_format>plain</log_format>
 </logging>

 <remote>
  <connection>secure</connection>
  <port>1514</port>
```

```
  <protocol>tcp</protocol>
  <queue_size>131072</queue_size>
</remote>

<rootcheck>
  <disabled>no</disabled>
  <check_files>yes</check_files>
  <check_trojans>yes</check_trojans>
  <check_dev>yes</check_dev>
  <check_sys>yes</check_sys>
  <check_pids>yes</check_pids>
  <check_ports>yes</check_ports>
  <check_if>yes</check_if>
  <frequency>43200</frequency>
  <rootkit_files>etc/rootcheck/rootkit_files.txt</rootkit_files>
  <rootkit_trojans>etc/rootcheck/rootkit_trojans.txt</rootkit_trojans>
  <skip_nfs>yes</skip_nfs>
  <ignore>/var/lib/containerd</ignore>
  <ignore>/var/lib/docker/overlay2</ignore>
</rootcheck>

<syscheck>
  <disabled>no</disabled>
  <frequency>43200</frequency>

  <alert_new_files>yes</alert_new_files>
  <directories>/etc,/usr/bin,/usr/sbin</directories>
  <directories>/bin,/sbin,/boot</directories>
  <ignore>/etc/mtab</ignore>
  <ignore>/etc/hosts.deny</ignore>
```

```
    <ignore>/etc/mail/statistics</ignore>
    <ignore>/etc/random-seed</ignore>
    <ignore>/etc/random.seed</ignore>
    <ignore>/etc/adjtime</ignore>
    <ignore>/etc/httpd/logs</ignore>
    <ignore>/etc/utmpx</ignore>
    <ignore>/etc/wtmpx</ignore>
    <ignore>/etc/cups/certs</ignore>
    <ignore>/etc/dumpdates</ignore>
    <ignore>/etc/svc/volatile</ignore>
    <ignore type="sregex">.log$|.swp$</ignore>
    <nodiff>/etc/ssl/private.key</nodiff>
    <skip_nfs>yes</skip_nfs>
    <skip_dev>yes</skip_dev>
    <skip_proc>yes</skip_proc>
    <skip_sys>yes</skip_sys>
    <process_priority>10</process_priority>
    <max_eps>50</max_eps>
    <synchronization>
      <enabled>yes</enabled>
      <interval>5m</interval>
      <max_eps>10</max_eps>
    </synchronization>
  </syscheck>
  <wodle name="syscollector">
    <disabled>no</disabled>
    <interval>1h</interval>
    <scan_on_start>yes</scan_on_start>
    <hardware>yes</hardware>
    <os>yes</os>
    <network>yes</network>
```

```
  <packages>yes</packages>
  <ports all="no">yes</ports>
  <processes>yes</processes>
  <synchronization>
    <max_eps>10</max_eps>
  </synchronization>
</wodle>

<sca>
  <enabled>yes</enabled>
  <scan_on_start>yes</scan_on_start>
  <interval>12h</interval>
  <skip_nfs>yes</skip_nfs>
</sca>

<vulnerability-detection>
  <enabled>yes</enabled>
  <index-status>yes</index-status>
  <feed-update-interval>60m</feed-update-interval>
</vulnerability-detection>


<localfile>
  <log_format>command</log_format>
  <command>df -P</command>
  <frequency>360</frequency>
</localfile>

<localfile>
  <log_format>full_command</log_format>
```

```
 <command>last -n 20</command>
 <frequency>360</frequency>
</localfile>

<localfile>
 <log_format>syslog</log_format>
 <location>/var/log/syslog</location>
</localfile>

<localfile>
 <log_format>syslog</log_format>
 <location>/var/ossec/logs/active-responses.log</location>
</localfile>

<localfile>
 <log_format>syslog</log_format>
 <location>/var/log/dpkg.log</location>
</localfile>

<ruleset>
 <decoder_dir>ruleset/decoders</decoder_dir>
 <rule_dir>ruleset/rules</rule_dir>
 <rule_exclude>0215-policy_rules.xml</rule_exclude>
 <list>etc/lists/audit-keys</list>
 <list>etc/lists/amazon/aws-eventnames</list>
 <list>etc/lists/security-eventchannel</list>
```

```
  <decoder_dir>etc/decoders</decoder_dir>
  <rule_dir>etc/rules</rule_dir>
</ruleset>

<rule_test>
  <enabled>yes</enabled>
  <threads>1</threads>
  <max_sessions>64</max_sessions>
  <session_timeout>15m</session_timeout>
</rule_test>

<auth>
  <disabled>no</disabled>
  <port>1515</port>
  <use_source_ip>no</use_source_ip>
  <purge>yes</purge>
  <use_password>no</use_password>
  <ciphers>HIGH:!ADH:!EXP:!MD5:!RC4:!3DES:!CAMELLIA:@STRENGTH</ciphers>
  <ssl_verify_host>no</ssl_verify_host>
  <ssl_manager_cert>etc/sslmanager.cert</ssl_manager_cert>
  <ssl_manager_key>etc/sslmanager.key</ssl_manager_key>
  <ssl_auto_negotiate>no</ssl_auto_negotiate>
</auth>

<cluster>
  <name>wazuh</name>
  <node_name>node01</node_name>
  <node_type>master</node_type>
  <key></key>
```

```
    <port>1516</port>
    <bind_addr>0.0.0.0</bind_addr>

    <nodes>
      <node>NODE_IP</node>
    </nodes>
    <hidden>no</hidden>
    <disabled>yes</disabled>
  </cluster>

<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/auth.log</location>
</localfile>



<remote>
  <connection>syslog</connection>
  <protocol>udp</protocol>
  <port>514</port>
</remote>

</ossec_config>
```

Inside <ossec_config>, add:

