

# Laporan Tugas Praktek

## Penerapan dekoder di Wazuh



Mr.Foxzie

Noctra Lupra Bootcamp Trainee – Cybersecurity & Penetration Testing  
IDN NETWORKERS

## Whoami?

Saya Mr. Foxzie, seorang praktisi keamanan siber dengan fokus pada penetration testing dan bug bounty hunting. Aktivitas peretasan yang saya lakukan semata-mata untuk eksplorasi dan kesenangan (hack for fun), dengan tujuan menemukan dan melaporkan celah keamanan secara etis.

## Pendahuluan

**Wazuh** adalah platform keamanan *open-source* yang powerful. Sederhananya, Wazuh ini seperti satpam super canggih untuk sistem komputer dan jaringan kita. Wazuh bekerja sebagai *Security Information and Event Management (SIEM)* dan *Host-based Intrusion Detection System (HIDS)*. Artinya, Wazuh bisa:

- **Mengumpulkan dan Menganalisis Log:** Wazuh mengumpulkan log (catatan aktivitas) dari berbagai sumber seperti server, laptop, aplikasi, dan perangkat jaringan. Lalu, semua log ini dianalisis untuk mendeteksi hal-hal aneh dan mencurigakan.
- **Mendeteksi Ancaman:** Wazuh bisa mengenali tanda-tanda serangan siber, aktivitas *malware*, dan upaya peretasan.
- **Memantau Integritas File:** Wazuh akan memberi tahu kita jika ada file penting di sistem yang diubah tanpa izin.
- **Merespons Insiden:** Jika ada ancaman terdeteksi, Wazuh bisa melakukan tindakan otomatis, misalnya memblokir alamat IP penyerang.

Dengan Wazuh, kita bisa memantau keamanan infrastruktur IT kita secara terpusat, jadi lebih mudah untuk mengelola dan merespons insiden keamanan.

tipe wazuh

WAZUH\_VERSION="v4.12.0"

WAZUH\_REVISION="rc1"

WAZUH\_TYPE="server"

setup wazuh pada lab

saya menggunakan spesifikasi sistem wazuh adalah



```
NAME="Amazon Linux"
VERSION="2023"
ID="amzn"
ID_LIKE="fedora"
VERSION_ID="2023"
PLATFORM_ID="platform:al2023"
PRETTY_NAME="Amazon Linux 2023.7.20250428"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2023"
HOME_URL="https://aws.amazon.com/linux/amazon-linux-2023/"
DOCUMENTATION_URL="https://docs.aws.amazon.com/linux/"
SUPPORT_URL="https://aws.amazon.com/premiumsupport/"
BUG_REPORT_URL="https://github.com/amazonlinux/amazon-linux-2023"
VENDOR_NAME="AWS"
VENDOR_URL="https://aws.amazon.com/"
SUPPORT_END="2029-06-30"
bash-5.2#
```

#### a. install docker

Berikut list sederhana install Docker di Arch Linux:

1. Update sistem  
► `sudo pacman -Syu`
2. Install Docker  
► `sudo pacman -S docker`
3. Enable dan start Docker service  
► `sudo systemctl enable --now docker`
4. (Opsional) Tambahkan user ke grup docker  
► `sudo usermod -aG docker \$USER`  
► `newgrp docker`
5. Cek versi dan test container  
► `docker --version`  
► `docker run hello-world`
6. (Opsional) Install Docker Compose v1  
► `sudo pacman -S docker-compose`

#### b. pulling wazuh

Setelah Docker siap, kita tinggal mengambil *image* Wazuh dan menjalankannya.

