École Nationale des Sciences Appliquées
Tétouan

Cybersecurity and embedded systems , GCSE

Advanced Security Operations Center LAB

Prepared by :

BOUSSETA HATIM

Contents

July 10, 2025

1 Introduction

This document outlines the architecture of our Security Operations Center (SOC) lab environment, which integrates the Wazuh XDR/SIEM platform with a comprehensive security toolkit. The lab features a complete Wazuh deployment including agents (deployed on Windows Domain Controller, Ubuntu, and Parrot systems), manager, indexer, and dashboard components. These are integrated with the ELK stack for enhanced log analysis capabilities. The environment further incorporates network security tools including Snort IDS/IPS and PfSense firewall, complemented by Zeek for network traffic analysis. For threat intelligence and automation, the lab leverages Shuffle for SOAR capabilities, The Hive for incident response, MISP for threat intelligence sharing, and Cortex for automated analysis. This integrated setup provides a realistic environment for security monitoring, threat detection, and incident response training.

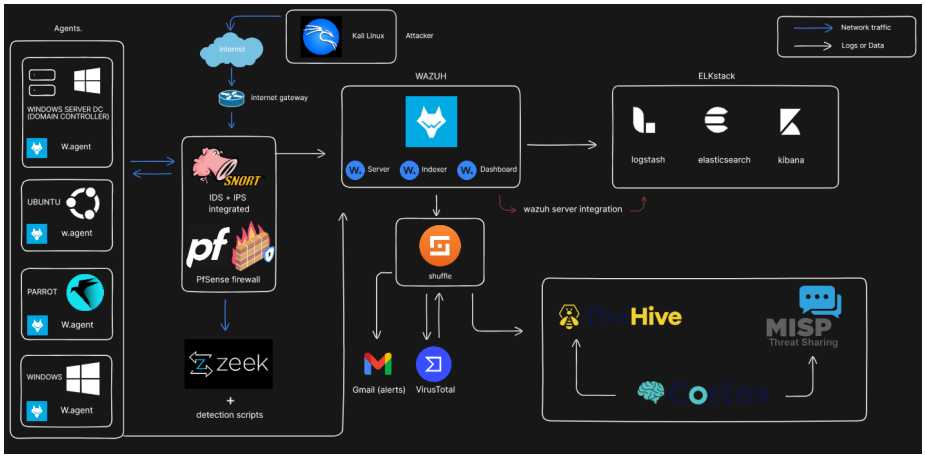


Figure 1: SOC Lab Architecture Diagram

2 Wazuh

2.1 What is Wazuh?

Wazuh is a free and open source security platform that unifies XDR and SIEM capabilities .

2.2 Wazuh Architecture

The Wazuh system in our lab consists of:

- **Agents**: Lightweight components installed on endpoints (Windows, Linux systems)
- **Server**: Central component that processes and analyzes data from agents
- **Elastic Stack integration**: For data storage (Elasticsearch), visualization (Kibana), and processing (Logstash)

3 ELK Stack

3.1 What is the ELK Stack?

The ELK Stack is a powerful combination of three open-source technologies (Elasticsearch, Logstash, and Kibana) that provides centralized logging and data analytics capabilities. In our SOC lab, it serves as the data processing backbone for Wazuh.

3.2 ELK Stack Components

The ELK implementation in our lab consists of:

- **Elasticsearch**: Distributed search and analytics engine that:
  - Stores and indexes security data
  - Enables fast search operations
  - Provides scalability for large datasets
- **Logstash**: Data processing pipeline that:
  - Ingests logs from multiple sources
  - Filters and normalizes log data
  - Enriches events with additional context
- **Kibana**: Visualization platform that:
  - Creates interactive dashboards
  - Provides security analytics
  - Enables investigation through the Wazuh plugin

4 PfSense and SNORT Integration

4.1 What are PfSense and SNORT?

PfSense is an open-source firewall and router platform, while SNORT is a powerful open-source intrusion detection/prevention system (IDS/IPS). In our SOC lab, they are integrated to provide comprehensive network security.

4.2 PfSense and SNORT Architecture

The integrated network security system consists of:

- **PfSense Firewall**:
  - Provides stateful packet inspection
  - Manages network address translation (NAT)
  - Offers VPN capabilities
  - Serves as traffic filtering gateway
- **SNORT IDS/IPS**:
  - Performs real-time traffic analysis
  - Detects various network attacks
  - Uses rule-based detection methods
  - Provides prevention capabilities (in IPS mode)

5 Zeek and Detection Scripts

5.1 What is Zeek?

Zeek (formerly Bro) is a powerful network analysis framework that provides comprehensive network traffic monitoring and metadata generation. Unlike traditional IDS solutions, Zeek functions primarily as a network traffic analyzer that requires custom scripts to implement detection capabilities.

5.2 Zeek Implementation

Our SOC lab deployment consists of:

- **Core Analysis Engine**:
  - Protocol analysis (HTTP, DNS, SSL/TLS, etc.)
  - Network flow reconstruction
  - Metadata generation (conn.log, http.log, etc.)
- **Custom Detection Scripts**:

  - Transform Zeek into detection capability with:
    * Signature-based threat detection
    * Behavioral anomaly scripts
    * Protocol violation checks
  - Generate security alerts in notice.log

6 Threat Intelligence & Automation

6.1 Shuffle

- **Purpose**: Open-source SOAR (Security Orchestration, Automation, and Response) platform
- **Implementation**:
  - Automates SOC workflows
  - Integrates with Wazuh, TheHive ...
  - Executes playbooks for incident response

6.2 TheHive

- **Purpose**: Incident response platform
- **Implementation**:
  - Case management for security incidents
  - Integration with Cortex for analysis
  - Synchronization with MISP for IOCs
  - Receives alerts from Wazuh

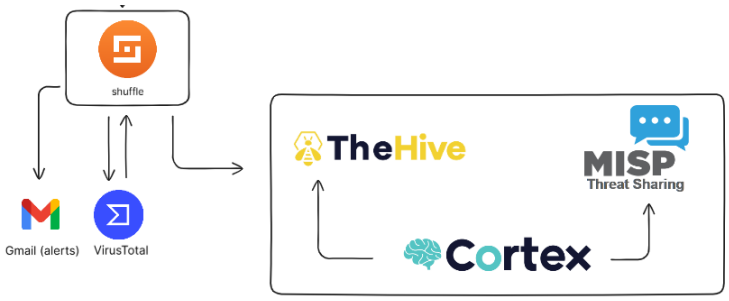
6.3 MISP

- **Purpose**: Threat intelligence sharing platform
- **Implementation**:
  - Stores and shares IOCs (Indicators of Compromise)
  - Integrates with TheHive for case enrichment

6.4 Cortex

- **Purpose**: Automated analysis engine
- **Implementation**:
  - Processes observables from TheHive
  - Returns enriched threat data
  - Supports custom analysis modules

6.5 Integration Flow



7 Lab Machines Architecture

7.1 Protected Network

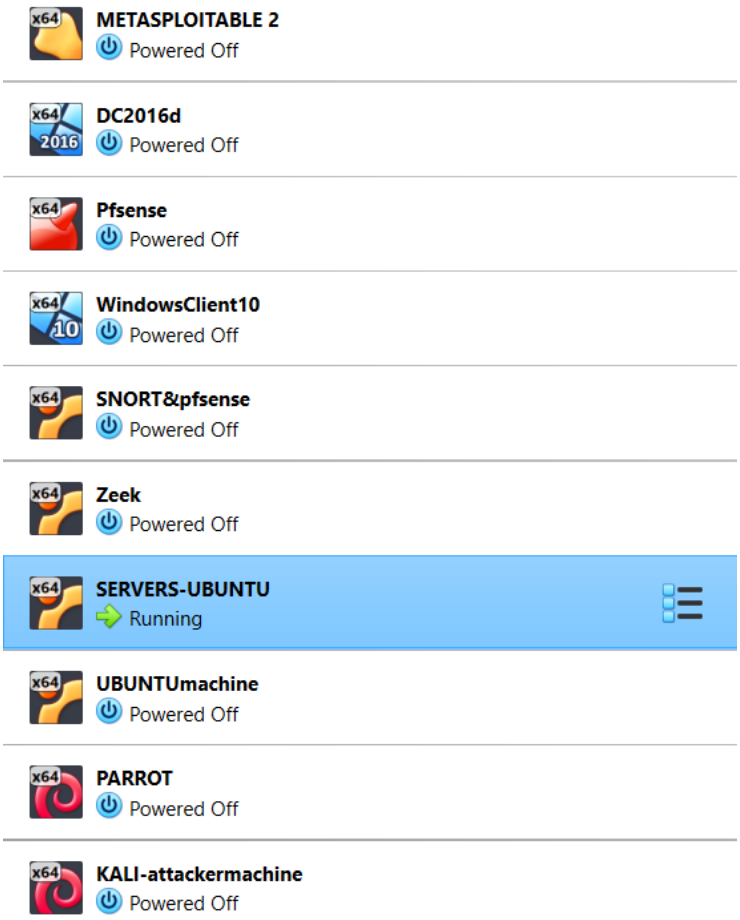
- **Domain Controller (Windows Server)**:
  - Runs Active Directory services
  - Hosts critical organizational resources
    - Wazuh agent with enhanced monitoring
    - Windows Defender ATP integration
    - Specialized audit policies
- **Ubuntu Workstation**:
  - Standard Linux endpoint for daily operations
    - Wazuh agent with:
      - System inventory monitoring
      - Vulnerability detection
      - Log collection (syslog/auth.log)
      - File integrity monitoring
      - Malware detection
- **Parrot Security**:
  - Used for defensive security testing
    - Wazuh agent with hardening
- **Windows Workstations**:
  - Standard employee endpoints
    - Wazuh agent with EDR features

7.2 Attacker Machine

- **External Kali Linux**:
  - Located outside protected network
  - Simulates external threats
  - Capabilities:
    - Penetration testing tools
    - C2 frameworks
    - Exploit development environment
  - Connection Methods:
    - VPN connections (for testing)
    - Simulated phishing campaigns
    - Web application attacks

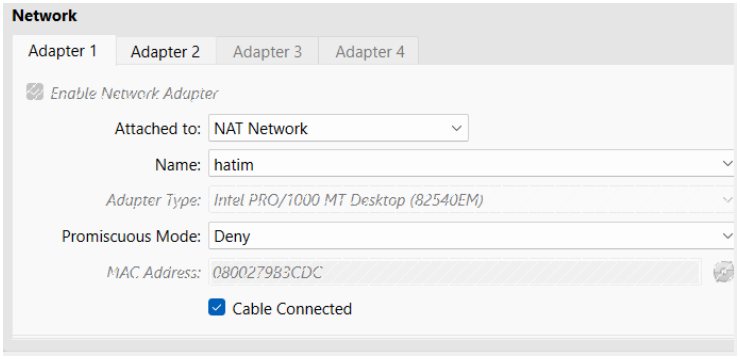
7.3 Network Segmentation

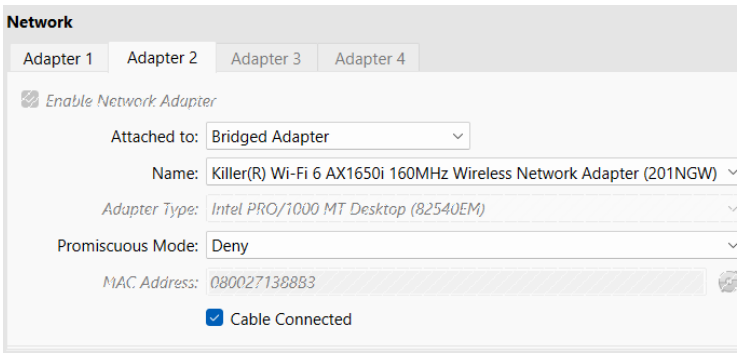
- **Isolation**:
  - Protected network behind PfSense firewall
  - IDS/IPS monitoring all ingress/egress
- **Monitoring**:
  - All internal traffic analyzed by Zeek
  - Wazuh agents report to central manager
  - SNORT as IDS and IPS.



8 VirtualBox Network Setup

The lab environment uses VirtualBox networking with two distinct configurations. For all internal protected machines (Domain Controller, Ubuntu, Parrot, and Windows workstations), I created a NAT network named "HATIM" to establish an isolated internal network ,also ,also a Bridged Adapter connected to the host's WiFi interface for wifi access. for wifi access.





9 PfSense Firewall Configuration

9.1 Network Setup

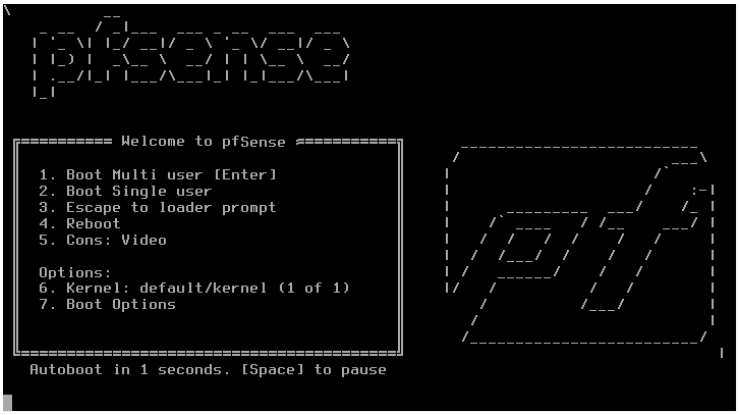
The PfSense firewall serves as the secure gateway .