```
<localfile>
  <log_format>syslog</log_format>
```

<location>/var/log/cisco_switch.log</location>

</localfile>

Restart

Wazuh:

sudo systemctl restart wazuh-manager

This tells Wazuh to read Cisco logs written to /var/log/cisco_switch.log.

Enable rsyslog to Accept Cisco Syslog Messages

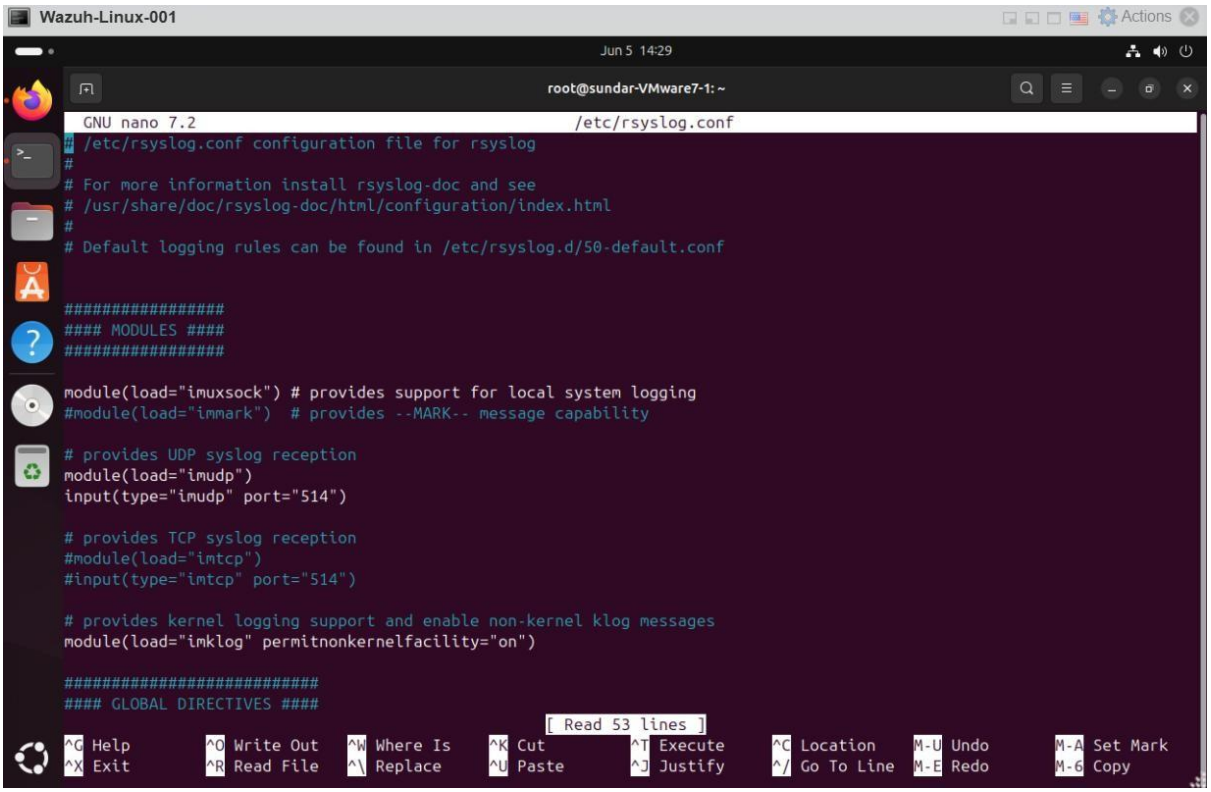
Make sure rsyslog is configured to receive logs over

UDP port 514: sudo nano /etc/rsyslog.conf

Uncomment:

module(load="imudp")

input(type="imudp" port="514")



Create rsyslog Rule for

Cisco Logs sudo nano

/etc/rsyslog.d/cisco.conf Add:

if ($fromhost-ip == '192.168.1.30') then