- Clone Repotori Wazuh Docker: Kita akan mengambil file konfigurasi Docker dari repositori resmi Wazuh. "git clone https://github.com/wazuh/wazuh-docker.git -b v4.12.0"
- Jalankan Wazuh: Gunakan Docker Compose untuk menjalankan semua komponen Wazuh. command "docker-compose up -d"
- Jika tidak ada install dulu docker compose disini saya menggunakan pacman -S docker-compose
- Cek Kontainer: Pastikan semua kontainer Wazuh berjalan dengan baik.

```
apple .../wazuh-docker/single-node 1 HEAD ① 13:18
$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2b8963d23961 wazuh/wazuh-dashboard:4.12.0 "/entrypoint.sh" 2 hours ago Up 2 hours 443/tcp, 0.0.0.0:443→5601/tcp, [::]:443-single-node-wazuh.dashboard-1
b78a9e2ed4a5 wazuh/wazuh-indexer:4.12.0 "/entrypoint.sh open..." 2 hours ago Up 2 hours 0.0.0.0:9200→9200/tcp, [::]:9200-single-node-wazuh.indexer-1
dc547bec097b wazuh/wazuh-manager:4.12.0 "/init" 2 hours ago Up 2 hours 0.0.0.0:1514→1514-1515/tcp, [::]:1514-1515→1514-1515/tcp, [::]:1515-single-node-wazuh.manager-1
14-1515→1514-1515/tcp, 0.0.0.0:514→514/udp, 0.0.0.0:55000→55000/tcp, [::]:55000→55000/tcp, 1516/tcp single-node-wazuh.manager-1
2888416fc894 nwodtuh5/exegol:free "/bin/bash ./exegol..." 39 hours ago Exited (0) 14 hours ago exegol-f0x
```

anda bisa menginstall sertifikat bila diperlukan

```
-r----- 1 999 adm 1784 Jul 22 11:23 root-ca-manager.key
apple .../wazuh-docker/single-node 1 HEAD ① 11:24
$ docker-compose -f generate-indexer-certs.yml run --rm generator

WARN[0000] /home/whoami/wazuh/wazuh-docker/single-node/generate-indexer-certs.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] Found orphan containers ([single-node-wazuh.dashboard-1 single-node-wazuh.manager-1 single-node-wazuh.indexer-1]) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphans flag to clean it up.
The tool to create the certificates exists in the in Packages bucket
22/07/2025 03:25:12 INFO: Generating the root certificate.
22/07/2025 03:25:13 INFO: Generating Admin certificates.
22/07/2025 03:25:13 INFO: Admin certificates created.
22/07/2025 03:25:13 INFO: Generating Wazuh indexer certificates.
22/07/2025 03:25:13 INFO: Wazuh indexer certificates created.
22/07/2025 03:25:13 INFO: Generating Filebeat certificates.
22/07/2025 03:25:14 INFO: Wazuh Filebeat certificates created.
22/07/2025 03:25:14 INFO: Generating Wazuh dashboard certificates.
22/07/2025 03:25:14 INFO: Wazuh dashboard certificates created.
Moving created certificates to the destination directory
Changing certificate permissions
Setting UID indexer and dashboard
```

tampilan awal setelah login dengan user admin dan password SecretPassword

The screenshot shows the Wazuh Google Chrome interface. At the top, there's a navigation bar with tabs like Beran, Tugas, Dock, Deplo, Searc, NGIN, Writin, Install, Settin, and a search bar. Below the navigation bar is the main dashboard area. It features several cards:

- AGENTS SUMMARY:** This instance has no agents registered. Please deploy agents to begin monitoring your endpoints. Includes a "Deploy new agent" button.
- LAST 24 HOURS ALERTS:** Critical severity: 0 (Rule level 12 to 14); High severity: 0 (Rule level 7 to 11); Medium severity: 45 (Rule level 0 to 6); Low severity: 141 (Rule level 0 to 6).
- ENDPOINT SECURITY:** Configuration Assessment: Scan your assets as part of a configuration assessment audit. Malware Detection: Check indicators of compromise triggered by malware infections or cyberattacks.
- THREAT INTELLIGENCE:** Threat Hunting: Browse through your security alerts, identifying issues and threats in your environment. Vulnerability Detection: Discover what applications in your environment are affected by well-known vulnerabilities. MITRE ATT&CK: Explore security alerts.

selain itu jika ingin melihat debug log bisa menuliskan perintah 'docker logs single-node-wazuh.dashboard-1

