



3-6-2025

Proyecto Implementación de Firewall

Ciberseguridad IV

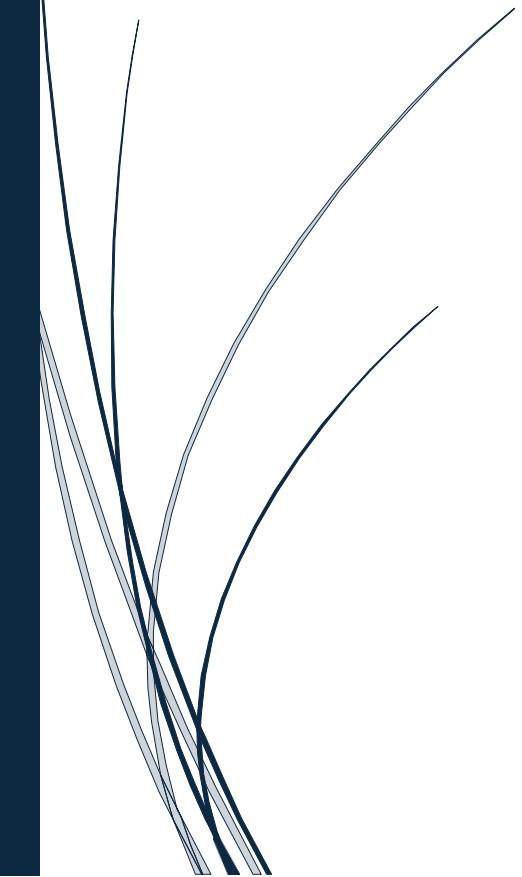
Profesor: Erick Pérez

AGUSTÍN SÁNCHEZ

MURIEL JARAMILLO

GILBERTO RAMOS

ABNER BALLESTEROS



I SEMESTRE 2025

Manual de configuración de VLANs — OPNsense + Proxmox

Introducción

En este laboratorio se implementó una infraestructura de red segmentada sobre una plataforma de virtualización utilizando Proxmox como hipervisor principal y OPNsense como firewall centralizado.

El objetivo fue construir un entorno seguro, aislado y administrable, replicando un esquema similar al de una red empresarial, aplicando segmentación por VLANs, configuración de firewall granular, control de acceso remoto seguro, y publicación de servicios web internos mediante NGINX Proxy Manager.

En este escenario, la red LAN cumple la función de **zona DMZ**, donde residen los servicios públicos o de acceso cruzado (como el NGINX Proxy Manager), mientras las VLANs 10 y 20 representan redes de usuarios o servicios internos debidamente segmentados.

El trabajo que presentamos es una simulación controlada de segmentación de red LAN implementando:

- Virtualización completa de infraestructura.
- Segmentación lógica con VLANs.
- Separación de zonas de seguridad: LAN, DMZ y segmentos de usuario.
- Firewall granular de capa 3 y 4.
- Seguridad perimetral avanzada.

Objetivos

- Virtualizar completamente el entorno de red sobre Proxmox.
- Implementar un firewall OPNsense como núcleo de control de tráfico.
- Segmentar las redes internas mediante VLANs:
 - VLAN10 → red interna para clientes Linux Mint.
 - VLAN20 → red interna para servidores Debian.
 - LAN (DMZ) → Servicios expuestos controlados
- Utilizar la LAN como DMZ dedicada para los servicios web públicos.
- Configurar un NGINX Proxy Manager en la DMZ (LAN) para gestionar futuros servicios web.
- Implementar políticas de firewall estrictas:
 - Permitir tráfico HTTP/HTTPS de clientes internos hacia el NGINX.
 - Limitar y bloquear accesos innecesarios (por ejemplo: ICMP externo, SSH desde redes no autorizadas).