/var/log/cisco_switch.log & stop

Restart rsyslog:

sudo systemctl restart rsyslog

This routes incoming logs from Cisco switch (IP 192.168.1.30) to a dedicated file.

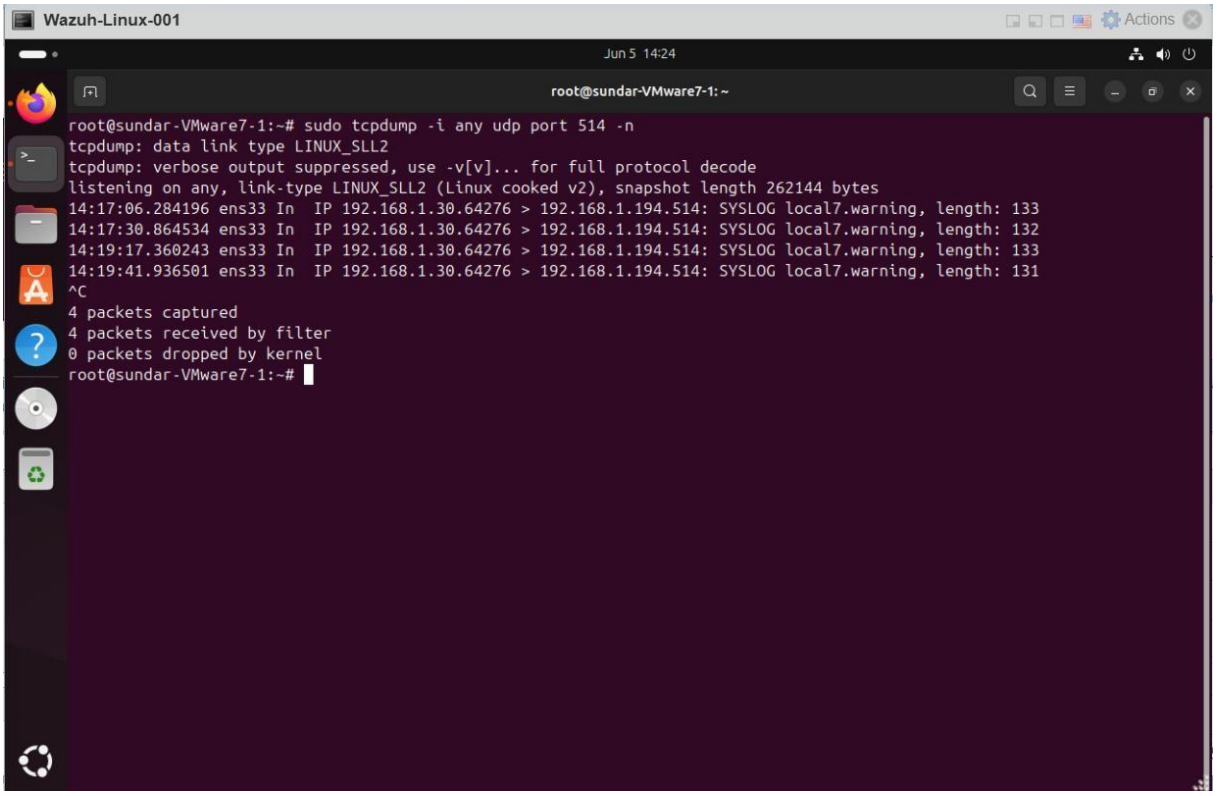
Verify Logs from Cisco Switch

Monitor incoming packets:

sudo tcpdump -i any udp port 514

-n Example:

IP 192.168.1.30.65398 > 192.168.1.194.514: SYSLOG local7.warning



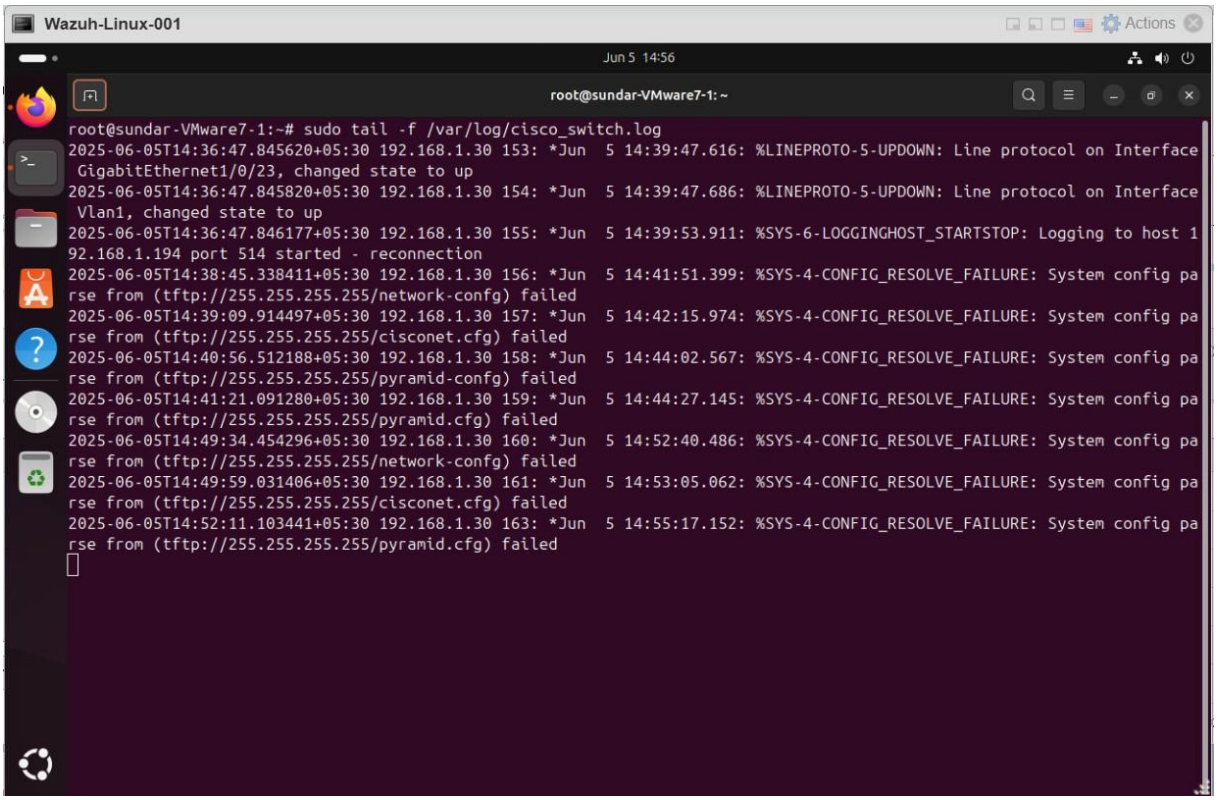
Check the log file:

sudo tail -f

/var/log/cisco_switch.log

Example logs:

%SYS-4-CONFIG_RESOLVE_FAILURE: System config parse from (tftp://...) failed



