

Blue Team

Caso práctico

Gerard Díaz Hoyos

KeepCoding
Bootcamp de Ciberseguridad
5a edición

En la presente práctica, se pretende documentar la creación de la siguiente infraestructura de red:



La cual, deberá cumplir los siguientes requisitos:

- Creación de un **PfSense** en *bridge* que conecte 3 redes, **LAN**, **DMZ** y **DMZ_2**, éstas como red interna.
- Un equipo **W11** en **LAN**, un **stack ELK** en **DMZ** y un grupo de *honeypots* en **DMZ_2**.
- Queremos transmitir los *logs* de los *honeypots* al **ELK stack**, pero los *honeypots* no deben tener acceso a las otras redes(solo para transmitir *logs*) y deben ser accesibles desde la red **WAN**.
- El servidor **ELK** debe almacenar y poder visualizar los diferentes *logs* de los *honeypots*.
- El **W10** debe poder conectarse a **ELK** vía **Kibana**.

Para ello desplegaremos varias máquinas virtuales que compondrán básicamente el laboratorio de pruebas pertinente. En este caso, utilizaremos la herramienta de virtualización **VirtualBox** en la que tendremos las siguientes máquinas:

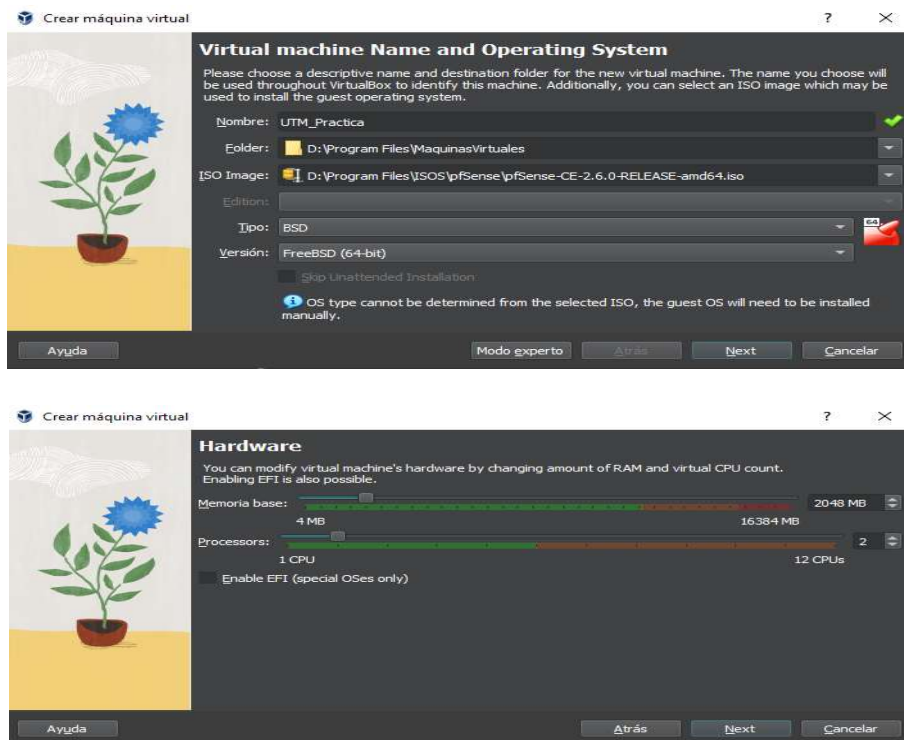
- **PfSense**, que será el **UTM** y el que representará la **red WAN o de Internet**.
- **Windows 11**, representando la **red LAN o Local**.
- **Kali Linux**, como red **DMZ_H**, en la que instalaremos un **IDS** y un **Honeypot**.

Y necesitaremos el siguiente software:

- **Elasticsearch**
- **Honeypot Cowrie**
- **OpenVPN Connect**
- **Suricata**

PfSense

Nos descargamos la [ISO](#) desde la página web oficial y creamos nuestra máquina virtual de este modo (escogemos el nombre que mejor la identifique, escogemos la ruta en la que se instalará, cargamos la ISO, seleccionamos que sea tipo BSD y en la versión FreeBSD (64-bit) y le dedicamos la potencia y disco duro que creamos conveniente, teniendo en cuenta que no necesitará demasiados recursos).



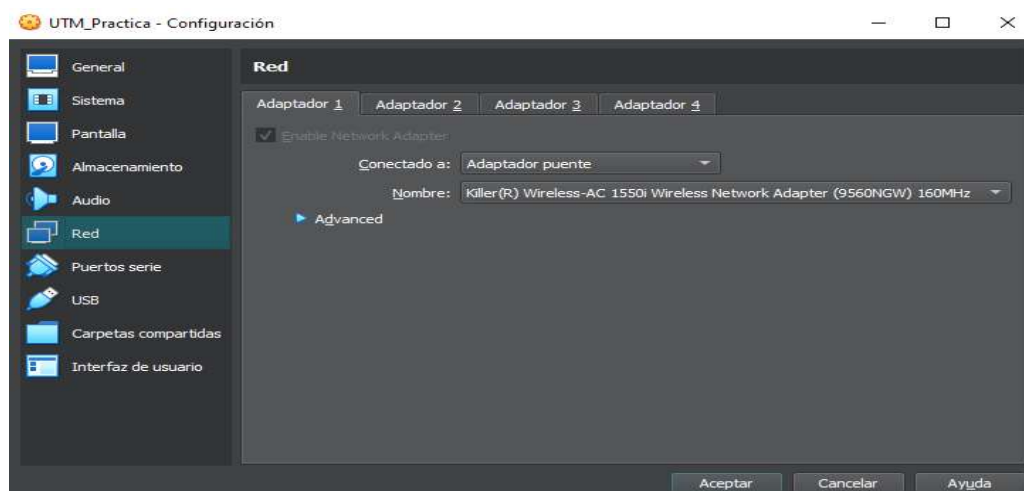


Y ya la tendremos disponible:

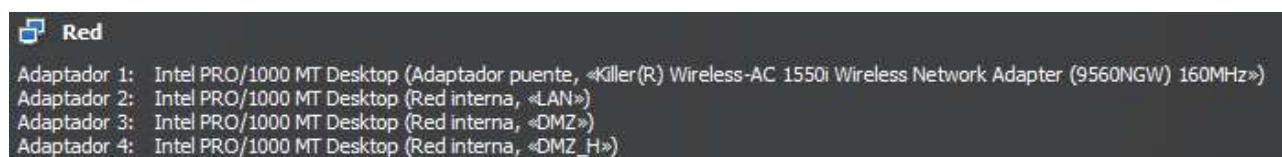


Configuración de la máquina virtual:

Aquí nos encargaremos de configurar la red UTM, dentro del panel de redes y estableceremos los 4 adaptadores que tenemos disponibles:



Deberemos acabar este paso teniendo la siguiente configuración:

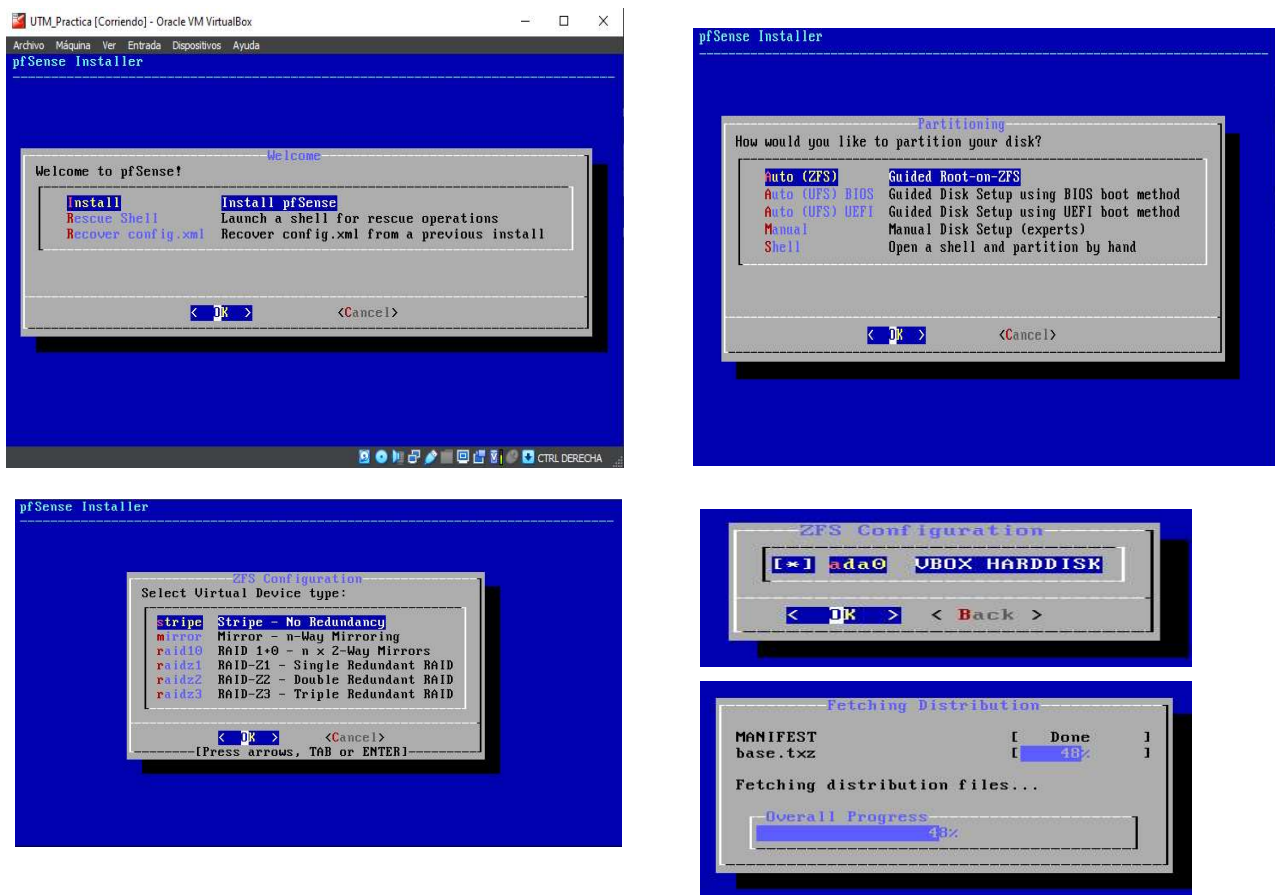


Como vemos, PfSense representará la red WAN y las otras 3 “ramas” serán la red interna, siendo 2 de ellas DMZ con funciones específicas que se comentarán posteriormente y el Windows la red LAN.