- Habilitar acceso remoto seguro mediante Tailscale.
-

Paso 1 — Crear las Interfaces de Red en Proxmox

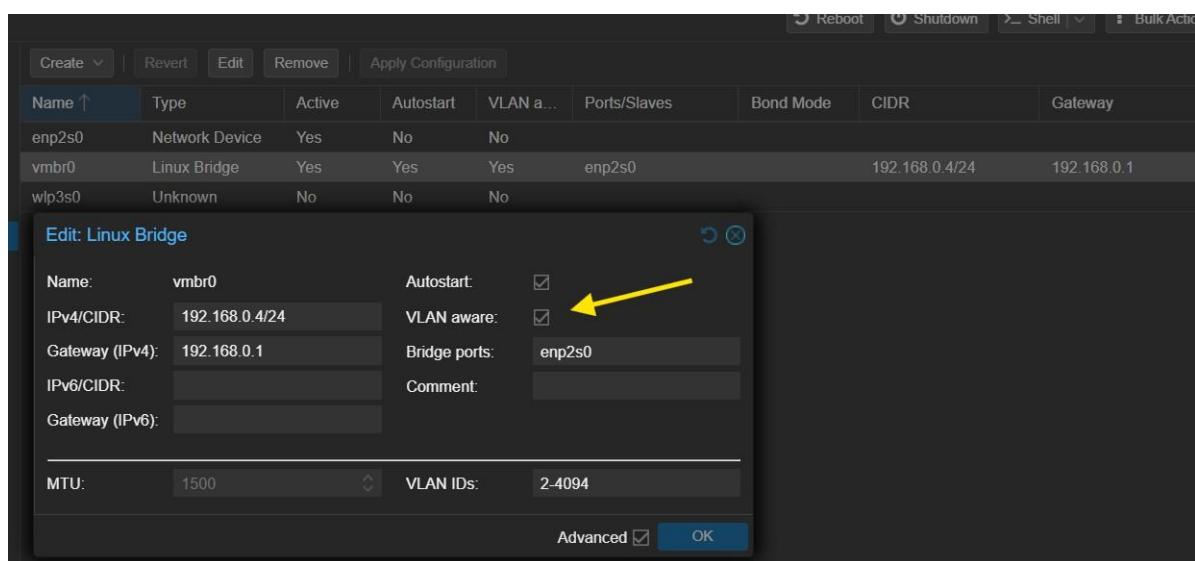
Activar VLAN Aware en el Bridge de Proxmox

Menú en Proxmox:

Datacenter → Node → Network → Editar vmbr0

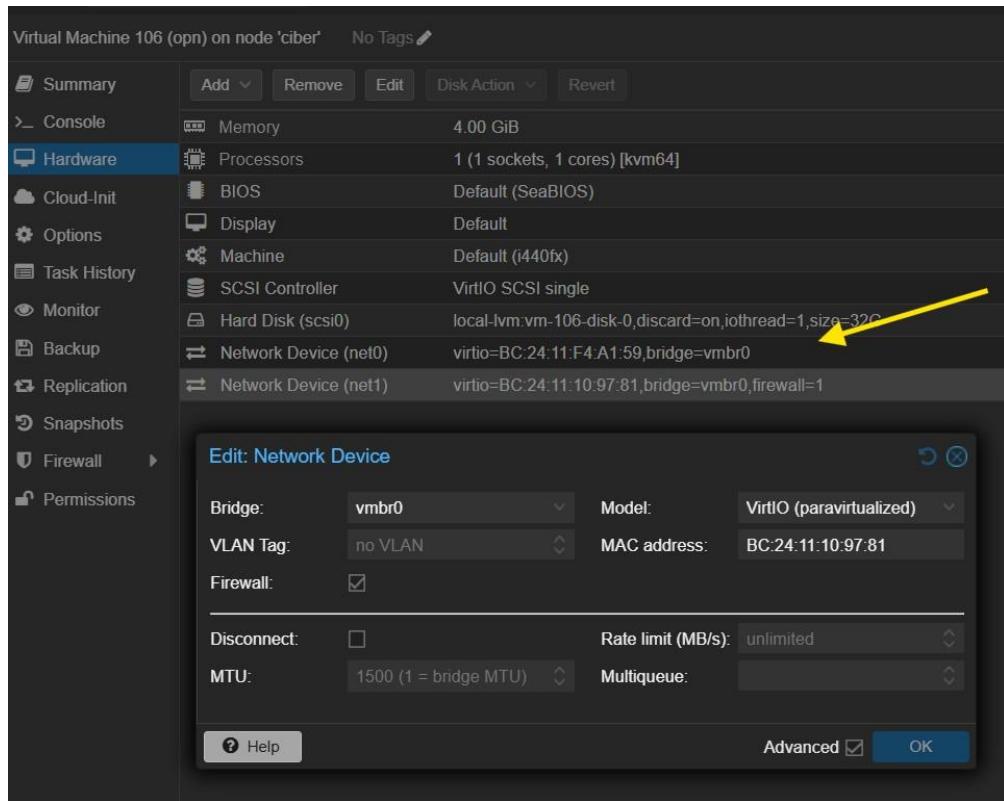
Marcar la opción VLAN aware

Esto permite que el bridge vmbr0 entienda los tags de VLAN que asignamos en las máquinas virtuales. Sin esto, las VLANs no pasan correctamente hacia OPNsense.