Confirm Wazuh Alerting on

Switch Logs sudo tail -f

/var/ossec/logs/alerts/alerts.log

Expected:

cisco.facility:

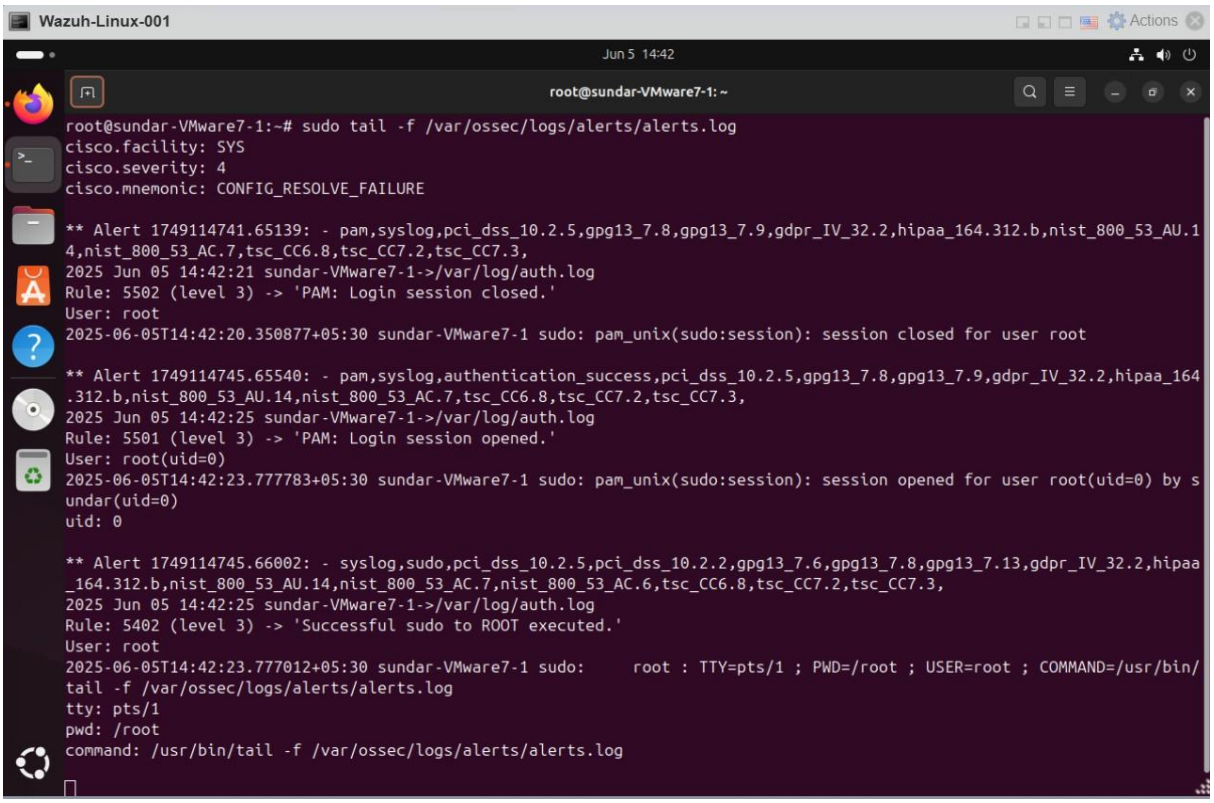
SYS

cisco.severity

: 4

cisco.mnemonic: CONFIG_RESOLVE_FAILURE

This confirms Wazuh is parsing and generating alerts.



Install Filebeat OSS 7.10.2 for Log Shipping

Filebeat forwards Wazuh logs to Elasticsearch.

curl -sO https://packages.wazuh.com/4.7/filebeat-oss-7.10.2-

x86_64.deb sudo dpkg -i filebeat-oss-7.10.2-x86_64.deb

Filebeat Configuration

sudo nano

/etc/filebeat/filebeat.yml Set

output and Kibana:

output.elasticsearch:

hosts: ["http://localhost:9200"]

setup.kibana:

host:

"localhost:5601"

Enable Wazuh

module:

sudo filebeat modules enable

wazuh Start Filebeat:

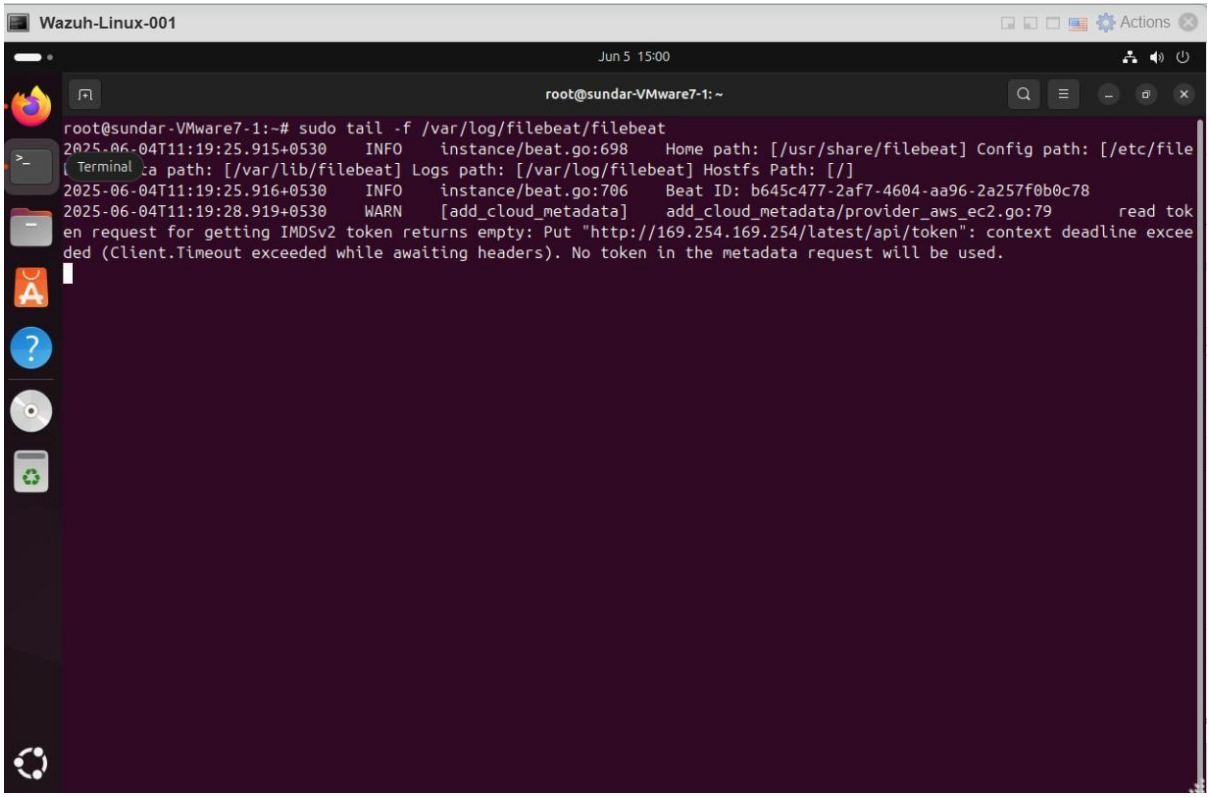
sudo systemctl enable filebeat --now

Verify the Setup

Check each component:

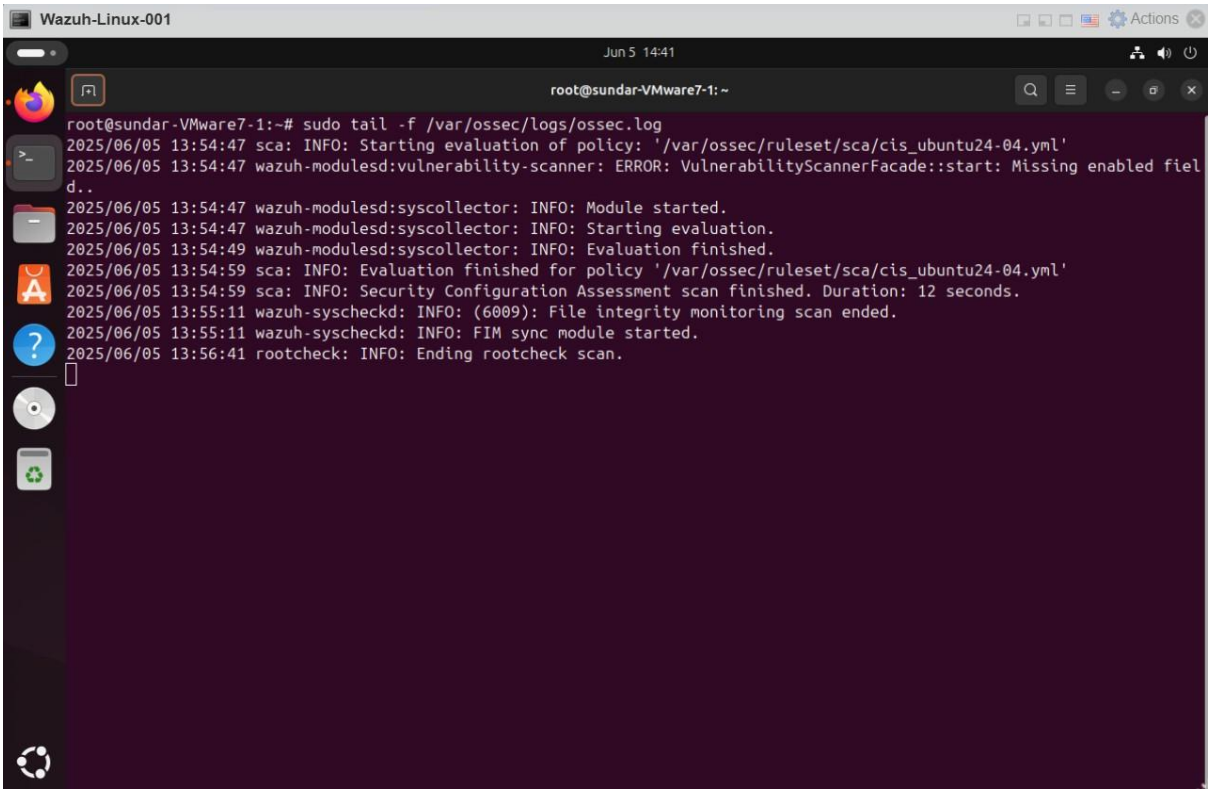
Filebeat logs:

sudo tail -f /var/log/filebeat/filebeat



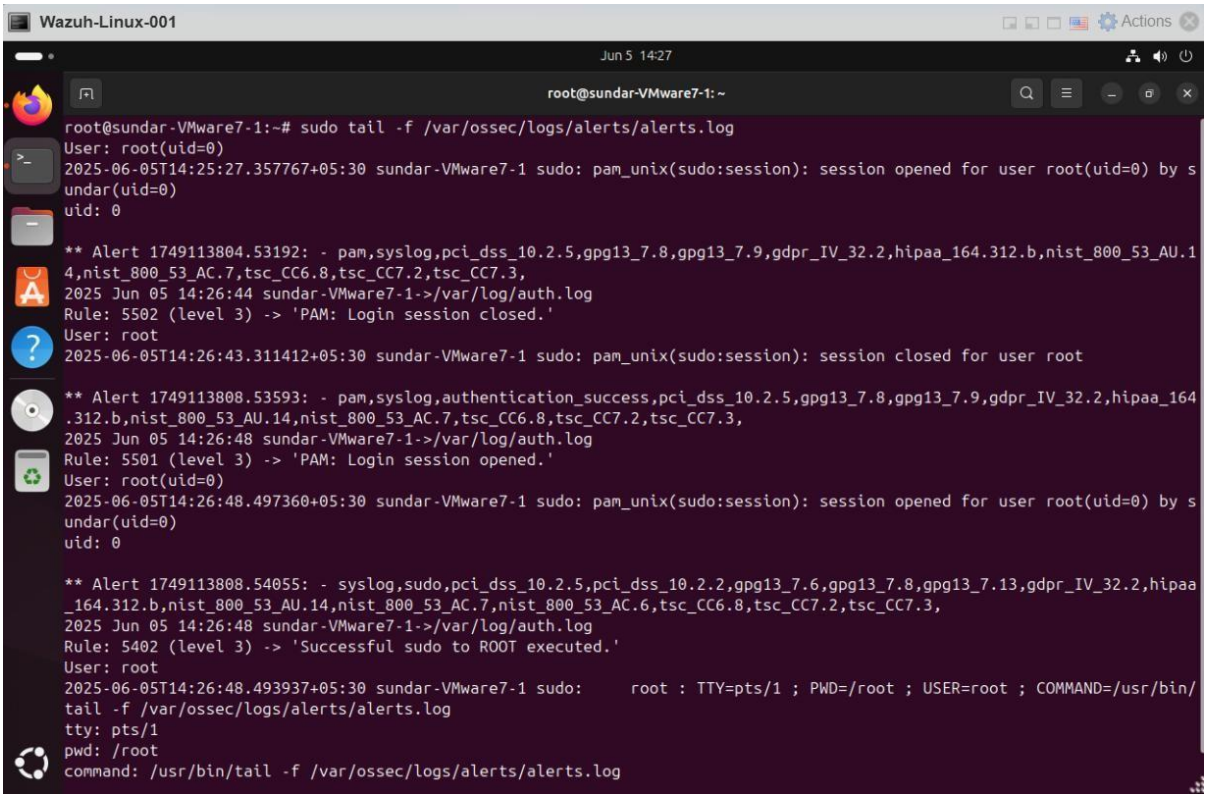
Wazuh internal logs:

sudo tail -f /var/ossec/logs/ossec.log



Wazuh alerts:

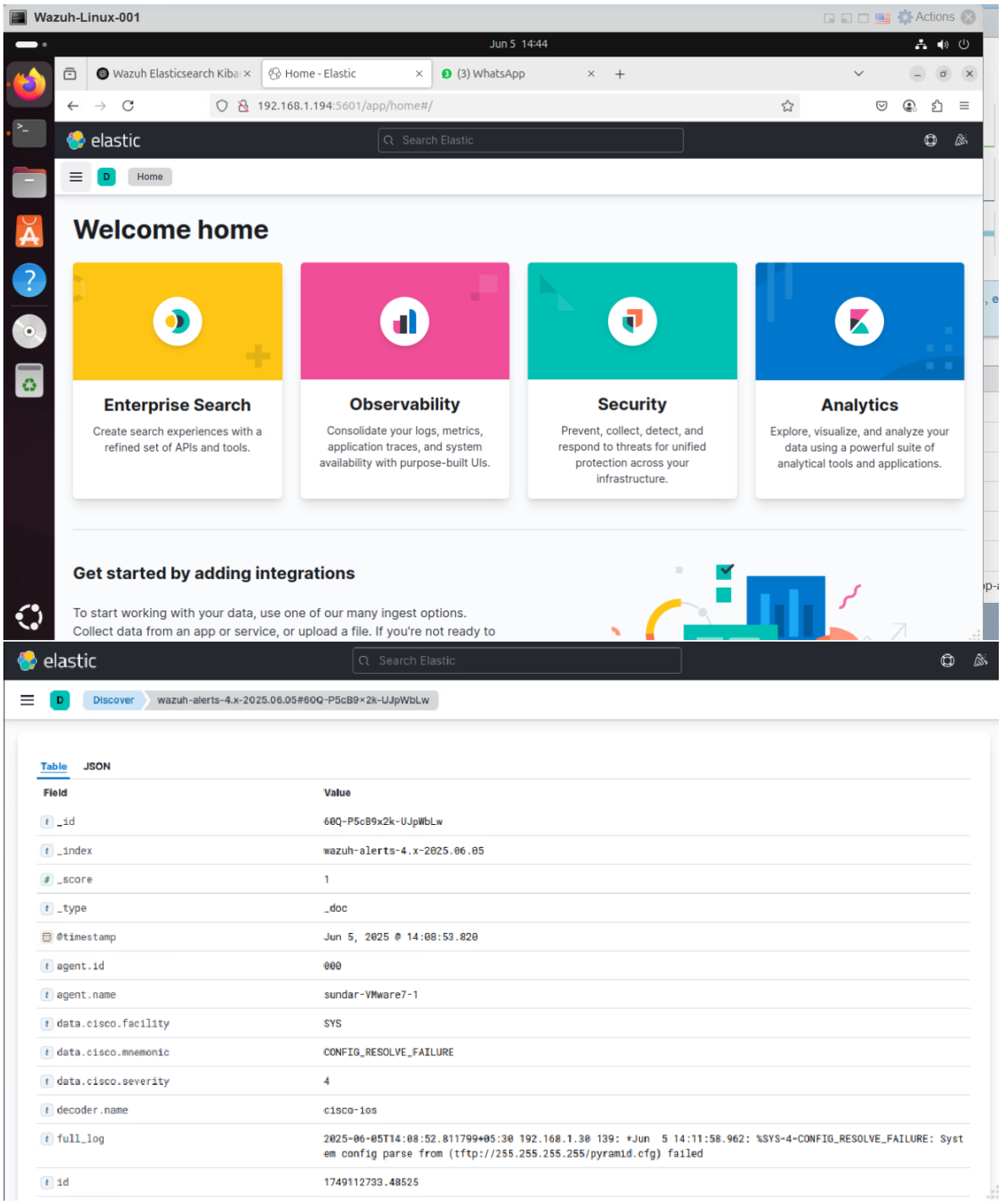
sudo tail -f /var/ossec/logs/alerts/alerts.log

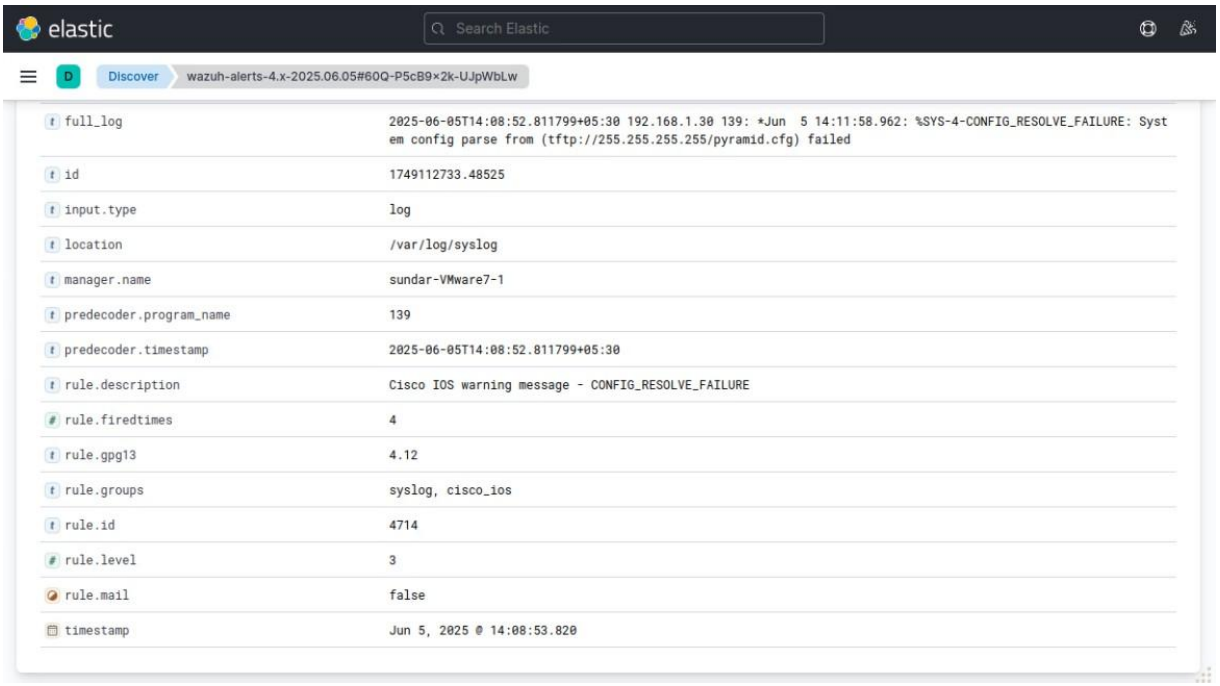


Kibana Discover:

Use Kibana to search logs:

- Navigate to **Discover** tab
- Search syslog, pam, or Cisco messages
- Create visualizations if needed





Devices Tested

- **Cisco Catalyst 1000 Series** (IP: 192.168.1.30)

• FORTINET LOGS

FortiGate Configuration (Syslog Forwarding)

On the FortiGate web interface:

1. **Go to Log & Report → Log Settings**
2. **Under Syslog Servers, configure:**
   - **Server: 192.168.1.194 (IP of Wazuh server)**
   - **Port: 514**
   - **Format: Default**
   - **Facility: local7**
   - **Enable log types: Traffic, Event, UTM logs, etc.**

Firewall CLI