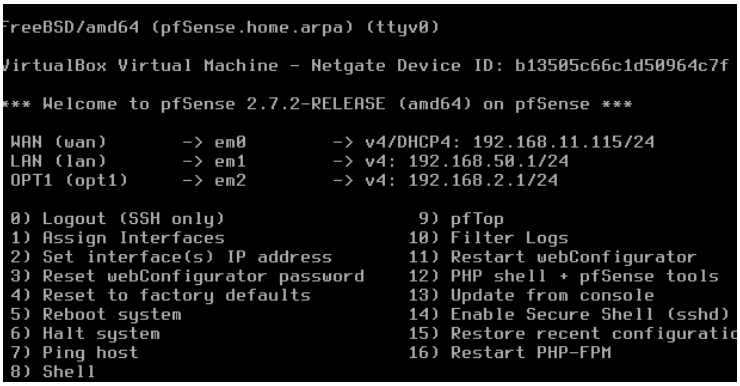
- **WAN Interface**:
  - Connected to VirtualBox (bridged adapter)
  - No inbound rules (default deny)
- **LAN Interface**:
  - Internal network "hatim" (192.168.50.0/24)
  - DHCP server enabled for lab machines
  - Default allow policy for outbound traffic

9.2 Security Rules

- **Firewall Rules**:
  - Block all inbound traffic from bridged network

– Allow internal communication between lab machines
– Restrict ICMP to internal network only



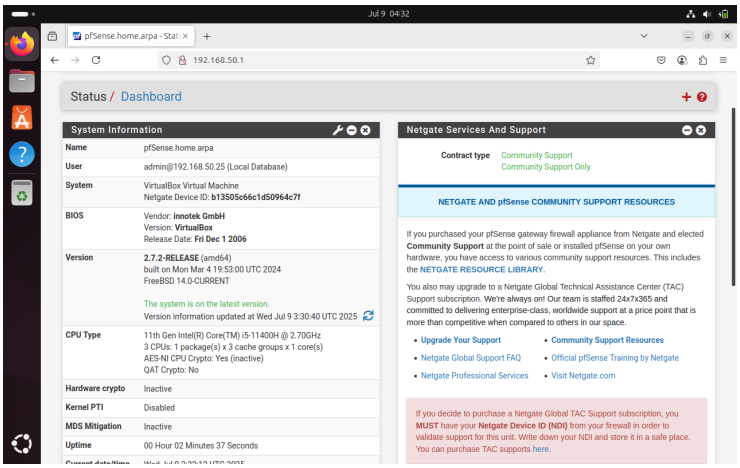


- **IDS/IPS Integration**:
  - Snort package installed and configured
  - Monitoring both WAN and LAN interfaces
  - Blocking known malicious patterns

10 Snort IDS/IPS Configuration in PfSense

10.1 Minimum Setup Procedure

1. **Package Installation**:
   - Access PfSense web interface
   - Install Snort package from Package Manager

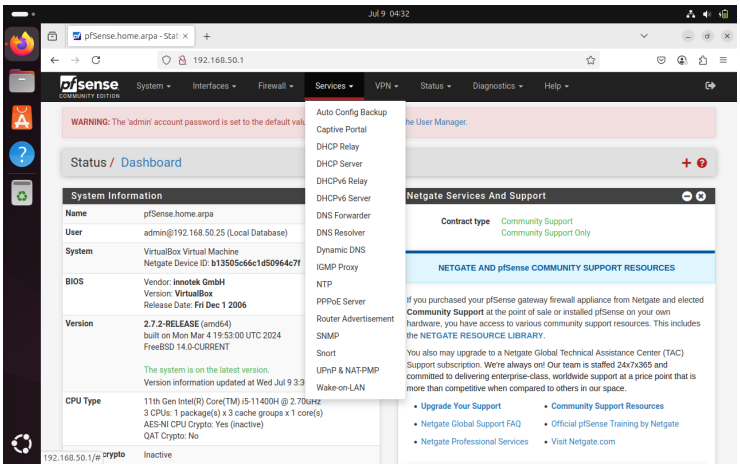


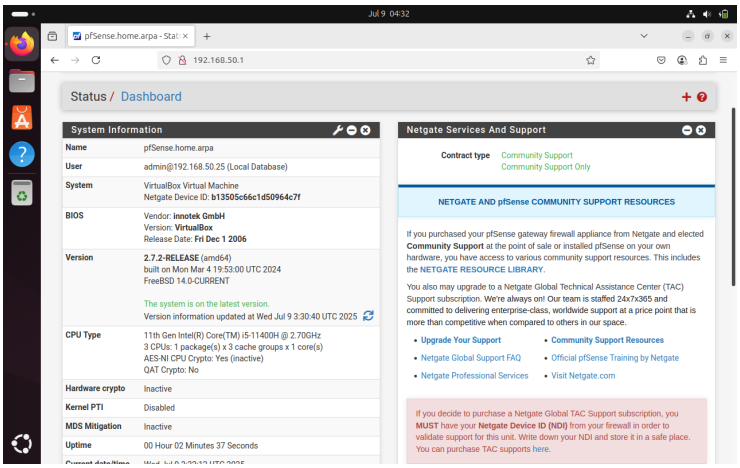
2. **Global Settings**:
   - Enable GPLv2 Community Ruleset
   - Set Rules Update Interval to 1 day
   - Set Block Host Interval to 1 day
3. **Rules Update**:
   - Navigate to Updates tab
   - Download latest rule sets
4. **Pass List Setup**:
   - Create new pass list
   - Save configuration

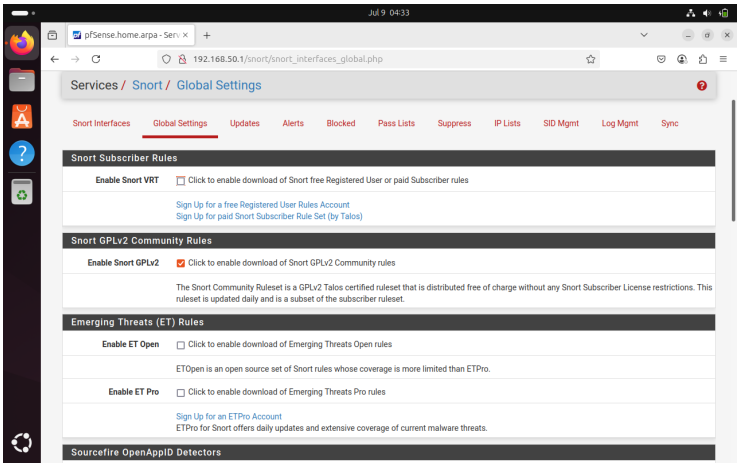
5. **Interface Configuration**:
    - Add both WAN and LAN interfaces
    - For each interface:
        - Check "Block Offenders" option
        - Enable GPLv2 Community Rules
        - Enable packet capture
        - Select created pass list in Home Net field

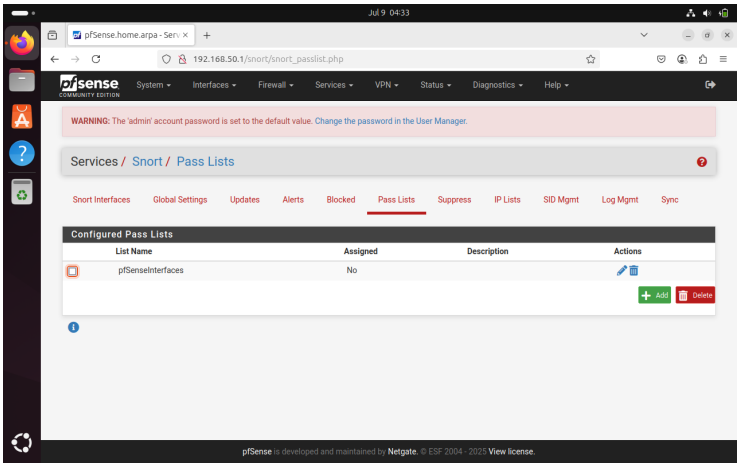
10.2 Verification

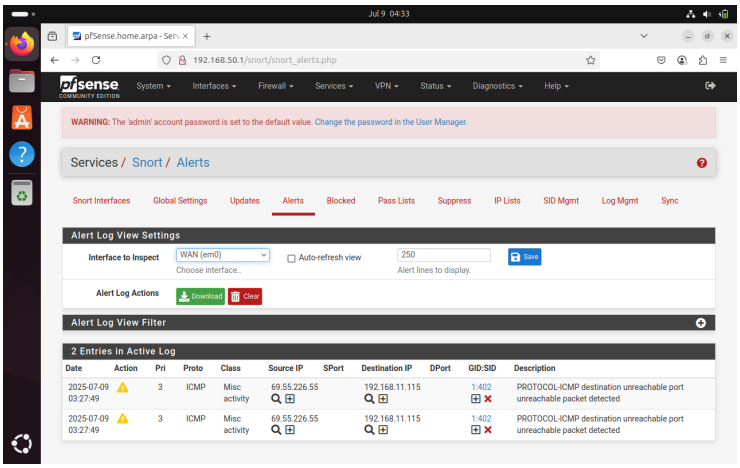
- Confirm Snort service is running
- Check Alerts tab for detected events
- Verify blocking functionality

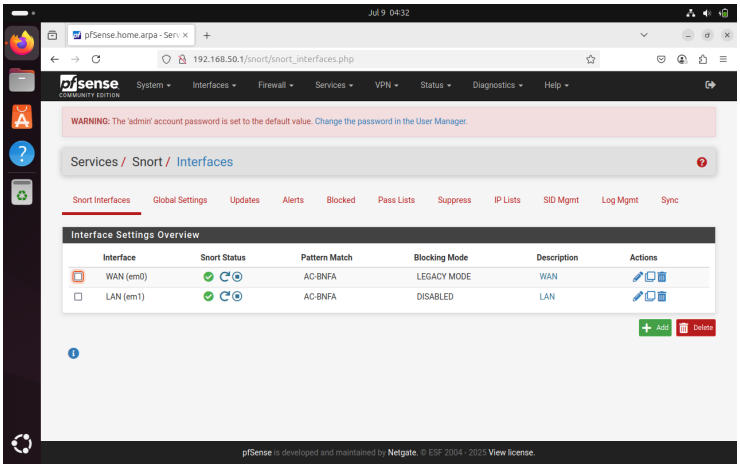



Jul 9 04:32
pfSense.home.arpa - Stat
192.168.50.1
Status / Dashboard
System Information
Name pfSense.home.arpa
User admin@192.168.50.25 (Local Database)
System VirtualBox Virtual Machine
Netgate Device ID: b13505c66c1d50964c7f
BIOS Vendor: innotek GmbH
Version: VirtualBox
Release Date: Fri Dec 1 2006
Version 2.7.2-RELEASE (amd64)
built on Mon Mar 4 19:53:00 UTC 2024
FreeBSD 14.0-CURRENT
The system is on the latest version.
Version information updated at Wed Jul 9 3:30:40 UTC 2025
CPU Type 11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz
3 CPUs: 1 package(s) x 3 cache groups x 1 core(s)
AES-NI CPU Crypto: Yes (inactive)
QAT Crypto: No
Hardware crypto Inactive
Kernel PTI Disabled
MDS Mitigation Inactive
Uptime 00 Hour 02 Minutes 37 Seconds
Netgate Services And Support
Contract type Community Support
Community Support Only
NETGATE AND pfSense COMMUNITY SUPPORT RESOURCES
If you purchased your pfSense gateway firewall appliance from Netgate and elected Community Support at the point of sale or installed pfSense on your own hardware, you have access to various community support resources. This includes the NETGATE RESOURCE LIBRARY.
You also may upgrade to a Netgate Global Technical Assistance Center (TAC) Support subscription. We're always on! Our team is staffed 24x7x365 and committed to delivering enterprise-class, worldwide support at a price point that is more than competitive when compared to others in our space.
Upgrade Your Support
Community Support Resources
Netgate Global Support FAQ
Official pfSense Training by Netgate
Netgate Professional Services
Visit Netgate.com
If you decide to purchase a Netgate Global TAC Support subscription, you MUST have your Netgate Device ID (NDI) from your firewall in order to validate support for this unit. Write down your NDI and store it in a safe place. You can purchase TAC supports here.


