MANUAL BOOK

PBL-RKS412 Pengembangan SIEM dan Simulasi Game-Based Security Operation Center (SOC) Blue Team

Disusun oleh :

4332301038	Rey Sastria Harianja
4332301055	Akyasa Fikri Ramadhan
4332311044	Adhitya Pramadhan
4332311041	Hasbi Hakim
4332301057	Desty Silva Dewi
4332311011	Nohiro Hazel Nayottama R.H

JURUSAN TEKNIK INFORMATIKA

PRODI REKAYASA KEAMANAN SIBER

POLITEKNIK NEGERI BATAM 2025

4.1 Instalasi Web Server

1. Update & upgrade

```
pbl306@webserver:~$ sudo apt-get update
[sudo] password for pbl306:
Hit:1 http://id.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://id.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:4 https://ppa.launchpadcontent.net/ondrej/php/ubuntu jammy InRelease
Get:5 http://id.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Fetched 384 kB in 4s (94.8 kB/s)
Reading package lists... Done
pbl306@webserver:~$
```

```
pbl306@webserver:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  php8.2 php8.2-cli php8.2-common php8.2-curl php8.2-fpm php8.2-gd php8.2-intl php8.2-mbstring
  php8.2-mysql php8.2-opcache php8.2-readline php8.2-xml php8.2-zip
13 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 6,017 kB of archives.
After this operation, 2,048 B of additional disk space will be used.
Do you want to continue? [Y/n] y
```

2. Install nginx dan config nginx

```
pbl306@webserver:~$ sudo apt-get install nginx
[sudo] password for pbl306:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.18.0-6ubuntu14.5).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pbl306@webserver:~$
```

```
pbl306@webserver:~$ sudo nano /etc/nginx/sites-enabled/website
```

```
  GNU nano 6.2                          /etc/nginx/sites-enabled/website
server {
    listen 443 ssl;
    server_name 192.168.182.200;

    ssl_certificate /etc/nginx/ssl/Sertif-306/PBL-306/fullchain.pem;
    ssl_certificate_key /etc/nginx/ssl/Sertif-306/PBL-306/HTTPS/ServerSSl.key;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers  HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers   on;

    #ssl_client_certificate /root/client_rootca_intermediate.crt;
    #ssl_client_certificate /etc/nginx/ssl/Sertif-306/PBL-306/Client/client.crt;
    #ssl_verify_client on;

    root /var/www/html/bagisto-306-main/public;
    index index.html index.htm index.php;

    charset utf-8;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    error_page 404 /index.php;

    location ~ \.php$ {
        fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~ /\.(?!well-known).* {
        deny all;
    }
}
```

```
pbl306@webserver:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
pbl306@webserver:~$
```

```
pbl306@webserver:~$ sudo systemctl restart nginx
pbl306@webserver:~$
```

3. Install Mysql-Client

```
pbl306@webserver:~$ sudo apt-get install mysql-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mysql-client is already the newest version (8.0.40-0ubuntu0.22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pbl306@webserver:~$
```

4. Install Php

```
pbl306@webserver:~$ sudo apt-get install php-mysql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  php8.3-mysql
The following NEW packages will be installed:
  php-mysql php8.3-mysql
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 139 kB of archives.
After this operation, 480 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

5. Setting IP pada Web Server

```
pbl306@webserver:~$ sudo nano /etc/netplan/00-installer-config.yaml
```

```
  GNU nano 6.2                        /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens33:
      dhcp4: no
      addresses: [192.168.182.200/24]
      routes:
        - to: default
          via: 192.168.182.2
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
  version: 2
```

4.2 Instalasi Database

1. Update dan upgrade

```
pbl306@database:~$ sudo apt-get update
[sudo] password for pbl306:
Hit:1 http://id.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://id.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://id.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Fetched 384 kB in 2s (195 kB/s)
Reading package lists... Done
pbl306@database:~$
```

```
pbl306@database:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  sosreport ubuntu-advantage-tools ubuntu-pro-client-l10n
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
pbl306@database:~$
```

2. Install mysql-server

```
pbl306@database:~$ sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mysql-server is already the newest version (8.0.40-0ubuntu0.22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
pbl306@database:~$
```

3. Secure pada mysql-server agar mengamankan user rootnya

```
pbl306@database:~$ sudo mysql_secure_installation
```

4. Memberikan hak akses dan password di db-server

```
pbl306@database:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 53
Server version: 8.0.40-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

5. Setting IP untuk database-server

4.3 Menghubungkan database dan web server

1. Konfigurasi pada database

```
pbl306@database:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
  GNU nano 6.2                     /etc/mysql/mysql.conf.d/mysqld.cnf *
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user            = mysql
# pid-file      = /var/run/mysqld/mysqld.pid
# socket        = /var/run/mysqld/mysqld.sock
# port          = 3306
# datadir       = /var/lib/mysql


# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir                = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
mysqlx-bind-address     = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size         = 16M
# max_allowed_packet    = 64M
# thread_stack          = 256K

# thread_cache_size       = -1

# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover-options  = BACKUP

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo
```

2. Restart mysql

```
pbl306@database:~$ sudo systemctl restart mysql
pbl306@database:~$
```

3. Membuat database baru

```
mysql> CREATE USER 'webserver'@'192.168.182.200' IDENTIFIED BY 'Admin@306';
Query OK, 0 rows affected (1.76 sec)

mysql>
```

4. Berikan hak privilages pada database supaya dapat mengakses database yang baru dibuat

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

5. Berikan hak privilages pada database supaya dapat mengakses database yang baru dibuat

```
mysql> GRANT SELECT ON Bagisto.* TO 'webserver'@'192.168.182.200' ;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

4.4 Konfiguarsi pada web server

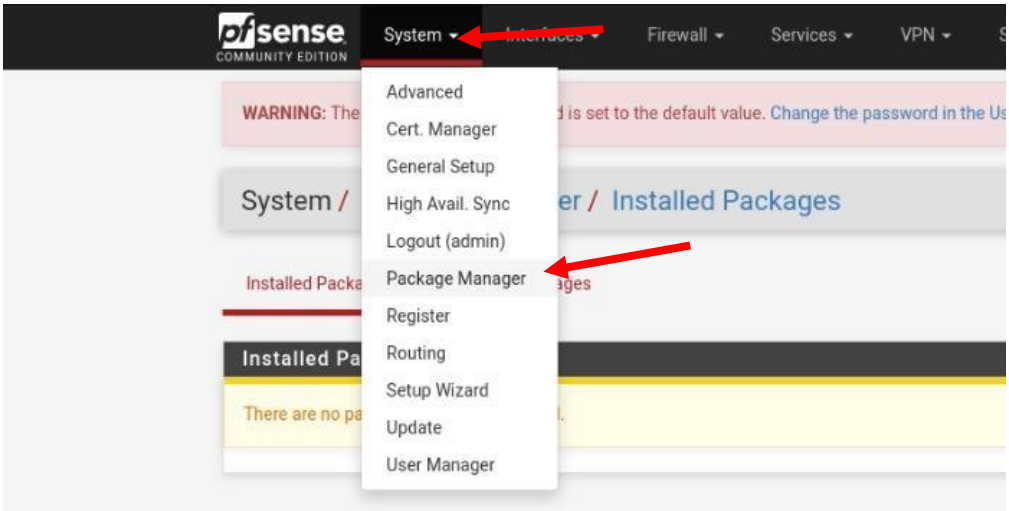
1. Pada web server login ke mysql dengan menggunakan command **mysql -u user -h ip Database -p.** Agar dapat mengakses database yang sudah dibikin.

```
pbl306@webserver:~$ sudo mysql -u webserver -h 192.168.182.199 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.40-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

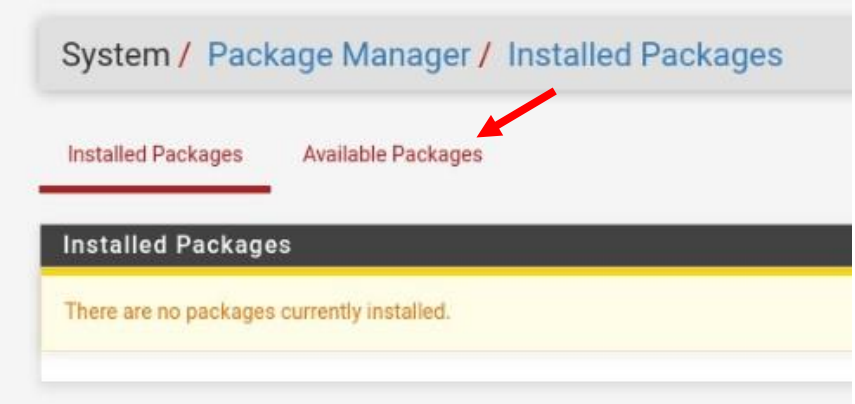
4.5 Instalasi IDS/IPS

1. Instalasi Snort

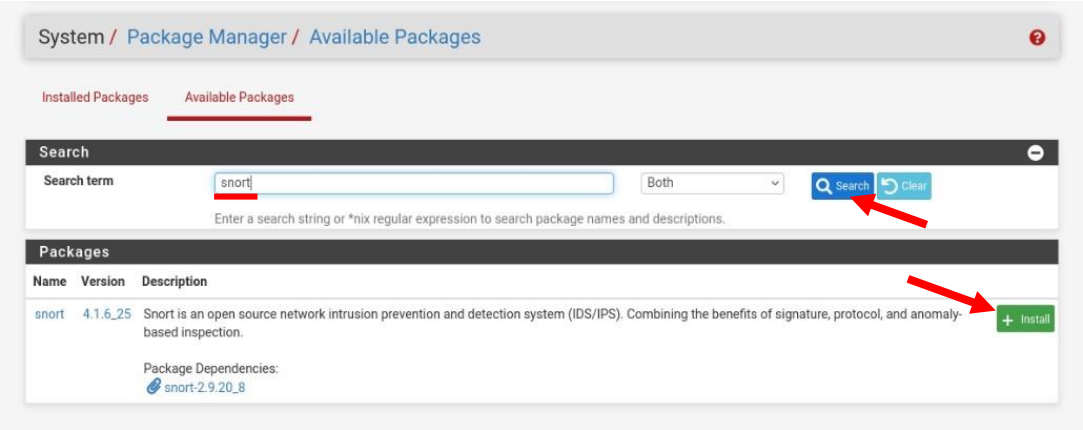
   Pada bar navigasi pilih **System**, lalu tekan **Package Manager**, seperti pada Gambar.



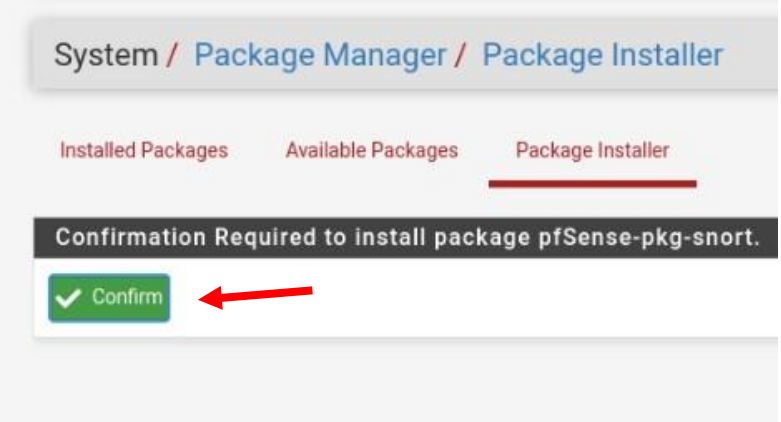
Lalu pilih **Available Packages**, seperti pada Gambar



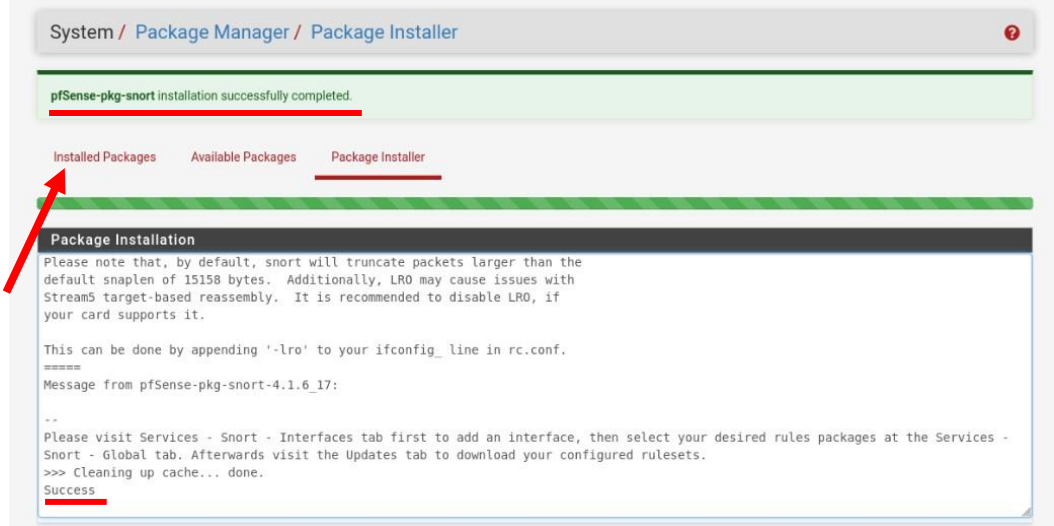
Kemudian ketik pada search bar dengan kata "**snort**". Lalu tekan tombol Search, seperti pada Gambar dibawah. Kemudian akan ditampilkan Packages snort, klik tombol Install



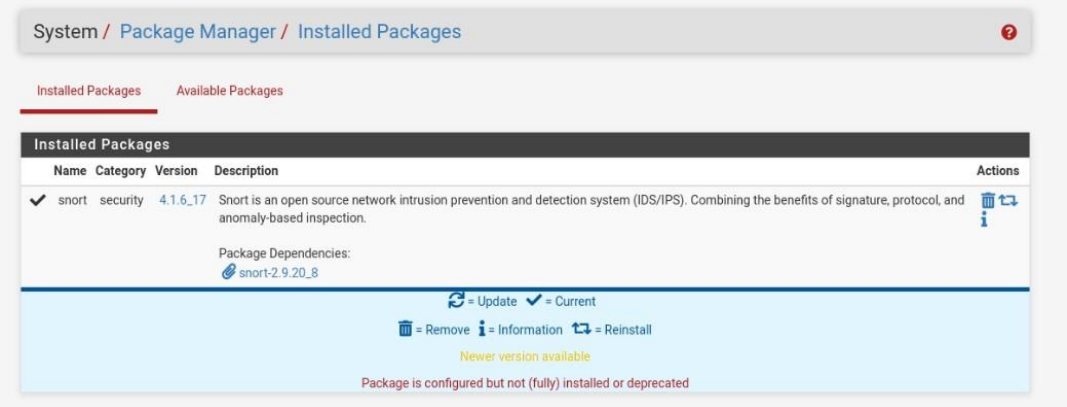
Kemudian tekan tombol **Confirm.**



Tunggu hingga proses instalasi selesai, jika sudah selesai akan tampil notifikasi **Success**. Selanjutnya, kembali ke **Installed Packages.**

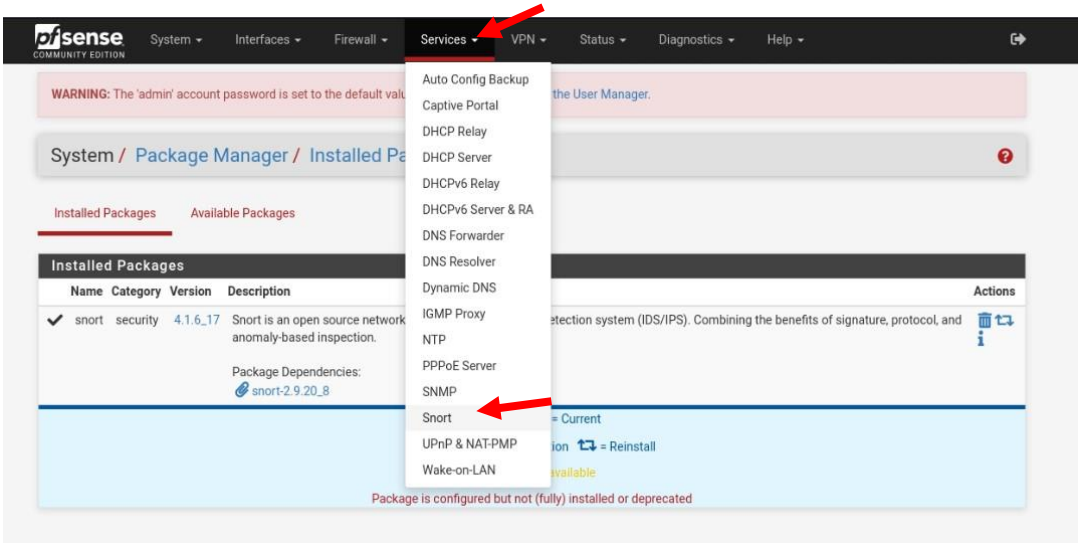


Terlihat bahwa packages snort sudah terinstall pada pfsense.

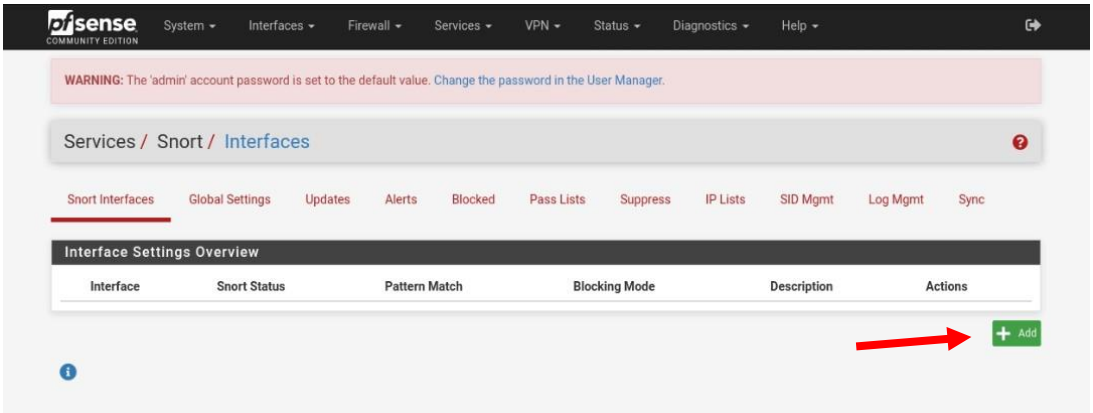


2. Konfigurasi Snort

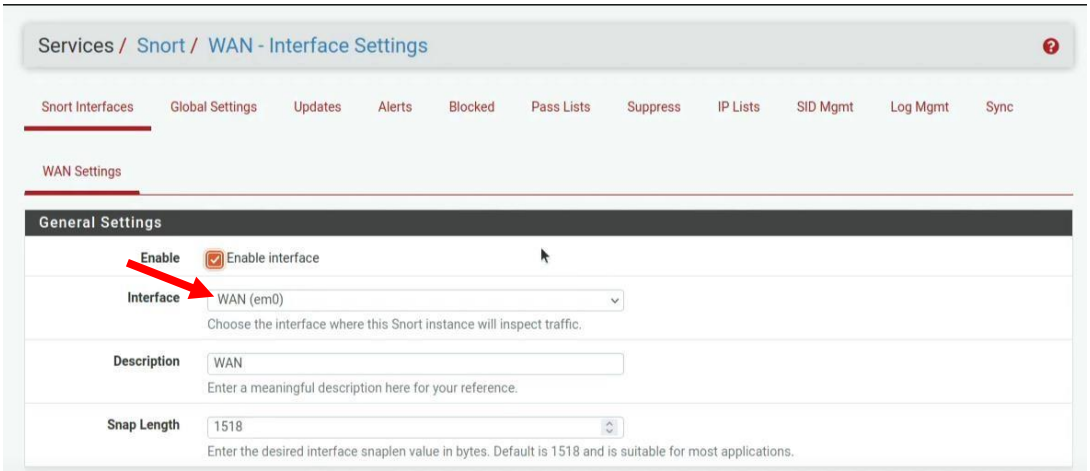
Setelah instalasi snort berhasil, snort sudah bisa dikonfigurasi. Pada bar navigasi pilih **Services**, lalu pilih **Snort**.



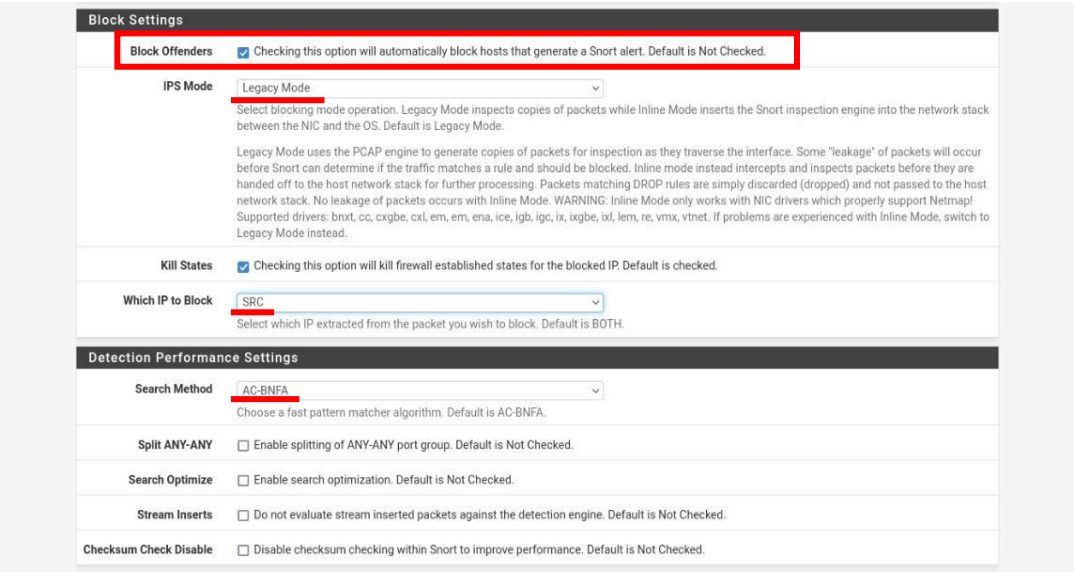
Saat mengakses layanan snort, pengguna akan langsung diarahkan pada menu **Interfaces**. Buatlah interface baru dengan mengklik tombol **Add**.