(Hay que recordar seleccionar, dentro de la pestaña de “Almacenamiento”, la unidad de *pfSense-CE 2.6.0* y marcar la casilla de *CD/DVD vivo*)

Instalación de PfSense:

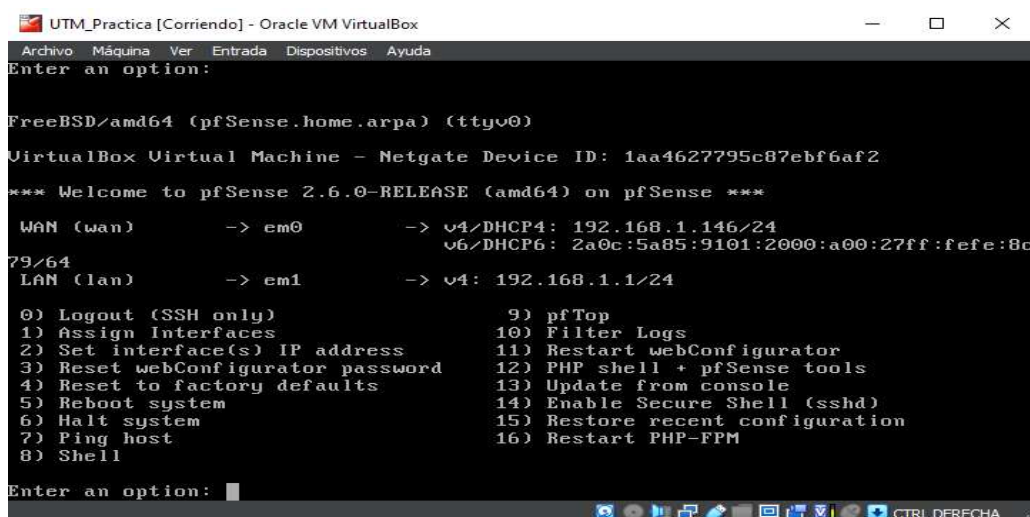
En este paso, iniciaremos la máquina virtual recién creada y comenzará un proceso de instalación de PfSense, en el que seguiremos los pasos siguientes:



Se reiniciará el PfSense y tendremos en cuenta de seleccionar la opción “eliminar disco de la unidad virtual”, dentro de la pestaña de dispositivos del menú de VirtualBox, para que arranque correctamente.

Configuración PfSense:

Una vez instalado, ya estaremos en el panel de control con el menú central, en el cual empezaremos a configurar las distintas redes:



Lo que primero que habrá que hacer será **asignar las interfaces** correspondientes. 4 “ramas” de nuestra infraestructura de red, 4 interfaces:

```
Enter the WAN interface name or 'a' for auto-detection
(em0 em1 em2 em3 or a): em0
```

```
Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(em1 em2 em3 a or nothing if finished): em1
```

```
Enter the Optional 1 interface name or 'a' for auto-detection
(em2 em3 a or nothing if finished): em2
```

```
Enter the Optional 2 interface name or 'a' for auto-detection
(em3 a or nothing if finished): em3
```

```
WAN -> em0
LAN -> em1
OPT1 -> em2
OPT2 -> em3
```

Seguidamente, estableceremos las IPs de las interfaces, con la opción 2 de “*Set interface(s) IP address*”:

Escogemos la **IP 192.168.100.254** para nuestro Windows o red LAN, con *subnet* o rango 24.

```
Enter the new LAN IPv4 address. Press <ENTER> for none:
> 192.168.100.254
```

```
Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0     = 8
```

```
Enter the new LAN IPv4 subnet bit count (1 to 32):
> 24
```

Trabajaremos con **Ipv4** (por ser el protocolo más extendido aún a día de hoy) y, por ello, no habilitaremos ninguna IP en protocolo Ipv6.

Habilitaremos el servidor DHCP, que nos asignará direcciones entre los valores que establezcamos a continuación.

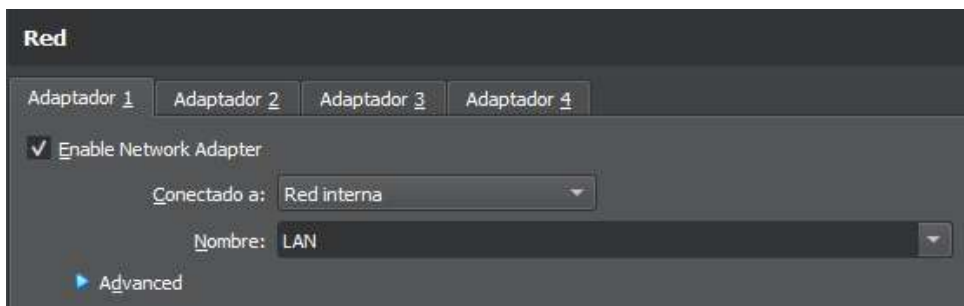
El rango escogido será de **192.168.100.100** a **192.168.100.200**.