```
config log syslogd setting
    set status enable
    set server <WAZUH_IP>
    set mode udp
    set port 514
```

```
    set facility local7
end

config log syslogd filter
    set severity information
    set forward-traffic enable
    set local-traffic enable
    set multicast-traffic enable
    set sniffer-traffic enable
    set anomaly enable
    set voip enable
end
```

Configure rsyslog on Wazuh Server
Edit /etc/rsyslog.conf and ensure these lines are uncommented:
bash
CopyEdit
module(load="imudp")
input(type="imudp" port="514")
Then create a new rule file:
bash
CopyEdit
sudo nano /etc/rsyslog.d/fortigate.conf
Add:
bash
CopyEdit
if ($fromhost-ip == '192.168.1.1') then /var/log/fortigate.log
& stop
Replace 192.168.1.1 with your actual FortiGate IP address.
Restart rsyslog:
bash
CopyEdit
sudo systemctl restart rsyslog

🛠️ Configure Wazuh to Parse FortiGate Logs
Edit /var/ossec/etc/ossec.conf and add:
xml
CopyEdit
<localfile>
** <log_format>syslog</log_format>**
** <location>/var/log/fortigate.log</location>**
</localfile>
Then restart Wazuh:
bash
CopyEdit
sudo systemctl restart wazuh-manager

📡 Validate Logs & Alerts

To check if logs are being received:

bash

CopyEdit

```
sudo tail -f /var/log/fortigate.log
```

To confirm Wazuh is generating alerts:

bash

CopyEdit

```
sudo tail -f /var/ossec/logs/alerts/alerts.log
```

You should see entries like:

makefile

CopyEdit

```
rule.id: 81633
rule.description: Fortigate: App passed by firewall.
app: Google.Chat
action: pass
srcip: 192.168.1.58
dstip: 142.251.222.142
```

Kibana Integration

Open Kibana (http://<wazuh-ip>:5601), go to Discover, and search:

kql

CopyEdit