Pastikan interface yang akan digunakan adalah interface **WAN.**

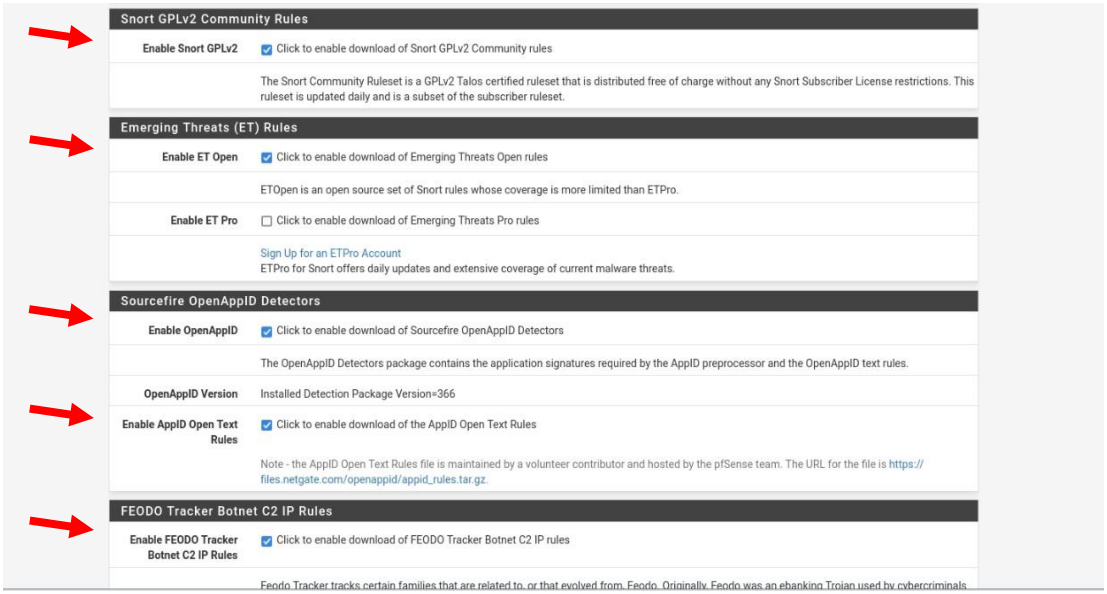


Kemudian pada Block Settings, centang **Block Offenders** dan pilih **Legacy Mode** pada IPS Mode. Biarkan Kill States dicentang, kemudian pada Which IP to Block pilih opsi **SRC** (Source) agar IPS hanya memblokir source packet dan tidak memblokir destination, yaitu layanan milik kita. Lalu pada Detection Performance Settings, pilih opsi **AC-BNFA** (default) pada Search Method.



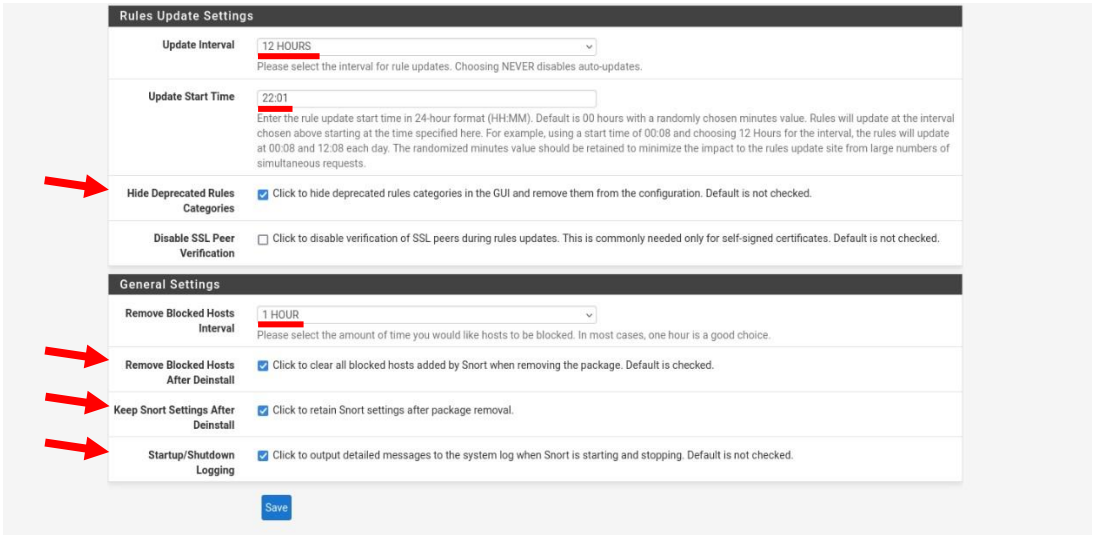
Pada tahap ini centanglah berapa rules seperti Gambar dibawah ini, jangan centang pada snort VRT karena fitur tersebut merupakan fitur enterpise atau berbayar.

Gunakan Community Rules yang sudah disediakan secara gratis.

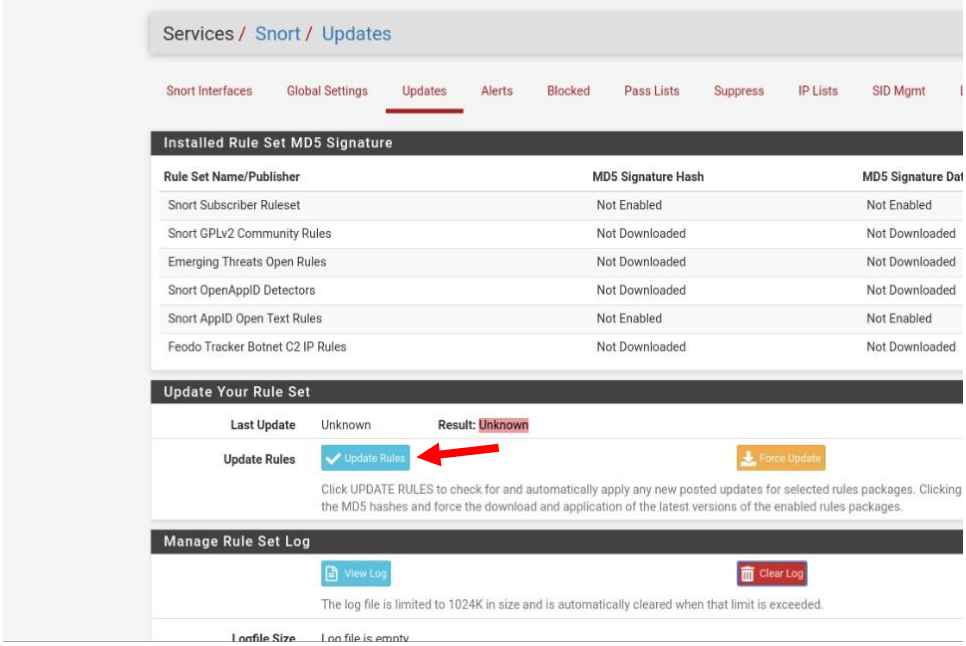


Pada Rules Update Settings, Opsi **Update Interval** digunakan untuk menerapkan waktu interval untuk pengupdate-an rules, disini kami menerapkan update setiap 12 jam. Selanjutnya, **Update Start Time** kami terapkan setiap pukul 22:01.

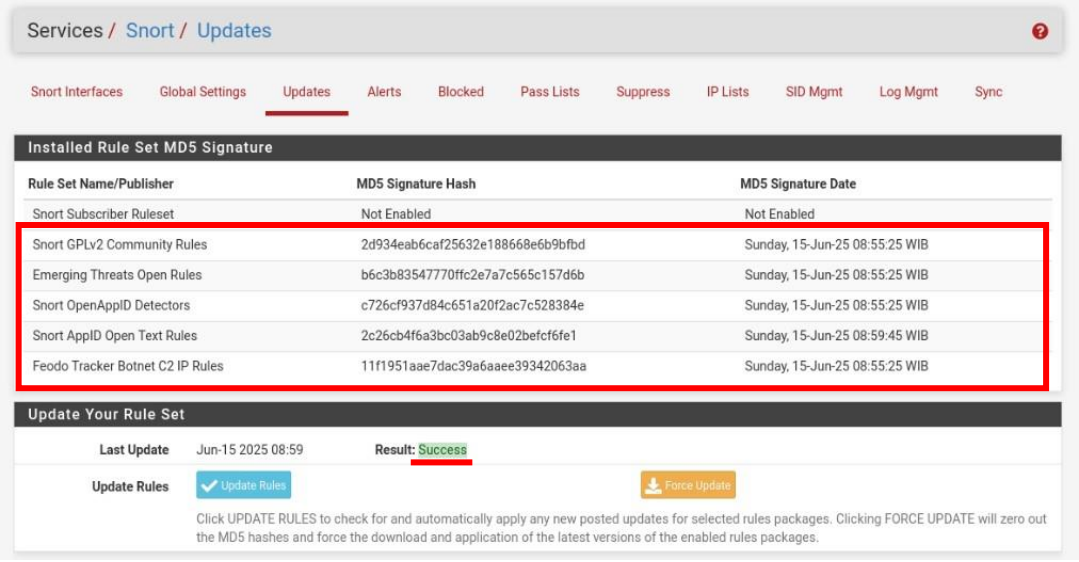
Pada General Settings**, Removed Blocked Host Interval** kami tetapkan pemblokiran sumber serangan selama 1 jam.



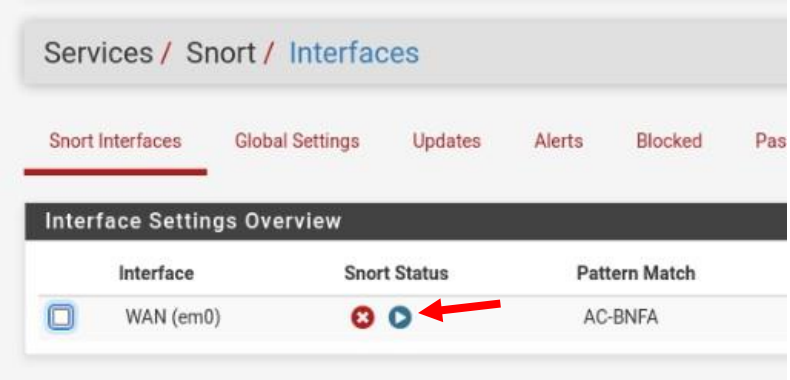
Selanjutnya, akses menu **Updates**. Terlihat bahwa beberapa rules belum diaktifkan. Untuk mengaktifkan, tekan tombol **Update Rules**.



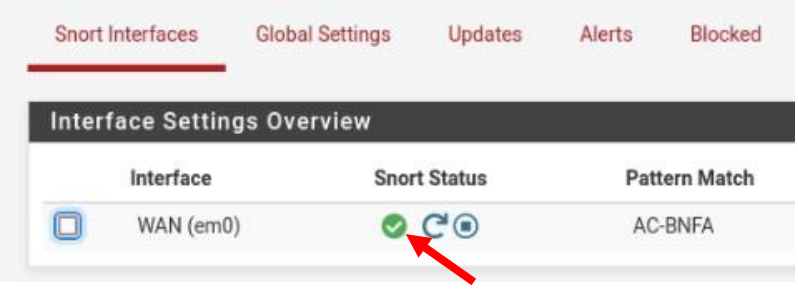
Setelah **Update Rules**, maka rules sebelumnya sudah langsung berjalan.



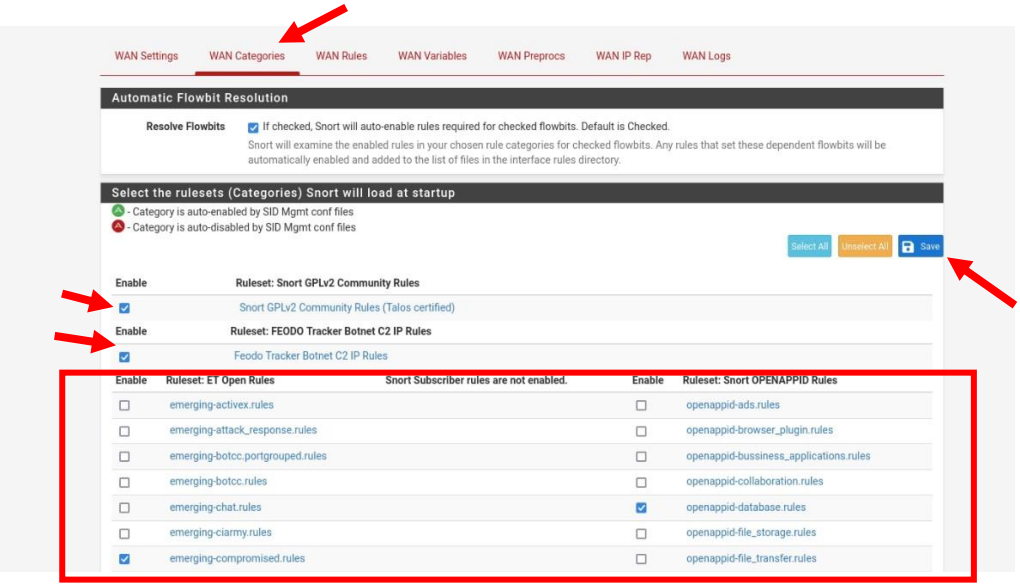
Kemudian kembali ke menu **Snort Interfaces**. Terlihat bahwa interface belum aktif, tekan tombol start seperti pada Gambar dibawah**.**



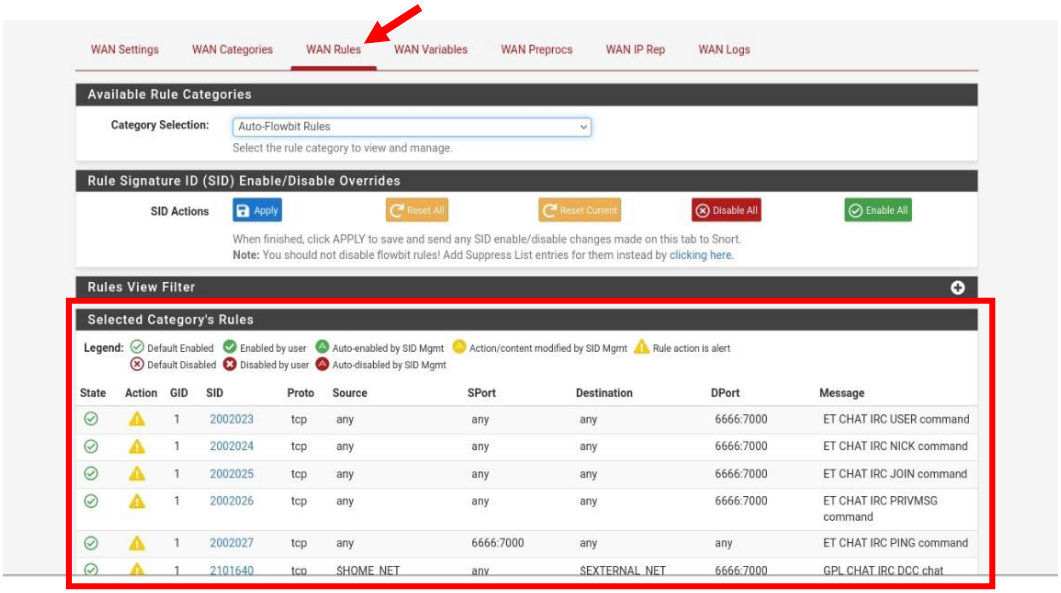
Tunggu hingga status menjadi aktif seperti pada Gambar dibawah**.**



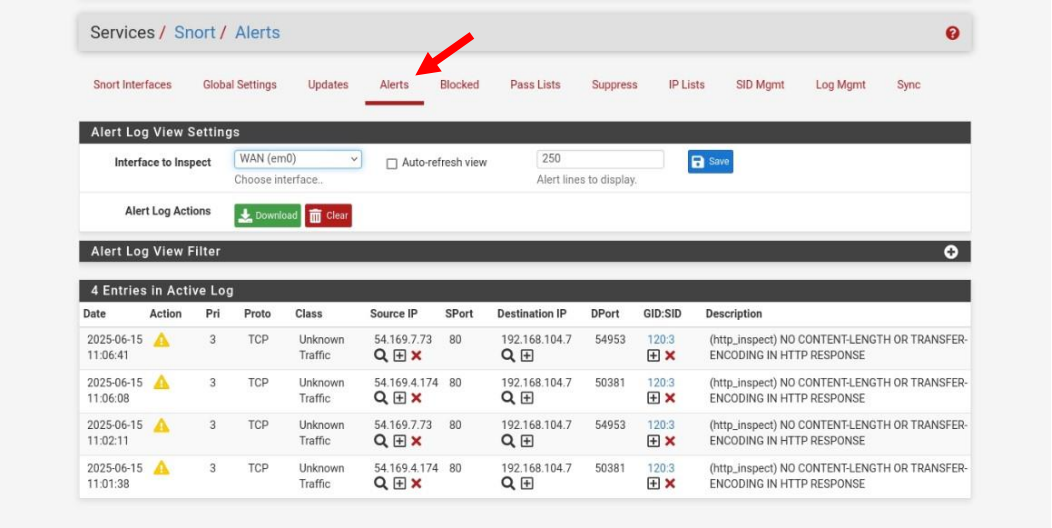
Akses menu WAN Categories. Ini adalah tahap pengaktifan beberapa ruleset yang diinginkan. Centanglah ruleset yang ingin diterapkan pada IPS. Lalu tekan **Save.**



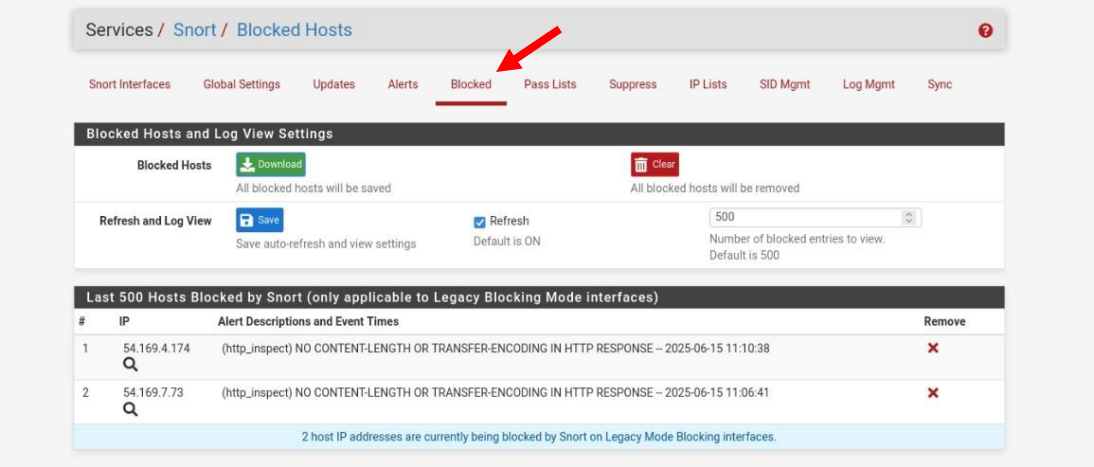
Akses menu WAN Rules. Terlihat pada Gambar dibawah bahwa sudah teraplikasi beberapa rules yang berfokus pada protokol tcp.



Keluar dari Interface WAN, lalu akses menu **Alerts** untuk melihat log alerts yang masuk.



Akses menu **Blocked** untuk melihat lalu lintas yang diblok oleh snort.



4.6 Instalasi Jaringan

1. Persiapan Jaringan Dasar

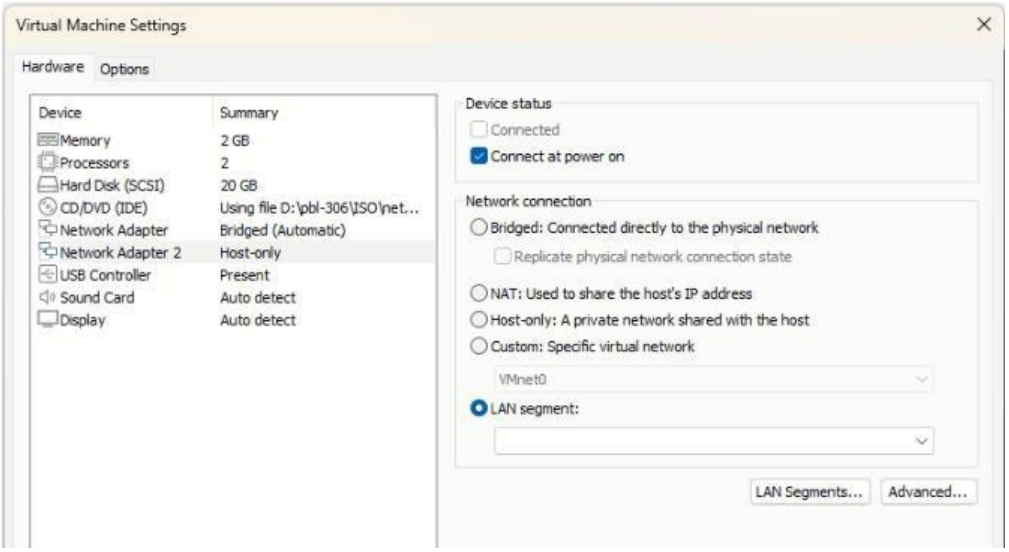
- Pembuatan Virtual Machine PfSense

Tambahkan 2 Network Adapter

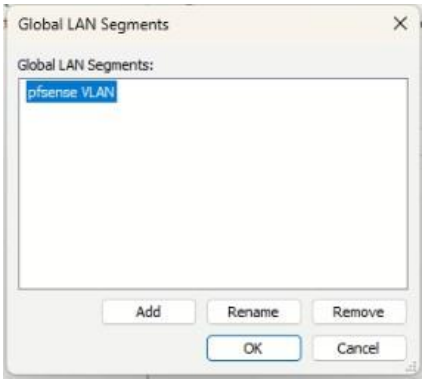
- Network Adapter 1 - NAT (akan diubah menjadi Bridge)
- Network Adapter 2 - Host-Only (Akan diubah menjadi LAN Segment)

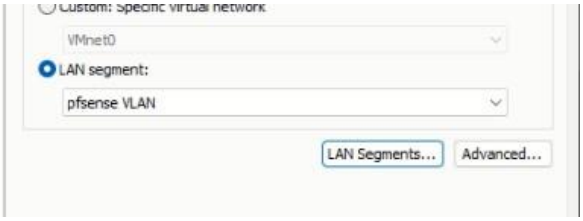
- Membuat LAN Segment

Pada Network Adapter 2 Pilih **LAN segment:** , lalu pilih tombol **LAN Segments**