Y quedará establecida la IP escogida y podremos empezar a movernos a la interfaz gráfica de PfSense desde la máquina virtual de Windows, escribiendo directamente la IP en el navegador.

```
The IPv4 LAN address has been set to 192.168.100.254/24
You can now access the webConfigurator by opening the following URL in your web
browser:
      http://192.168.100.254/
```

```
WAN (wan)      -> em0      -> v4/DHCP4: 192.168.1.146/24
                                   v6/DHCP6: 2a0c:5a85:9101:2000:a00:27ff:fefe:8c
79/64
LAN (lan)      -> em1      -> v4: 192.168.100.254/24
OPT1 (opt1)    -> em2      ->
OPT2 (opt2)    -> em3      ->
```

Antes de nada, desde el panel de configuración de VirtualBox, cambiaremos la red que tenemos predeterminada a NAT por Red Interna y LAN, para que el tráfico vaya etiquetándose de tal manera que llegue a esta red.



Interfaz gráfica de PfSense:

Iniciamos Windows.

Primero hacemos una comprobación rutinaria, observando si está establecida la IP que hemos seleccionado en los pasos anteriores. Abrimos la terminal y usamos el comando *ipconfig*:

```
C:\Users\User>ipconfig

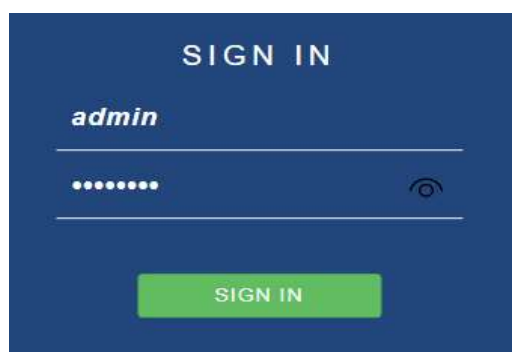
Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : home.arpa
    Link-local IPv6 Address . . . . . : fe80::2daf:7b00:c56f:babb%5
    IPv4 Address. . . . . : 192.168.100.100
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.100.254
```

Nos aparece la misma IP.

Entramos al navegador *Microsoft Edge* e ingresamos la dirección <https://192.168.100.254>. Introducimos las credenciales por defecto (admin/pfsense) y entramos a un panel de control y configuración gráfico:



Después de pasar de las 2 primeras ventanas informativas, llegaremos a un panel de configuración de **información general**, donde estableceremos el nombre del *host* y del dominio, las DNS primaria y secundaria y habilitaremos la casilla de *Override DNS* (ya por defecto como tal) para poder sobrescribirlos si nos envían un DHCP.

(Los servidores DNS pertenecen a *Cloudflare* – 1.1.1.1 – y a *Google* – 8.8.8.8 –)

General Information

On this screen the general pfSense parameters will be set.

Hostname

UTM

EXAMPLE: myserver

Domain

home

EXAMPLE: mydomain.com

The default behavior of the DNS Resolver will ignore manually configured DNS servers for client queries and query root DNS servers directly. To use the manually configured DNS servers below for client queries, visit Services > DNS Resolver and enable DNS Query Forwarding after completing the wizard.

Primary DNS Server

1.1.1.1

Secondary DNS Server

8.8.8.8

Override DNS

☒

Allow DNS servers to be overridden by DHCP/PPP on WAN

En la siguiente ventana, simplemente estableceremos la zona horario como *Europe/Madrid*.

En los siguientes pasos, prácticamente todo permanecerá por defecto. Nos aseguraremos, no obstante, de que no estén marcadas las casillas que bloquean redes privadas y *bogons* (puesto que ya crearemos una VPN para este tipo de funciones).

Finalizaremos los 9 pasos correspondientes a esta configuración de la *Información General* y confirmaremos los cambios realizados; ya estaremos en la pantalla principal de PfSense con todas las opciones disponibles.

pfSense

COMMUNITY EDITION

System

Interfaces

Firewall

Services

VPN

Status

Diagnostics

Help

WARNING: The 'admin' account password is set to the default value. [Change the password in the User Manager.](#)

Status / Dashboard

System Information

Name

pfSense.home.arpa

User

admin@192.168.100.100 (Local Database)

System

VirtualBox Virtual Machine

Netgate Device ID: 1aa4627795c87ebf6af2

Netgate Services And Support

Contract type

Community Support

Community Support Only

NETGATE AND pfSense COMMUNITY SUPPORT RESOURCES

Vamos a habilitar las otras 2 interfaces de red: la **DMZ** y la **DMZ_H**; lo haremos desde la pestaña “*Interfaces*” y **OPT1** y **OPT2** respectivamente, habilitándolas primeramente marcando la casilla del inicio.

OPT1:

General Configuration

Enable

☒ Enable interface

Description

DMZ

Enter a description (name) for the interface here.

IPv4 Configuration Type

Static IPv4

IPv6 Configuration Type

None

Static IPv4 Configuration

IPv4 Address

192.168.90.1

/ 24

IPv4 Upstream gateway

None

+ Add a new gateway

If this interface is an Internet connection, select an existing Gateway from the list or add a new one using the "Add" button.