En el nodo de Proxmox, dentro de la VM de OPNsense:

- Asignar dos adaptadores de red:
 - **Net0** (WAN) → conectado al vmbr0 (red física o NAT según el escenario).
 - **Net1** (LAN y trunk para VLANs) → también conectado al vmbr0.



Paso 2 — Instalación base de OPNsense

- Iniciar la VM de OPNsense.
- Durante el primer arranque, asignar interfaces:
 - **WAN:** vtnet0 (normalmente Proxmox lo asigna bien).
 - **LAN:** vtnet1

Aquí no tomé captura porque tuve que volver a asignar las interfaz, pero debe aparecer la VTNET1

```
Valid interfaces are:
vtnet0      bc:24:11:f4:a1:59 VirtIO Networking Adapter
vtnet0_1    00:00:00:00:00:00 VLAN tag 10, parent interface vtne
vtnet0_2    00:00:00:00:00:00 VLAN tag 20, parent interface vtne

If you do not know the names of your interfaces, you may choose to use
auto-detection. In that case, disconnect all interfaces now before
hitting 'a' to initiate auto detection.

Enter the WAN interface name or 'a' for auto-detection: █
```

Paso 3 — Confirmación de IP iniciales

- WAN recibe IP por DHCP (ejemplo: 192.168.0.29)

- LAN queda en: 192.168.1.1/24

Aquí aparecen las VLANS pero es porque despues que ingrese a la webui cree las VLANS

```
*** OPNsense.localdomain: OPNsense 25.1 (amd64) ***

LAN (vtnet1)      -> v4: 192.168.1.1/24
OPT1 (vlan01)     -> v4: 192.168.10.1/24
OPT2 (vlan02)     -> v4: 192.168.20.1/24
WAN (vtnet0)      -> v4/DHCP4: 192.168.0.29/24
                           v6/DHCP6: ::be24:11ff:fef4:a159/64

HTTPS: sha256 DB 23 73 0A A2 74 35 20 9F C4 46 C2 68 40 A1 EA
       63 85 3D 3B AA F2 13 07 B0 0C 1D FD 7D 8B 21 62

0) Logout                      7) Ping host
1) Assign interfaces            8) Shell
2) Set interface IP address    9) pfTop
3) Reset the root password    10) Firewall log
4) Reset to factory defaults   11) Reload all services
5) Power off system            12) Update from console
6) Reboot system                13) Restore a backup

Enter an option: █
```

Paso 4 — Ingreso a la WEBGUI de OPNsense

- Desde tu PC o una VM dentro del rango 192.168.1.x, ingresar a:

<https://192.168.1.1>

- Usuario: root
- Password: opnsense

Antes de entrar a la WEBUI debemos ponerle la VLAN TAG 1 a la máquina que vamos a usar

ment 8.4.1 Search

Virtual Machine 102 (Mint) on node 'ciber' No Tags

Summary Add Remove Edit Disk Action Revert

Console

Hardware

Memory 4.00 GiB

Processors 1 (1 sockets, 1 cores) [host]

BIOS Default (SeaBIOS)

Display Default

Machine Default (i440fx)

SCSI Controller VirtIO SCSI single

Hard Disk (scsi0) local-lvm:vm-102-disk-0,iothread=1,size=32G

Network Device (net0) virtio=BC:24:11:47:8D:C5,bridge=vmbr0,firewall=1,tag=1

Edit: Network Device

Bridge: vmbr0 Model: VirtIO (paravirtualized)