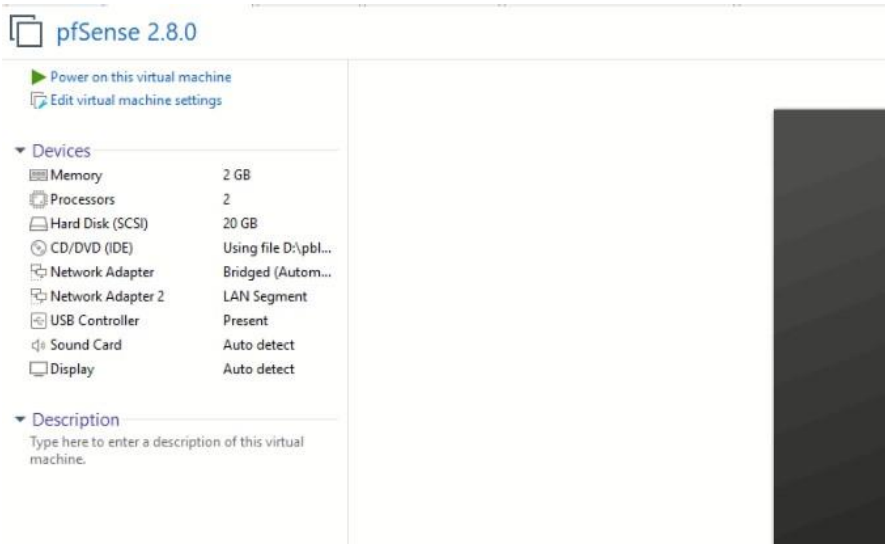
Tambahkan segmen baru dan beri nama misalnya **pfsense VLAN**, Klik OK, lalu assign LAN Segment ke Network Adapter 2

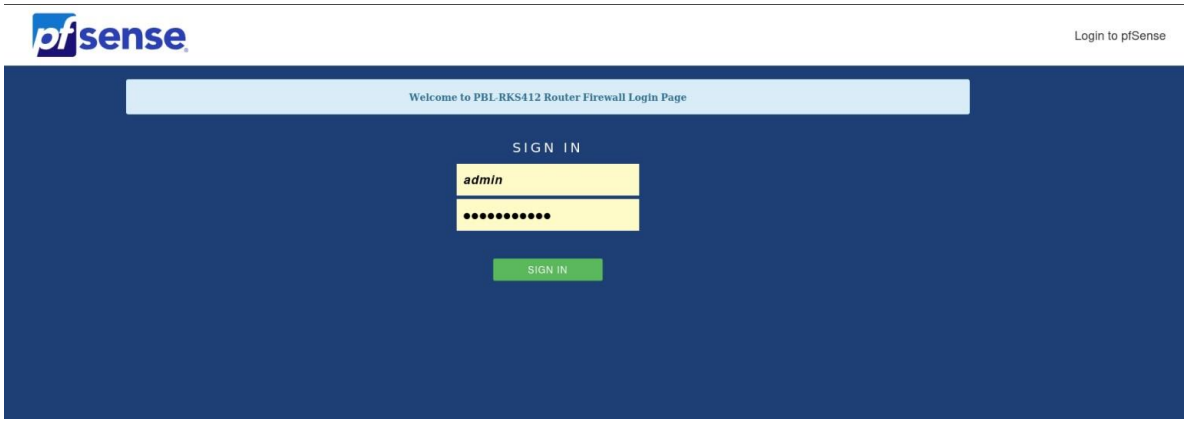




- Akses Web GUI PfSense

  Jalankan VM PfSense, lalu masuk ke Web GUI menggunakan username dan password default



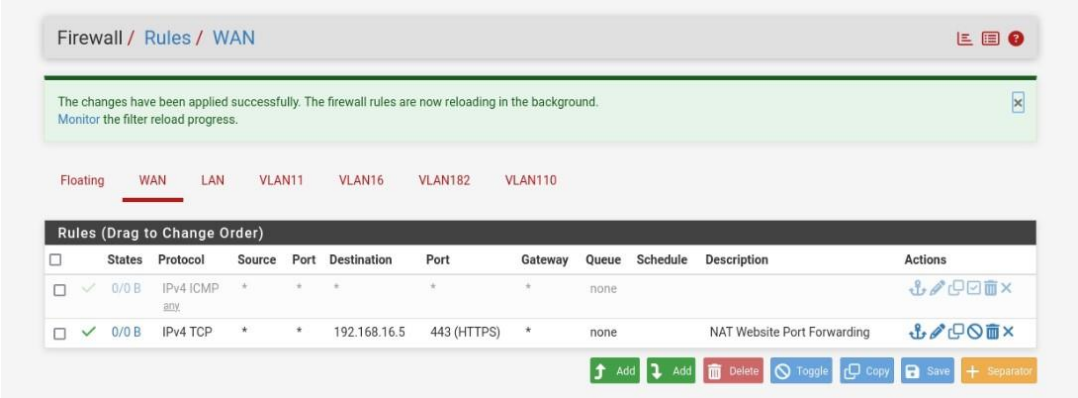


- Konfigurasi Firewall

  Masuk ke **Firewall -> Rules -> WAN**, lalu tambahkan rule port forwarding dengan konfigurasi:

  - Protocol: IPv4 TCP
  - Source: Any
  - Destination: 192.168.16.5 (Proxy server)
  - Port: 443 (HTTPS)
  - Action: Pass

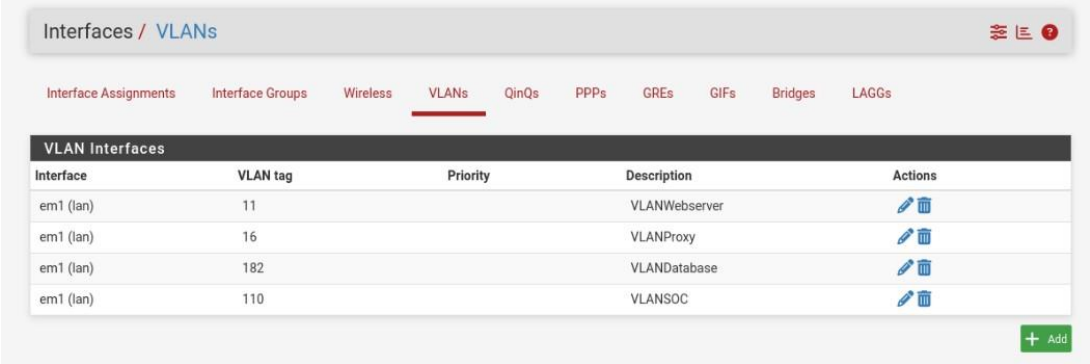
  Description: NAT Website Port Forwarding



2. Pembuatan VLAN di PfSense

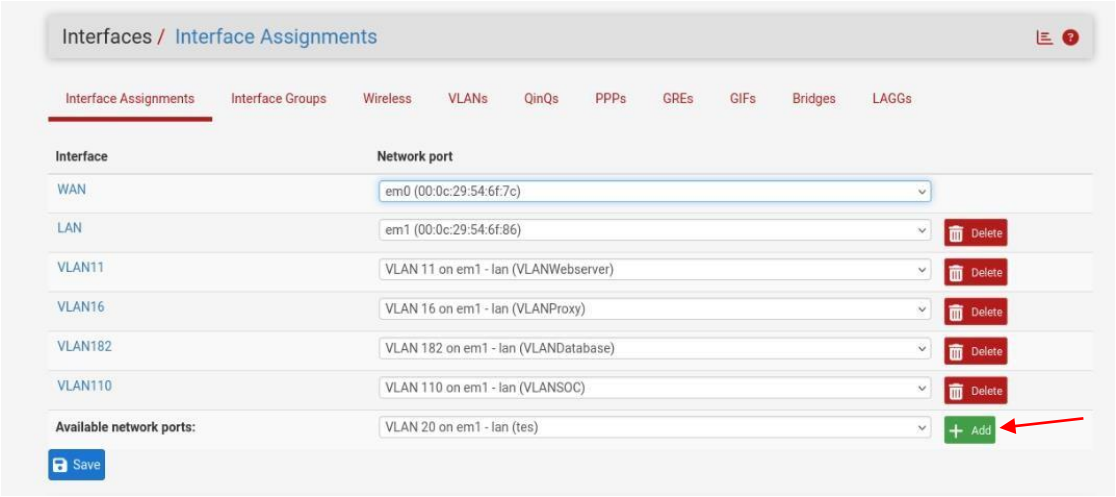
- Menambahkan VLAN

  Akses menu **Interfaces → Assignments → VLANs**, Klik tombol **+ Add.**



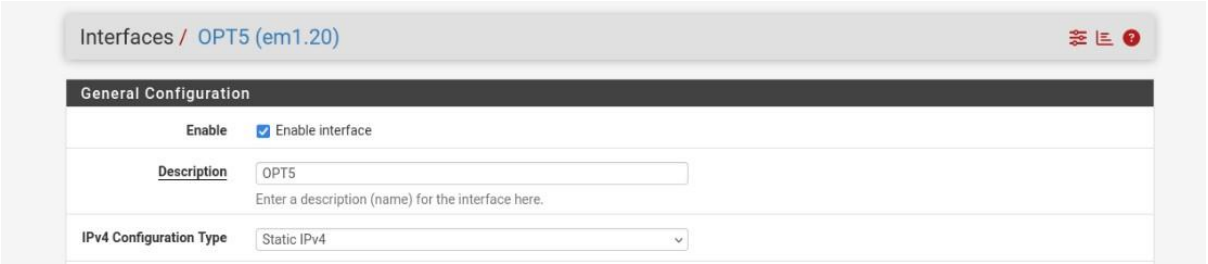
- Assign VLAN ke Interface

  Akses **Interface -> Assignments.** Assign interface VLAN yang telah dibuat

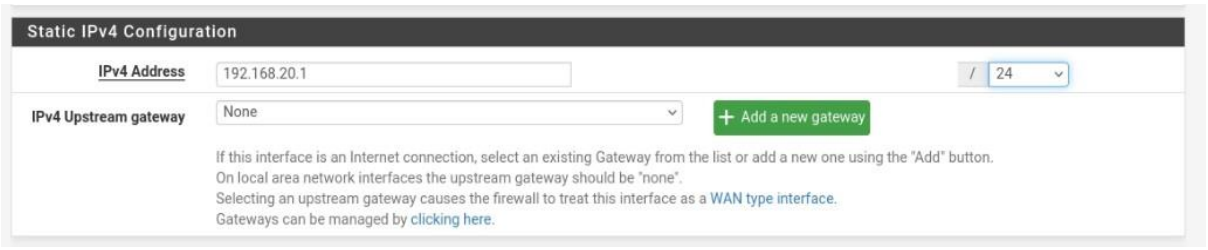


Masuk ke interface tersebut, aktifkan opsi **Enable Interface**. Ubah **IPv4**

Configuration Type menjadi **Static IPv4**

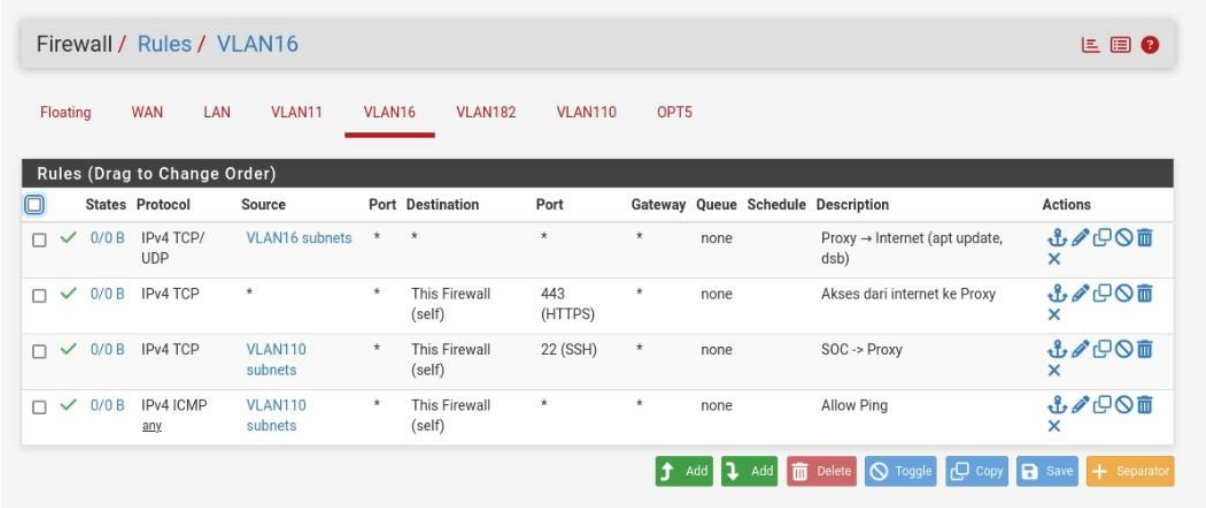


Masukkan IP Address sesuai gateway VLAN, lalu klik **Save**



3. Konfigurasi Firewall VLAN

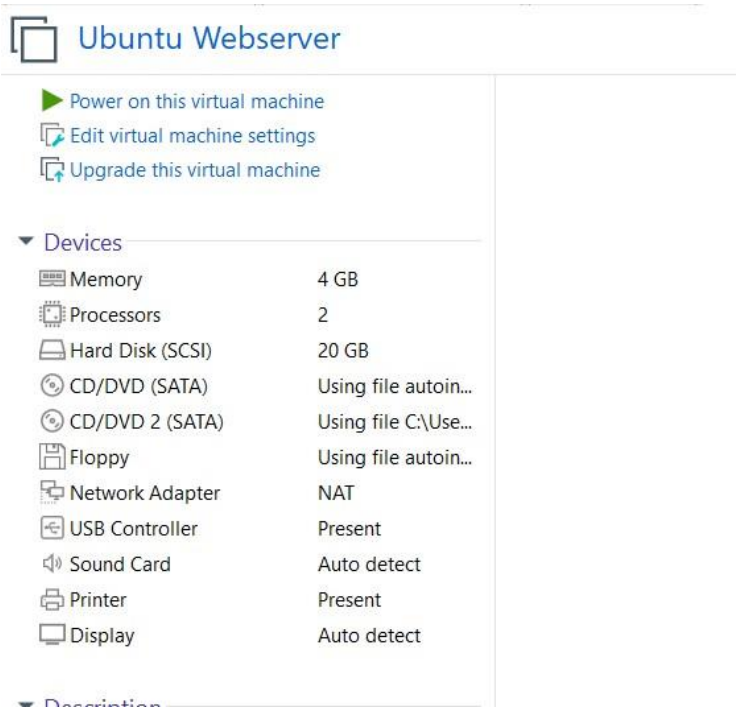
   Masuk ke **Firewall → Rules → [Nama VLAN]** (misalnya VLAN16).



4. Konfigurasi Koneksi VLAN pada Virtual Machine

   Kami menggunakan NetworkManager untuk mengatur VLAN, Jika virtual machine anda menggunakan OS CentOS seperti Rocky linux, tidak diperlukan penginstallan NetworkManager. Jika virtual machine anda menggunakan OS Debian/Ubuntu, maka diperlukan penginstallan NetworkManager terlebih dahulu.

   Pastikan network adapter menggunakan Bridge/NAT agar dapat mengakses internet



- Instalasi NetworkManager (Untuk Ubuntu/Debian)

  sudo apt update sudo apt install

  network-manager

- Konfigurasi Netplan

  Akses direktori konfigurasi:

  cd/etc/netplan

  Lihat file konfigurasi dengan ls, lalu edit:

  sudo nano [nama_file].yaml

```
pbl412@proxyserver:/$ cd /etc/netplan
pbl412@proxyserver:/etc/netplan$ ls
00-installer-config.yaml
pbl412@proxyserver:/etc/netplan$ sudo nano 00-installer-config.yaml
```

  Tambahkan atau ubah:

  renderer: NetworkManager

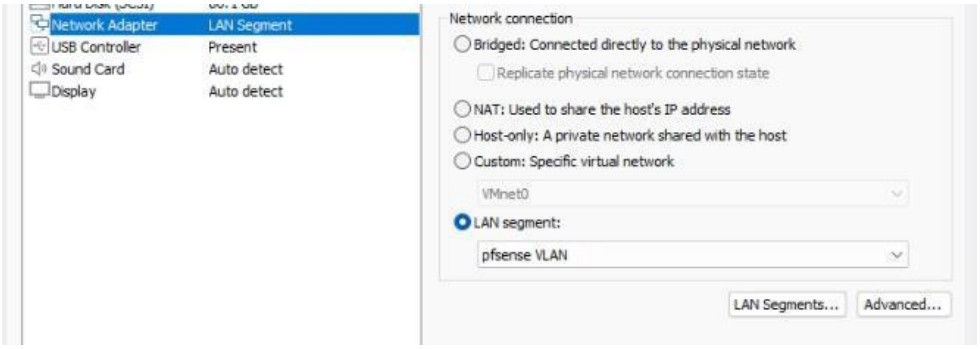
  dhcp4: no

```
network:
  renderer: NetworkManager
  version: 2
```

  Simpan, lalu jalankan: sudo netplan apply

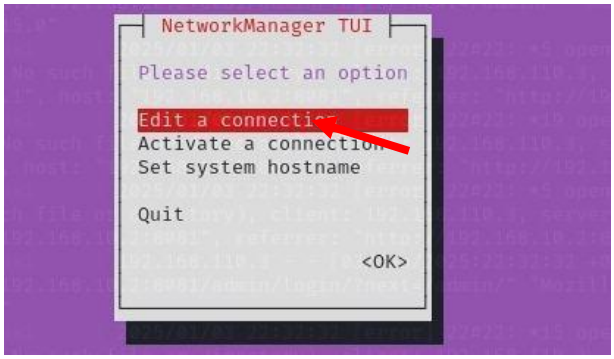
- Ubah Adapter dan Restart VM

  Ubah network adapter virtual machine menjadi **LAN Segment LAN**, lalu **restart**.

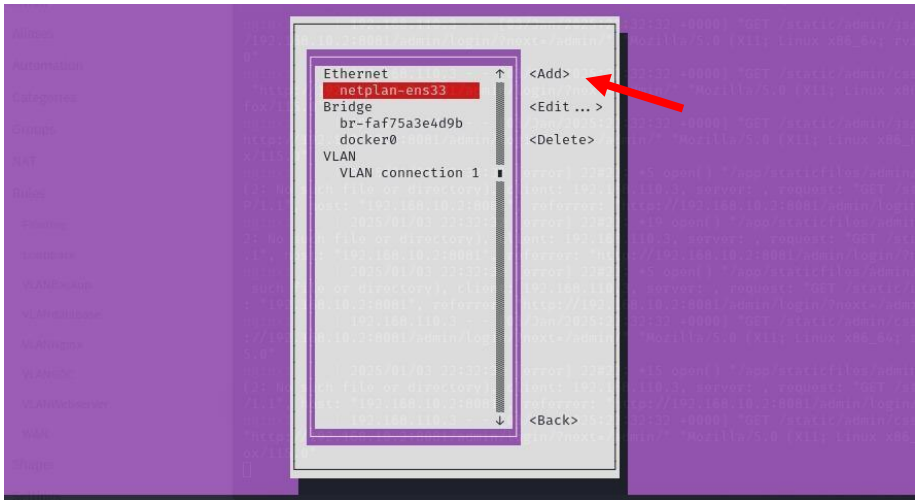


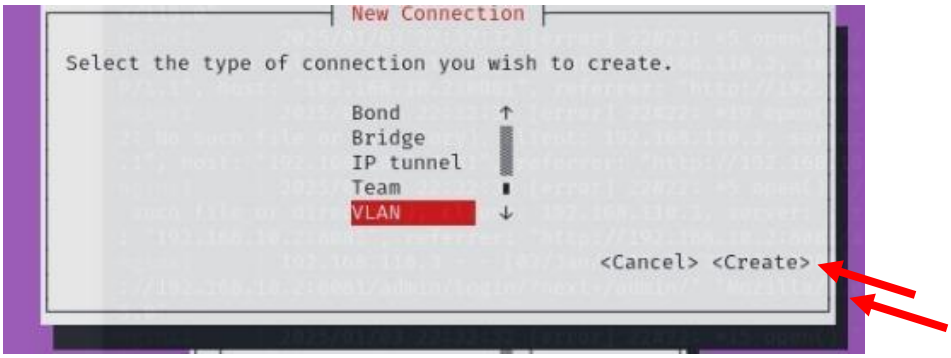
- Konfigurasi VLAN dengan nmtui

  Jalankan: nmtui, Pilih **Edit a connection.**



Jika belum ada, buat VLAN: Pilih **Add**, Scroll ke bawah, pilih **VLAN**, lalu tekan **Create**



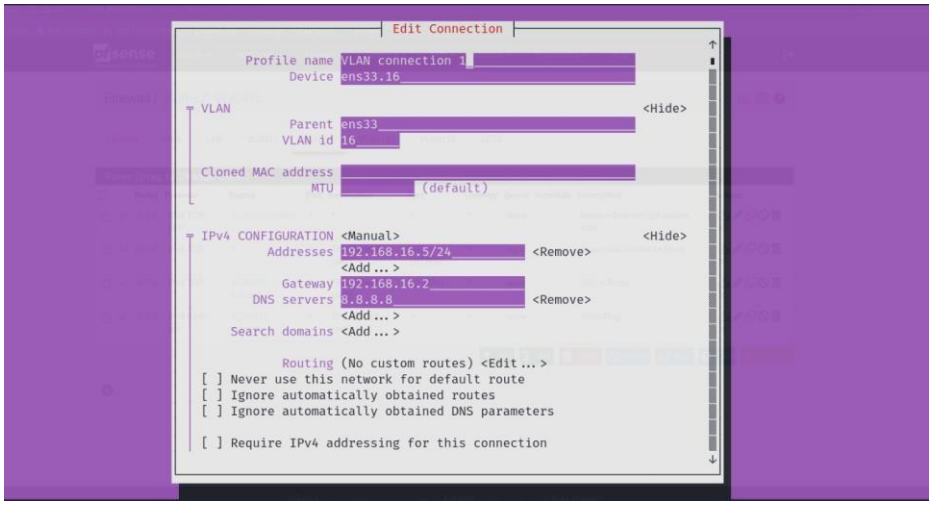


Device: ens33.16 (Interface_parent.VLAN_ID)