OPT2:

General Configuration

Enable

☒ Enable interface

Description

DMZ_H

Enter a description (name) for the interface here.

IPv4 Configuration Type

Static IPv4

IPv6 Configuration Type

None

Static IPv4 Configuration

IPv4 Address

192.168.50.1

/ 24

IPv4 Upstream gateway

None

+ Add a new gateway

If this interface is an Internet connection, select an existing Gateway from the list or add a new one using the "Add" button.
 On local area network interfaces the upstream gateway should be "none".
 Selecting an upstream gateway causes the firewall to treat this interface as a [WAN type interface](#).
 Gateways can be managed by [clicking here](#).

PfSense ya nos mostrará los cambios efectuados:

```

WAN (wan)      -> em0      -> v4/DHCP4: 192.168.1.146/24
                v6/DHCP6: 2a0c:5a85:9101:2000:a00:27ff:fe78:925f
LAN (lan)      -> em1      -> v4: 192.168.100.254/24
DMZ (opt1)     -> em2      -> v4: 192.168.90.1/24
DMZ_H (opt2)   -> em3      -> v4: 192.168.50.1/24
  
```

| Interfaces ⚙️ - ✕ | | | |
|---|---|-------------------------|---|
|  WAN | ↑ | 1000baseT <full-duplex> | 192.168.1.147 2a0c:5a85:9101:2000:a00:27ff:fe78:925f |
|  LAN | ↑ | 1000baseT <full-duplex> | 192.168.100.254 |
|  DMZ | ↑ | 1000baseT <full-duplex> | 192.168.90.1 |
|  DMZ_H | ↑ | 1000baseT <full-duplex> | 192.168.50.1 |

Para que estas dos interfaces de red (DMZ y DMZ_H) tengan conexión a Internet será necesario habilitarles un **servidor DHCP** ; además, el DHCP nos permite asignar IPs concretas sólo sabiendo la MAC del equipo en cuestión. Desde la pestaña de **Servicios y DHCP Server**:

Services / DHCP Server / DMZ

LAN DMZ DMZ_H

General Options

Enable

☒ Enable DHCP server on DMZ interface

BOOTP

☐ Ignore BOOTP queries

Deny unknown clients

Allow all clients

When set to **Allow all clients**, any DHCP client will get an IP address within this scope/range on this interface. If set to **Allow known clients from any interface**, any DHCP client with a MAC address listed on **any** scope(s)/interface(s) will get an IP address. If set to **Allow known clients from only this interface**, only MAC addresses listed below (i.e. for this interface) will get an IP address within this scope/range.

Ignore denied clients

☐ Denied clients will be ignored rather than rejected.
This option is not compatible with failover and cannot be enabled when a Failover Peer IP address is configured.

Ignore client identifiers

☐ If a client includes a unique identifier in its DHCP request, that UID will not be recorded in its lease.
This option may be useful when a client can dual boot using different client identifiers but the same hardware (MAC) address. Note that the resulting server behavior violates the official DHCP specification.

Subnet

192.168.90.0

Subnet mask

255.255.255.0

Available range

192.168.90.1 - 192.168.90.254

Range

192.168.90.100

192.168.90.200

FromTo

Other Options

Gateway

192.168.90.1

The default is to use the IP on this interface of the firewall as the gateway. Specify an alternate gateway here if this is not the correct gateway for the network. Type "none" for no gateway assignment.

Servers

WINS servers

WINS Server 1

WINS Server 2

DNS servers

1.1.1.1

8.8.8.8

DNS Server 3

Hemos establecido el rango de IPs, los DNS y la puerta de enlace para la interfaz DMZ.

Haremos lo mismo con la interfaz DMZ_H:

LAN

DMZ

DMZ_H

General Options

Enable

☒ Enable DHCP server on DMZ_H interface

BOOTP

☐ Ignore BOOTP queries

Deny unknown clients

Allow all clients

When set to **Allow all clients**, any DHCP client will get an IP address within this scope/range on this interface. If set to **Allow known clients from any interface**, any DHCP client with a MAC address listed on **any** scope(s)/interface(s) will get an IP address. If set to **Allow known clients from only this interface**, only MAC addresses listed below (i.e. for this interface) will get an IP address within this scope/range.

Ignore denied clients

☐ Denied clients will be ignored rather than rejected.
This option is not compatible with failover and cannot be enabled when a Failover Peer IP address is configured.

Ignore client identifiers

☐ If a client includes a unique identifier in its DHCP request, that UID will not be recorded in its lease.
This option may be useful when a client can dual boot using different client identifiers but the same hardware (MAC) address. Note that the resulting server behavior violates the official DHCP specification.

Subnet

192.168.50.0

Subnet mask

255.255.255.0

Available range

192.168.50.1 - 192.168.50.254

Range

192.168.50.100

From

192.168.50.200

To

DNS servers

1.1.1.1

8.8.8.8

Gateway

192.168.50.1

The default is to use the IP on this interface of the firewall as the gateway. Specify an alternate gateway here if this is not the correct gateway for the network. Type "none" for no gateway assignment.