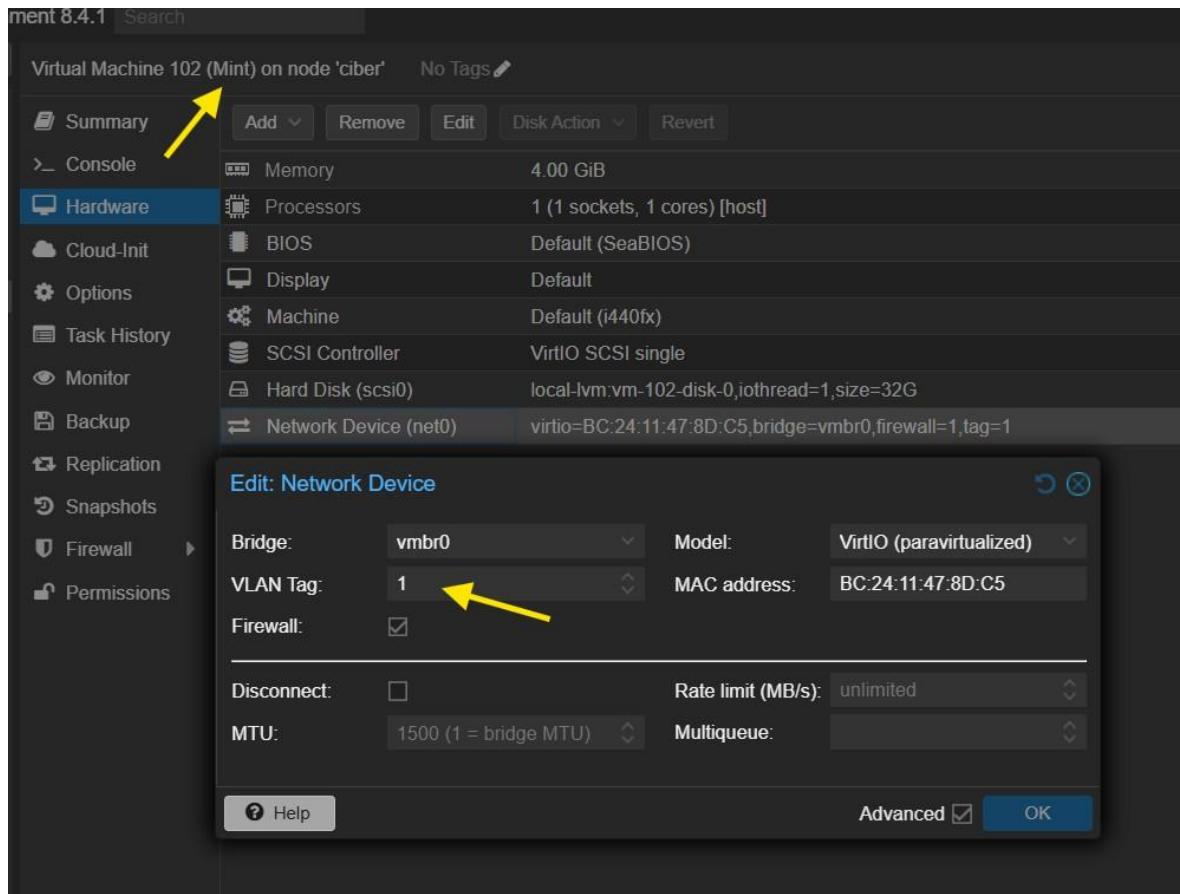
VLAN Tag: 1 MAC address: BC:24:11:47:8D:C5

Firewall:

Disconnect: Rate limit (MB/s): unlimited

MTU: 1500 (1 = bridge MTU) Multiqueue:

Help Advanced OK



Login | OPNsense — Mozilla Firefox

Login | OPNsense 192.168.1.1

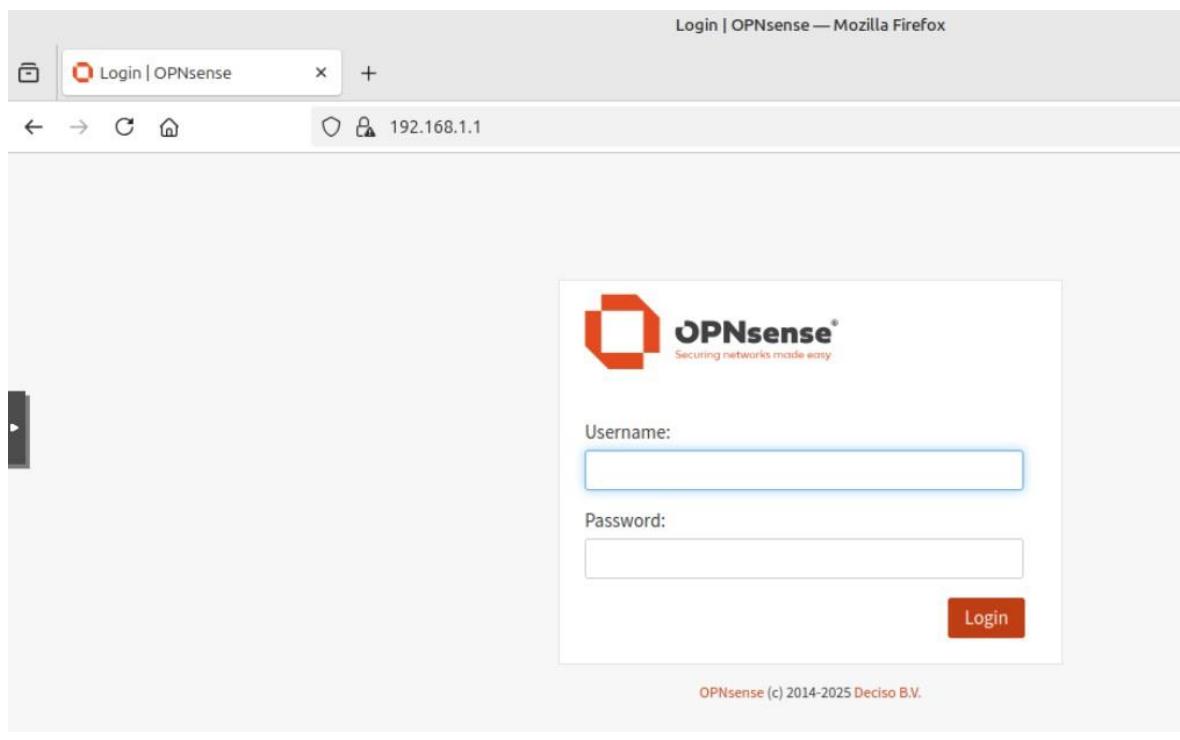
OPNsense®
Securing networks made easy

Username:

Password:

Login

OPNsense (c) 2014-2025 Deciso B.V.



Paso 5 — Creación de VLANs en OPNsense

Menú: Interfaces → Devices → VLAN

- **Parent Interface:** vtnet1 (LAN)
- **VLAN10:** Tag 10 → Descripción: VLAN10
- **VLAN20:** Tag 20 → Descripción: VLAN20

Device	Parent	Tag	PCP	Description
vlan01 [OPT1]	vtnet1 (bc:24:11:10:97:81) [LAN]	10	Best Effort (0, default)	VLAN10
vlan02 [OPT2]	vtnet1 (bc:24:11:10:97:81) [LAN]	20	Best Effort (0, default)	VLAN20

Edit Vlan

advanced mode	full help
Device	vlan01
Parent	vtnet1 (bc:24:11:10:97:81) [LAN]
VLAN tag	10
VLAN priority	Best Effort (0, default)
Description	VLAN10

Cancel Save

Paso 6 — Asignación de Interfaces

Menú: Interfaces → Assignments

- Agregar:
 - **vlan01 (Tag 10)** como OPT1
 - **vlan02 (Tag 20)** como OPT2

Interface	Identifier	Device
[LAN]	lan	vtnet1 (bc:24:11:10:97:81)
[OPT1]	opt1	vlan01 VLAN10 (Parent: vtnet1, Tag: 10)
[OPT2]	opt2	vlan02 VLAN20 (Parent: vtnet1, Tag: 20)
[WAN]	wan	vtnet0 (bc:24:11:f4:a1:59)

Paso 7 — Activación de Interfaces

Menú: Interfaces → OPT1 / OPT2

- Habilitar interface
- IPv4 Configuration Type: **Static IPv4**

Configuración:

Interfaz	IP
OPT1 (VLAN10)	192.168.10.1/24
OPT2 (VLAN20)	192.168.20.1/24

Interfaces: [OPT1]

Basic configuration

- Enable Enable Interface
- Lock Prevent interface removal
- Identifier opt1
- Device vlan01
- Description OPT1

Generic configuration

- Block private networks
- Block bogon networks
- IPv4 Configuration Type

Static IPv4 configuration

- IPv4 address 24
- IPv4 gateway rules

Buttons: Save Cancel

Paso 8 — Configuración de DHCP para cada VLAN

Menú: Services → DHCPv4

- VLAN10 → Range: 192.168.10.100 - 192.168.10.200

- VLAN20 → Range: 192.168.20.100 - 192.168.20.200

The screenshot shows the OPNsense Firewall configuration interface. On the left, a sidebar lists various services: VPN, Services (selected), Captive Portal, DHCRelay, Dnsmasq DNS, Intrusion Detection (disabled), ISC DHCPv4 (selected), [LAN], [OPT1] (selected), [OPT2], Leases, Log File, ISC DHCPv6, Kea DHCP, Monit, Network Time, OpenDNS, Unbound DNS, and Power. The main panel displays the configuration for the ISC DHCPv4 service on the OPT1 interface. It includes fields for Enable (checked), Deny unknown clients (unchecked), Ignore Client UIDs (unchecked), Subnet (192.168.10.0), Subnet mask (255.255.255.0), Available range (192.168.10.1 - 192.168.10.254), Range (from 192.168.10.100 to 192.168.10.200), Additional Pools, WINS servers, and DNS servers.