Jul 9 04:33
pfSense.home.arpa - Serv
192.168.50.1/snort/snort_interfaces_global.php
Services / Snort / Global Settings
Snort Interfaces
Global Settings
Updates
Alerts
Blocked
Pass Lists
Suppress
IP Lists
SID Mgmt
Log Mgmt
Sync
Snort Subscriber Rules
Enable Snort VRT
Click to enable download of Snort free Registered User or paid Subscriber rules
Sign Up for a free Registered User Rules Account
Sign Up for paid Snort Subscriber Rule Set (by Talos)
Snort GPLv2 Community Rules
Enable Snort GPLv2
Click to enable download of Snort GPLv2 Community rules
The Snort Community Ruleset is a GPLv2 Talos certified ruleset that is distributed free of charge without any Snort Subscriber License restrictions. This ruleset is updated daily and is a subset of the subscriber ruleset.
Emerging Threats (ET) Rules
Enable ET Open
Click to enable download of Emerging Threats Open rules
ETOpen is an open source set of Snort rules whose coverage is more limited than ETPro.
Enable ET Pro
Click to enable download of Emerging Threats Pro rules
Sign Up for an ETPro Account
ETPro for Snort offers daily updates and extensive coverage of current malware threats.
Sourcefire OpenAppID Detectors


Jul 9 04:33
pfSense.home.arpa - Serv
192.168.50.1/snort/snort_passlist.php
pfSense COMMUNITY EDITION
System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager.
Services / Snort / Pass Lists
Snort Interfaces
Global Settings
Updates
Alerts
Blocked
Pass Lists
Suppress
IP Lists
SID Mgmt
Log Mgmt
Sync
Configured Pass Lists
List Name
Assigned
Description
Actions
pfSenseInterfaces
No
Add
Delete
pfSense is developed and maintained by Netgate. © ESF 2004 - 2025 View license.


Jul 9 04:33
pfSense.home.arpa - Serv
192.168.50.1/snort/snort_alerts.php
System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager.
Services / Snort / Alerts
Snort Interfaces
Global Settings
Updates
Alerts
Blocked
Pass Lists
Suppress
IP Lists
SID Mgmt
Log Mgmt
Sync
Alert Log View Settings
Interface to Inspect
WAN (em0)
Choose interface..
Auto-refresh view
250
Alert lines to display.
Save
Alert Log Actions
Download
Clear
Alert Log View Filter
2 Entries in Active Log
Date
Action
Pri
Proto
Class
Source IP
SPort
Destination IP
DPort
GID:SID
Description
2025-07-09 03:27:49
3
ICMP
Misc activity
69.55.226.55
192.168.11.115
1:402
PROTOCOL-ICMP destination unreachable port unreachable packet detected
2025-07-09 03:27:49
3
ICMP
Misc activity
69.55.226.55
192.168.11.115
1:402
PROTOCOL-ICMP destination unreachable port unreachable packet detected


Jul 9 04:32
pfSense.home.arpa - Serv
192.168.50.1/snort/snort_interfaces.php
System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager.
Services / Snort / Interfaces
Snort Interfaces
Global Settings
Updates
Alerts
Blocked
Pass Lists
Suppress
IP Lists
SID Mgmt
Log Mgmt
Sync
Interface Settings Overview
Interface
Snort Status
Pattern Match
Blocking Mode
Description
Actions
WAN (em0)
AC-BNFA
LEGACY MODE
WAN
LAN (em1)
AC-BNFA
DISABLED
LAN
Add
Delete
pfSense is developed and maintained by Netgate. © ESF 2004 - 2025 View license.

11 Wazuh Configuration

The Wazuh server is co-installed with the ELK stack (Elasticsearch, Logstash, and Kibana) on a single Ubuntu machine, creating an all-in-one security monitoring solution.

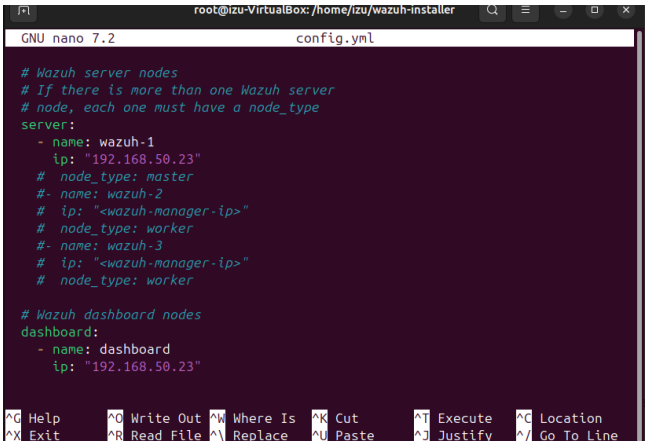
11.1 Manual Installation

11.1.1 Certificate Setup

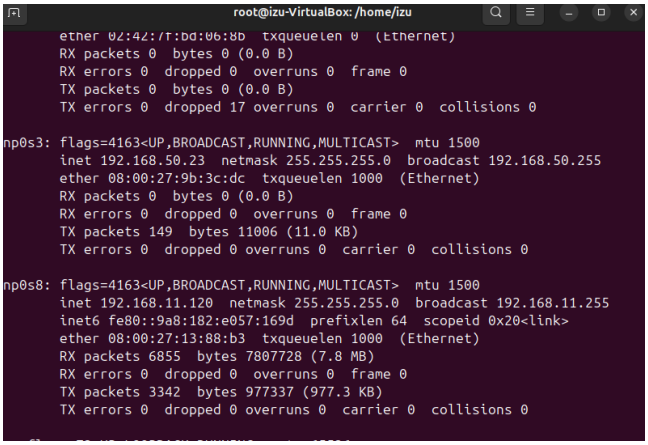
```
# Create working directory
mkdir wazuh-installer && cd wazuh-installer

# Download certificate tools
curl -sO https://packages.wazuh.com/4.7/wazuh-certs-tool.sh
curl -sO https://packages.wazuh.com/4.7/config.yml
```

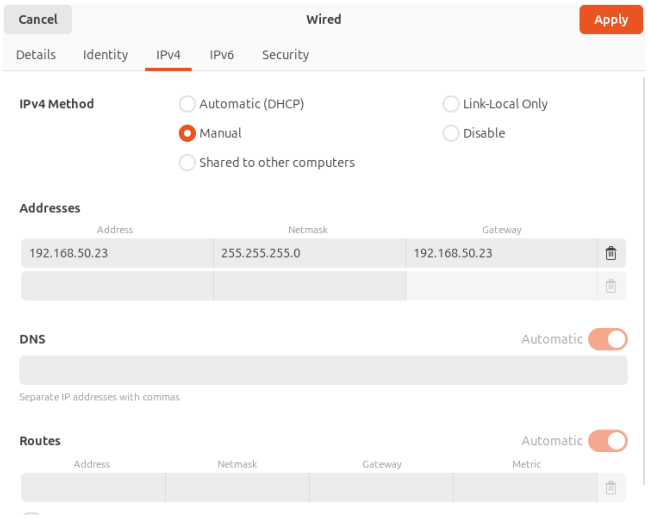
Edit ./config.yml and replace the node names and IP values with the corresponding names and IP addresses. You need to do this for all Wazuh server, Wazuh indexer, and Wazuh dashboard nodes. Add as many node fields as needed.



Use the ip a command to retrieve your server's IP . mine is 192.168.50.23



To ensure stable operation and avoid connection issues, I configured a static IP address for the Wazuh server before beginning the installation.




```
# Generate certificates (after editing config.yml)
bash ./wazuh-certs-tool.sh -A

# Package certificates
tar -cvf wazuh-certificates.tar -C ./wazuh-certificates/ .
rm -rf ./wazuh-certificates
```

11.2 Nodes Installation

11.2.1 Install Package Dependencies

```
apt-get install debconf adduser procps
```

11.2.2 Add Wazuh Repository

1. Install prerequisite packages:

   ```
   apt-get install gnupg apt-transport-https
   ```

2. Import Wazuh GPG key:

   ```
   curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | \
   gpg --no-default-keyring \
   --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg \
   --import && chmod 644 /usr/share/keyrings/wazuh.gpg
   ```

3. Add repository to sources list:

   ```
   echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] \
   https://packages.wazuh.com/4.x/apt/ stable main" | \
   tee -a /etc/apt/sources.list.d/wazuh.list
   ```

4. Update package information:

   ```
   apt-get update
   ```

12 Wazuh Indexer Installation and Configuration

12.1 Package Installation

```
apt-get -y install wazuh-indexer
```

12.2 Configuration

Edit `/etc/wazuh-indexer/opensearch.yml` with these critical settings:

- `network.host`: your-node-ip (matches config.yml) mine is 192.168.50.23
- `node.name`: node-1 (as defined in config.yml)
- `cluster.initial_master_nodes`: ["node-1"]

12.3 Certificate Deployment

```
NODE_NAME=node-1
mkdir /etc/wazuh-indexer/certs
tar -xf ./wazuh-certificates.tar -C /etc/wazuh-indexer/certs/ \
./$NODE_NAME.pem ./$NODE_NAME-key.pem ./admin.pem ./admin-key.pem ./root-ca.pem
mv -n /etc/wazuh-indexer/certs/$NODE_NAME.pem /etc/wazuh-indexer/certs/indexer.pem
mv -n /etc/wazuh-indexer/certs/$NODE_NAME-key.pem /etc/wazuh-indexer/certs/indexer-key.pem
chmod 500 /etc/wazuh-indexer/certs
chmod 400 /etc/wazuh-indexer/certs/*
chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs
```

12.4 Service Management

```
systemctl daemon-reload
systemctl enable wazuh-indexer
systemctl start wazuh-indexer
systemctl status wazuh-indexer  # Verify running status
```

12.5 Cluster Initialization

```
/usr/share/wazuh-indexer/bin/indexer-security-init.sh
```

12.6 Verification

Test the installation:

```
curl -k -u admin:admin https://<WAZUH_INDEXER_IP>:9200
curl -k -u admin:admin https://<WAZUH_INDEXER_IP>:9200/_cat/nodes?v
```

- **Security Note**: Remove certificate archive after deployment:

  ```
  rm -f ./wazuh-certificates.tar
  ```

- **Success Indicator**: Both curl commands should return valid JSON responses showing node status

13 Wazuh Server Installation

13.1 Overview

The Wazuh server processes agent data, generates security alerts, and manages agent configurations. With the indexer installed, proceed with server setup.

Wazuh server installation process is divided into two stages.

-Wazuh server node installation -Cluster configuration for multi-node deployment

13.2 Installation Process

1. **Install Wazuh Manager**:

   ```
   apt-get -y install wazuh-manager
   ```

2. **Enable and Start Services**:

   ```
   systemctl daemon-reload
   systemctl enable wazuh-manager
   systemctl start wazuh-manager
   ```

3. **Verify Installation**:

   ```
   systemctl status wazuh-manager
   ```

   Expected output should show `active (running)` status.

14 Filebeat Installation and Configuration

14.1 Installation

```
apt-get -y install filebeat
```

14.2 Configuration

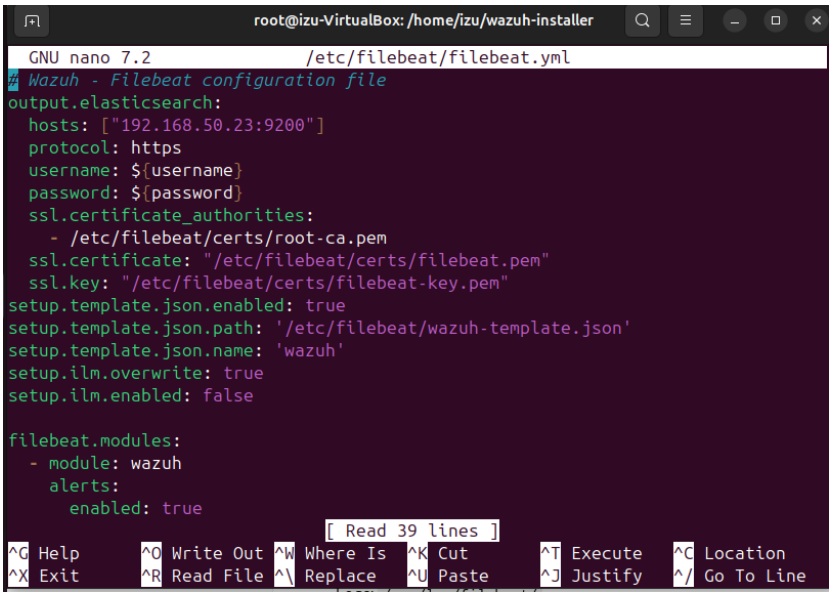
1. Download configuration template:

   ```
   curl -so /etc/filebeat/filebeat.yml \
   https://packages.wazuh.com/4.7/tpl/wazuh/filebeat/filebeat.yml
   ```

2. Edit `/etc/filebeat/filebeat.yml`:

   - Set `hosts: ["<your-indexer-ip>:9200"]`
   - Uncomment and configure:

     ```
     protocol: https
     username: ${username}
     password: ${password}
     ```



3. Secure credentials:

```
filebeat keystore create
echo admin | filebeat keystore add username --stdin --force
echo admin | filebeat keystore add password --stdin --force
```

4. Download Wazuh resources:

```
curl -so /etc/filebeat/wazuh-template.json \
https://raw.githubusercontent.com/wazuh/wazuh/v4.7.2/extensions/elasticsearch/
7.x/wazuh-template.json
chmod go+r /etc/filebeat/wazuh-template.json

curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.3.tar.gz |
\
tar -xvz -C /usr/share/filebeat/module
```