IP Address: 192.168.16.5/24

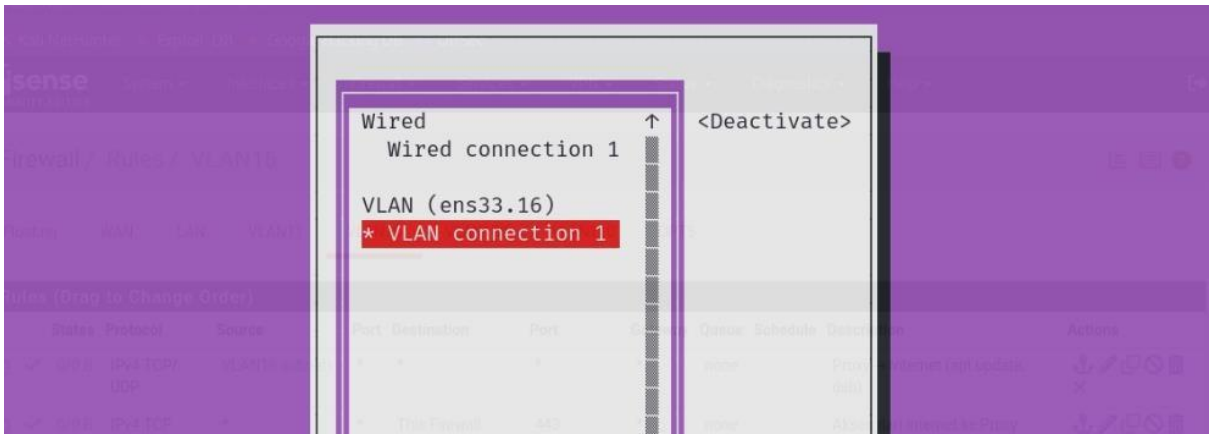
Gateway: 192.168.16.2

DNS Servers: 8.8.8.8



Setelah selesai, aktifkan koneksi VLAN melalui **Activate a connection**

Nonaktifkan Wired, lalu aktifkan VLAN



Verifikasi IP dengan: ip a

```
3: ens33.16@ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 00:0c:29:0d:3a:30 brd ff:ff:ff:ff:ff:ff
    inet 192.168.16.5/24 brd 192.168.16.255 scope global noprefixroute ens33.16
       valid_lft forever preferred_lft forever
    inet6 fe80::d9f1:20e2:637:8ab4/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Maka VLAN sudah terhubung di virtual machine

- Konfigurasi Port Forwarding di PfSense (NAT)

Untuk mengarahkan akses dari internet ke server lokal (misalnya proxy server HTTPS), ikuti langkah berikut:

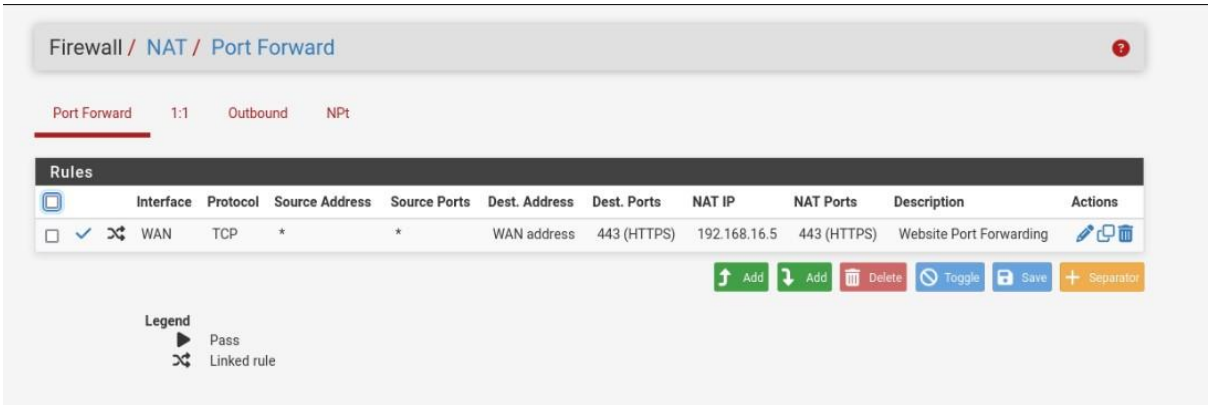
Masuk ke menu **Firewall → NAT → Port Forward**

Klik tombol **+ Add** untuk membuat rule baru

Isi konfigurasi sebagai berikut:

- Interface: WAN
- Protocol: TCP
- Destination: WAN address
- Destination Port Range: HTTPS (443)
- Redirect Target IP: 192.168.16.5 (NAT IP)
- Redirect Target Port: 443 HTTPS (NAT Ports)

Description: Website Port Forwarding



Centang "Filter rule association": pilih Add associated filter rule

Klik **Save**, lalu **Apply Changes.**

4.7 Otomasi Backup Database

1. Instalasi Rclone

   Pada server database, instal terlebih dahulu paket rclone dengan perintah:

   sudo dpkg -i rclone-current-linux-amd64.deb

```
root@database:/# sudo dpkg -i rclone-current-linux-amd64.deb
Selecting previously unselected package rclone.
(Reading database ... 113141 files and directories currently installed.)
Preparing to unpack rclone-current-linux-amd64.deb ...
Unpacking rclone (1.70.2) ...
Setting up rclone (1.70.2) ...
Processing triggers for man-db (2.10.2-1) ...
root@database:/# sudo apt install fuse3
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
fuse3 is already the newest version (3.10.5-1build1).
fuse3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 42 not upgraded.
```

Kemudian instal dependensi yang diperlukan, yaitu fuse: sudo apt install fuse3

```
root@database:/# sudo apt install fuse3
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
fuse3 is already the newest version (3.10.5-1build1).
fuse3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 42 not upgraded.
```

2. Konfigurasi Rclone

   Jalankan perintah berikut untuk memulai konfigurasi:

   rclone config

- Ketik n untuk membuat remote baru.
- Masukkan nama remote sesuai keinginan (nama ini akan digunakan saat proses mounting).

```
root@database:/# rclone config
2025/07/08 07:10:44 NOTICE: Config file "/root/.config/rclone/rclone.conf" not found
 - using defaults
No remotes found, make a new one?
n) New remote
s) Set configuration password
q) Quit config
n/s/q> n

Enter name for new remote.
name> pbl412
```

- Saat diminta memilih jenis storage, pilih Google Drive (nomor 22)
- Biarkan kosong saat diminta memasukkan client_id dan client_secret.

```
Storage> 22

Option client_id.
Google Application Client Id
Setting your own is recommended.
See https://rclone.org/drive/#making-your-own-client-id for how to create your own.
If you leave this blank, it will use an internal key which is low performance.
Enter a value. Press Enter to leave empty.
client_id>

Option client_secret.
OAuth Client Secret.
Leave blank normally.
Enter a value. Press Enter to leave empty.
client_secret>
```

- Pilih scope 1 (Full access).

```
   \ (drive.file)
   / Allows read and write access to the Application Data folder.
 4 | This is not visible in the drive website.
   \ (drive.appfolder)
   / Allows read-only access to file metadata but
 5 | does not allow any access to read or download file content.
   \ (drive.metadata.readonly)
scope> 1
```

- Untuk opsi Edit advanced config, pilih n (No).
- Untuk opsi autentikasi menggunakan auto config, pilih n.

```
Edit advanced config?
y) Yes
n) No (default)
y/n> n

Use web browser to automatically authenticate rclone with remote?
 * Say Y if the machine running rclone has a web browser you can use
 * Say N if running rclone on a (remote) machine without web browser access
If not sure try Y. If Y failed, try N.

y) Yes (default)
n) No
y/n> n
```

Pada tahap ini, mengharuskan pengguna untuk menjalankan perintah yang membutuhkan mesin yang memiliki web browser. Salin perintah ini.

```
For this to work, you will need rclone available on a machine that has
a web browser available.
For more help and alternate methods see: https://rclone.org/remote_setup/
Execute the following on the machine with the web browser (same rclone
version recommended):
        rclone authorize "drive" "eyJzY29wZSI6ImRyaXZlIn0"
```

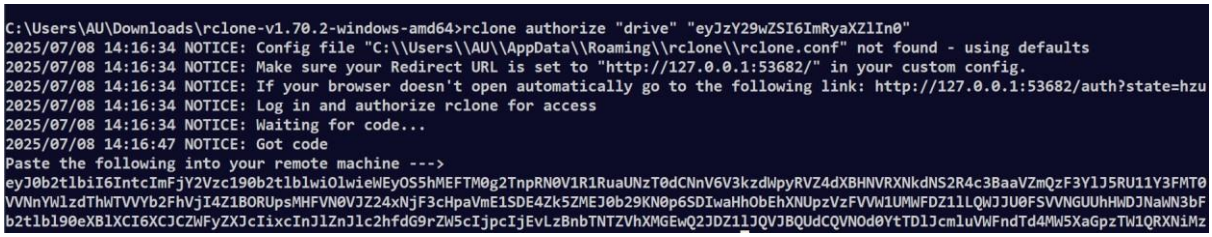
3. Otorisasi Rclone di Windows

   Karena proses autentikasi memerlukan web browser, lakukan langkah berikut di mesin Windows: Instal Rclone dari situs resmi.

   Buka Command Prompt dan masuk ke direktori tempat rclone berada.

Jalankan perintah berikut (yang sebelumnya disalin dari terminal server):

rclone authorize “drive” “**koderandom**”

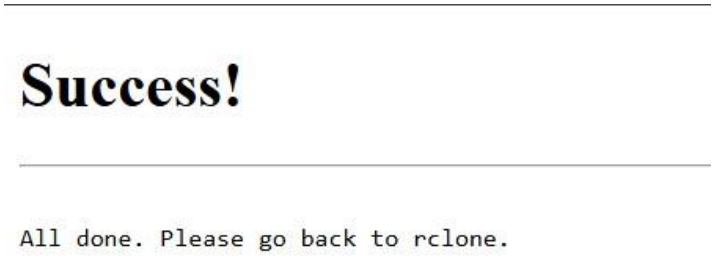


Salin lah config_token yang muncul untuk di tempelkan ke terminal database.

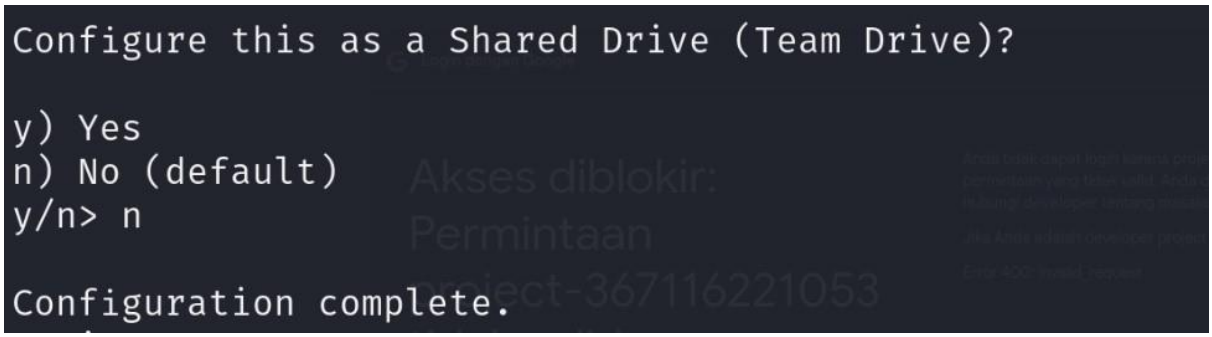
4. Selessaikan Knfigurasi di Server

Kembali ke terminal database, tempel config_token yang diminta.

Kembali ke windows, maka akan muncul bahwa rclone sudah sukses.



Kembali ke terminal database, ketik n .



Ketik Y. untuk menyimpan konfigurasi remote.

```
Keep this "pbl412" remote?
y) Yes this is OK (default)
e) Edit this remote
d) Delete this remote
y/e/d> y
```

5. Mount Google Drive

- Buat direktori mount point terlebih dahulu, misalnya:

  mkdir /home/backup

- Kemudian lakukan mounting: rclone mount **pbl412: /home/backup &**

```
root@database:/home/pbl306# cd /home/backup
root@database:/home/backup# rclone mount pbl412: /home/backup &
[1] 2522
```

6. Otomasi Backup dengan Bash Script dan Corn

- Akses direktori berikut:

  cd /usr/local/bin

- Buat file bash script, misalnya backup_bagisto.sh, dan isi sesuai kebutuhan backup.

```
root@database: /home/backup
File  Actions  Edit  View  Help
  GNU nano 6.2                    /usr/local/bin/backup_bagisto.sh
#!/bin/bash

# Nama database
DB_NAME="Bagisto"
# Lokasi backup lokal
BACKUP_DIR="/home/backup"
# Nama file backup pakai tanggal
DATE=$(date +%Y-%m-%d)
BACKUP_FILE="$BACKUP_DIR/bagisto_backup_$DATE.sql"

# Backup database (tanpa password karena root tidak pakai password)
mysqldump -u root $DB_NAME > "$BACKUP_FILE"

# Upload ke Google Drive remote bernama 'pbl412', folder 'bagisto-backup'
rclone copy "$BACKUP_FILE" pbl412:bagisto-backup --ignore-existing -v
```

- Ubah permission agar bisa dieksekusi:

  chmod +x backup.sh

- Tambahkan cron job

  Lakukan otomasi dengan menggunakan cronjob, ketik crontab -e lalu buatlah jadwal yang di inginkan.

  Contoh cronjob untuk menjalankan backup setiap hari Minggu pukul 23.00 dan menyimpan log.

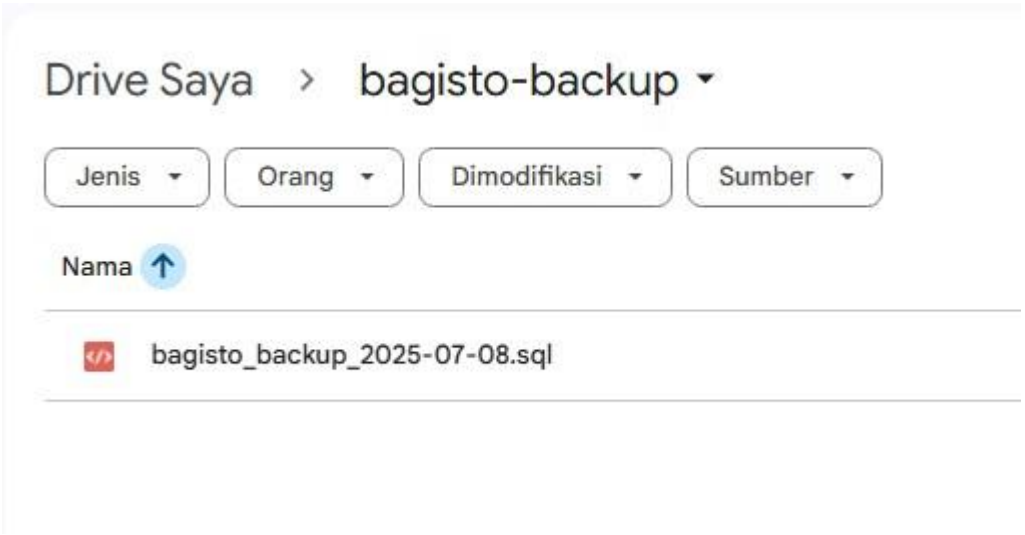
```
#
# For more information see the manual pages of crontab(5) and cron(8)
#
0 23 * * 0 /usr/local/bin/backup_bagisto.sh >> /var/log/bagisto_backup.log 2>&1
```

Untuk pengujian, bisa diatur agar berjalan setiap menit:

* * * * * /usr/local/bin/backup_bagisto.sh

7. Verifikasi

   Tunggu sesuai waktu yang dijadwalkan, kemudian buka Google Drive Anda. Jika berhasil, file backup dari server akan muncul dan tersinkronisasi dengan akun Google Drive.



4.8 install The Hive dan Cortex

1. Inatall docker engine

   “ sudo apt install docker “

2. Install docker compose

   “ sudo apt install docker-compose “

3. Membuat direktori tempat the hive dan cortex berjalan

   “ mkdir thehive-cortex “

4. Membuat file konfigurasi docker compose

   “ sudo nano docker-compose.yml “

5. Isi file konfigurasi docker compose

```
version: "3.7"

services:
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.17.18
    environment:
      - discovery.type=single-node
      - xpack.security.enabled=false
      - cluster.name=hive
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms1g -Xmx1g"
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - esdata:/usr/share/elasticsearch/data
    networks:
      - hive

  cortex:
    image: thehiveproject/cortex:3.1.1
    ports:
      - "9001:9001"
    environment:
      - jobDirectory=/tmp/cortex-jobs
      - cortex.external-url=http://localhost:9001
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - /tmp/cortex-jobs:/tmp/cortex-jobs
    depends_on:
      - elasticsearch
    networks:
```

```
      - hive

  thehive:
    image: strangebee/thehive:5.2
    ports:
      - "9000:9000"
    environment:
      - "THEHIVE_elasticsearch.hosts=[\"http://elasticsearch:9200\"]"
      - "THEHIVE_cortex.cortex-1.url=http://cortex:9001"
      - "THEHIVE_cortex.cortex-1.name=Cortex"
    depends_on:
      - elasticsearch
      - cortex
    networks:
      - hive

volumes:
  esdata:
  cortex-data:

networks:
  hive:
    driver: bridge
```

6. Install the hive dan cortex
   “ sudo docker-compose up -d “

7. Melihat container yang berjalan dan mengetahui tiap-tiap port yang di gunakan oleh the hive dan cortex
   “ sudo docker ps”

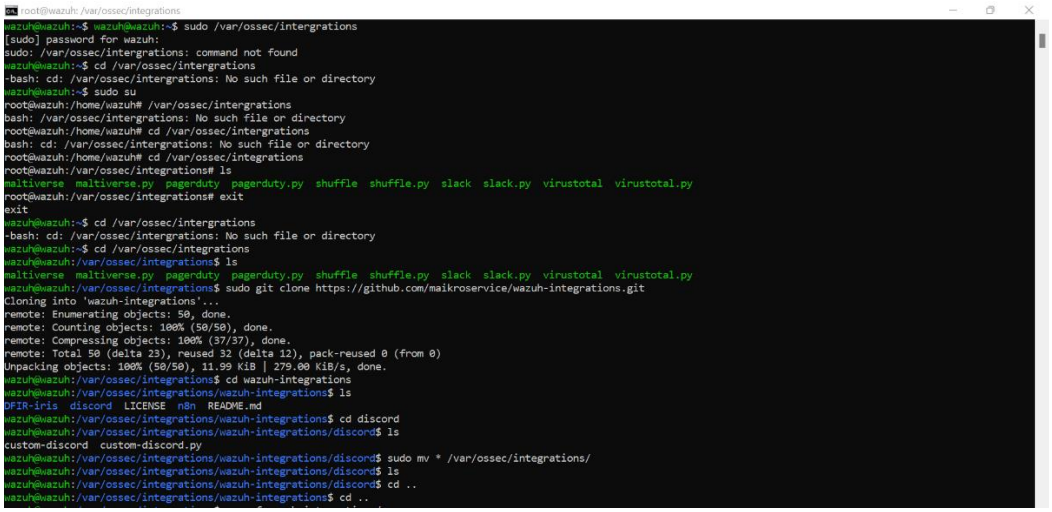
4.9 Integrasi The Hive dan Wazuh

1. Install Wazuh
   curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash ./wazuh-install.sh -a

2. Cloning Repo

   Menclone repo GitHub berisi skrip integrasi pihak ketiga untuk Wazuh. Lalu masuk ke *cd wazuh-integrations/discords/* dan melihat isi folder *discord/* ada file *custom-discord.py* dan mencoba memindahkannya menggunakan perintah

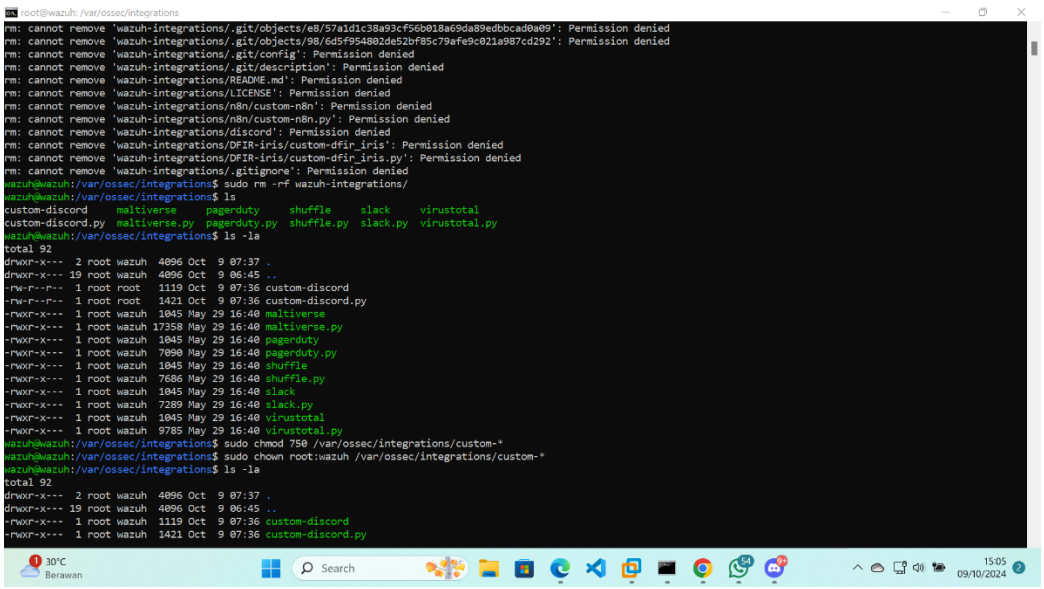
   sudo mv * /var/ossec/integrations/



   Dan selanjutnya menghapus.

3. Menghapus dan Set Permission

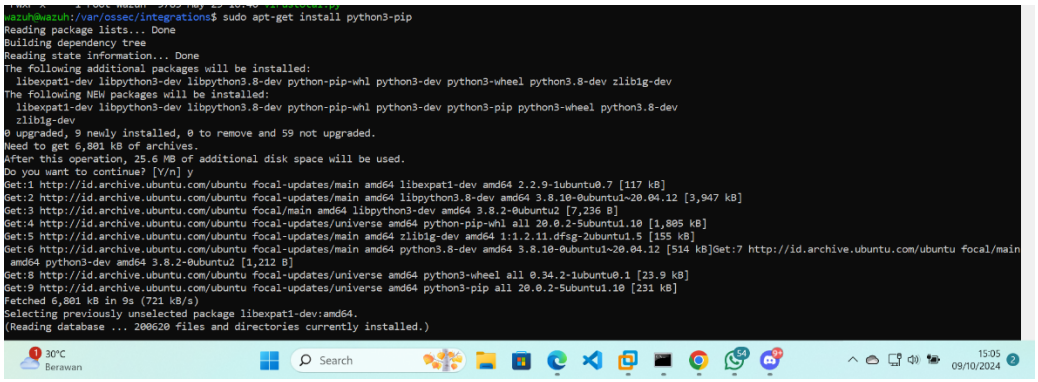
   Gunakan perintah sudo rm -rf wazuh-integrations/, lalu cek isi folder masih ada atau tidak. Kemudian kita mengatur permission agar hanya owner (root) bisa baca, tulis, dan eksekusi: wazuh hanya bis abaca dan eksekusi.



4. Install Python3

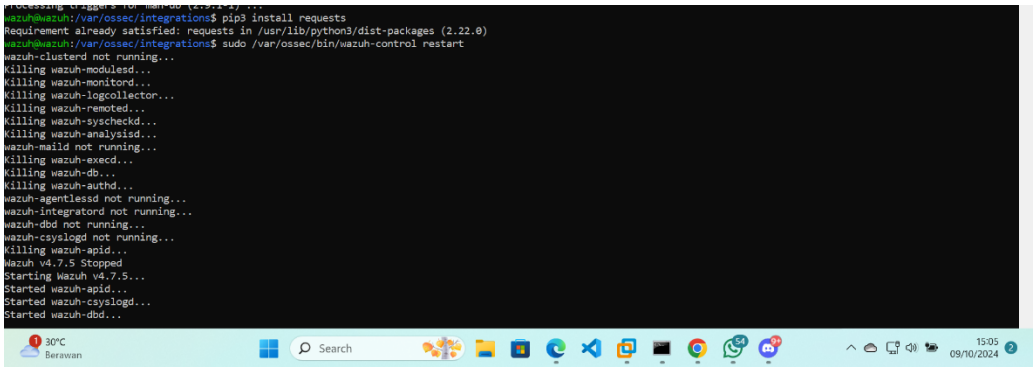
   Untuk bisa install library python (misalnya requests) yang dibutuhkan skrip integrasi. Sistem otomatis menginstall depedensi tambahan seperti python3-dev, python3-wheel, zlibig-dev, dll.

sudo apt-get install python3-pip



5. Install library dan restart Wazuh

   Diperlukan agar python script bisa kirim HTTP request ke discord.



4.9 install The Hive dan Cortex

8. Inatall docker engine

   “ sudo apt install docker “

9. Install docker compose

   “ sudo apt install docker-compose “

10. Membuat direktori tempat the hive dan cortex berjalan

    “ mkdir thehive-cortex “

11. Membuat file konfigurasi docker compose

    “ sudo nano docker-compose.yml “

12. Isi file konfigurasi docker compose

    version: "3.7"