### c. Setting

```
/wazuh/Resource
> # Salin direktori log ke dalam direktori /tmp di dalam kontainer manager
docker cp /home/whoami/Documents/wazuh/Resource/asdaads.log/ single-node-wazuh.manager-1:/tmp/asdaads.log/
docker cp /home/whoami/Documents/wazuh/Resource/asdasedasad-walwe.log/ single-node-wazuh.manager-1:/tmp/asdasedasad-walwe.log/
Successfully copied 15.5MB to single-node-wazuh.manager-1:/tmp/asdaads.log/
Successfully copied 5.44MB to single-node-wazuh.manager-1:/tmp/asdasedasad-walwe.log/

/wazuh/Resource
> docker exec -it single-node-wazuh.manager-1 bash
bash-5.2
```

memberikan izin ke wazuh untuk melihat log di custom /temp/

```
g":"Ending rootcheck scan.,"decoder":{"name":"rootcheck"},"o
bash-5.2# ls -l /tmp/
total 0
drwxr-xr-x 1 1000 1000 64 Jul 25 03:37 asdaads.log
drwxr-xr-x 1 1000 1000 86 Jul 25 03:37 asdasedasad-walwe.log
bash-5.2# chown -R wazuh:wazuh /tmp/asdaads.log/
chown -R wazuh:wazuh /tmp/asdasedasad-walwe.log/
bash-5.2# exit
```

File tuning dan pengertiannya

*File tuning* di Wazuh adalah proses penyesuaian *rules* dan *decoder* agar lebih sesuai dengan kebutuhan spesifik lingkungan kita. Secara *default*, Wazuh sudah punya banyak *rules* dan *decoder*, tapi kadang kita perlu membuat yang baru atau memodifikasi yang sudah ada untuk:

- Mendeteksi ancaman yang lebih spesifik.
- Mengurangi *false positive* (alarm palsu).
- Mendapatkan informasi yang lebih detail dari log.

Proses ini biasanya dilakukan dengan mengedit file di `/var/ossec/etc/decoders/` dan `/var/ossec/etc/rules/`.

memasukkan file dir ke ossec

```
<localfile>
<location>/tmp/openstack_normal1.log</location>
<log_format>syslog</log_format>
</localfile>

<localfile>
<location>/tmp/openstack_abnormal.log</location>
<log_format>syslog</log_format>
</localfile>

</ossec_config>
bash-5.2# 
```

## 1. ruleset dan dekodeer

### Pengertian Ruleset dan Decoder

Di Wazuh, decoder dan rules adalah dua komponen utama dalam analisis log.

- Decoder: Tugasnya adalah "menerjemahkan" log mentah yang tidak terstruktur menjadi format yang lebih mudah dibaca dan dipahami oleh Wazuh. Decoder akan mengekstrak informasi penting dari log, seperti alamat IP, nama pengguna, jenis serangan, dan lain-lain.
- Rules: Setelah log diterjemahkan oleh decoder, rules akan menganalisis informasi yang sudah diekstrak tersebut. Rules berisi kondisi-kondisi tertentu, dan jika kondisi tersebut terpenuhi, maka Wazuh akan menghasilkan sebuah alert (peringatan).

Sederhananya: Decoder memecah log, Rules menganalisis hasilnya

#### a. atomic ruleset

Istilah "Atomic Ruleset" lebih umum digunakan pada produk keamanan lain seperti Atomicorp. Dalam konteks Wazuh, ini bisa diartikan sebagai sekumpulan *rules* yang sangat spesifik dan terfokus untuk mendeteksi satu jenis ancaman atau aktivitas tertentu dengan sangat akurat.