14.3 Certificate Deployment

```
mkdir /etc/filebeat/certs
tar -xf ./wazuh-certificates.tar -C /etc/filebeat/certs/ \
./$NODE_NAME.pem ./$NODE_NAME-key.pem ./root-ca.pem
mv -n /etc/filebeat/certs/$NODE_NAME.pem /etc/filebeat/certs/filebeat.pem
mv -n /etc/filebeat/certs/$NODE_NAME-key.pem /etc/filebeat/certs/filebeat-key.pem
```

```
chmod 500 /etc/filebeat/certs
chmod 400 /etc/filebeat/certs/*
chown -R root:root /etc/filebeat/certs
```

14.4 Service Management

```
systemctl daemon-reload
systemctl enable filebeat
systemctl start filebeat
filebeat test output  # Verify connection
```

14.5 Troubleshooting

- **SSL Errors**: Expected with self-signed certificates
- **Verification**: Check service status with:

  ```
  journalctl -u filebeat --no-pager -n 50
  ```

- **Success**: Service shows "active (running)" status

15 Wazuh Dashboard Installation

15.1 Overview

The Wazuh dashboard provides a web-based interface for:

- Security event visualization
- Vulnerability monitoring
- File integrity analysis
- Configuration assessment
- Cloud infrastructure monitoring
- Compliance reporting

15.2 Installation Process

1. **Install Dependencies**:

   ```
   apt-get install debhelper tar curl libcap2-bin
   ```

2. **Install Dashboard Package**:

```
apt-get -y install wazuh-dashboard
```

3. **Configuration**: Edit `/etc/wazuh-dashboard/opensearch_dashboards.yml`:
   - Set `server.host: "0.0.0.0"`
   - Configure `opensearch.hosts: ["https://<your-indexer-ip>:9200"]`
4. **Certificate Deployment**:

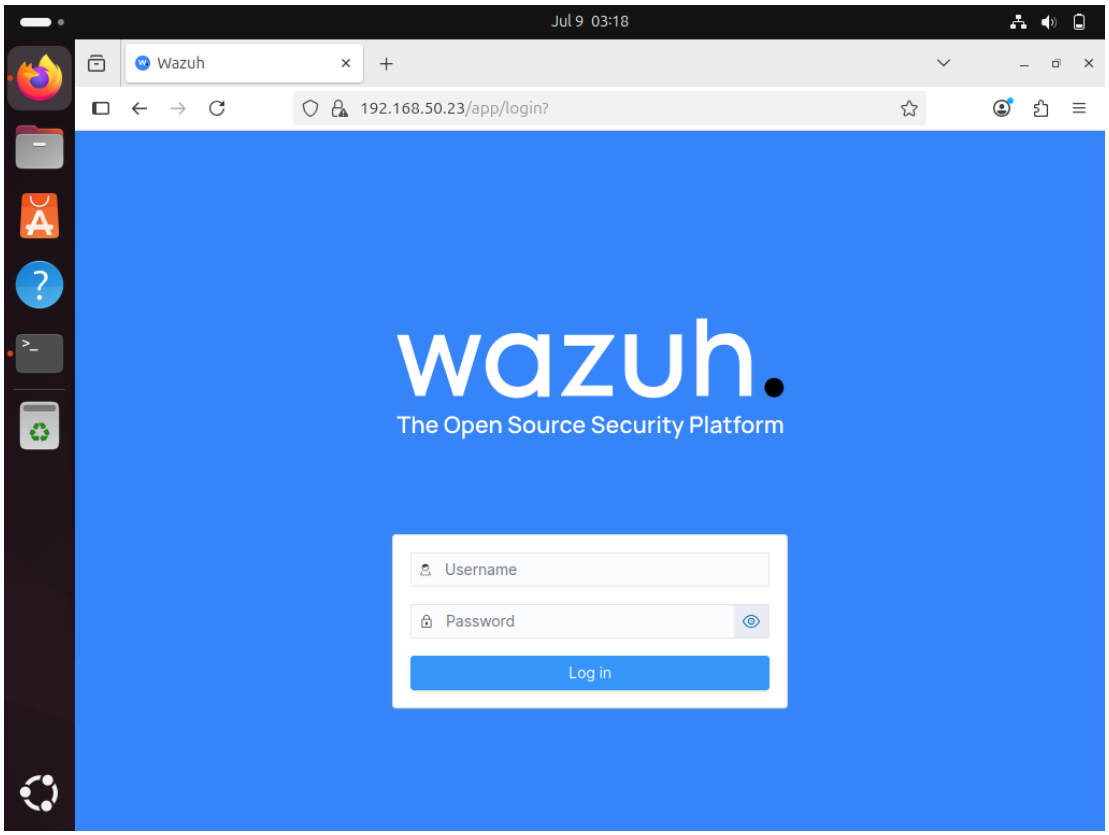
```
mkdir /etc/wazuh-dashboard/certs
tar -xf ./wazuh-certificates.tar -C /etc/wazuh-dashboard/certs/ \
./$NODE_NAME.pem ./$NODE_NAME-key.pem ./root-ca.pem
mv /etc/wazuh-dashboard/certs/$NODE_NAME.pem \
/etc/wazuh-dashboard/certs/dashboard.pem
mv /etc/wazuh-dashboard/certs/$NODE_NAME-key.pem \
/etc/wazuh-dashboard/certs/dashboard-key.pem
chmod 500 /etc/wazuh-dashboard/certs
chmod 400 /etc/wazuh-dashboard/certs/*
chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs
```

5. **Service Management**:

```
systemctl daemon-reload
systemctl enable wazuh-dashboard
systemctl start wazuh-dashboard
```

15.3 Accessing the Dashboard

- URL: `https://<your-server-ip>`
- Default credentials: admin/admin



15.4 Troubleshooting

- **Missing Alerts Index**:

```
curl https://raw.githubusercontent.com/wazuh/wazuh/v4.5.2/extensions/
elasticsearch/7.x/wazuh-template.json | \
curl -X PUT "https://<your-ip>:9200/_template/wazuh" \
-H 'Content-Type: application/json' -d @- -u admin:admin -k
```

- **SSL Errors**: Expected with self-signed certificates

15.5 Security Hardening

```
/usr/share/wazuh-indexer/plugins/opensearch-security/tools/\
wazuh-passwords-tool.sh --change-all \
--admin-user <new-user> --admin-password <new-password>
```

- Store credentials securely
- Consider IP whitelisting for dashboard access

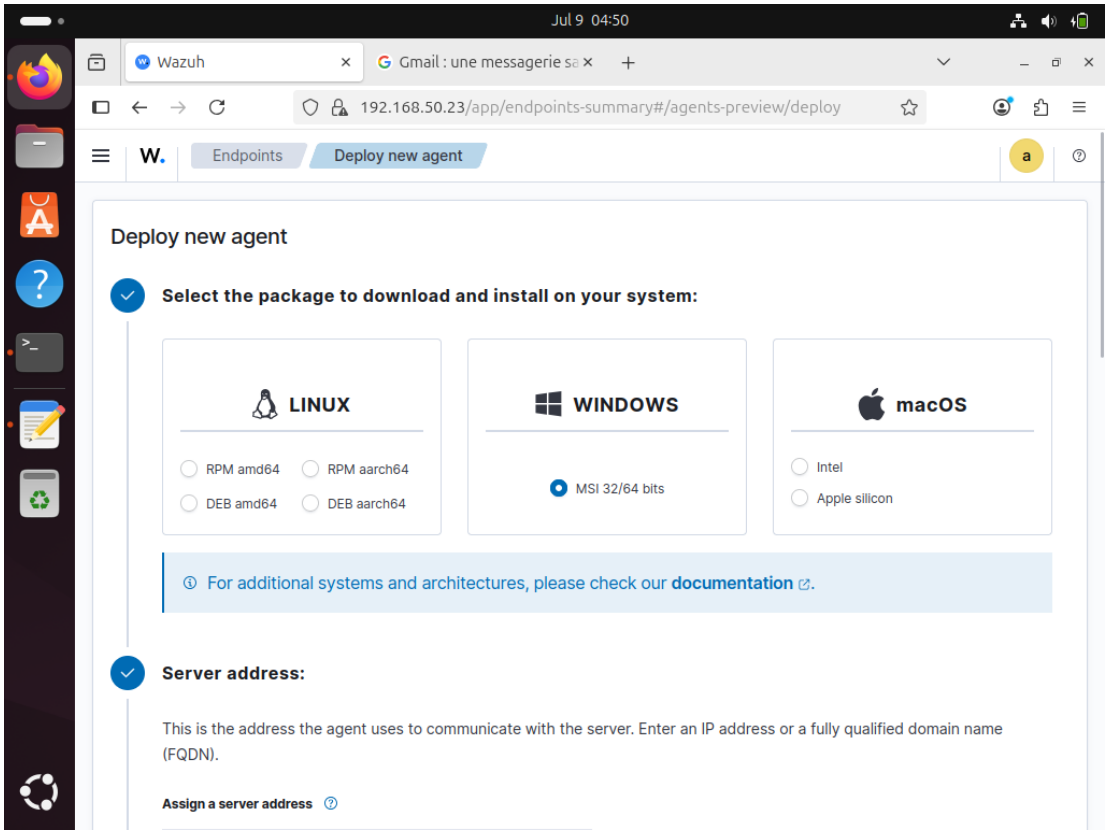
16 Adding agents

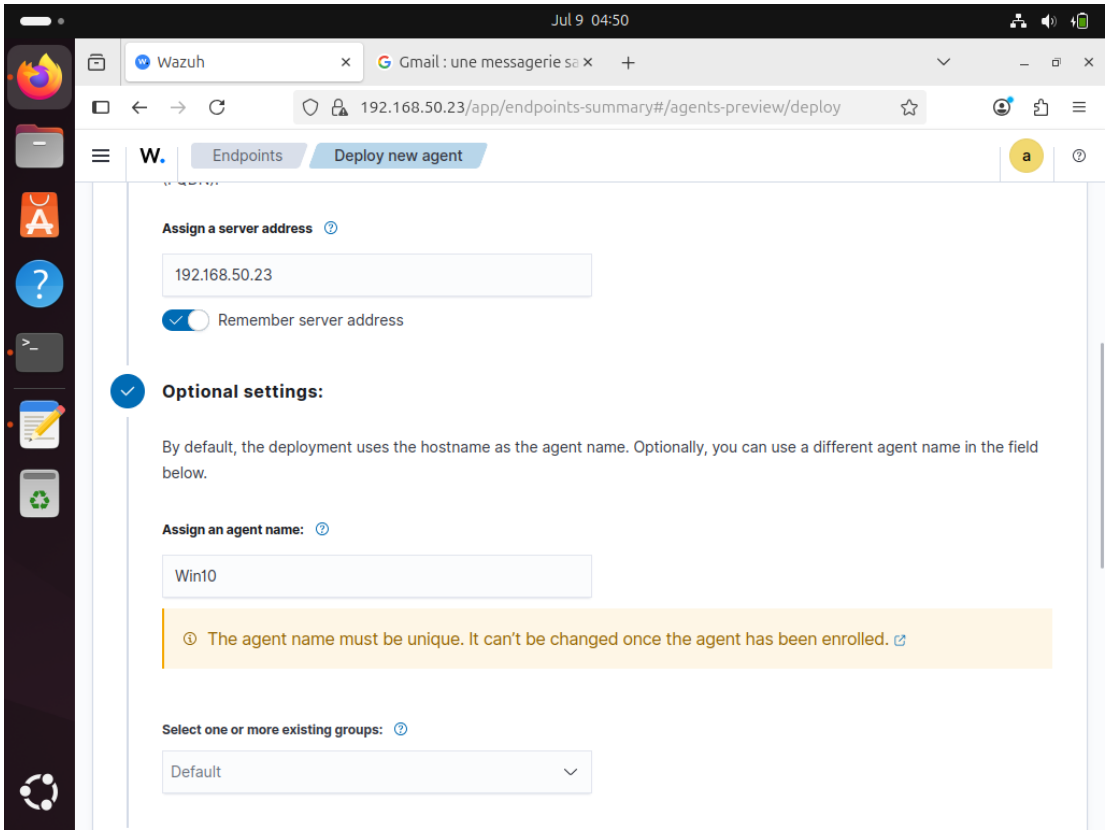
Before agent installation, ensure:

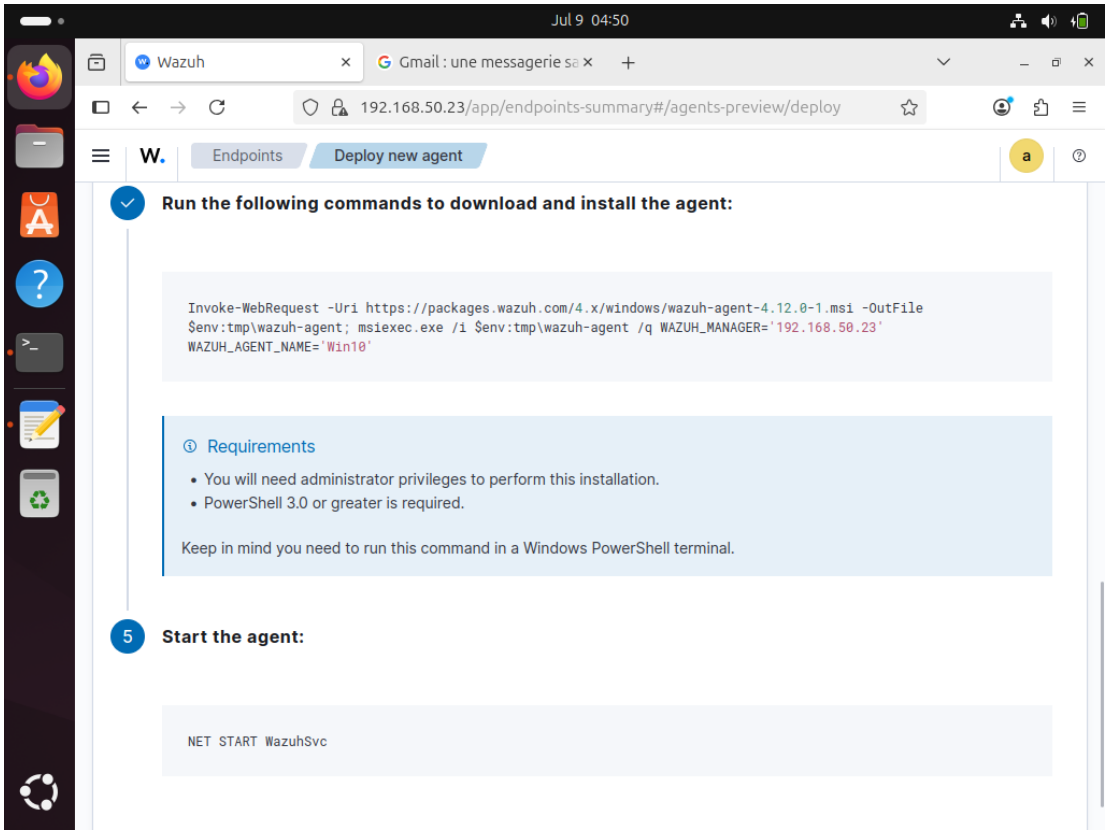
- Network connectivity exists between agent and server

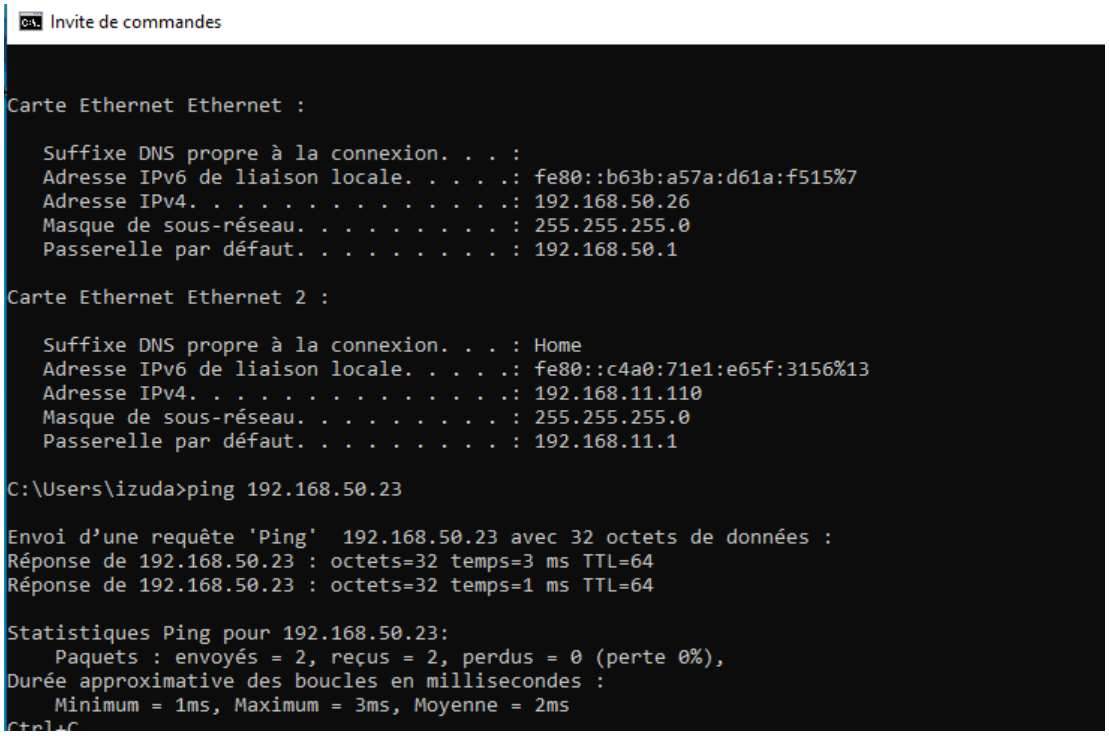
16.1 Note :

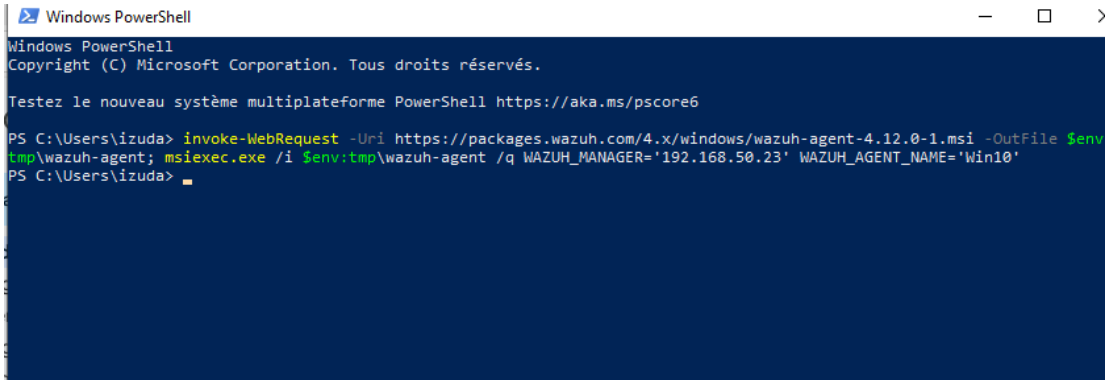
The same agent installation process applies to Ubuntu workstations, Parrot Security OS, and Windows Domain Controllers, with packages available through apt for Linux systems and MSI installers for Windows, all requiring connectivity to the Wazuh server.

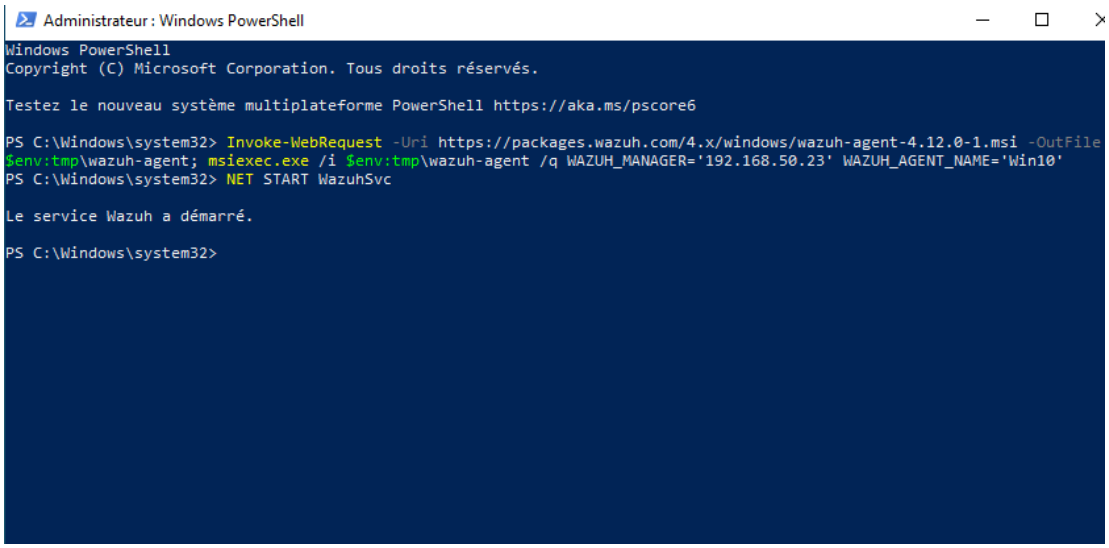

Jul 9 04:50
Wazuh
Gmail : une messagerie sa
192.168.50.23/app/endpoints-summary#/agents-preview/deploy
W.
Endpoints
Deploy new agent
a
Deploy new agent
Select the package to download and install on your system:
LINUX
RPM amd64
RPM aarch64
DEB amd64
DEB aarch64
WINDOWS
MSI 32/64 bits
macOS
Intel
Apple silicon
For additional systems and architectures, please check our documentation.
Server address:
This is the address the agent uses to communicate with the server. Enter an IP address or a fully qualified domain name (FQDN).
Assign a server address


Jul 9 04:50
Wazuh
Gmail : une messagerie sa
192.168.50.23/app/endpoints-summary#/agents-preview/deploy
W.
Endpoints
Deploy new agent
a
Assign a server address
192.168.50.23
Remember server address
Optional settings:
By default, the deployment uses the hostname as the agent name. Optionally, you can use a different agent name in the field below.
Assign an agent name:
Win10
The agent name must be unique. It can't be changed once the agent has been enrolled.
Select one or more existing groups:
Default


Jul 9 04:50
Wazuh
Gmail : une messagerie sa
192.168.50.23/app/endpoints-summary#/agents-preview/deploy
W.
Endpoints
Deploy new agent
a
Run the following commands to download and install the agent:
Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.12.0-1.msi -OutFile $env:tmp\wazuh-agent; msiexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='192.168.50.23' WAZUH_AGENT_NAME='Win10'
Requirements
You will need administrator privileges to perform this installation.
PowerShell 3.0 or greater is required.
Keep in mind you need to run this command in a Windows PowerShell terminal.
5
Start the agent:
NET START WazuhSvc


Invite de commandes
Carte Ethernet Ethernet :
Suffixe DNS propre à la connexion. . . :
Adresse IPv6 de liaison locale. . . . .: fe80::b63b:a57a:d61a:f515%7
Adresse IPv4. . . . . .: 192.168.50.26
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . : 192.168.50.1
Carte Ethernet Ethernet 2 :
Suffixe DNS propre à la connexion. . . : Home
Adresse IPv6 de liaison locale. . . . .: fe80::c4a0:71e1:e65f:3156%13
Adresse IPv4. . . . . .: 192.168.11.110
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . : 192.168.11.1
C:\Users\izuda>ping 192.168.50.23
Envoi d'une requête 'Ping' 192.168.50.23 avec 32 octets de données :
Réponse de 192.168.50.23 : octets=32 temps=3 ms TTL=64
Réponse de 192.168.50.23 : octets=32 temps=1 ms TTL=64
Statistiques Ping pour 192.168.50.23:
Paquets : envoyés = 2, reçus = 2, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
Minimum = 1ms, Maximum = 3ms, Moyenne = 2ms
Ctrl+C


Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.
Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6
PS C:\Users\izuda> invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.12.0-1.msi -OutFile $env
tmp\wazuh-agent; msiexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='192.168.50.23' WAZUH_AGENT_NAME='Win10'
PS C:\Users\izuda>


Administrateur : Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.
Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6
PS C:\Windows\system32> Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.12.0-1.msi -OutFile $env:tmp\wazuh-agent; msiexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='192.168.50.23' WAZUH_AGENT_NAME='Win10'
PS C:\Windows\system32> NET START WazuhSvc
Le service Wazuh a démarré.
PS C:\Windows\system32>

17 ELK Stack Installation on Ubuntu

17.1 Repository Setup

```
sudo apt update
sudo apt install gnupg2 apt-transport-https -y
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | \
sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/elasticsearch-keyring.gpg
echo "deb https://artifacts.elastic.co/packages/8.x/apt stable main" | \
sudo tee /etc/apt/sources.list.d/elk-8.list
sudo apt update
```