```
services:
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.17.18
    environment:
      - discovery.type=single-node
      - xpack.security.enabled=false
      - cluster.name=hive
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms1g -Xmx1g"
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - esdata:/usr/share/elasticsearch/data
    networks:
      - hive

  cortex:
    image: thehiveproject/cortex:3.1.1
    ports:
      - "9001:9001"
    environment:
      - jobDirectory=/tmp/cortex-jobs
      - cortex.external-url=http://localhost:9001
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - /tmp/cortex-jobs:/tmp/cortex-jobs
    depends_on:
      - elasticsearch
    networks:
      - hive
```

```
  thehive:
    image: strangebee/thehive:5.2
    ports:
      - "9000:9000"
    environment:
      - "THEHIVE_elasticsearch.hosts=[\"http://elasticsearch:9200\"]"
      - "THEHIVE_cortex.cortex-1.url=http://cortex:9001"
      - "THEHIVE_cortex.cortex-1.name=Cortex"
    depends_on:
      - elasticsearch
      - cortex
    networks:
      - hive

volumes:
  esdata:
  cortex-data:

networks:
  hive:
    driver: bridge
```

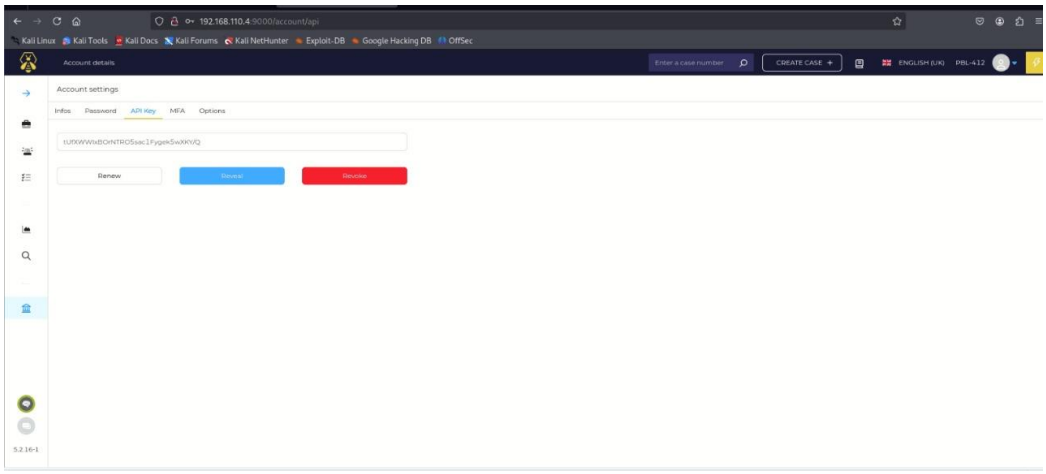
13. Install the hive dan cortex
    “ sudo docker-compose up -d “

14. Melihat container yang berjalan dan mengetahui tiap-tiap port yang di gunakan oleh the hive dan cortex
    “ sudo docker ps”

5.0 Integrasi Wazuh

```
<integration>
  <name>custom-w2thive</name>
  <hook_url>http://192.168.110.4:9000</hook_url>
  <api_key>tUfXWWlxBOrNTRO5sac1Fygek5wXKY/Q</api_key>
  <alert_format>json</alert_format>
  <level>10</level>
</integration>
```

Ini untuk mengambil API key wazuh



5.1 Install Grafana

1. Install docker engine

   “ sudo apt install docker-ce “

2. Install docker compose”

   “ sudo apt install docker-compose “

3. Buat direktori tempat menjalankan Grafana dan Prometheus

   “ mkdir monitoring “

4. Buat file konfigurasi docker-compose untuk Grafana dan Prometheus

   “ sudo nano docker-compose.yml “

5. Isi file konfigurasi docker compose

   version: '3'

   services:
     prometheus:
       image: prom/prometheus
       container_name: prometheus
       volumes:
         - ./prometheus.yml:/etc/prometheus/prometheus.yml
       command:

```
    - --config.file=/etc/prometheus/prometheus.yml
   ports:
    - "9090:9090"
   networks:
    - monitoring

  node_exporter:
   image: prom/node-exporter
   container_name: node_exporter
   ports:
    - "9100:9100"
   networks:
    - monitoring

  grafana:
   image: grafana/grafana:latest
   container_name: grafana
   volumes:
    - grafana_data:/var/lib/grafana
   ports:
    - "3000:3000"
   networks:
    - monitoring

volumes:
  grafana_data:

networks:
  monitoring:
```

6. Install atau menjalankan Grafana dan Prometheus
   “ sudo docker-compose up -d “

7. Perintah untuk melihat container apa yang berjalan dan melihat port yang sedang di gunakan

“ sudo docker ps”

5.2 Konfigurasi Wazuh dengan Virustotal

1. Di bagian konfigurasi wazuh tambahkan

```
<integration>
  <name>virustotal</name>
  <api_key>eff72cfb14d2002c88ce2e72f0121557e73307be98f6d0ebd5a0d9e03d80b0fd</api_key> <!-- Replace with your VirusTotal API key -->
  <group>syscheck</group>
  <alert_format>json</alert_format>
</integration>
```

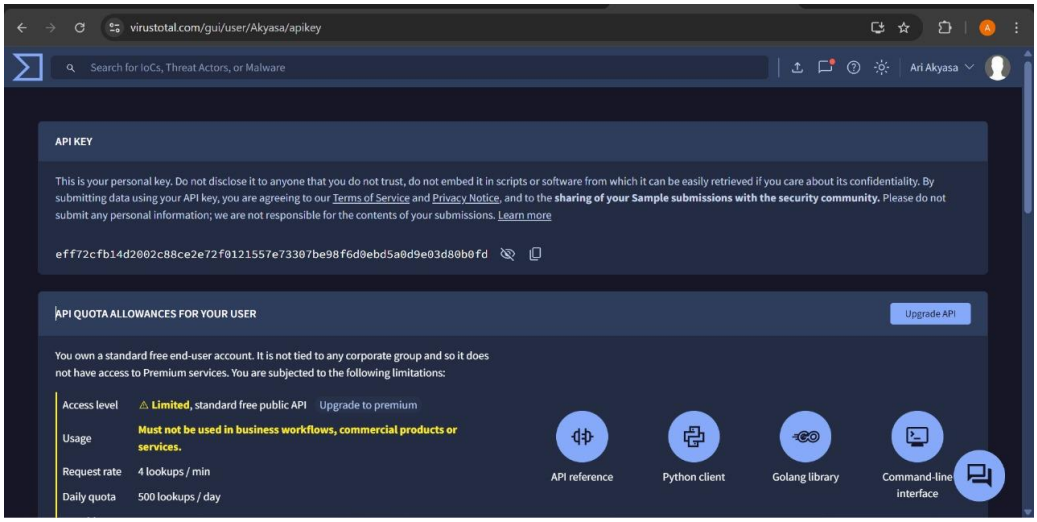
Setelah itu klik save dan restart manager di pojok kanan atas

2. Di bagian Group di wazuh

Selanjutnya klik save di pojok kanan atas

3. Ambil API Key

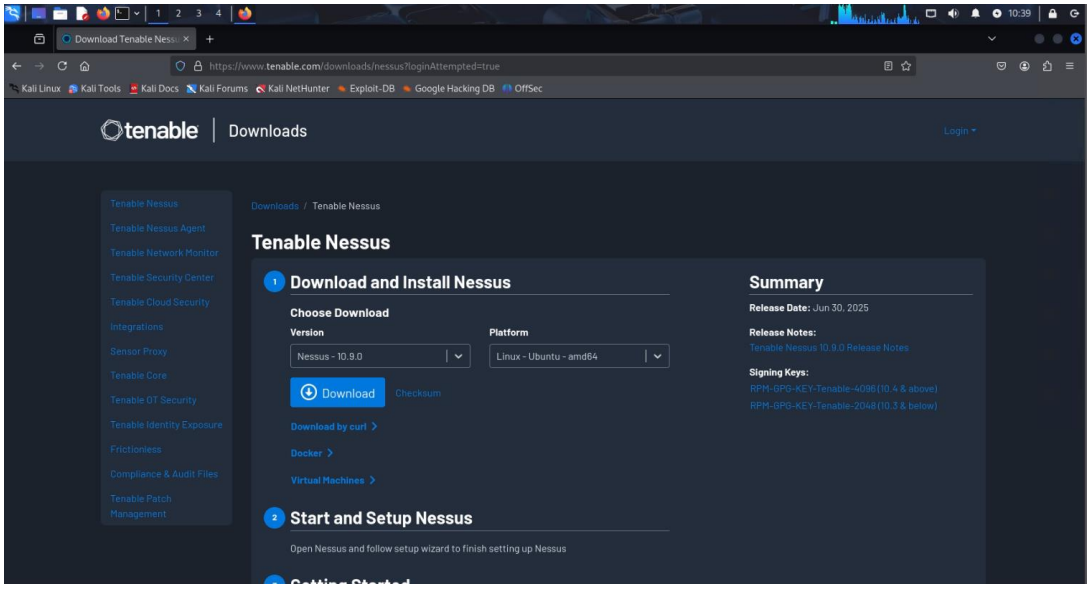
Diisni untuk mengambil API key virus total untuk menyambungkan ke Wazuh tapi sebelum buat akun virustotalnya dulu baru mengambil API keynya.



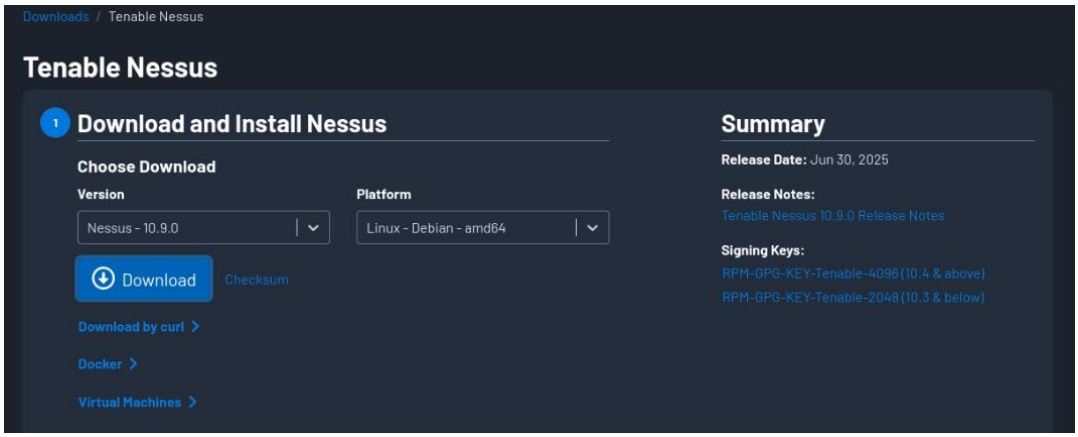
5.3 Vulnerability Assessment (Nessus)

1. Unduh Nessus

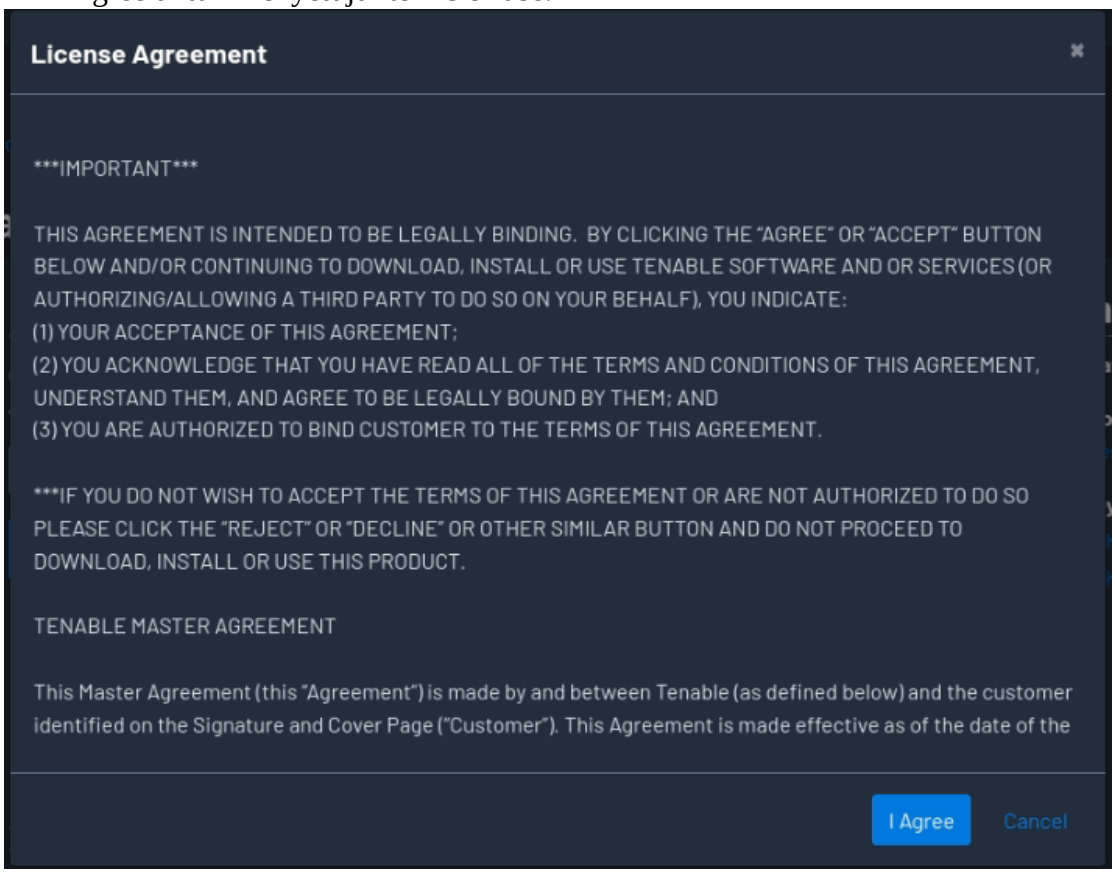
https://www.tenable.com/downloads/nessus



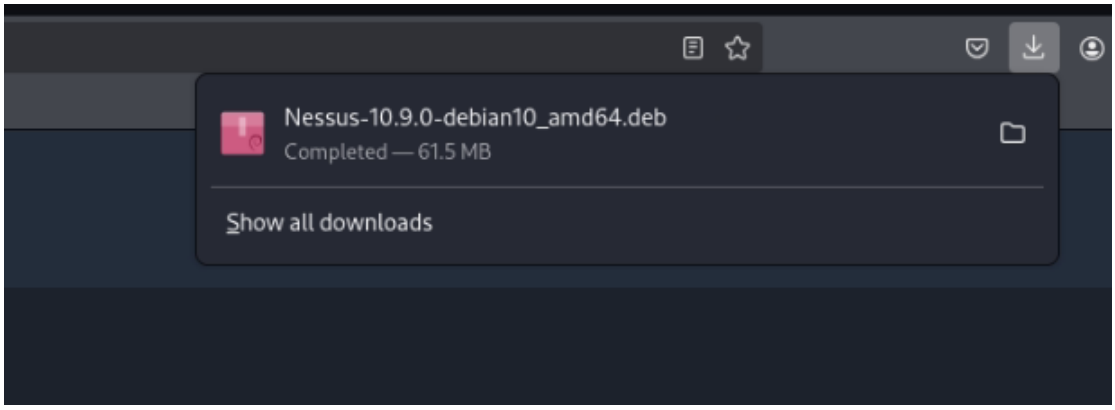
Pilih platform Debian.



Klik Agree untuk menyetujui terms of use.

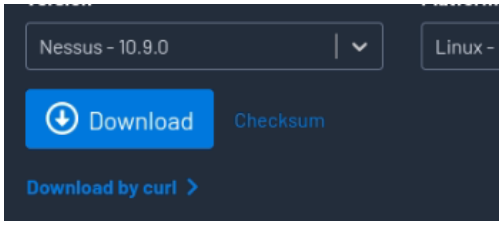


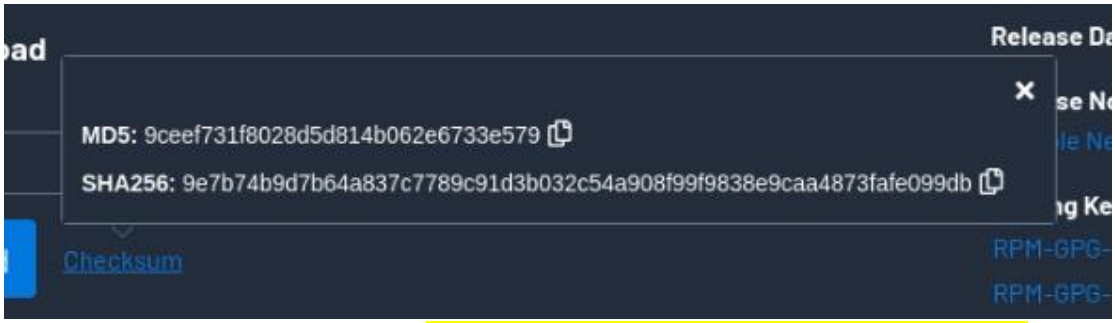
Setelah proses unduhan selesai, simpan file installer .deb di direktori Downloads.



2. Verifikasi Checksum

Salin **Checksum**

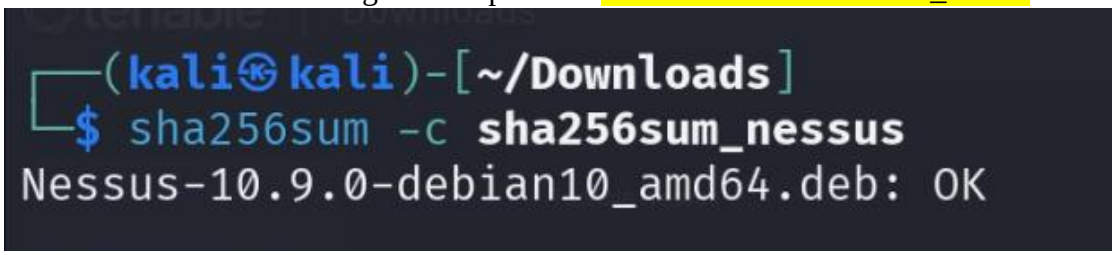




Pada direktori Downloads, ketik echo ***[checksum sha256]*** > sha256sum_nessus

File sudah dibuat

Lalu lakukan checksum dengan ketik perintah sha256sum -c sha256sum_nessus



3. Instalasi Nessus

Jalankan perintah berikut:
sudo apt install ./Nessus-10.9.0-debian10_amd64.deb

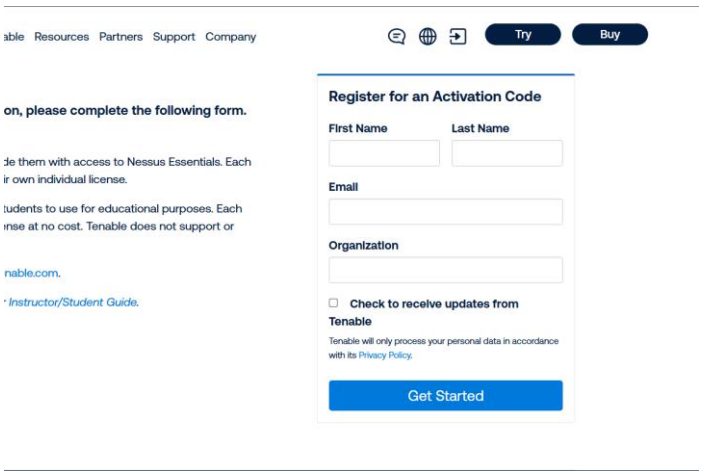
```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo apt install ./Nessus-10.9.0-debian10_amd64.deb
[sudo] password for kali:
Note, selecting 'nessus' instead of './Nessus-10.9.0-debian10_amd64.deb'
Installing:
  nessus

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 1316
  Download size: 0 B / 64.5 MB
  Space needed: 0 B / 60.1 GB available

Get:1 /home/kali/Downloads/Nessus-10.9.0-debian10_amd64.deb nessus amd64 10.9.0 [64.5
MB]
Selecting previously unselected package nessus.
(Reading database ... 408594 files and directories currently installed.)
RSA_Decrypt : (KAT_AsymmetricCipher) : Pass
RSA_Decrypt : (KAT_AsymmetricCipher) : Pass
INSTALL PASSED
Unpacking Nessus Scanner Core Components ...