Paso 9 — Configuración de NAT (modo automático)

Menú: Firewall → NAT → Outbound

- Se mantiene en: **Automatic outbound NAT rule generation**

Se generan automáticamente las reglas para:

- 192.168.1.0/24
- 192.168.10.0/24
- 192.168.20.0/24

Paso 10 — Ajuste final en Proxmox (muy importante)

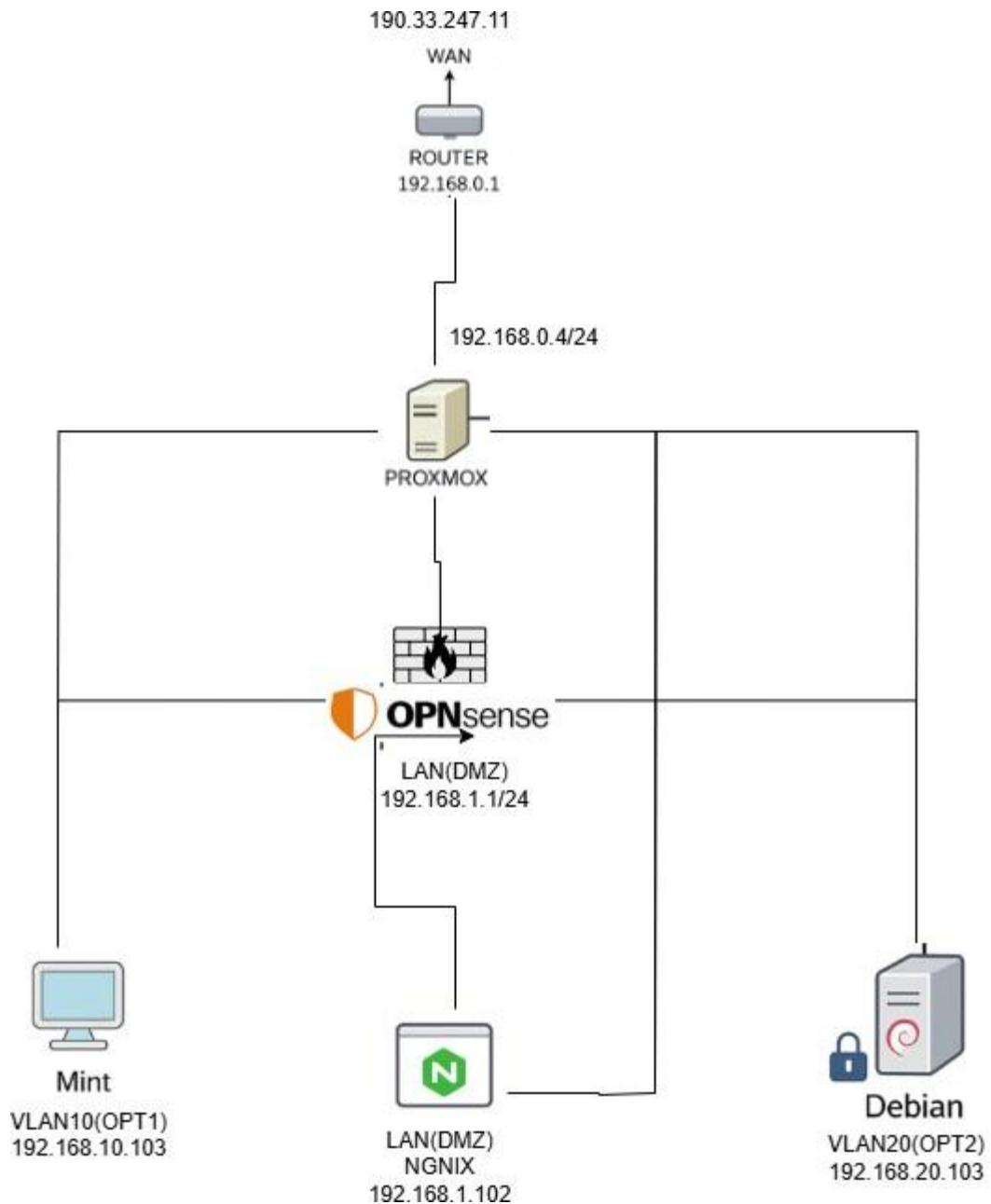
- Confirmar que las VMs cliente en VLAN10 o VLAN20 tengan:
 - **Bridge:** vmbr0
 - **VLAN Tag:** 10 o 20 (según la VM)
 - + **IMPORTANTE:** No poner ningún tag extra en la VM de OPNsense (solo en las VMs cliente).