17.2 Elasticsearch Installation

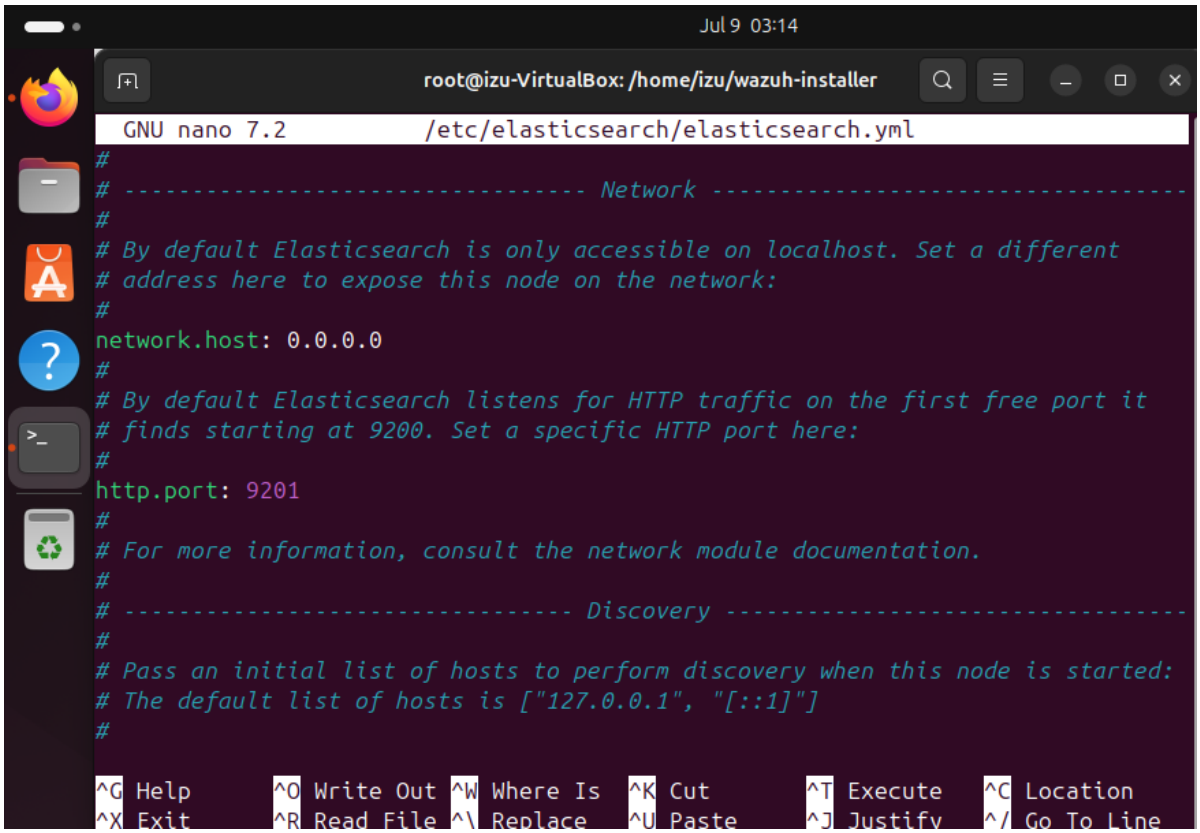
```
sudo apt install elasticsearch -y
```

Key post-install notes:

- Security auto-configured with TLS
- Default superuser 'elastic' with generated password (try to take a note for your password)
- Single-node cluster configuration

17.3 Configuration

1. Edit `/etc/elasticsearch/elasticsearch.yml`:
    - Set `network.host: 0.0.0.0`
    - Verify `cluster.initial_master_nodes`
- Modified default port from 9200 to 9201
- Prevents conflict with Wazuh indexer (uses 9200)
- Ensures both services run simultaneously

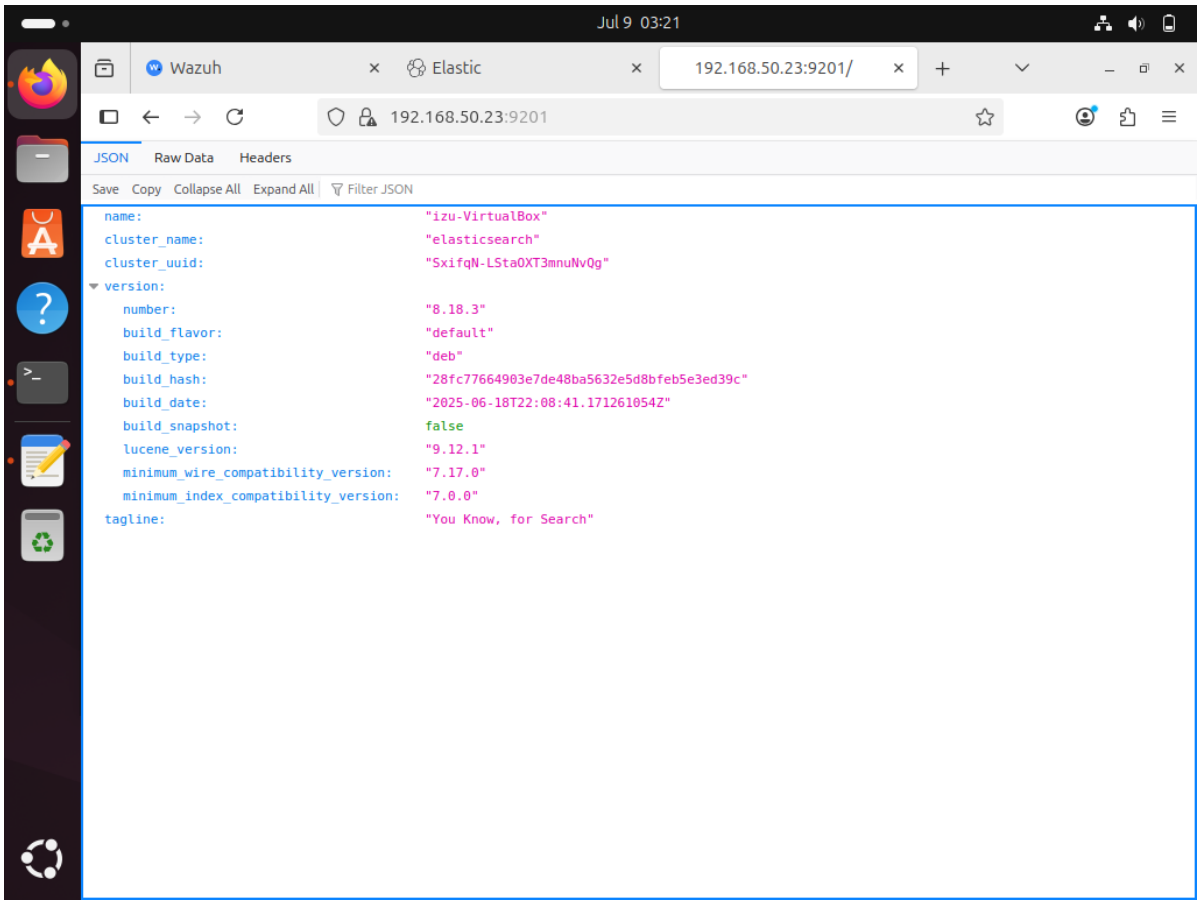

Jul 9 03:14
root@izu-VirtualBox: /home/izu/wazuh-installer
GNU nano 7.2 /etc/elasticsearch/elasticsearch.yml
---------- Network ----------
By default Elasticsearch is only accessible on localhost. Set a different
address here to expose this node on the network:
network.host: 0.0.0.0
By default Elasticsearch listens for HTTP traffic on the first free port it
finds starting at 9200. Set a specific HTTP port here:
http.port: 9201
For more information, consult the network module documentation.
---------- Discovery ----------
Pass an initial list of hosts to perform discovery when this node is started:
The default list of hosts is ["127.0.0.1", "[::1]"]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

17.4 Service Management

```
sudo systemctl daemon-reload
sudo systemctl enable --now elasticsearch.service
```

17.5 Verification

```
curl -k -XGET https://localhost:9201 -u elastic \
--cacert /etc/elasticsearch/certs/http_ca.crt
systemctl status elasticsearch
sudo journalctl -u elasticsearch -f
```



18 Kibana Configuration

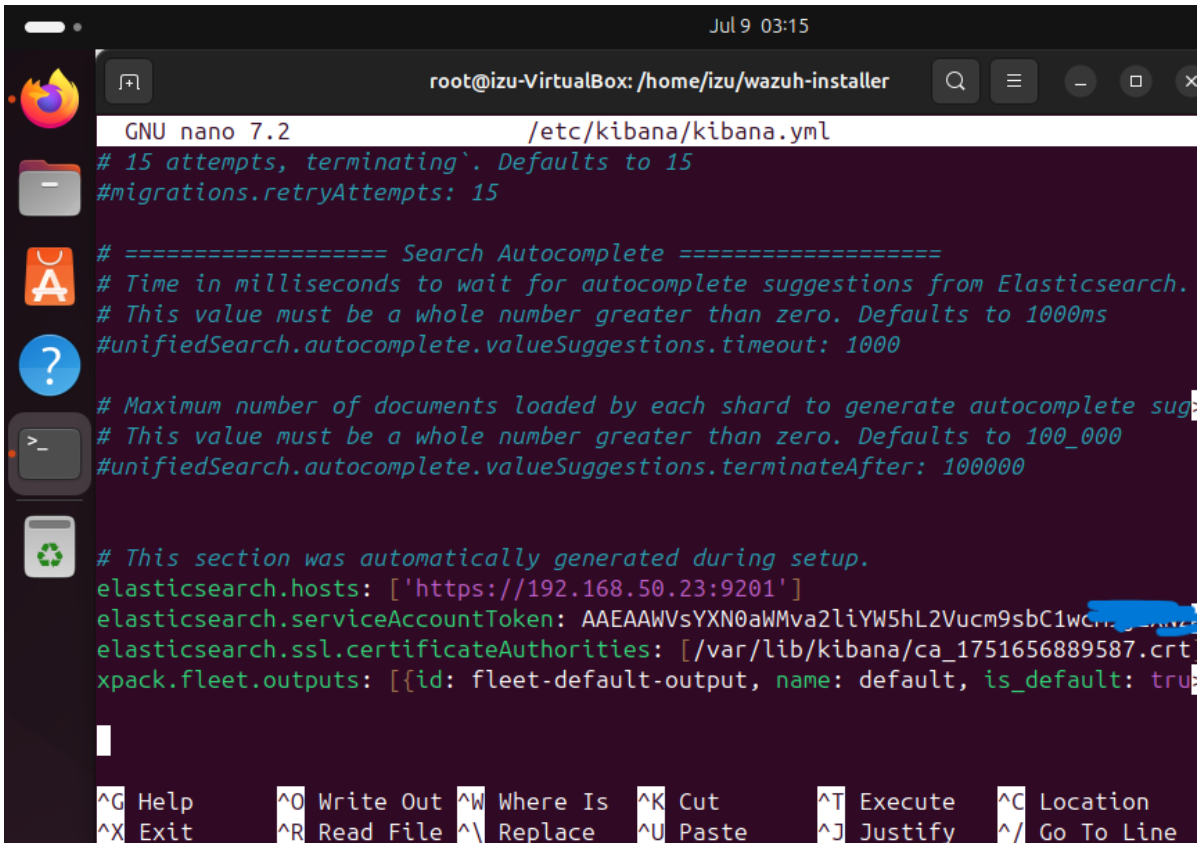
18.1 Network Settings

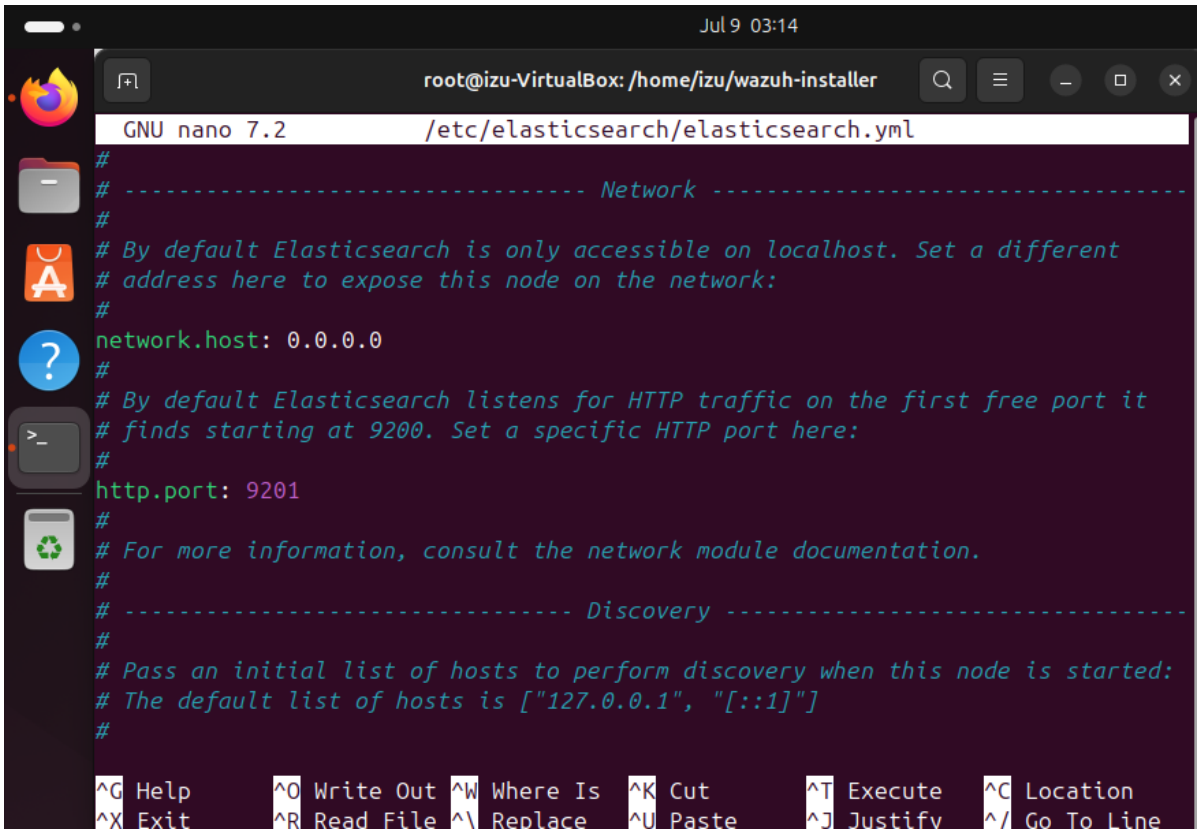
- Edit `/etc/kibana/kibana.yml`:

```
server.host: "192.168.50.23"  # Replace with your server IP
```