 - You can start Nessus Scanner by typing /bin/systemctl start nessusd.service
 - Then go to https://NESSUS_HOSTNAME_OR_IP:8834/ to configure your scanner

Notice: Download is performed unsandboxed as root as file '/home/kali/Downloads/Nessus
-10.9.0-debian10_amd64.deb' couldn't be accessed by user '_apt'. - pkgAcquire::Run (13
: Permission denied)
```

Jika terjadi error, maka ulangi sekali lagi

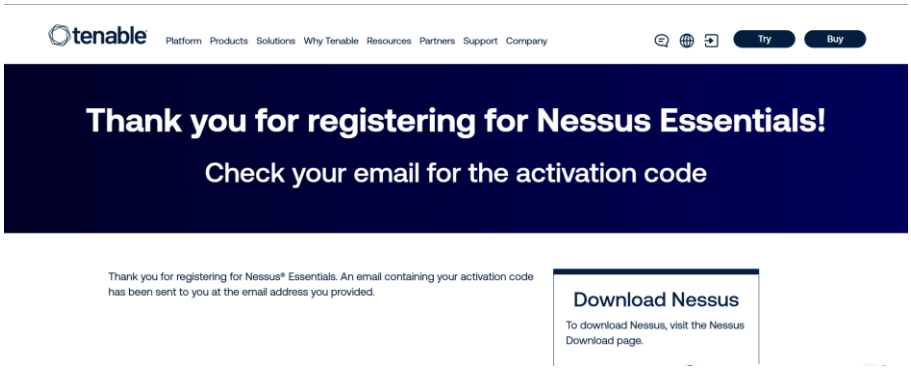
```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo apt install ./Nessus-10.9.0-debian10_amd64.deb
Note, selecting 'nessus' instead of './Nessus-10.9.0-debian10_amd64.deb'
nessus is already the newest version (10.9.0).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1316