Y en sendas interfaces, repararemos en la parte final de la ventana de configuración, donde podremos añadir en cada una un **DHCP estático** (“Add” “*DHCP Static Mappings for this interface*”). Esto nos habilitará la posibilidad de configurar de manera específica un determinado cliente, basándonos en su dirección MAC.

Deberemos especificar la *Dirección MAC* (que en este caso está bastante automatizado y sólo deberemos darle con el ratón al botón de “*Copy My MAC*”), especificaremos el identificador y el nombre del *Host* como **elastic/honey**, volveremos a especificar los DNS (1.1.1.1 y 8.8.8.8), asignaremos las IPs **192.168.90.225/192.168.50.225** y las puertas de enlace **192.168.90.1/192.168.50.1**.

Obtendremos la siguiente información:

DMZ:

| DHCP Static Mappings for this Interface (total: 1) | | | | | |
|--|-------------------|-----------|----------------|----------|---|
| Static ARP | MAC address | Client Id | IP address | Hostname | Description |
| | 08:00:27:cc:67:0e | elastic | 192.168.90.225 | elastic |   |

DMZ H:

| DHCP Static Mappings for this Interface (total: 1) | | | | | |
|--|-------------------|-----------|----------------|----------|---|
| Static ARP | MAC address | Client Id | IP address | Hostname | Description |
| | 08:00:27:cc:67:0e | honey | 192.168.50.225 | honey |   |

En este punto podríamos hacer unas pequeñas comprobaciones rápidas para ver si, desde Windows, detecta correctamente las configuraciones que hemos ido haciendo. Nos situamos en el panel de opciones de red de VirtualBox, desde la ventana de la máquina virtual de Windows 11 y cambiamos la red interna a DMZ (en vez de LAN); en la consola de Windows – con el comando *ipconfig* – veremos como la IP es diferente, es la que hemos establecido en los últimos pasos.

```
C:\Users\User>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : home
    Link-local IPv6 Address . . . . . : fe80::2daf:7b00:c56f:babb%5
    IPv4 Address. . . . . : 192.168.90.225
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.90.1
```

Lo mismo ocurre si cambiamos a DMZ_H:

```
C:\Users\User>ipconfig

Windows IP Configuration






Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : home
    Link-local IPv6 Address . . . . . : fe80::2daf:7b00:c56f:babb%5
    IPv4 Address. . . . . : 192.168.50.225
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.50.1
```

Creación de una VPN:

Antes de crear la VPN como tal, tendremos que realizar algunos pasos antes.

Por ejemplo, será necesario instalar un paquete. PfSense permite localizar diversos paquetes o softwares disponibles de usos variados e instalarlos directamente desde la interfaz gráfica (ésta es una de las diferencias con respecto a los *Firewall* simples, ya que estos no tienen esta opción). Para ello, nos vamos a la pestaña “*System*” y “*Package Manager*”. En concreto, el que necesitamos instalar se llama **openvpn-client-export**; lo encontraremos fácilmente en el buscador que hay integrado en la ventana de “*Available packages*”.

| | | | |
|---|-------|---|---|
| openvpn-client-export | 1.6_9 | Allows a pre-configured OpenVPN Windows Client or Mac OS X's Viscosity configuration bundle to be exported directly from pfSense. |  |
| Package Dependencies: | | | |
|  openvpn-client-export-2.5.8  openvpn-2.5.4_1  zip-3.0_1  p7zip-16.02_3 | | | |

Este paquete se utiliza para exportar la configuración del cliente OpenVPN desde pfSense en un archivo de configuración que puede ser utilizado por un cliente OpenVPN para conectarse a la red VPN configurada.

Le daremos a *Install* y esperaremos unos minutos hasta que se instale correctamente.

El siguiente paso será crear un **servidor DNS** con PfSense (esto lo hacemos, sobre todo, para el tráfico interno de nuestra red).

Para ello nos iremos esta vez a la pestaña de “*Services*”, “*DNS Resolver*” y “*General settings*”. Nos desplazaremos a la parte inferior de la pantalla en la que estamos y buscaremos el apartado “*Host Overrides*”; le daremos al botón *Add* para comenzar la configuración:

Host Override Options

Host

homedns

Name of the host, without the domain part
e.g. enter "myhost" if the full domain name is "myhost.example.com"

Domain

home.local

Parent domain of the host
e.g. enter "example.com" for "myhost.example.com"

IP Address

192.168.100.100

IPv4 or IPv6 comma-separated addresses to be returned for the host
e.g.: 192.168.100.100 or fd00:abcd::
or list 192.168.1.3,192.168.4.5,fc00:123::3

Description

Servidor Windows

A description may be entered here for administrative reference (not parsed).

This page is used to override the usual lookup process for a specific host. A host is defined by its name and parent domain (e.g., 'somesite.google.com' is entered as host='somesite' and parent domain='google.com'). Any attempt to lookup that host will automatically return the given IP address, and any usual external lookup server for the domain will not be queried. Both the name and parent domain can contain 'non-standard', 'invalid' and 'local' domains such as 'test', 'nas.home.arpa', 'mycompany.localdomain', or '1.168.192.in-addr.arpa', as well as usual publicly resolvable names such as 'www' or 'google.co.uk'.