- Allow external access on port 5601:

```
sudo ufw allow 5601/tcp
```




Jul 9 03:14
root@izu-VirtualBox: /home/izu/wazuh-installer
GNU nano 7.2 /etc/elasticsearch/elasticsearch.yml
---------- Network ----------
By default Elasticsearch is only accessible on localhost. Set a different
address here to expose this node on the network:
network.host: 0.0.0.0
By default Elasticsearch listens for HTTP traffic on the first free port it
finds starting at 9200. Set a specific HTTP port here:
http.port: 9201
For more information, consult the network module documentation.
---------- Discovery ----------
Pass an initial list of hosts to perform discovery when this node is started:
The default list of hosts is ["127.0.0.1", "[::1]"]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

18.2 Service Management

```
sudo systemctl enable --now kibana
systemctl status kibana  # Verify running status
sudo journalctl -u kibana -f  # View logs
```

subsectionToken Generation To establish a secure connection between Kibana and Elasticsearch:

```
/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana
```

Sample Output :

19 Accessing Kibana Web Interface

19.1 Port Verification

```
ss -altnp | grep :5601
```

Expected output:

```
LISTEN 0 511 192.168.50.23:5601 0.0.0.0:* users:(("node",pid=4108,fd=18))
```

19.2 Access Instructions

1. Retrieve the activation URL:

   ```
   sudo journalctl -u kibana
   ```

   Pay attention to line; Go to http://192.168.50.23:5601/?code=xxxxx

2. Or get verification code:

   ```
   /usr/share/kibana/bin/kibana-verification-code
   ```

3. Access in browser:
   - URL: `http://<your-server-ip>:5601`
   - Default port: 5601

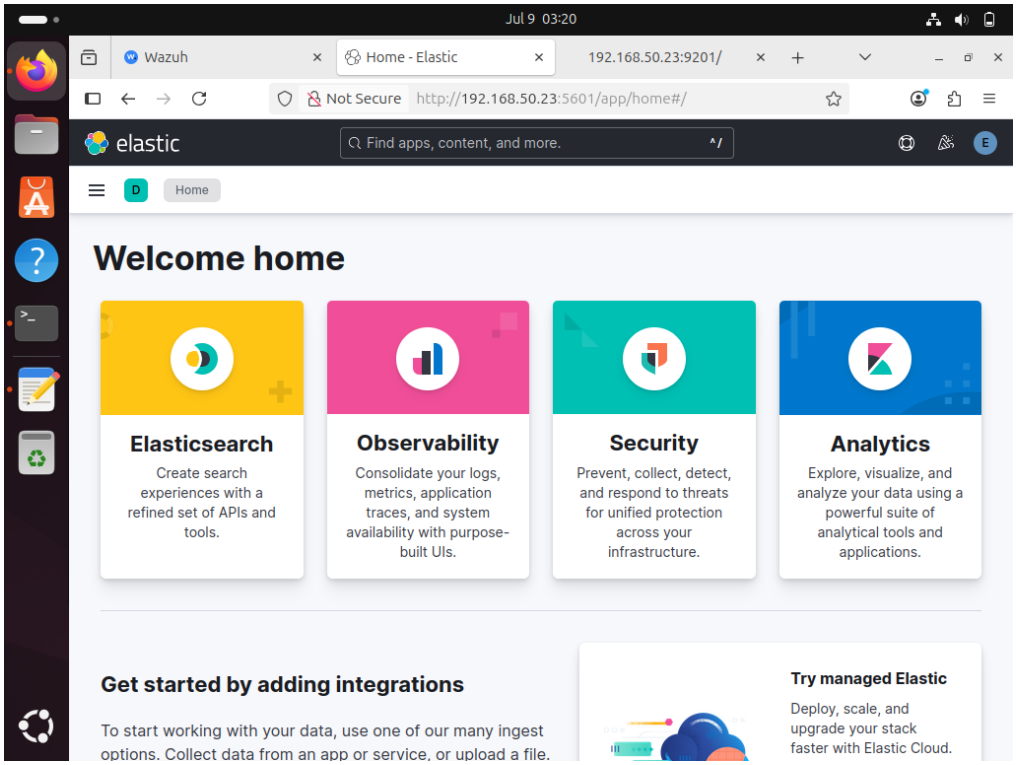
19.3 Firewall Configuration

```
sudo ufw allow 5601/tcp
# Or for iptables:
sudo iptables -A INPUT -p tcp --dport 5601 -j ACCEPT
```

19.4 Troubleshooting

- If connection fails:
  1. Verify Kibana service is running
  2. Check firewall rules
  3. Confirm IP address in `kibana.yml`
- View real-time logs:

  ```
  sudo journalctl -u kibana -f
  ```


Jul 9 03:20
Wazuh
Home - Elastic
192.168.50.23:9201/
Not Secure
http://192.168.50.23:5601/app/home#/
elastic
Find apps, content, and more.
Home
Welcome home
Elasticsearch
Create search experiences with a refined set of APIs and tools.
Observability
Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.
Security
Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.
Analytics
Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.
Get started by adding integrations
To start working with your data, use one of our many ingest options. Collect data from an app or service, or upload a file.
Try managed Elastic
Deploy, scale, and upgrade your stack faster with Elastic Cloud.

20 Logstash Installation

sudo apt install logstash

21 Logstash Elasticsearch Plugin

21.1 Plugin Installation

```
sudo /usr/share/logstash/bin/logstash-plugin install logstash-output-elasticsearch
```

21.2 Verification

- List installed plugins:

  ```
  sudo /usr/share/logstash/bin/logstash-plugin list
  ```

- Check plugin documentation:

  ```
  sudo /usr/share/logstash/bin/logstash-plugin --help
  ```

21.3 Configuration Example

Add to your Logstash pipeline (`/etc/logstash/conf.d/output.conf`):

```
output {
  elasticsearch {
    hosts => ["https://localhost:9201"]
    user => "elastic"
    password => "yourpassword"
    cacert => "/etc/elasticsearch/certs/http_ca.crt"
    index => "logstash-%{+YYYY.MM.dd}"
  }
}
```

21.4 Troubleshooting

- Verify plugin installation:

  ```
  sudo /usr/share/logstash/bin/logstash-plugin list | grep elasticsearch
  ```

- Update all plugins:

  ```
  sudo /usr/share/logstash/bin/logstash-plugin update
  ```

- Check connection to Elasticsearch:

```
curl -k -u elastic:yourpassword https://localhost:9201
```

22 Wazuh Alerts Elasticsearch Integration

22.1 Index Template Setup

1. Download Wazuh index template:

```
sudo wget -qO /etc/logstash/wazuh-template.json \
https://packages.wazuh.com/integrations/elastic/4.x-8.x/
dashboards/wz-es-4.x-8.x-template.json
```

2. Configure certificate access:

```
sudo cp /etc/elasticsearch/certs/http_ca.crt /etc/logstash/
sudo chown logstash:logstash /etc/logstash/http_ca.crt
```

23 Logstash Configuration File

Create `/etc/logstash/conf.d/wazuh.conf`:

```
input {
  file {
    # Unique identifier for this input
    id => "wazuh_alerts"

    # Parse JSON formatted logs
    codec => "json"

    # Processing options
    start_position => "beginning"  # Start reading from beginning of file
    stat_interval => "1 second"    # Check for updates every second
    mode => "tail"                 # Only read new lines added to file

    # Path to Wazuh alerts file
    path => "/var/ossec/logs/alerts/alerts.json"

    # Disable Elastic Common Schema compatibility
    ecs_compatibility => "disabled"
  }
}

output {
  elasticsearch {
    # Elasticsearch connection details
    hosts => ["https://localhost:9201"]
    index => "wazuh-alerts-4.x-%{+YYYY.MM.dd}"

    # Security settings
    user => "elastic"
    password => "your_password_here"
    ssl => true
    cacert => "/etc/logstash/http_ca.crt"

    # Template management
    template => "/etc/logstash/wazuh-template.json"
    template_name => "wazuh"
    template_overwrite => true
  }
}
```

23.1 Important Notes

- Replace `your_password_here` with your actual Elasticsearch password
- Ensure the certificate path is correct for your system
- Verify Logstash has read permissions on the alerts.json file
- The template file must be downloaded before use:

```
sudo wget -O /etc/logstash/wazuh-template.json \
https://packages.wazuh.com/integrations/elastic/4.x-8.x/
dashboards/wz-es-4.x-8.x-template.json
```

23.2 Security Recommendations

- Create dedicated Elasticsearch user with minimal privileges
- Store credentials in Logstash keystore:

```
sudo /usr/share/logstash/bin/logstash-keystore create
sudo /usr/share/logstash/bin/logstash-keystore add ES_USER
sudo /usr/share/logstash/bin/logstash-keystore add ES_PASS
```

23.3 Service Management

```
sudo systemctl restart logstash
sudo journalctl -u logstash -f  # Verify logs
```

23.4 Verification

1. Check Elasticsearch indices:

```
curl -k -u elastic:yourpassword https://localhost:9201/
_cat/indices/wazuh*?v
```

2. Test template application:

```
curl -k -u elastic:yourpassword https://localhost:9201/
_template/wazuh?pretty
```

24 Finalizing Logstash-Wazuh Integration

24.1 Permissions Configuration

1. Add Logstash user to Wazuh group:

```
sudo usermod -aG wazuh logstash
```

2. Set proper certificate ownership:

```
sudo chown logstash: /etc/logstash/http_ca.crt
sudo chmod 640 /etc/logstash/http_ca.crt
```

24.2 Configuration Validation

```
sudo -u logstash /usr/share/logstash/bin/logstash \
--path.settings /etc/logstash/ -t
```

Expected output: `Configuration OK`

24.3 Service Management

```
sudo systemctl enable --now logstash
systemctl status logstash  # Verify running status
```

24.4 Log Monitoring

```
sudo journalctl -f -u logstash  # Real-time logs
```

24.5 Kibana Access

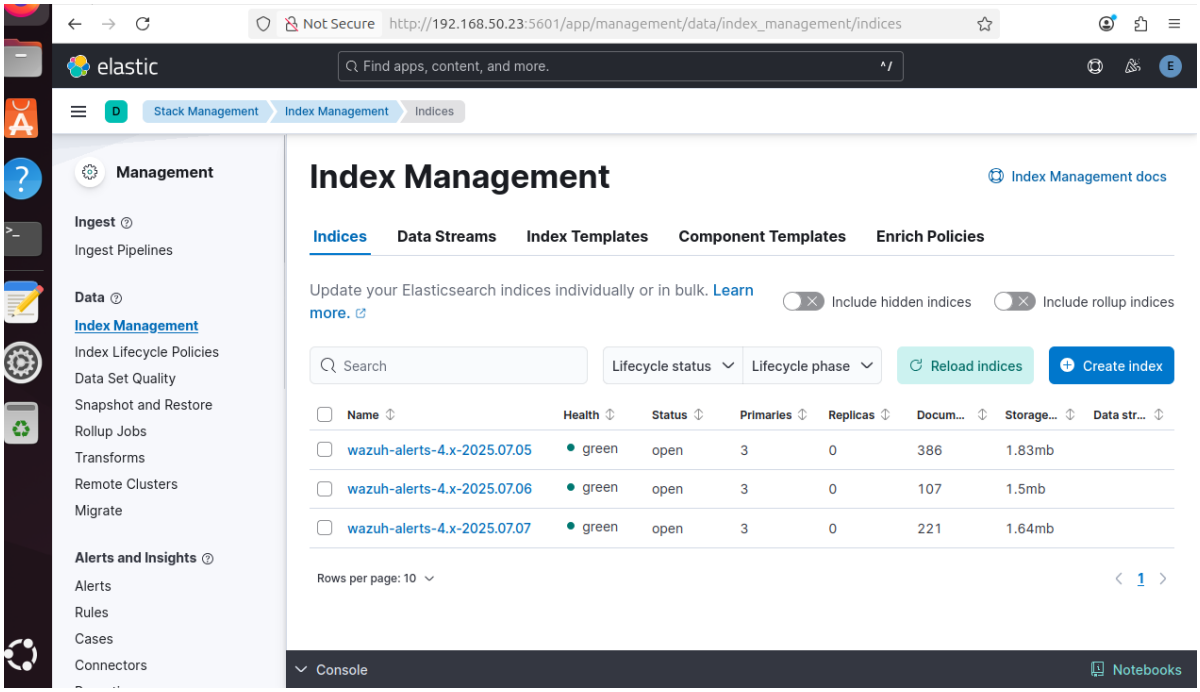
- Open firewall port:

```
sudo ufw allow 5601/tcp
```

- Access URL: `http://<server-IP>:5601`
- Default credentials: elastic/[generated-password]

24.6 Verification Steps

1. Check index creation in Kibana:
    - Navigate to: `Management > Stack Management > Data > Index Management`
    - Verify `wazuh-alerts-4.x-*` indices exist