Karakteristiknya adalah:

- **Sangat Spesifik:** Dibuat untuk satu tujuan yang jelas.
- **Tingkat Akurasi Tinggi:** Dirancang untuk meminimalkan *false positive*.
- **Modular:** Bisa digabungkan dengan *rules* lain untuk membentuk skenario deteksi yang kompleks.

berikut adalah ruleset yang disesuaikan

```

bash-5.2# cat /var/ossec/etc/rules/openstack_rules.xml
<group name="nova,">
<!-- A. Decode As LVL-0 -->
<!-- 1. nova-api -->
<rule id="100100" level="3">
  <decoded_as>nova-api-log</decoded_as>
  <description>OpenStack Nova API base log</description>
</rule>

<!-- 2. nova-compute -->
<rule id="100200" level="0">
  <decoded_as>nova-computed-log</decoded_as>
  <description>OpenStack Nova Compute base log</description>
</rule>

<!-- B. Alert - LOG_LEVEL LVL-1,5,8 -->
<!-- 1. nova-api -->
<rule id="100101" level="3">
  <if_sid>100100</if_sid>
  <field name="log_level">INFO</field>
  <description>Nova Compute INFO log</description>
</rule>

<rule id="100102" level="8">
  <if_sid>100100</if_sid>
  <field name="log_level">WARNING</field>
  <description>Nova API WARNING log</description>
</rule>

<rule id="100103" level="8">
  <if_sid>100100</if_sid>
  <field name="log_level">ERROR</field>
  <description>Nova API ERROR log</description>
</rule>

<!-- 2. nova-compute -->
<rule id="100201" level="3">
  <if_sid>100200</if_sid>
  <field name="log_level">INFO</field>
  <description>Nova Compute INFO log</description>
</rule>

<rule id="100202" level="5">
  <if_sid>100200</if_sid>
  <field name="log_level">WARNING</field>
  <description>Nova Compute WARNING log</description>
</rule>

<rule id="100203" level="8">
  <if_sid>100200</if_sid>
  <field name="log_level">ERROR</field>
  <description>Nova Compute ERROR log</description>
</rule>
</group>
bash-5.2#

```

## b. siblings decoder

*Siblings decoder* (decoder bersaudara) adalah sebuah strategi di Wazuh di mana beberapa *decoder* bekerja secara paralel pada level hierarki yang sama. Ini berbeda dengan hubungan *parent-child* (induk-anak) di mana satu *decoder* memicu *decoder* lainnya.

*Siblings decoder* sangat berguna ketika kita berhadapan dengan log yang kompleks dan membutuhkan beberapa strategi *decoding* yang berbeda. Dengan pendekatan ini, kita bisa

memecah proses *decoding* menjadi beberapa bagian yang lebih kecil dan mudah dikelola, sehingga meningkatkan keterbacaan dan fleksibilitas.