Lo que estamos haciendo, al fin y al cabo, es que cada vez que un equipo de nuestra red LAN vaya al dominio declarado, se le mande directamente a la IP especificada.

| Host Overrides | | | | |
|----------------|-----------------------|-----------------------|------------------|---|
| Host | Parent domain of host | IP to return for host | Description | Actions |
| homedns | home.local | 192.168.100.100 | Servidor Windows |   |

Otro paso que podemos realizar para aumentar la seguridad de nuestra infraestructura de red sería habilitar el HTTPS (SSL/TLS). Lo haríamos desde “*System*”, “*Advanced*”. Simplemente deberemos marcar la casilla correspondiente y guardar los cambios.

webConfigurator

Protocol

☐ HTTP

☒ HTTPS (SSL/TLS)

SSL/TLS Certificate

webConfigurator default (6432e633b8608)

Certificates known to be incompatible with use for HTTPS are not included in this list.

Seguiremos por crear un **certificado**.

En primer lugar, crearemos una **CA (Autoridad Certificadora)**; lo haremos desde la pestaña “System”, “Certificate Manager”, “CAs” y al botón de **Añadir**.

Create / Edit CA

Descriptive name

home_certificate

Method

Create an internal Certificate Authority

Trust Store

☐ Add this Certificate Authority to the Operating System Trust Store

When enabled, the contents of the CA will be added to the trust store so that they will be trusted by the operating system.

Randomize Serial

☐ Use random serial numbers when signing certificates

When enabled, if this CA is capable of signing certificates then serial numbers for certificates signed by this CA will be automatically randomized and checked for uniqueness instead of using the sequential value from Next Certificate Serial.

Internal Certificate Authority

Key type

RSA

2048

The length to use when generating a new RSA key, in bits.
The Key Length should not be lower than 2048 or some platforms may consider the certificate invalid.

Digest Algorithm

sha256

The digest method used when the CA is signed.
The best practice is to use an algorithm stronger than SHA1. Some platforms may consider weaker digest algorithms invalid

Lifetime (days)





3650

Common Name

home_certificate

También especificaremos información relativa a nuestra situación geográfica.

Como se puede observar, entre otras cosas, se está definiendo el tipo de encriptado (*RSA-SHA 256*) que usará la certificación.

| Certificate Authorities | | | | | | | |
|-------------------------|----------|-------------|--------------|--|--|--------|---|
| Name | Internal | Issuer | Certificates | Distinguished Name | | In Use | Actions |
| home_certificate | ✓ | self-signed | 0 | ST=Barcelona, O=home, L=Barcelona, CN=home_certificate, C=ES | | |     |
| | | | | Valid From: Sun, 09 Apr 2023 21:56:39 +0200 | | | |
| | | | | Valid Until: Wed, 06 Apr 2033 21:56:39 +0200 | | | |

A partir de ahora ya podremos crear nuestros certificados.

Para ello, dentro de la misma ventana, nos dirigiremos a “Certificates” y le daremos al botón de **Añadir**:

| Add/Sign a New Certificate | |
|--|--|
| Method | Create an internal Certificate |
| Descriptive name | vpn.home.local |
| Internal Certificate | |
| Certificate authority | home_certificate |
| Key type | RSA |
| | 2048 The length to use when generating a new RSA key, in bits. The Key Length should not be lower than 2048 or some platforms may consider the certificate invalid. |
| Digest Algorithm | sha256 The digest method used when the certificate is signed. The best practice is to use an algorithm stronger than SHA1. Some platforms may consider weaker digest algorithms invalid. |
| Lifetime (days) | 365 The length of time the signed certificate will be valid, in days. Server certificates should not have a lifetime over 398 days or some platforms may consider the certificate invalid. |
| Common Name | vpn.home.local |
| The following certificate subject components are optional and may be left blank. | |
| Country Code | ES |
| State or Province | Barcelona |
| City | Barcelona |
| Organization | home |

Es importante escoger “*Server Certificate*” en el siguiente campo:

| | |
|------------------|--------------------|
| Certificate Type | Server Certificate |
|------------------|--------------------|

Add type-specific usage attributes to the signed certificate. Used for placing usage restrictions on, or granting abilities to, the signed certificate.

Automáticamente, identifica el CA que hemos creado justo en el paso anterior.

El propio PfSense nos recomienda que tengamos en cuenta que los certificados no deberían tener una validez superior a los 398 días; por eso, sería recomendable cambiar el valor por defecto que nos aparece (3650 días) y establecer, por ejemplo, un año.

Y como ya tenemos un certificado para el servidor de VPN, podemos proceder a crearlo. Lo haremos desde la pestaña “*VPN*”, “*OpenVPN*”, “*Servers*”:

| General Information | |
|----------------------------|--|
| Description | VPN to LAN A description of this VPN for administrative reference. |
| Disabled | <input type="checkbox"/> Disable this server Set this option to disable this server without removing it from the list. |
| Mode Configuration | |
| Server mode | Remote Access (SSL/TLS + User Auth) |
| Backend for authentication | Local Database |
| Device mode | tun - Layer 3 Tunnel Mode *tun* mode carries IPv4 and IPv6 (OSI layer 3) and is the most common and compatible mode across all platforms. *tap* mode is capable of carrying 802.3 (OSI Layer 2.) |

Endpoint Configuration

Protocol

TCP on IPv4 only

Interface

WAN

The interface or Virtual IP address where OpenVPN will receive client connections.