Not Secure
http://192.168.50.23:5601/app/management/data/index_management/indices
elastic
Find apps, content, and more.
Stack Management
Index Management
Indices
Management
Ingest
Ingest Pipelines
Data
Index Management
Index Lifecycle Policies
Data Set Quality
Snapshot and Restore
Rollup Jobs
Transforms
Remote Clusters
Migrate
Alerts and Insights
Alerts
Rules
Cases
Connectors
Index Management
Index Management docs
Indices
Data Streams
Index Templates
Component Templates
Enrich Policies
Update your Elasticsearch indices individually or in bulk. Learn more.
Include hidden indices
Include rollup indices
Search
Lifecycle status
Lifecycle phase
Reload indices
Create index
Name
Health
Status
Primaries
Replicas
Docum...
Storage...
Data str...
wazuh-alerts-4.x-2025.07.05 green open 3 0 386 1.83mb
wazuh-alerts-4.x-2025.07.06 green open 3 0 107 1.5mb
wazuh-alerts-4.x-2025.07.07 green open 3 0 221 1.64mb
Rows per page: 10
1
Console
Notebooks

2. Confirm data flow:

```
curl -k -u elastic:yourpassword \
https://localhost:9201/wazuh-alerts-4.x-*/_count?pretty
```

24.7 Troubleshooting

- If no alerts appear:
  1. Verify Wazuh is generating alerts at `/var/ossec/logs/alerts/alerts.json`
  2. Check Logstash has read permissions
  3. Confirm Elasticsearch connection in Logstash logs

25 Shuffle Installation

25.1 Prerequisites

- Docker and Docker Compose installed

1. Clone the Shuffle repository:

```
git clone https://github.com/Shuffle/Shuffle.git
cd Shuffle
docker-compose up -d
```

link : https://shuffler.io/docs/configuration/

25.2 Accessing Shuffle

- Web Interface: `http://<your-server-ip>:3001`

26 The Hive Installation

26.1 Prerequisites Installation

```
sudo apt update
sudo apt install -y wget gnupg apt-transport-https git ca-certificates \
curl jq software-properties-common lsb-release python3-pip iproute2
```

26.2 Java Installation

```
sudo apt install -y openjdk-11-jre-headless
echo "JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64" | sudo tee -a /etc/environment
echo "ES_JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64" | sudo tee -a /etc/environment
source /etc/environment
sudo update-java-alternatives --jre-headless -s java-1.11.0-openjdk-amd64
```

26.3 Cassandra Installation

```
wget -qO - https://downloads.apache.org/cassandra/KEYS | \
sudo gpg --dearmor > /etc/apt/trusted.gpg.d/cassandra-archive.gpg
echo "deb https://downloads.apache.org/cassandra/debian 40x main" | \
sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list
sudo apt update
sudo apt install -y cassandra
```

26.4 Cassandra Configuration

```
sudo sed -i '/cluster_name/s/Test Cluster/thehive/' /etc/cassandra/cassandra.yaml
sudo systemctl restart cassandra
sudo systemctl enable cassandra
```

26.5 Elasticsearch Configuration

```
sudo sed -i '/cluster.name/s/^#//;s/my-application/thehive/' \
/etc/elasticsearch/elasticsearch.yml
echo -e "-Xms1g\n-Xmx1g\n-Dlog4j2.formatMsgNoLookups=true" | \
sudo tee /etc/elasticsearch/jvm.options.d/jvm.options
sudo systemctl restart elasticsearch
```

26.6 The Hive Installation

```
wget -qO- https://archives.strangebee.com/keys/strangebee.gpg | \
sudo gpg --dearmor > /etc/apt/trusted.gpg.d/strangebee-archive-keyring.gpg
echo 'deb https://deb.strangebee.com thehive-5.x main' | \
sudo tee /etc/apt/sources.list.d/strangebee.list
sudo apt update
sudo apt install -y thehive
```

26.7 The Hive Configuration

```
sudo sed -i 's/cluster-name = thp/cluster-name = thehive/' \
/etc/thehive/application.conf
sudo systemctl restart thehive
sudo systemctl enable thehive
```

26.8 Verification

- Check services:

  ```
  systemctl status cassandra elasticsearch thehive
  ```

- Access The Hive:

- URL: http://server-ip:9000
- Default credentials: admin@thehive.local - secret

27 MISP Installation via Docker

27.1 Prerequisites

```
sudo apt update
sudo apt install -y docker.io docker-compose
sudo systemctl enable docker
sudo usermod -aG docker $USER
```

Note: Log out and back in to apply group changes.

27.2 Deployment

1. Clone the repository:

   ```
   git clone https://github.com/MISP/misp-docker.git
   cd misp-docker
   ```

2. Configure environment:

   ```
   cp template.env .env
   nano .env  # Edit configuration
   ```

3. Build and start containers:

   ```
   docker-compose build
   docker-compose up -d
   ```

4. Monitor logs:

   ```
   docker-compose logs -f
   ```

27.3 Initial Configuration

- Access: `http://localhost` (or configured domain)
- Default credentials: admin@admin.test - admin
- Critical steps:
  - Change admin password immediately
  - Configure organization details
  - Set base URL in server settings

27.4 Security Hardening

- **SSL Configuration**:

```
a2enmod rewrite ssl
```

- **Firewall Rules**:

```
sudo ufw allow ssh
sudo ufw allow https
sudo ufw enable
```

- **Database Security**:

```
docker exec -it misp-db mysql_secure_installation
```

27.5 Maintenance

- Update containers:

```
docker-compose pull
docker-compose up -d --build
```

- Backup procedure:

```
docker exec misp-db mysqldump -u root -p misp > misp_backup.sql
```

27.6 Verification

- Check running services:

```
docker ps
curl -k https://localhost
```

- Test API access:

```
curl -H "Authorization: YOUR_API_KEY" \
https://localhost/attributes/restSearch
```

28 Cortex Installation and Configuration

28.1 Repository Setup

```
wget -qO- "https://raw.githubusercontent.com/TheHive-Project/Cortex/master/PGP-PUBLIC-KEY" |
sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/cortex.gpg
echo 'deb https://deb.thehive-project.org release main' | \
sudo tee -a /etc/apt/sources.list.d/thehive-project.list
sudo apt update
```

28.2 Installation

```
sudo apt install -y cortex
```

28.3 Configuration

1. Generate secret key:

   ```
   sudo sed -i "/play.http.secret.key/s/^#//;s/\*\*\*CHANGEME\*\*\*/`cat /dev/urandom
   tr -dc 'a-zA-Z0-9' | fold -w 64 | head -n 1`/" /etc/cortex/application.conf
   ```

2. Configure Elasticsearch connection (`/etc/cortex/application.conf`):

   ```
   search {
     index = cortex
     uri = "http://127.0.0.1:9200"
     # Optional authentication:
     # username = "elastic"
     # password = "yourpassword"
   }
   ```

28.4 Service Management

```
sudo systemctl enable --now cortex
systemctl status cortex  # Verify running status
```

28.5 Web Interface

- Open firewall port:

  ```
  sudo ufw allow 9001/tcp
  ```

- Access URL: `http://<server-ip>:9001`
- Initial setup:

1. Complete database migration prompt
2. Create admin account
3. Configure organization details

28.6 Verification

- Check service logs:

```
journalctl -u cortex -f
```

- Test API access:

```
curl http://localhost:9001/api/analyzer
```

29 Shuffle Webhook Integration with Wazuh , gmail , VirusTotal , theHive

link : https://github.com/megatrongodlike/Automated-Threat-Detection-with-Wazuh-Shuffle-and-TheHive/blob/main/README.md

30 TheHive , CORTEX , MISP (integration)

https://youtu.be/F9aCAYwP9do?si=h5etgWc–fbPFt2y

31 Zeek Network Monitoring Setup

31.1 Installation on Ubuntu

```
echo 'deb https://download.opensuse.org/repositories/security:/zeek/xUbuntu_22.04/ /' | sudo
curl -fsSL https://download.opensuse.org/repositories/security:zeek/xUbuntu_22.04/Release.ke
sudo apt update
sudo apt install zeek-7.0
```

32 Zeek Scripting Capabilities

For scripting try to learn basic zeek scripting.
mink : https://try.zeek.org//?example=hello

32.1 Core Functionality

Zeek scripts can:

- Detect security events (intrusions, malware, anomalies)
- Generate real-time alerts and logs

32.2 Detection Script Examples

32.2.1 SSH Brute Force Detection

```
event ssh_auth_failed(c: connection)
{
    if (|c$history| > 5) {
        NOTICE([$note=SSH::Bruteforcing,
                $conn=c,
                $msg=fmt("Brute force attempt from %s", c$id$orig_h)]);
    }
}
```

32.2.2 HTTP Exploit Detection

```
event http_request(c: connection, method: string, uri: string)
{
    if (/etc/passwd in uri || /wp-admin in uri) {
        NOTICE([$note=HTTP::Exploit_Attempt,
                $conn=c,
                $msg=fmt("Web exploit attempt: %s", uri)]);
    }
}
```

32.2.3 DNS Tunneling Detection

```
event dns_request(c: connection, msg: dns_msg, query: string, qtype: count)
{
    if (|query| > 50 && qtype == 16) { # TXT records
        NOTICE([$note=DNS::Tunneling,
                $conn=c,
                $msg="Possible DNS tunneling"]);
    }
}
```

32.3 Traffic Analysis

32.4 Advanced Features

- **Machine Learning**: Integrate with Python/R models via Broker
- **Threat Intelligence**: Match against IOC feeds
- **Protocol Decoding**: Custom protocol analyzers
- **File Analysis**: Extract/sandbox files

32.5 Script Deployment

```
# Load scripts at startup
echo '@load custom_detection' >> /opt/zeek/share/zeek/site/local.zeek
zeekctl deploy
```

33 zeek monitoring

34 Configuration Files

34.1 LAN Monitoring (/opt/zeek/scripts/lan_monitor.zeek)

```
@load policy/tuning/json-logs.zeek

global lan_bandwidth: table[addr] of count &default=0;
const lan_threshold = 100MB &redef;

event connection_state_remove(c: connection) {
    if (c$id$orig_h in 192.168.50.0/24) {
        lan_bandwidth[c$id$orig_h] += c$conn$orig_bytes;
        if (lan_bandwidth[c$id$orig_h] > lan_threshold) {
            NOTICE([$note=Bandwidth::LAN_Abuse,
                    $conn=c,
                    $identifier=fmt("LAN-%s", c$id$orig_h),
                    $msg=fmt("%s exceeded %s on enp0s3",
                            c$id$orig_h,
                            lan_threshold)]);
        }
    }
}
```

34.2 WAN Monitoring (/opt/zeek/scripts/wan_monitor.zeek)

```
@load policy/frameworks/notice/extend-email/hostnames.zeek

global wan_bandwidth: table[addr] of count &default=0;
const wan_threshold = 1GB &redef;

event connection_state_remove(c: connection) {
    if (c$id$resp_h !in 192.168.50.0/24) {  # Exclude LAN traffic
        wan_bandwidth[c$id$resp_h] += c$conn$resp_bytes;
        if (wan_bandwidth[c$id$resp_h] > wan_threshold) {
            NOTICE([$note=Bandwidth::WAN_Abuse,
                    $conn=c,
                    $identifier=fmt("WAN-%s", c$id$resp_h),
                    $msg=fmt("%s traffic via enp0s8: %s",
                            c$id$resp_h,
                            wan_bandwidth[c$id$resp_h])]);
        }
    }
}
```

35 Systemd Service Units

35.1 LAN Service (/etc/systemd/system/zeek-lan.service)

```
[Unit]
Description=Zeek LAN Monitor (enp0s3)
After=network.target

[Service]
Type=simple
ExecStart=/opt/zeek/bin/zeek -i enp0s3 -e 'redef lan_threshold=100MB;' /opt/zeek
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

35.2 WAN Service (/etc/systemd/system/zeek-wan.service)

```
[Unit]
Description=Zeek WAN Monitor (enp0s8)
After=network.target

[Service]
Type=simple
ExecStart=/opt/zeek/bin/zeek -i enp0s8 -e 'redef wan_threshold=1GB;' /opt/zeek/s
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

36 Deployment Commands

```
# Enable and start services
sudo systemctl daemon-reload
sudo systemctl enable --now zeek-lan.service
sudo systemctl enable --now zeek-wan.service

# Verify running instances
zeekctl status
```

37 Log Locations

- **LAN Logs**: /opt/zeek/logs/current/lan_notice.log

- **WAN Logs**: /opt/zeek/logs/current/wan_notice.log
- **Combined JSON**: /opt/zeek/logs/current/bandwidth.json

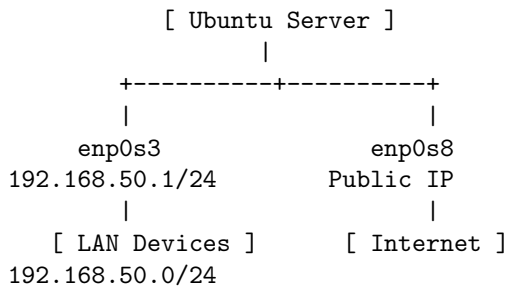


Figure 2: Network Topology with Zeek Monitoring