TOPOLOGÍA DE RED CON LOS SIGUIENTES ELEMENTOS:

Resumen de la topología final

Segmento	Interfaz	Subred	Equipo(s) principales
WAN	vtnet0	192.168.0.29/24	Salida a Internet (Proveedor ISP local)
LAN (DMZ)	vtnet1	192.168.1.0/24	NGINX Proxy Manager → 192.168.1.102
VLAN10 (OPT1)	vlan01 (sobre vtnet1)	192.168.10.0/24	Mint → 192.168.10.103
VLAN20 (OPT2)	vlan02 (sobre vtnet1)	192.168.20.0/24	Debian → 192.168.20.103
Tailscale	TLSC	100.93.156.12	Administración remota

- Todas estas redes están virtualizadas sobre un solo host Proxmox usando un solo bridge vmbr0 con VLAN Aware habilitado.



Reglas de Firewall — Implementación final

Resumen exacto de lo que vamos a hacer ahora:

Estado de red:

Segmento	Interfaz	Subred	Equipo(s) principales
WAN	vtnet0	192.168.0.29/24	Acceso Internet
LAN (vtnet1)	vtnet1	192.168.1.0/24	NGINX Proxy Manager → 192.168.1.102
VLAN10 (OPT1)	vlan01	192.168.10.0/24	Mint 192.168.10.103
VLAN20 (OPT2)	vlan02	192.168.20.0/24	Debian 192.168.20.103
Tailscale	TLSC	100.93.156.12	Administración remota

Permitir HTTP/HTTPS desde VLAN10 (OPT1) hacia NGINX DMZ (LAN)

Regla en OPT1 (VLAN10):

- Interfaz: OPT1
- Action: Pass
- Protocol: TCP/UDP
- Source: OPT1 net
- Destination: LAN red
- Destination Port: HTTP (80), HTTPS (443) (o 81 si el NPM escucha ahí como veo en tu nota)
- Description: Permitir HTTP/HTTPS hacia DMZ (NGINX)

Cortafuegos: Reglas: OPT1

Seleccionar categoría

<input type="checkbox"/>	Protocolo	Origen	Puerto	Destino	Puerto	Puerta de Enlace	Programar	Descripción <small>?</small>
<small>Reglas generadas automáticamente</small>								
<input type="checkbox"/>	IPv4+6 TCP/UDP	OPT1 red	*	LAN red	*	*	*	Permitir HTTP/HTTPS hacia DMZ (NGINX)

Bloquear pings externos (ICMP bloqueado desde OPT2 hacia el 192.168.1.102)

En OPNsense → Firewall → Rules → OPT2

- Action: Block

- **Interface:** OPT2
- **Protocol:** ICMP
- **Source:** OPT2 red
- **Destination:** LAN red
- **Description:** Bloquear pings externos

Cortafuegos: Reglas: OPT2

	Protocolo	Origen	Puerto	Destino	Puerto	Puerta de Enlace	Programar	Descripción
	IPv4+6 ICMP	OPT2 red	*	LAN red	*	*		bloquear del 20 al 1 dmz
	permitir	bloquear	rechazar	registro	entrada			primera coincidencia
	permitir (deshabilitado)	bloquear (deshabilitado)	rechazar (deshabilitado)	registro (deshabilitado)	salida			última coincidencia

Reglas generadas automáticamente

Programación Activa/Inactiva (clic para ver/editar)

Limitar tráfico SSH solo desde ciertas IPs

Supongamos que solo desde la OPT1 permitiremos hacer SSH a las VMs:

En OPNsense → Firewall → Rules → OPT1

- Action: bloquear
- Protocol: TCP
- Source: 192.168.10.103
- Destination: 192.168.1.102
- Destination port: 22 (SSH)
- Description: BLOQUER SSH

En todos los demás interfaces (LAN, DMZ):

- Bloqueas explícitamente el puerto 22.

Esto asegura que sólo las máquinas internas de administración puedan usar SSH.

Cortafuegos: Reglas: OPT1

Los cambios se aplicaron satisfactoriamente.