```
decoder.name: "fortigate-firewall-v5"
```

Create dashboards or alerts for:

- **Allowed/Dropped apps (e.g., Google.Chat)**
- **Blocked IPs or ports**
- **App usage by department/IP**

✅ Example FortiGate Log Parsed

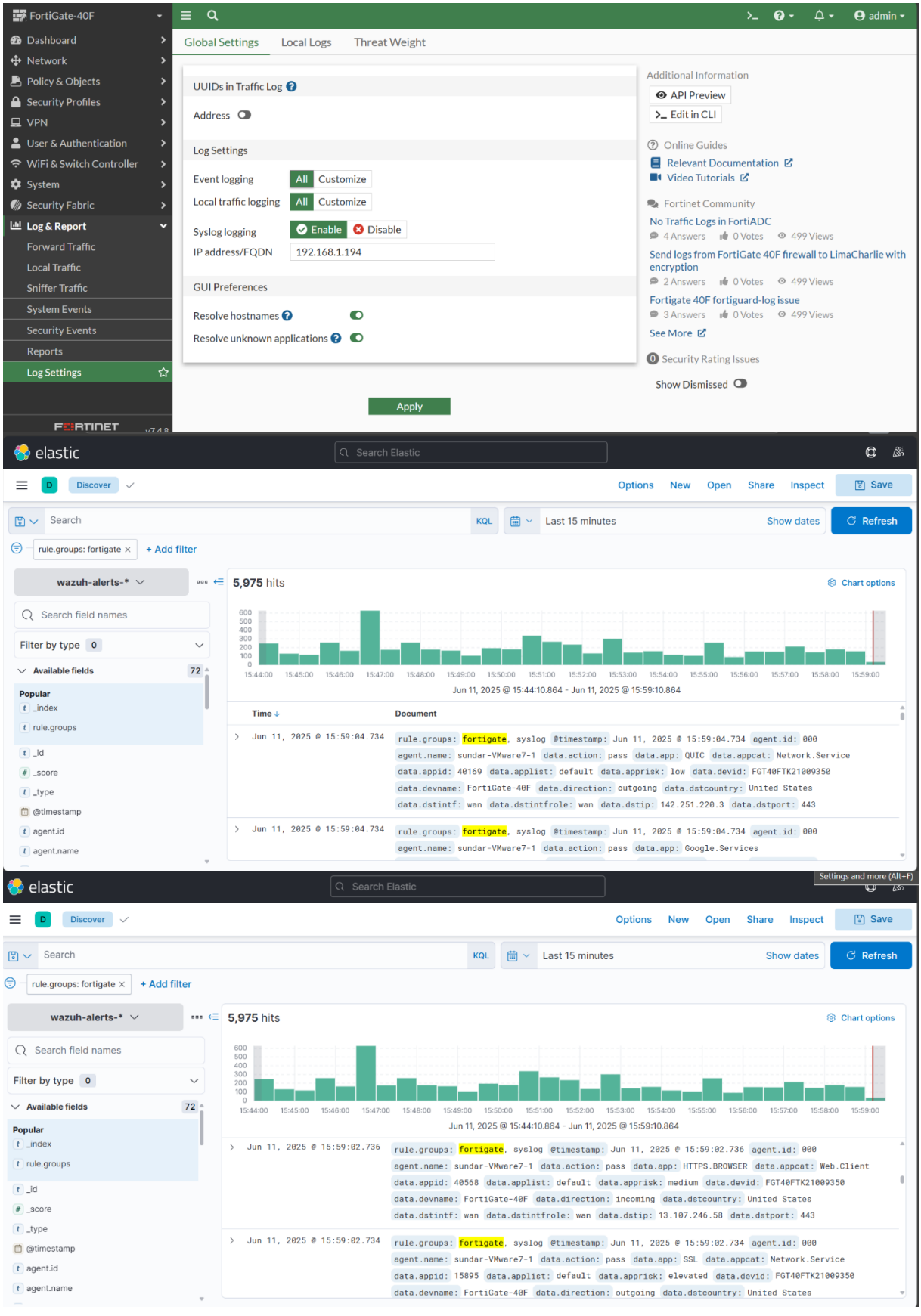
json

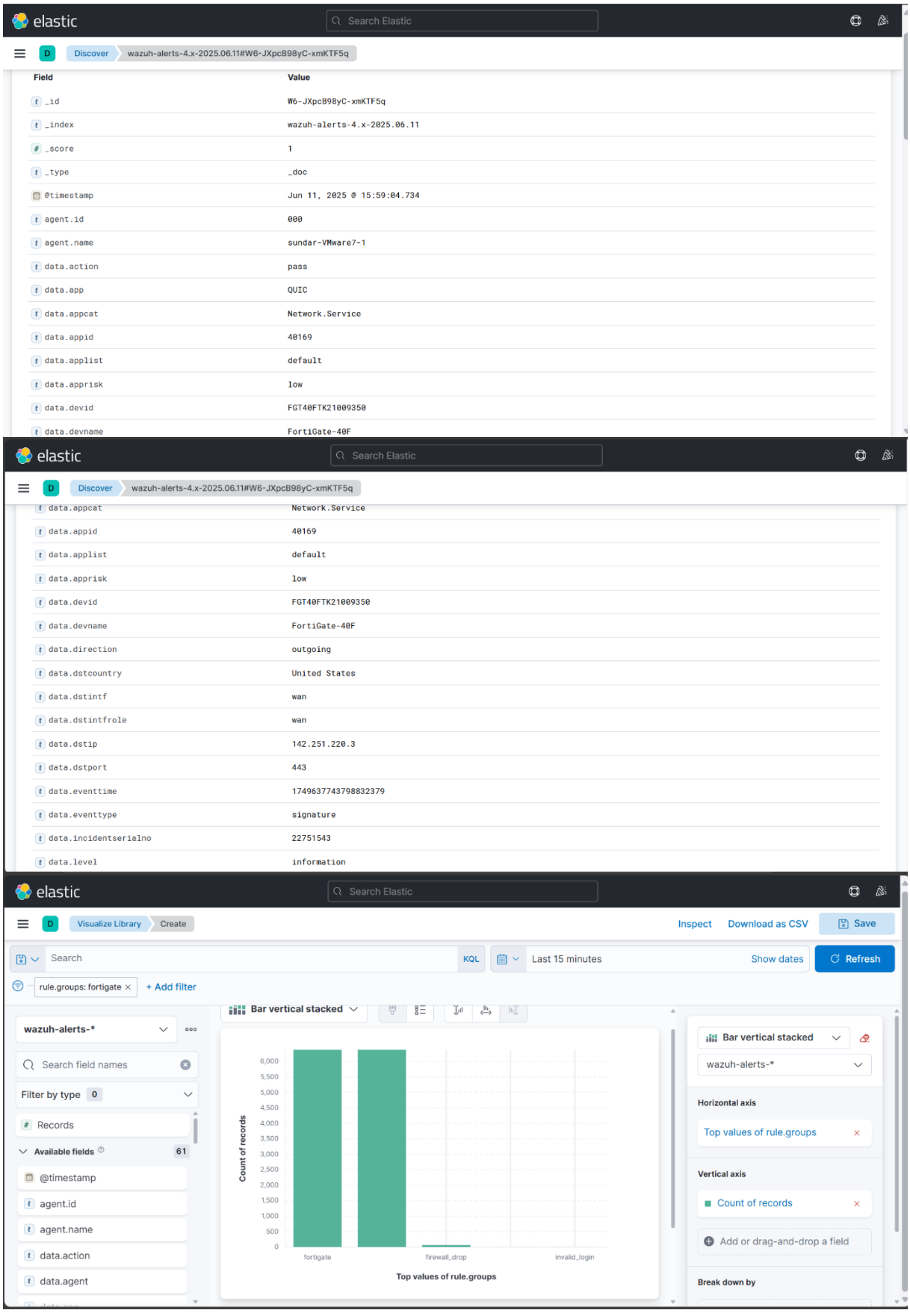
CopyEdit

```
{
  "app": "Google.Chat",
  "action": "pass",
  "srcip": "192.168.1.58",
  "dstip": "142.251.222.142",
  "proto": "17",
  "service": "HTTPS",
  "hostname": "chat.google.com",
  "decoder.name": "fortigate-firewall-v5",
  "rule.description": "Fortigate: App passed by firewall."
}
```

Tested Firewall Device

- **FortiGate-40F**
  - **IP: 192.168.1.1**
  - **syslog facility: local7**
  - **Transport: UDP port 514**





✅ Now Your Setup Supports:

- **Cisco Catalyst Switch Logging**
- **FortiGate Firewall Logging**
- **Wazuh Real-Time Alerting Centralized Visual ization with Kibana.**