**This document will serve as the central guide for your Next-Generation Firewall using Wazuh.**

**Next-Generation Firewall with IDS/IPS Using Wazuh**

# Project Overview

This project involves the development of a Next-Generation Firewall (NGFW) using Wazuh for both Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) functionalities. Wazuh serves as the primary monitoring and alerting tool, enabling real-time threat detection and automated response mechanisms to enhance network security.

# Key Features

- Real-time Threat Detection using Wazuh's monitoring and alert management.

- Custom Firewall Rule Creation for controlling and filtering network traffic.

- Intrusion Detection and Prevention by leveraging Wazuh’s capabilities to identify and respond to threats.

- Automated Incident Response with Wazuh’s active monitoring features.

# Tools & Technologies

- Wazuh: An open-source security platform for threat detection, integrity monitoring, and incident response.

- Docker: Containerized environment for easy deployment and management.

- Custom Wazuh Rules: Tailored rules for network monitoring and response.

- Amazon Alexa Echo Dot: Context-aware device used to enhance security monitoring (optional).

# Setup Instructions

## Prerequisites

- A Linux-based environment (e.g., Ubuntu 20.04 or later).

- Docker and Docker Compose installed.

- Wazuh Server and Wazuh Agent properly set up.

- Root or sudo access for configuring Wazuh.

## Step-by-Step Setup

### 1. Deploy Wazuh Using Docker

- Clone the Wazuh Docker repository:

```bash

git clone https://github.com/wazuh/wazuh-docker.git

cd wazuh-docker

```

- Edit the `docker-compose.yml` file to include your configuration.

- Launch the Wazuh stack:

```bash

docker-compose up -d

```

- Access the Wazuh Dashboard at `https://<your-server-ip>` (default credentials: `admin`/`wazuh`).

### 2. Install Wazuh Agent on Client Machines

- Download and install the Wazuh agent on the devices to be monitored:

```bash

sudo apt-get install wazuh-agent

```

- Configure the agent to communicate with the Wazuh server by editing the configuration file:

```bash

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

```

- Add the server's IP address under `<server>` tags.

### 3. Create Custom IDS/IPS Rules in Wazuh

- Create a custom rule file in `/var/ossec/etc/rules/`:

```bash

sudo nano /var/ossec/etc/rules/custom\_rules.xml

```

- Example rule to detect failed SSH login attempts:

```xml

<group name="ngfw\_custom\_rules">

<rule id="100001" level="10">

<decoded\_as>ssh</decoded\_as>

<field name="srcip">!</field>

<description>Multiple failed SSH login attempts detected.</description>

</rule>

</group>

```

- Save the file and restart the Wazuh agent:

```bash

sudo systemctl restart wazuh-agent

```

### 4. Configure Wazuh for Real-time Monitoring

- Enable log monitoring for specific applications (e.g., Apache, Nginx).

- Set up log forwarding from firewalls and network devices to Wazuh for centralized analysis.

- Use the Wazuh dashboard to monitor alerts and manage incidents.

### 5. Implement Automated Incident Response

- Configure active responses in Wazuh to block suspicious IPs.

- Go to the Wazuh dashboard, navigate to `Active Responses`, and set up actions like IP blocking or user lockout.

#### Usage

- Real-time Monitoring: Use the Wazuh dashboard for viewing alerts and tracking incidents.

- Custom Rule Management: Add or modify custom rules in `custom\_rules.xml` to tailor network defense.

- Automated Responses: Enable active responses to automatically mitigate detected threats.

#### Example Custom Rules

- Multiple Failed SSH Logins:

```xml

<rule id="100001" level="10">

<group>authentication\_failed,sshd</group>

<description>Multiple failed SSH login attempts detected.</description>

<frequency>5</frequency>

<same\_source\_ip />

<options>no\_full\_log</options>

</rule>

## Project Outcome

By implementing this NGFW solution with Wazuh, the following security improvements are achieved:

* **Enhanced Threat Detection** through customized rules and real-time monitoring.
* **Automated Threat Prevention** with active response mechanisms.
* **Centralized Security Management** using Wazuh's comprehensive dashboard and alerting system.

## Future Enhancements

* Integration of additional Wazuh modules for vulnerability detection.
* Refinement of custom rules to improve accuracy and reduce false positives.
* Explore machine learning integrations for anomaly detection.

## Contact

For any queries or contributions, please contact **Apoorva Karambelkar** at <apoorvaknk@gmail.com>