	Protocolo	Origen	Puerto	Destino	Puerto	Puerta de Enlace	Programar	Descripción
	IPv4 TCP	192.168.10.103/32	*	192.168.1.102	22 (SSH)	*		bloquear ssh de la opt1
	IPv4+6 TCP/UDP	OPT1 red	*	LAN red	*	*		Permitir HTTP/HTTPS hacia DMZ (NGINX)
	IPv4 TCP/UDP	OPT1 red	*	*	*	*		permitir salida a internet
	permitir			rechazar				prime
	bloquear							

Reglas generadas automáticamente

Simular ataques básicos con nmap

Ahora viene la validación:

Desde una máquina de pruebas (por ejemplo tu Debian o Mint):

```
# Intentar escanear el OPNsense directamente (debe estar blindado en WAN)
```

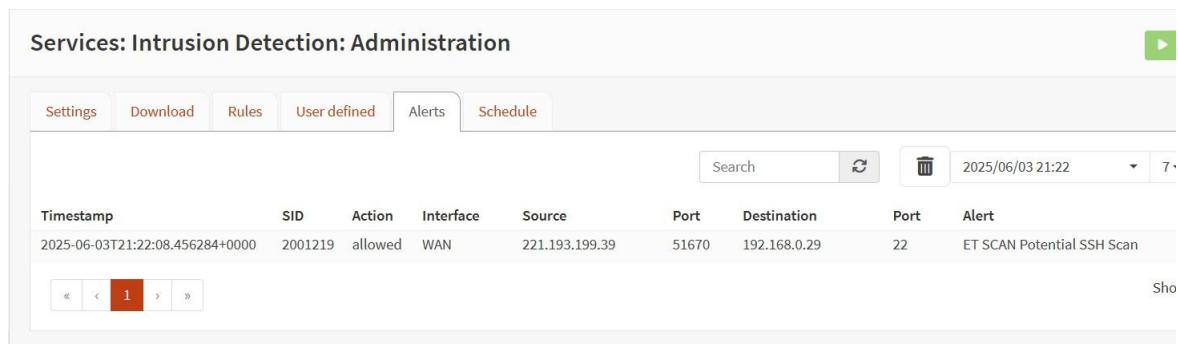
```
nmap 192.168.0.29
```

```
root@debi:~# nmap 192.168.0.29 ←
Starting Nmap 7.93 ( https://nmap.org ) at 2025-06-03 21:22 UTC
Nmap scan report for 192.168.0.29
Host is up (0.00024s latency).
All 1000 scanned ports on 192.168.0.29 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: BC:24:11:F4:A1:59 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 21.16 seconds
root@debi:~# █
```

¿Qué vemos en este log?

- Está detectando:
 - **ET SCAN Potential SSH Scan**
- Fuente: direcciones IP externas (ataques reales desde internet)
- Destino: 192.168.0.29 (que es tu interfaz WAN en la red interna de Proxmox/OPNsense)
- Acción: allowed (significa que lo detectó, pero no lo bloqueó).



The screenshot shows a web-based interface for managing intrusion detection alerts. At the top, there's a header bar with tabs: Settings, Download, Rules, User defined, Alerts (which is the active tab), and Schedule. Below the header is a search bar and a date/time filter set to 2025/06/03 21:22. A table displays a single row of data:

Timestamp	SID	Action	Interface	Source	Port	Destination	Port	Alert
2025-06-03T21:22:08.456284+0000	2001219	allowed	WAN	221.193.199.39	51670	192.168.0.29	22	ET SCAN Potential SSH Scan

At the bottom of the table, there are navigation buttons for page selection, with the number '1' highlighted in red.

Conclusión

La infraestructura construida cumplió con los objetivos de segmentación, control de tráfico y seguridad. A través de la correcta configuración de VLANs en Proxmox y OPNsense, se logró establecer zonas independientes de red, minimizando la exposición y riesgos entre segmentos internos.

La utilización de la LAN como DMZ permitió centralizar los servicios web de la organización, administrados mediante NGINX Proxy Manager, mientras que las VLAN10 y VLAN20 alojan los equipos de operación y servidores internos.

Además, la integración de Tailscale permitió obtener un acceso remoto seguro, cifrado punto a punto, con certificados TLS válidos y administración vía HTTPS, eliminando la necesidad de exponer el firewall directamente al público.

Finalmente, las políticas de firewall se ajustaron de forma precisa:

- El tráfico SSH está estrictamente controlado por VLAN.
- El tráfico ICMP externo bloqueado en la WAN.
- Los accesos HTTP/HTTPS permitidos de forma controlada hacia la DMZ.

El laboratorio refleja una topología robusta, escalable y muy cercana a escenarios productivos reales.