┌──(kali㉿kali)-[~/Downloads]
└─$
```

4. Aktivitasi dan Konfigurasi Awal

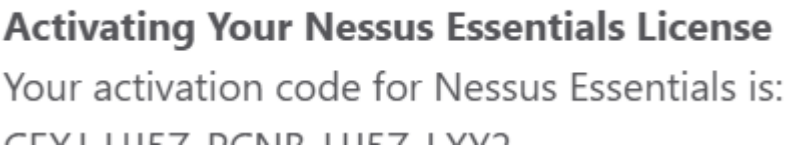
Dapatkan lisensi dari nessus dengan sign up menggunakan email



Cek email anda

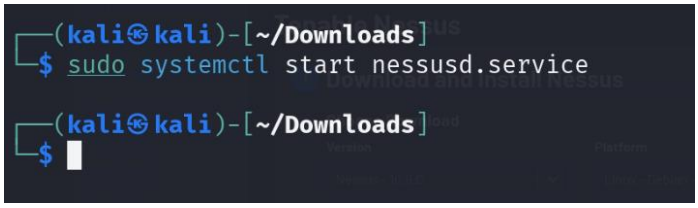


Lalu, salin kode aktivasi



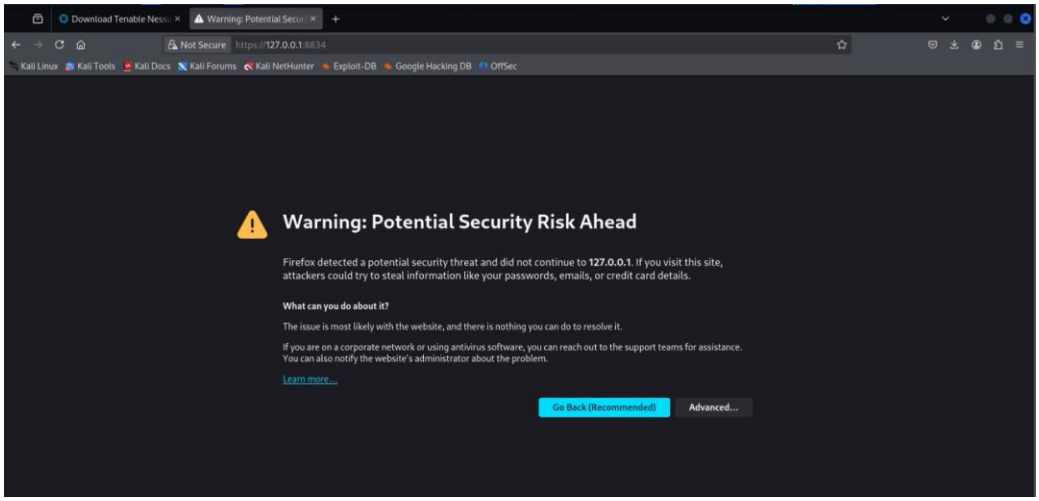
Mulai Layanan Nessus

sudo systemctl start nessusd.service

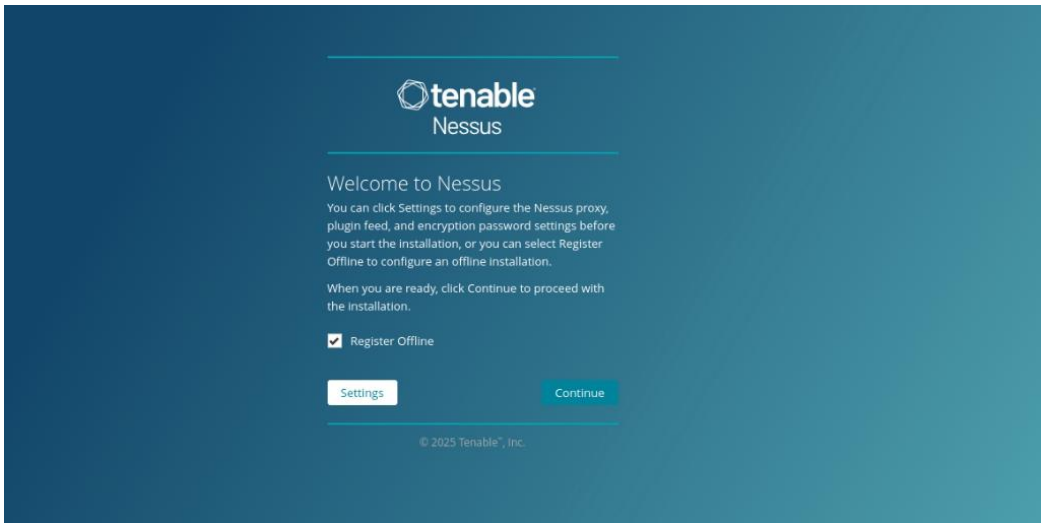


Akses Web Interface

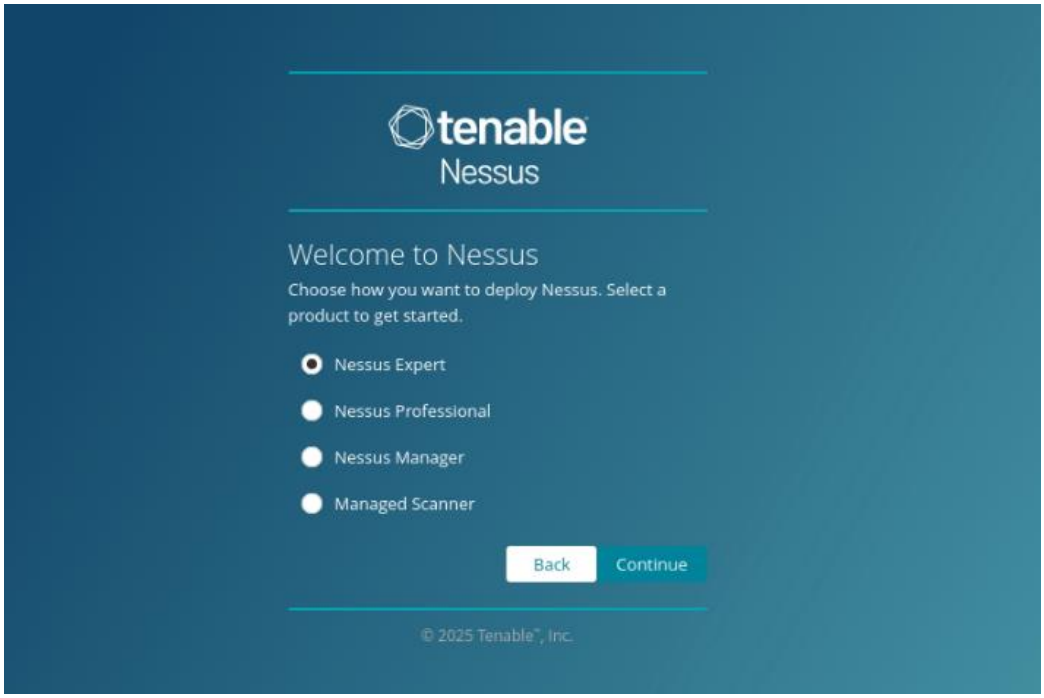
Lalu, akses https://127.0.0.1:8834 pada web browser dan klik Advanced -> Accept the risk lalu continue



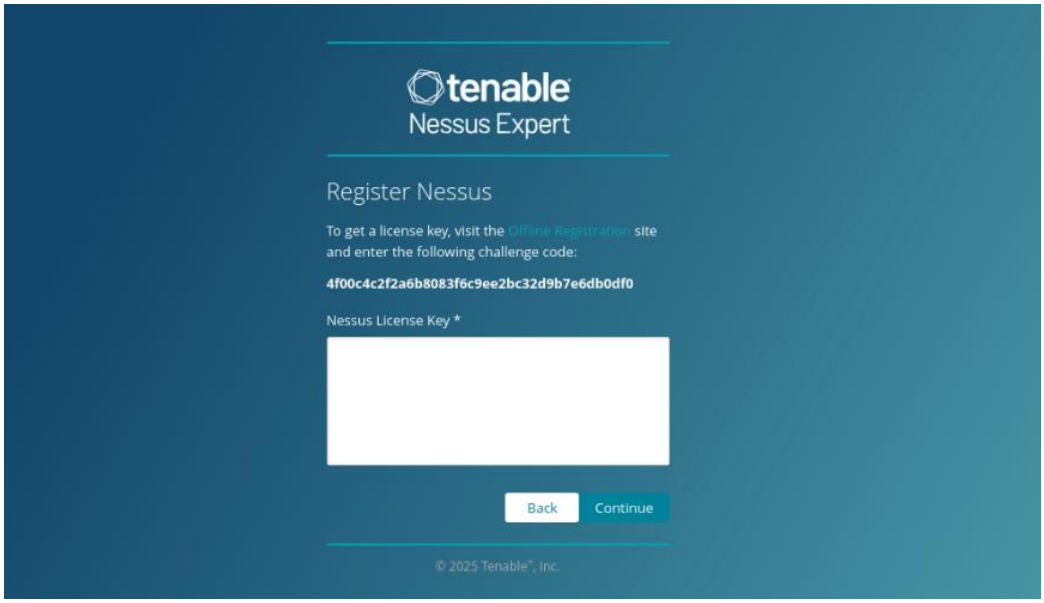
Pilih Offline Register -> Continue



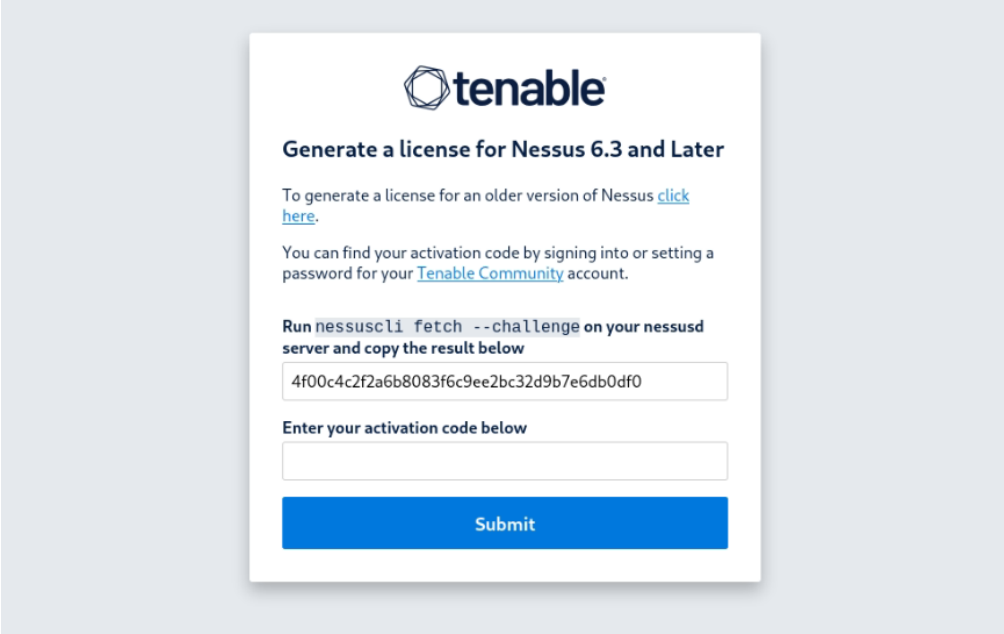
Biarkan pengaturan default, klik Continue



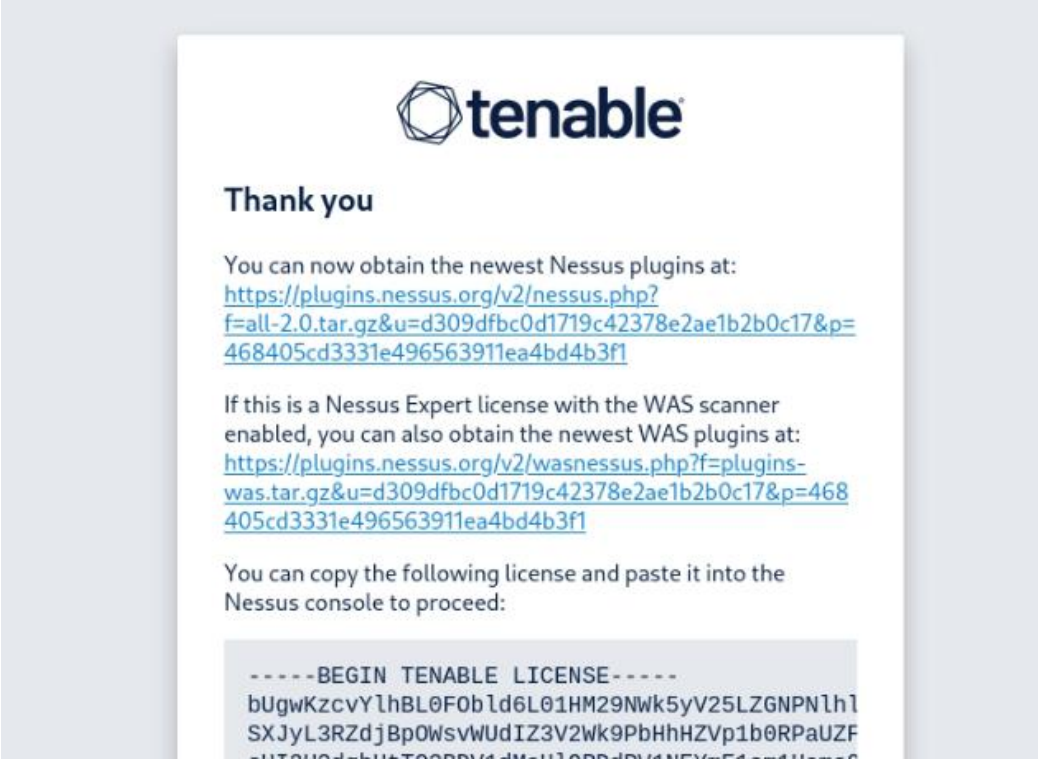
Buka link **offline registration site** dan salin challenge code dibawah berikut:



Masukkan challenge code + activation code dari email

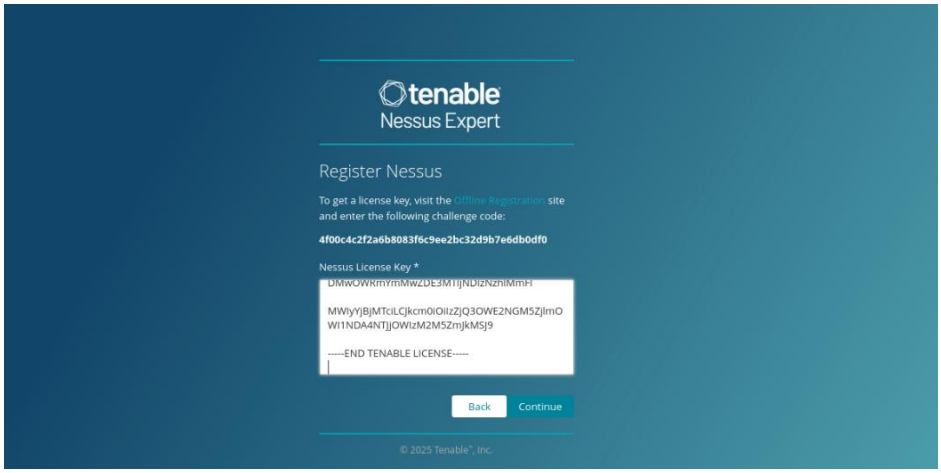


Submit dan salin license file yang dihasilkan

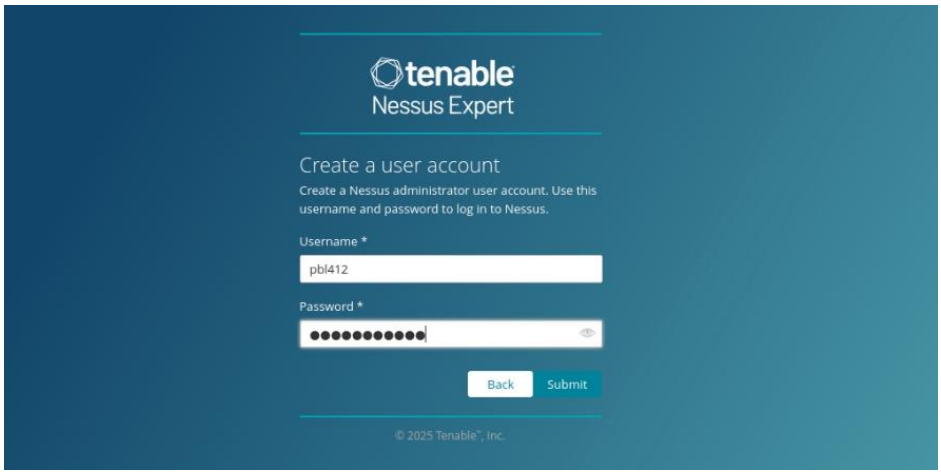


Kembali ke Web Nessus

Tempel license key, lalu tekan continue.



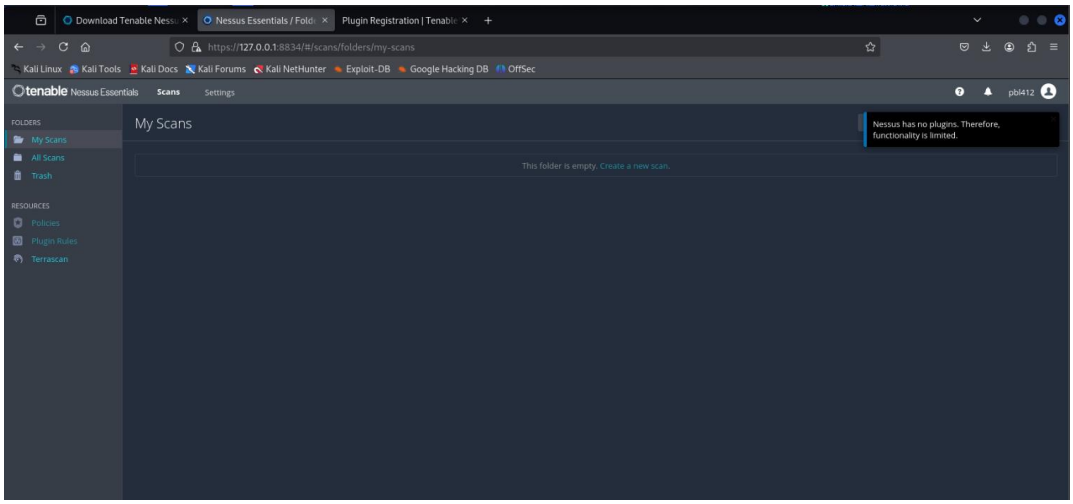
Lanjutkan dan buat akun Nessus



Submit dan tunggu hingga instalasi selesai

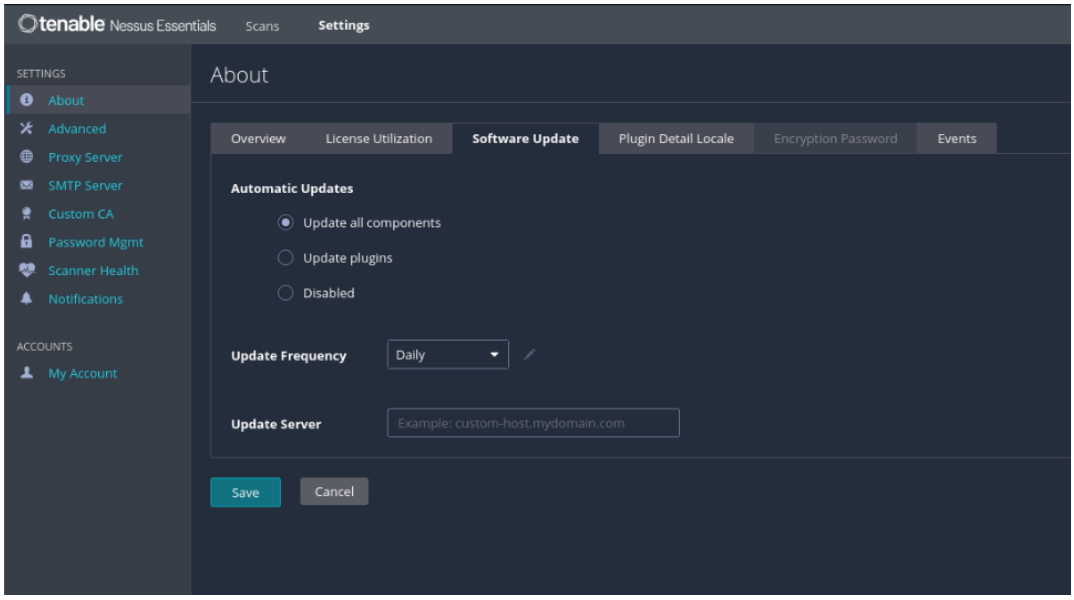
Update Plugin dan Software

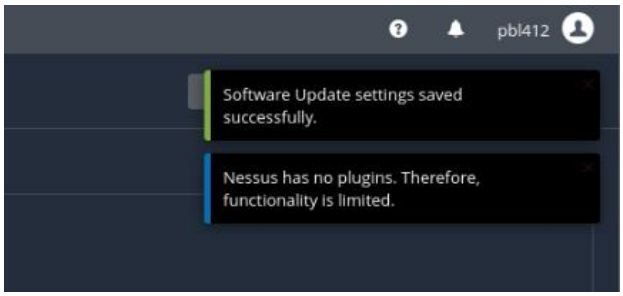
Kita sudah masuk ke halaman utama, lihat pada notifikasi bahwa nessus belum mempunyai plugins, sehingga fungsionalitasnya terbatas. Selanjutnya adalah penambahan plugins



Masuk ke Menu Settings

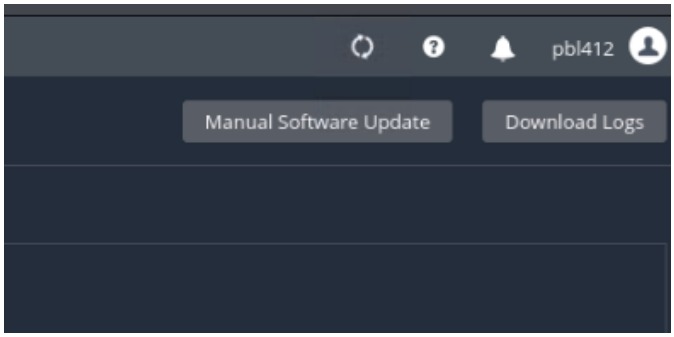
Pada navbar klik **Settings** -> **Software Updates.** Kemudian pilih opsi Update all components, lalu tekan Save.



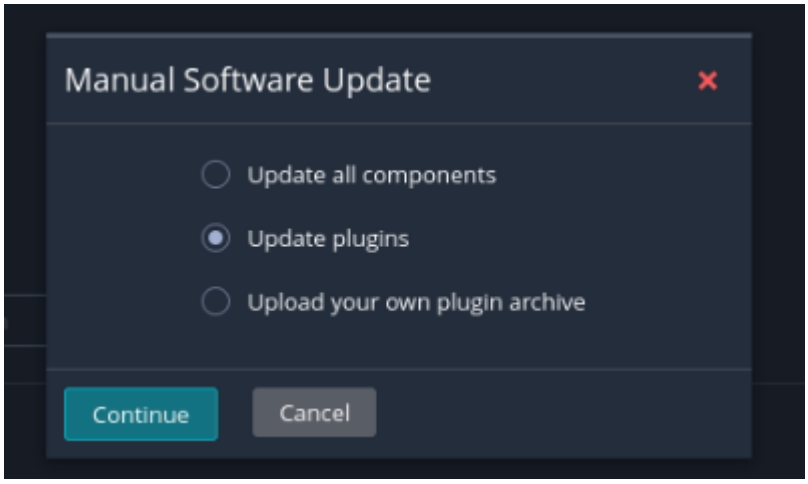


Update Plugin Secara Manual

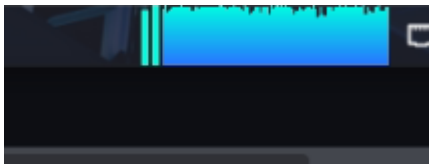
Selanjutnya klik Manual Software Updates

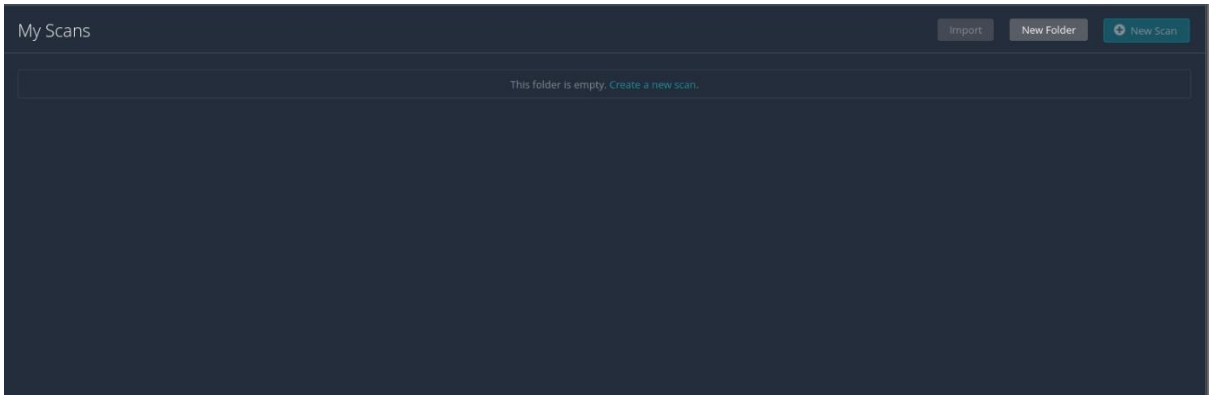


Ubah opsi menjadi update plugins



Tunggu beberapa saat, terlihat bahwa terjadi CPU Spike karena proses unduhan. Dan jika proses belum selesai, pada navbar Scans masih kosong.

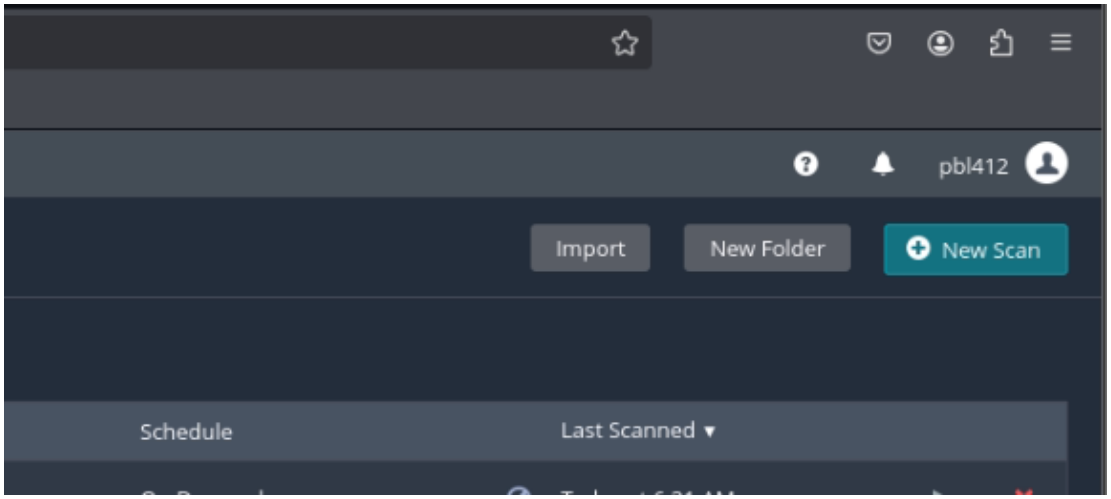




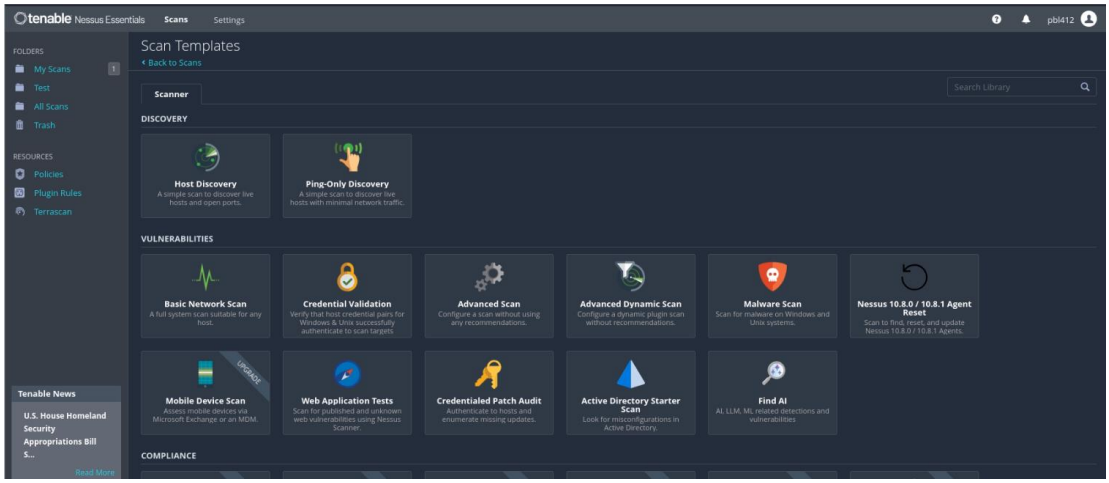
5.4 Melakukan Vulnerability Scanning

1. Membuat Scan
   Buka Nessus di browser: **https://127.0.0.1:8834,** Login menggunakan akun yang telah dibuat

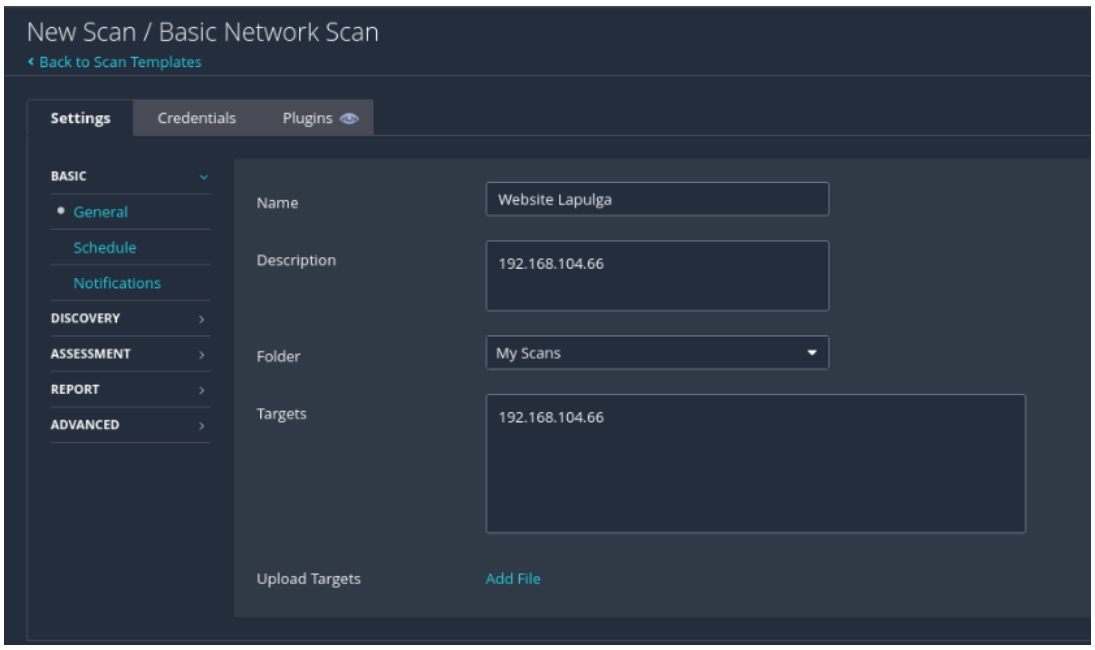
   Buat scan baru melalui menu **Scans → New Scan**



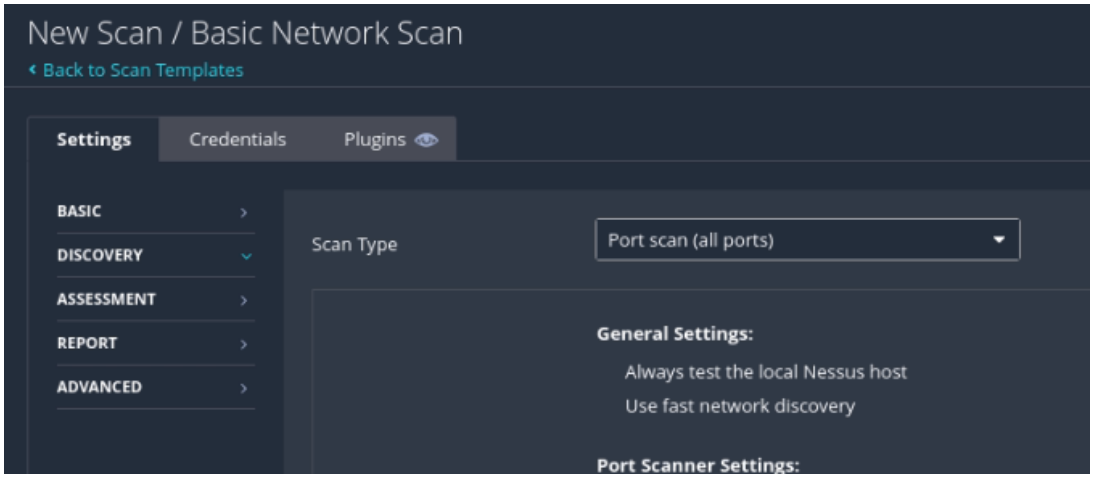
Pilih template **Basic Network Scan**



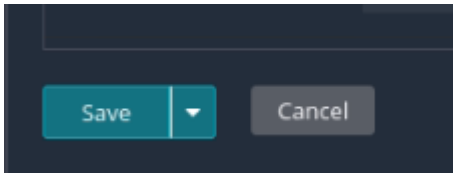
Masukkan target IP atau hostname

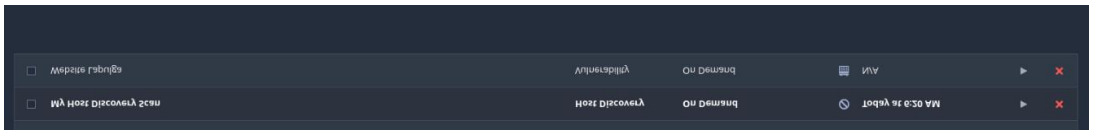


Buka Discovery, lalu pilih **Port scan (all ports)** pada Scan type.

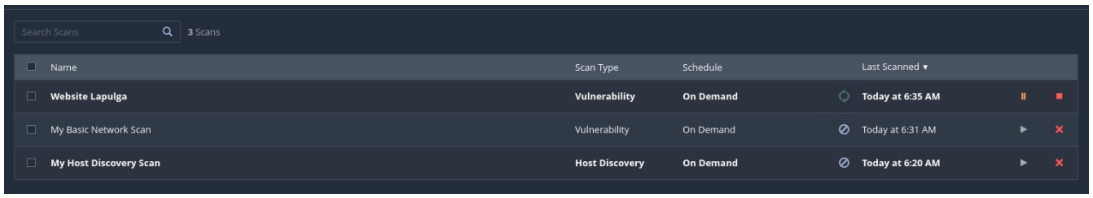


Klik **Save**, lalu klik Launch untuk memulai scanning

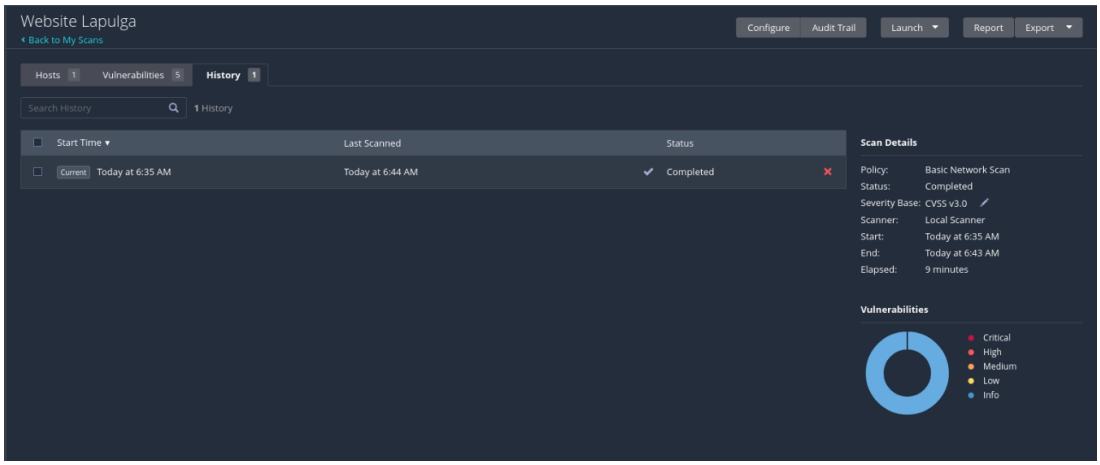




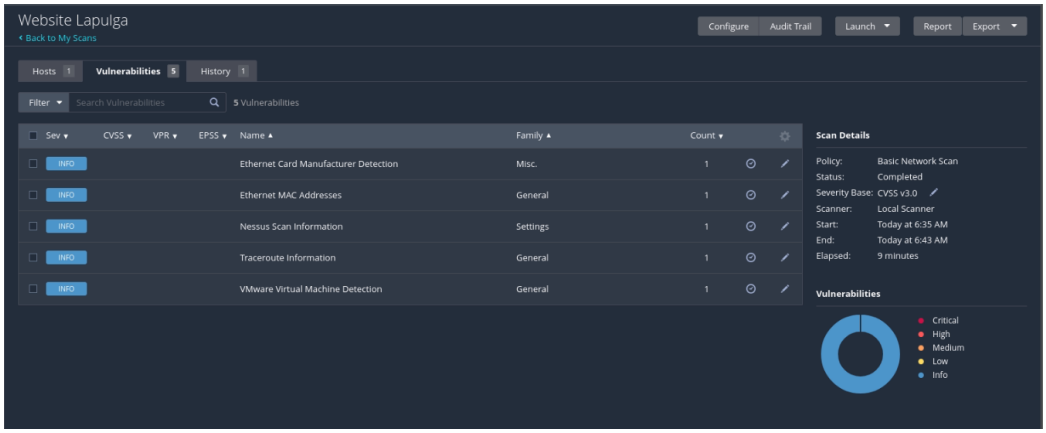
Tunggu hingga proses scanning selesai.



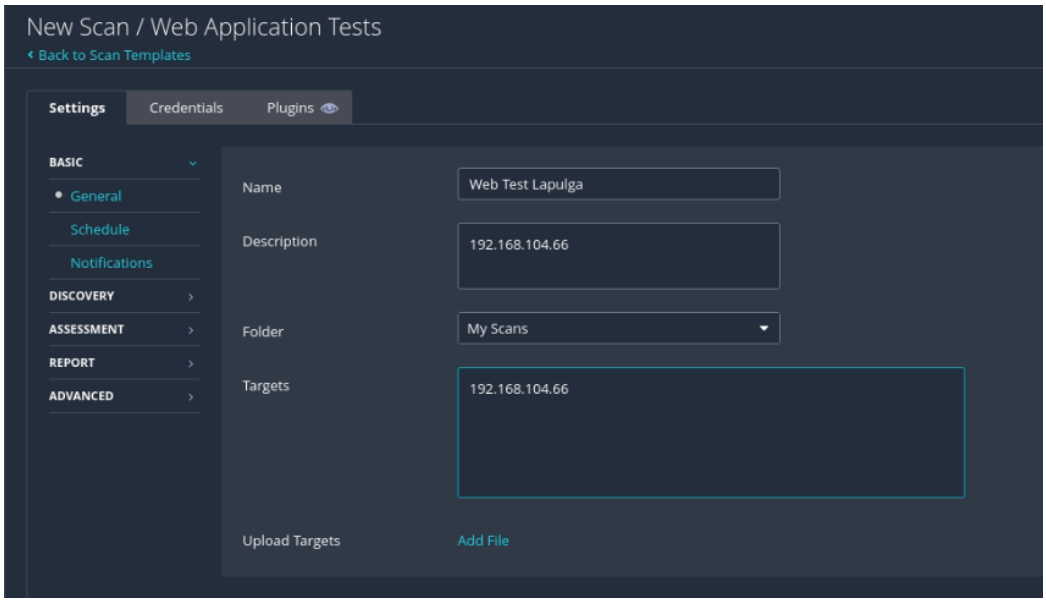
Jika Status Completed, klik untuk melihat hasil.



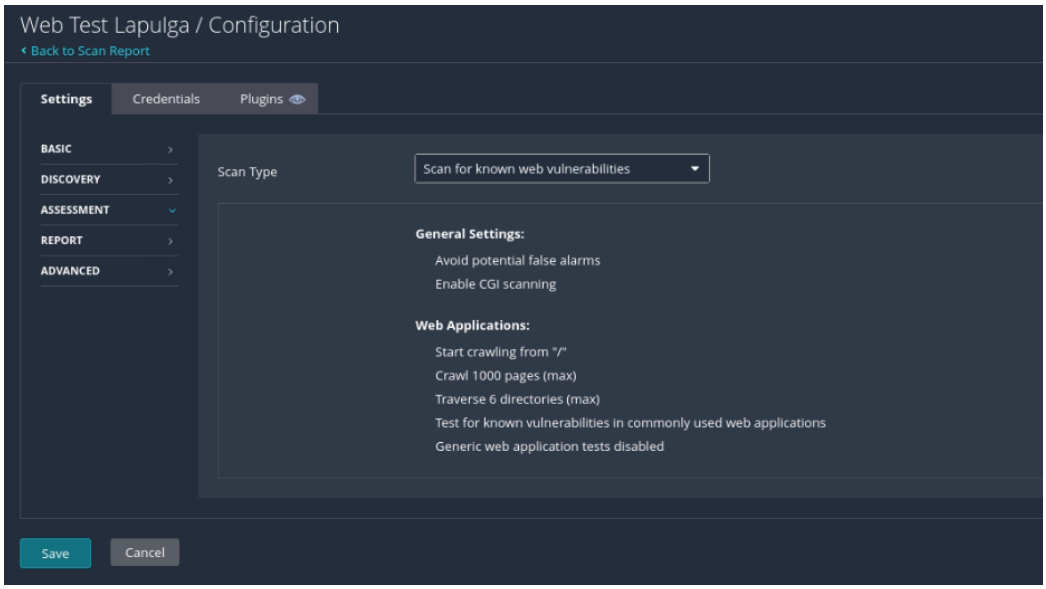
Hasil menunjukkan bahwa tidak ditemukan kerentanan pada network.



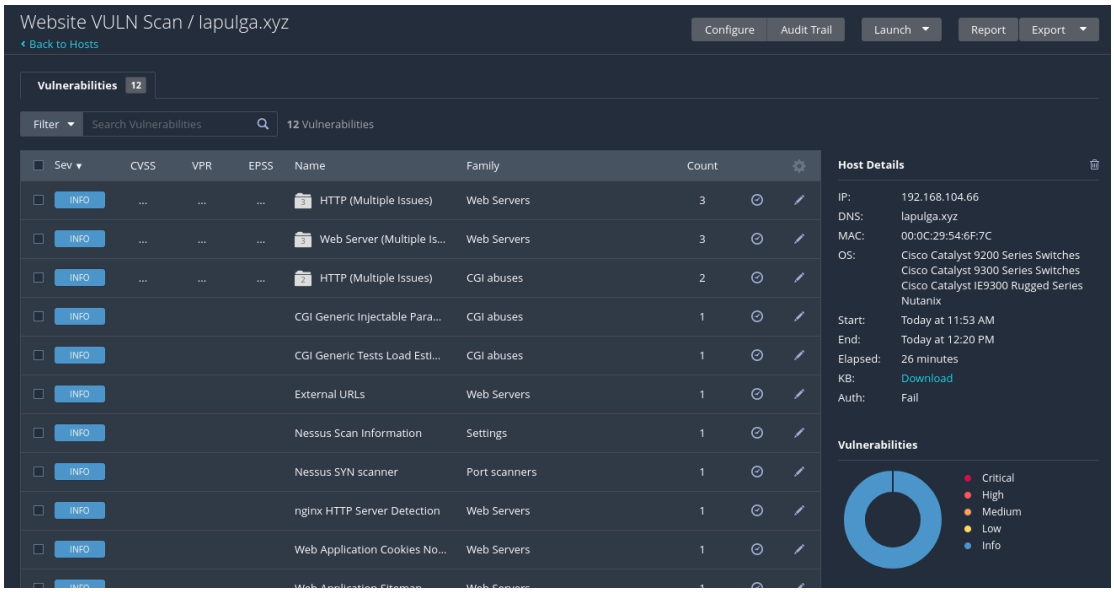
Selanjutnya Web Application vulnerability scanning untuk mencari kerentanan pada website. Masukkan IP Targets atau URL website.



Pilih Scafor known web vulnerabilities. (Anda bisa memilih yang lebih advance)



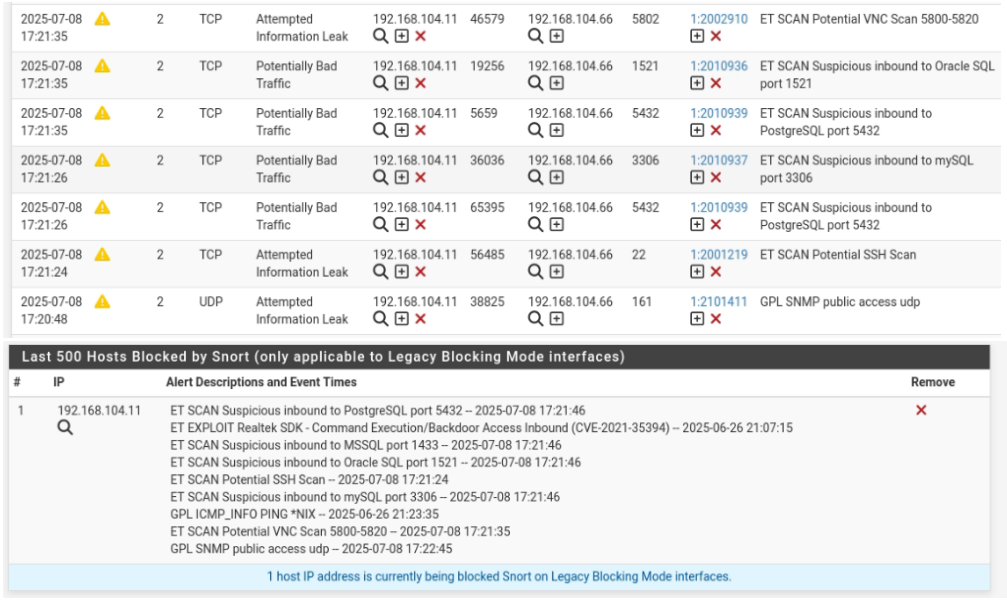
Hasil dari scanning adalah tidak ditemukannya vulnerabilities, namun hanya ditemukannya beberapa info yang tidak berbahaya, menunjukkan bahwa website ini sudah aman dari percobaan scanning.



2. Pengaruh Snort terhadap Proses Scanning

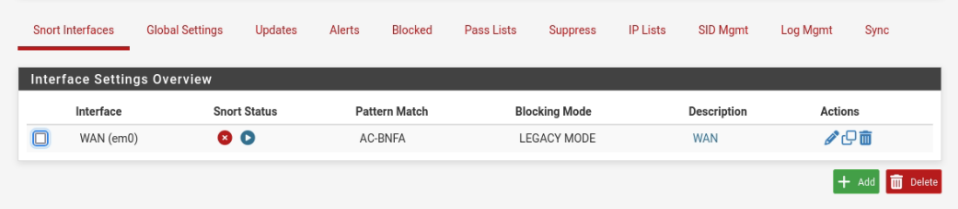
Untuk menguji efektivitas sistem deteksi intrusi (IDS), dilakukan dua kali scanning dengan kondisi Snort berbeda:

- Scan saat Snort Aktif
    - Snort dijalankan dalam mode blocking (inline)
    - Ketika Nessus melakukan scanning, sebagian besar request **diblokir** oleh Snort
    - Akibatnya, hasil scan menjadi tidak lengkap, dan Nessus gagal mendeteksi banyak layanan
    - Log Snort menunjukkan adanya alert dan blocking terhadap paket yang mencurigakan



- Scan saat Snort Dinonaktifkan

Snort dihentikan sementara



Nessus berhasil melakukan pemindaian penuh terhadap target
Tetapi tidak ditemukannya vulnerabilites.

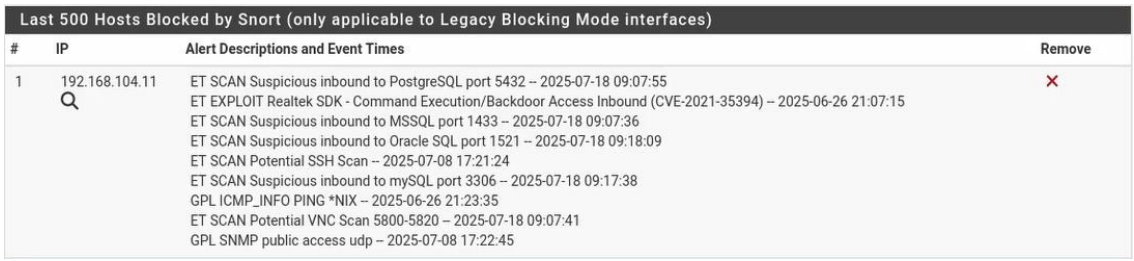
5.5 Penetration Testing Manual

1. Pengujian Menggunakan Nmap
   Perintah yang digunakan dalam proses scanning:
   nmap -sS -Pn -T2 --open -n -f **[IP TARGET]**

   Melakukan scanning port terbuka secara stealth, dengan kecepatan lambat, tanpa resolusi DNS, menghindari deteksi IDS/firewall, dan mengabaikan ping, agar bisa menyusup diam-diam ke sistem target.

   Hasil

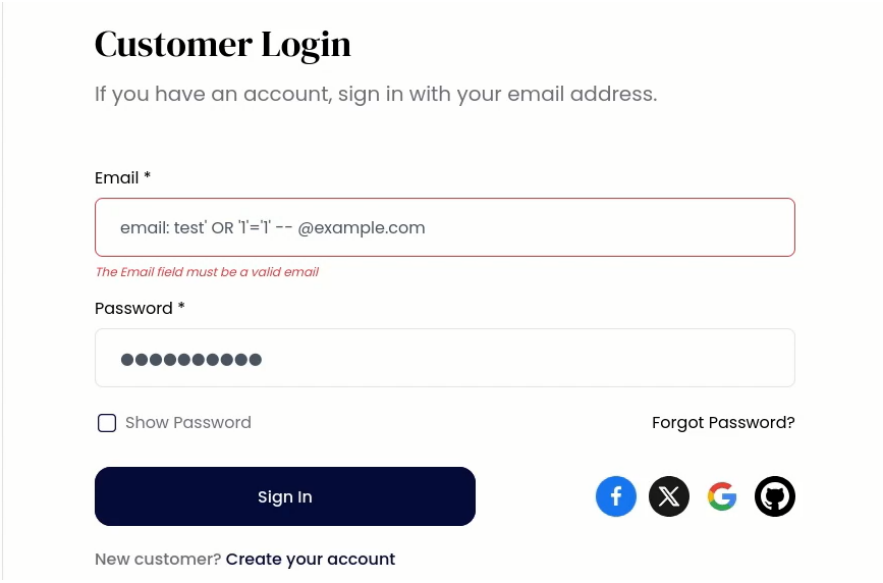
   Tidak dapat melakukan scanning, karena percobaan tetap terdeteksi oleh IDS/IPS Snort



2. Pengujian SQL Injection
   Pengujian dilakukan pada form login dan parameter URL dengan payload sederhana seperti: ' OR 1=1 --
   Karena login page menggunakan email pengguna maka kami mencoba membuat payload untuk mengakali inputan dengan email test' OR '1'='1' -- @example.com

   Hasil

   Tidak ditemukan tanda-tanda kerentanan SQL Injection



Tidak bisa ditembus langsung dengan payload, karena validasi format email mencegah payload dieksekusi.

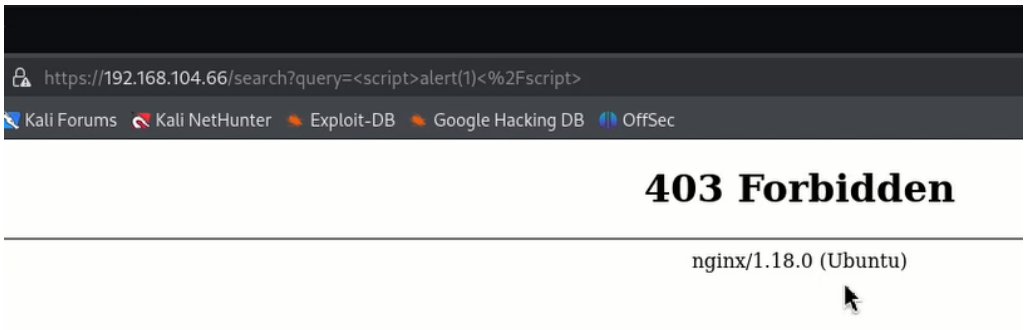
3. Pengujian Cross Site Scripting (XSS)
   Payload umum yang digunakan:

   <script>alert('XSS')</script>

   Diuji pada:

   - Form pencarian
   - Input URL

   Hasil



agent.name	rule.description	rule.level
proxyserver	XSS (Cross Site Scripting) attempt.	6
proxyserver	XSS (Cross Site Scripting) attempt.	6
proxyserver	XSS (Cross Site Scripting) attempt.	6
proxyserver	XSS (Cross Site Scripting) attempt.	6
proxyserver	XSS (Cross Site Scripting) attempt.	6
proxyserver	XSS (Cross Site Scripting) attempt.	6

- Input tidak mencetak kembali skrip ke halaman web
- Indikasi bahwa aplikasi telah menerapkan escape atau filter terhadap input
- Tidak ditemukan celah XSS yang berhasil dieksploitasi
- Percobaan XSS terdeteksi oleh Wazuh dan ditolak oleh WAF

5.6 Install Mod Security WAF on Nginx

1. Install Dependencies
   sudo apt update
   sudo apt install -y git build-essential libtool libpcre3 libpcre3-dev \
   libssl-dev zlib1g-dev libxml2 libxml2-dev libyajl-dev \
   pkgconf libcurl4-openssl-dev libgeoip-dev doxygen libpcre2-dev