comtoh decoding

```
bash-5.2# cat /var/ossec/etc/decoders/openstack_decoders.xml
<!-- LOG Parent -->
<decoder name="nova-api-log">
  <prematch type="pcre2">^nova-api\.\log</prematch>
</decoder>

<decoder name="nova-computed-log">
  <prematch type="pcre2">^nova-compute\.\log</prematch>
</decoder>

<!-- A. Decoder nova-api -->
<!-- 1. LOG HTTP -->
<decoder name="nova-api-log-http">
  <parent>nova-api-log</parent>
  <prematch type="pcre2">"(GET|POST|PUT|DELETE) .+ HTTP/[0-9\.]+</prematch>
  <regex type="pcre2">\.([0-9]{4}-[0-9]{2}-[0-9]{2}_[0-9]{2}:[0-9]{2}:[0-9]{2}) ([0-9]{4}-[0-9]{2}-[0-9]{2}:[0-9]{2}:([0-9]{4}-[0-9]{2}-[0-9]{2}):[0-9]{2}:[0-9]{3}) ([0-9]{4},6) (INFO|DEBUG|ERROR|WARNING|CRITICAL) ([\w\.]+) \\\
  [req-([a-f0-9\-\-]+)(?: ([^\-]+)*?)?](?: ([0-9\.,]+)+)? "^(GET|POST|PUT|DELETE) ([^ ]+) HTTP/([0-9\.]+)" status: (\d{3}) len: (\d+) time: ([0-9\.]+)</regex>
  <order>log_timestamp, event_timestamp, pid, log_level, module, request_id, ip, http_method, http_path, http_version, http_status, http_len, http_time</order>
</decoder>

<!-- 99. LOG Detail -->
<decoder name="nova-api-log-detail">
  <parent>nova-api-log</parent>
  <regex type="pcre2">\.(\d{4}-\d{2}-\d{2}_\d{2}:\d{2}:\d{2}) (\d{4}-\d{2}-\d{2}) \d{2}:\d{2}:\d{3} (\d+) (INFO|DEBUG|ERROR|WARNING|CRITICAL) ([\w\.]+)(?: \\[req-([a-f0-9\-\-]+)(?: ([^\-]+)*?)?](?: \\[instance: ([a-f0-9\-\-]+)])?)(?: (.+))</regex>
  <order>log_timestamp, event_timestamp, pid, log_level, module, request_id, instance_id, message</order>
</decoder>

<!-- B. Decoder nova-compute -->
<!-- 99. LOG Detail -->
<decoder name="nova-compute-log-detail">
  <parent>nova-computed-log</parent>
  <regex type="pcre2">\.(\d{4}-\d{2}-\d{2}_\d{2}:\d{2}:\d{2}) (\d{4}-\d{2}-\d{2}) \d{2}:\d{2}:\d{3} (\d+) (INFO|DEBUG|ERROR|WARNING|CRITICAL) ([\w\.]+)(?: \\[req-([a-f0-9\-\-]+)(?: ([^\-]+)*?)?](?: \\[instance: ([a-f0-9\-\-]+)])?)(?: (.+))</regex>
  <order>log_timestamp, event_timestamp, pid, log_level, module, request_id, instance_id, message</order>
</decoder>
bash-5.2#
```

kode sumber:

[https://github.com/ariafatah0711/idn\\_bootcamp/blob/main/task/week\\_9/2\\_fine\\_tunning\\_wazuh/README.md](https://github.com/ariafatah0711/idn_bootcamp/blob/main/task/week_9/2_fine_tunning_wazuh/README.md)

2. testing

a. dashboard wazuh

dalam dashboard wazuh tidak bisa langsung deteksi log yang di injek di tmp dan hanya bisa cek ruleset

The screenshot shows the 'Ruleset Test' section of the Wazuh UI. At the top, there's a header bar with the URL <https://192.168.1.13/app/ruleset-test#/wazuh-dev?tab=logtest>. Below the header, the title 'Ruleset Test' is displayed. A log entry is shown: '2025-07-25 13:31:00.000 98765 ERROR keystonemiddleware.auth\_token Bad response code while validating token: 503'. Below the log, a blue button labeled '▷ Test' is visible. To the right of the log, there's a red button labeled 'Clear session'. The main area contains processing logs:  
\*\*Phase 2: Completed decoding.  
No decoder matched.  
  
\*\*Phase 3: Completed filtering (rules).  
id: '1002'  
level: '2'  
description: 'Unknown problem somewhere in the system.'  
groups: '['syslog", "errors"]'  
firetimes: '1'  
gg13: '['4.3"]'  
mail: 'false'

### 3. reporting

Berdasarkan hasil pengujian di atas, laporan ini menunjukkan bahwa:

- Instalasi Berhasil:** Wazuh telah berhasil diinstal dan dijalankan menggunakan Docker.
- Konfigurasi Kustom Efektif:** Wazuh mampu memantau log dari direktori kustom setelah dilakukan konfigurasi pada `ossec.conf`.
- Decoder dan Rules Bekerja:** *Custom decoder* dan *rules* yang dibuat berhasil memproses log dan menghasilkan *alert* yang relevan di *dashboard*.

Dengan demikian, penerapan *decoder* di Wazuh telah berhasil dilakukan. Langkah selanjutnya adalah mengembangkan *ruleset* yang lebih komprehensif untuk meningkatkan kapabilitas deteksi ancaman sesuai dengan kebutuhan spesifik lingkungan IT yang dipantau.