```
root@ubuntu:/home/ubuntu# sudo apt install -y git build-essential libtool libpcre3 libpcre3-dev \
>     libssl-dev zlib1g-dev libxml2 libxml2-dev libyajl-dev \
>     pkgconf libcurl4-openssl-dev libgeoip-dev doxygen
```

2. Install ModSecurity (v3)
   cd /usr/local/src
   sudo git clone --depth 1 -b v3/master https://github.com/SpiderLabs/ModSecurity

```
root@ubuntu:/home/ubuntu# cd /usr/local/src
root@ubuntu:/usr/local/src# sudo git clone --depth 1 -b v3/master https://github.com/SpiderLabs/ModSecurity
Cloning into 'ModSecurity'...
remote: Enumerating objects: 870, done.
remote: Counting objects: 100% (870/870), done.
remote: Compressing objects: 100% (727/727), done.
remote: Total 870 (delta 502), reused 238 (delta 131), pack-reused 0 (from 0)
Receiving objects: 100% (870/870), 812.50 KiB | 2.11 MiB/s, done.
Resolving deltas: 100% (502/502), done.
```

   cd ModSecurity
   sudo git submodule init

```
root@ubuntu:/usr/local/src/ModSecurity# sudo git submodule init
Submodule 'bindings/python' (https://github.com/owasp-modsecurity/ModSecurity-Python-bindings.git) registered for path '
bindings/python'
Submodule 'others/libinjection' (https://github.com/libinjection/libinjection.git) registered for path 'others/libinject
ion'
Submodule 'others/mbedtls' (https://github.com/Mbed-TLS/mbedtls.git) registered for path 'others/mbedtls'
Submodule 'test/test-cases/secrules-language-tests' (https://github.com/owasp-modsecurity/secrules-language-tests) regis
tered for path 'test/test-cases/secrules-language-tests'
```

   sudo git submodule update

```
root@ubuntu:/usr/local/src/ModSecurity# sudo git submodule update
Cloning into '/usr/local/src/ModSecurity/bindings/python'...
Cloning into '/usr/local/src/ModSecurity/others/libinjection'...
Cloning into '/usr/local/src/ModSecurity/others/mbedtls'...
```

   sudo ./build.sh

```
root@ubuntu:/usr/local/src/ModSecurity# sudo ./build.sh
libtoolize: putting auxiliary files in '.'.
libtoolize: copying file './ltmain.sh'
libtoolize: putting macros in AC_CONFIG_MACRO_DIRS, 'build'.
libtoolize: copying file 'build/libtool.m4'
libtoolize: copying file 'build/ltoptions.m4'
```

sudo ./configure

```
root@ubuntu:/usr/local/src/ModSecurity# sudo ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a race-free mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
```

Sudo make

```
root@ubuntu:/usr/local/src/ModSecurity# sudo make
Making all in others
make[1]: Entering directory '/usr/local/src/ModSecurity/others'
/bin/bash ../libtool  --tag=CC   --mode=compile gcc -DHAVE_CONFIG_H -I. -I../src    -D LIBINJECTION_VERSION=\"\" -g -O
-MT libinjection/src/la-libinjection_html5.lo -MD -MP -MF libinjection/src/.deps/la-libinjection_html5.Tpo -c -o libin
ction/src/la-libinjection_html5.lo `test -f 'libinjection/src/libinjection_html5.c' || echo './'`libinjection/src/libi
ection_html5.c
libtool: compile:  gcc -DHAVE_CONFIG_H -I. -I../src -D LIBINJECTION_VERSION=\"\" -g -O2 -MT libinjection/src/la-libinj
tion_html5.lo -MD -MP -MF libinjection/src/.deps/la-libinjection_html5.Tpo -c libinjection/src/libinjection_html5.c  -
IC -DPIC -o libinjection/src/.libs/la-libinjection_html5.o
libtool: compile:  gcc -DHAVE_CONFIG_H -I. -I../src -D LIBINJECTION_VERSION=\"\" -g -O2 -MT libinjection/src/la-libinj
```

Sudo make install

```
root@ubuntu:/usr/local/src/ModSecurity# sudo make install
Making install in others
```

3. Install ModSecurity-Nginx Connector

   cd /usr/local/src

   sudo git clone --depth 1 https://github.com/SpiderLabs/ModSecurity-nginx.git

```
root@ubuntu:/usr/local/src# sudo git clone --depth 1 https://github.com/SpiderLabs/ModSecurity-nginx.git
Cloning into 'ModSecurity-nginx'...
remote: Enumerating objects: 59, done.
remote: Counting objects: 100% (59/59), done.
remote: Compressing objects: 100% (59/59), done.
remote: Total 59 (delta 12), reused 26 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (59/59), 1.12 MiB | 2.69 MiB/s, done.
Resolving deltas: 100% (12/12), done.
```

4. Rebuild or Install Nginx with ModSecurity Module

   nginx -v

```
root@ubuntu:/usr/local/src# nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
```

   wget http://nginx.org/download/nginx-1.18.0.tar.gz

```
root@ubuntu:/usr/local/src# wget http://nginx.org/download/nginx-1.18.0.tar.gz
--2025-07-19 07:13:56--  http://nginx.org/download/nginx-1.18.0.tar.gz
Resolving nginx.org (nginx.org)... 52.58.199.22, 3.125.197.172, 2a05:d014:5c0:2601:
Connecting to nginx.org (nginx.org)|52.58.199.22|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1039530 (1015K) [application/octet-stream]
Saving to: ‘nginx-1.18.0.tar.gz’

nginx-1.18.0.tar.gz          100%[=============================================

2025-07-19 07:14:00 (264 KB/s) - ‘nginx-1.18.0.tar.gz’ saved [1039530/1039530]
```

tar -xvzf nginx-1.18.0.tar.gz

```
root@ubuntu:/usr/local/src# tar -xvzf nginx-1.18.0.tar.gz
nginx-1.18.0/
nginx-1.18.0/auto/
nginx-1.18.0/conf/
nginx-1.18.0/contrib/
```

cd nginx-1.18.0

```
root@ubuntu:/usr/local/src/nginx-1.18.0# ls
auto  CHANGES  CHANGES.ru  conf  configure  contrib  html  LICENSE  man  README  src
```

sudo ./configure --with-compat --add-dynamic-module=../ModSecurity-nginx

```
root@ubuntu:/usr/local/src/nginx-1.18.0# sudo ./configure --with-compat --add-dynamic-module=../ModSecurity-nginx
checking for OS
```

Sudo make modules

```
root@ubuntu:/usr/local/src/nginx-1.18.0# sudo make modules
make -f objs/Makefile modules
make[1]: Entering directory '/usr/local/src/nginx-1.18.0'
cc -c -fPIC -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -Werror -g  -I src/core -I src/event -I src/ev
odules -I src/os/unix -I /usr/local/modsecurity/include -I objs -I src/http -I src/http/modules \
```

sudo mkdir -p /etc/nginx/modules

sudo cp /usr/local/src/nginx-1.18.0/objs/ngx_http_modsecurity_module.so /etc/nginx/modules/

```
root@ubuntu:/usr/local/src/nginx-1.18.0# sudo mkdir -p /etc/nginx/modules
root@ubuntu:/usr/local/src/nginx-1.18.0# sudo cp /usr/local/src/nginx-1.18.0/objs/ngx_http_modsecurity_module.so /etc/ng
inx/modules/
```

sudo nano /etc/nginx/nginx.conf

load_module /etc/nginx/modules/ngx_http_modsecurity_module.so;

```
  GNU nano 6.2                                    /etc/nginx/nginx.conf
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;
load_module /etc/nginx/modules/ngx_http_modsecurity_module.so;
```

sudo nginx -t

sudo systemctl restart nginx

```
root@ubuntu:/usr/local/src/nginx-1.18.0# sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ubuntu:/usr/local/src/nginx-1.18.0# sudo systemctl restart nginx
```

5. Salin dan konfigurasi modsecurity.conf

   sudo mkdir -p /etc/nginx/modsec

   sudo cp /usr/local/src/ModSecurity/modsecurity.conf-recommended /etc/nginx/modsec/modsecurity.conf

```
root@ubuntu:/usr/local/src/ModSecurity# sudo mkdir -p /etc/nginx/modsec
root@ubuntu:/usr/local/src/ModSecurity# sudo cp /usr/local/src/ModSecurity/modsecurity.conf-recommended /etc/nginx/modse
c/modsecurity.conf
```

sudo nano /etc/nginx/modsec/modsecurity.conf
ubah SecRuleEngine DetectionOnly menjadi SecRuleEngine On

```
  GNU nano 6.2                                /etc/nginx/modsec/modsecurity.conf
# -- Rule engine initialization ----------------------------------------------

# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
SecRuleEngine On


# -- Request body handling ---------------------------------------------------

# Allow ModSecurity to access request bodies. If you don't, ModSecurity
# won't be able to see any POST parameters, which opens a large security
# hole for attackers to exploit.
#
SecRequestBodyAccess On
```

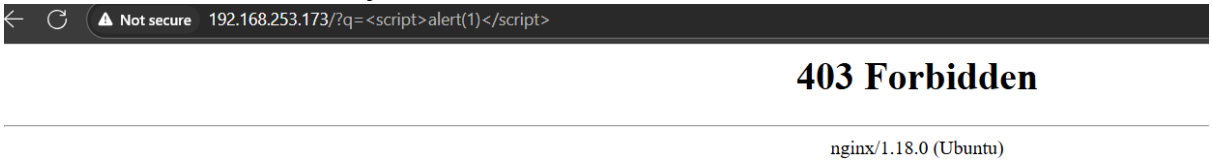
6. Tambah unicode.mapping
   sudo cp /usr/local/src/ModSecurity/unicode.mapping /etc/nginx/modsec/
7. Tambah konfigurasi di nginx.conf
   sudo nano /etc/nginx/nginx.conf
   Tambahkan:
   modsecurity on;
   modsecurity_rules_file /etc/nginx/modsec/modsecurity.conf;

```
  GNU nano 6.2                                  /etc/nginx/nginx.conf *
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;
load_module /etc/nginx/modules/ngx_http_modsecurity_module.so;

events {
        worker_connections 768;
        # multi_accept on;
}

http {

        ##
        # Basic Settings
        ##

        modsecurity on;
        modsecurity_rules_file /etc/nginx/modsec/modsecurity.conf;
```

8. Tes konfigurasi dan restart
   sudo nginx -t
   sudo systemctl restart nginx

```
root@ubuntu:/etc/nginx/modsec# sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ubuntu:/etc/nginx/modsec# sudo systemctl restart nginx
```

9. Hasil dari block mod security



Ketika ada serangan otomatis akan menjadi forbidden

10. Jika masih blm forbidden

Membuat file conf sendiri:
sudo nano /etc/nginx/modsec/main.conf
setelah itu menambahkan konfigurasi berikut
SecRuleEngine On
SecRequestBodyAccess On
SecResponseBodyAccess Off
SecRule REQUEST_URI "@contains script" "id:1234,phase:2,deny,status:403,msg:'XSS Attack Detected'"

```
  GNU nano 6.2                                  /etc/nginx/modsec/main.conf
SecRuleEngine On
SecRequestBodyAccess On
SecResponseBodyAccess Off
SecRule REQUEST_URI "@contains script" "id:1234,phase:2,deny,status:403,msg:'XSS Attack Detected'"
```

Setelah itu baru konfigurasi di file web nginx nya seperti berikut:
cd sites-enabled/
nano default (default Ganti ke nama website kalian)
tambahkan konfigurasi berikut di website kalian
modsecurity on;
modsecurity_rules_file /etc/nginx/modsec/main.conf;

```
  GNU nano 6.2                                          default
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        modsecurity on;
        modsecurity_rules_file /etc/nginx/modsec/main.conf;
```