Local port

1194

The port used by OpenVPN to receive client connections.

Cryptographic Settings

TLS Configuration

☒ Use a TLS Key

A TLS key enhances security of an OpenVPN connection by requiring both parties to have a common key before a peer can perform a TLS handshake. This layer of HMAC authentication allows control channel packets without the proper key to be dropped, protecting the peers from attack or unauthorized connections. The TLS Key does not have any effect on tunnel data.

☒ Automatically generate a TLS Key.

Peer Certificate Authority

home_certificate

Peer Certificate Revocation list

No Certificate Revocation Lists defined. One may be created here: [System > Cert. Manager](#)

OCSP Check

☐ Check client certificates with OCSP

Server certificate

vpn.home.local (Server: Yes, CA: home_certificate)

DH Parameter Length

2048 bit

Diffie-Hellman (DH) parameter set used for key exchange. [i](#)

La siguiente opción puede ayudar a mejorar notablemente la velocidad:

Hardware Crypto

Intel RDRAND engine - RAND

Tunnel Settings

IPv4 Tunnel Network

192.168.225.0/24

This is the IPv4 virtual network or network type alias with a single entry used for private communications between this server and client hosts expressed using CIDR notation (e.g. 10.0.8.0/24). The first usable address in the network will be assigned to the server virtual interface. The remaining usable addresses will be assigned to connecting clients.

IPv6 Tunnel Network

This is the IPv6 virtual network or network type alias with a single entry used for private communications between this server and client hosts expressed using CIDR notation (e.g. fe80::/64). The ::1 address in the network will be assigned to the server virtual interface. The remaining addresses will be assigned to connecting clients.

Redirect IPv4 Gateway

☐ Force all client-generated IPv4 traffic through the tunnel.

Redirect IPv6 Gateway

☐ Force all client-generated IPv6 traffic through the tunnel.

IPv4 Local network(s)

192.168.90.0/24, 192.168.50.0/24, 192.168.100.0/24




IPv4 networks that will be accessible from the remote endpoint. Expressed as a comma-separated list of one or more CIDR ranges or host/network type aliases. This may be left blank if not adding a route to the local network through this tunnel on the remote machine. This is generally set to the LAN network.

A modo de explicación de algunos de los parámetros seleccionados:

- el “*server mode*” va a solicitar credenciales y certificado (por eso escogemos el *SSL/TLS + User Auth*)
- el “*device mode*”, en el que hemos seleccionado “*tun – Layer 3 Tunnel Mode*” se refiere al proceso de construcción de un túnel – una red intermedia – que va a ser donde se conecten los clientes de la VPN.
- aparece la CA y el Certificado creados anteriormente.
- el “*IPv4 Tunnel Network*” se refiere a la red del túnel que estamos creando, donde se van a conectar los clientes.

- el “*IPv4 Local Network(s)*” designa a qué redes vamos a poder conectarnos de nuestra infraestructura interna (*rooting*).

Guardaremos los cambios efectuados y nos aparece la VPN creada:

| OpenVPN Servers | | | | | |
|-----------------|----------------------|------------------|--|-------------|---|
| Interface | Protocol / Port | Tunnel Network | Mode / Crypto | Description | Actions |
| WAN | TCP4 / 1194 (TUN) | 192.168.225.0/24 | Mode: Remote Access (SSL/TLS + User Auth) Data Ciphers: AES-256-GCM, AES-128-GCM, CHACHA20-POLY1305, AES-256-CBC Digest: SHA256 D-H Params: 2048 bits | VPN to LAN |    |

Pero ahora necesitaremos tener algún usuario para la VPN recién creada.

Para ellos nos vamos a dirigir al apartado “*System*”, “*User Manager*”, “*Users*” y comenzaremos **añadiendo** uno.

User Properties

Defined by

USER

Disabled

☐ This user cannot login

Username

vpn

Password

Full name

vpn

User's full name, for administrative information only

Expiration date

04/16/2024

Leave blank if the account shouldn't expire, otherwise enter the expiration date as MM/DD/YYYY

Custom Settings

☐ Use individual customized GUI options and dashboard layout for this user.

Group membership

admins

Not member of

Member of

>>

Move to "Member of" list

<<

Move to "Not member of" list

Hold down CTRL (PC)/COMMAND (Mac) key to select multiple items.

Certificate

☒ Click to create a user certificate

Create Certificate for User

Descriptive name

vpn

Certificate authority

home_certificate

Key type

RSA

2048

The length to use when generating a new RSA key, in bits.

The Key Length should not be lower than 2048 or some platforms may consider the certificate invalid.

Lo más destacado es asegurarnos de marcar la casilla que crea certificados de usuario (“*click to create a user certificate*”).

Ahora ya nos aparecerá el nuevo usuario vpn:

| Users | | | | | |
|--------------------------|----------|----------------------|--------|--------|---------|
| | Username | Full name | Status | Groups | Actions |
| <input type="checkbox"/> | admin | System Administrator | ✓ | admins | |
| <input type="checkbox"/> | vpn | vpn | ✓ | | |

Definiendo reglas:

Este paso establecerá las formas en cómo se permita el tráfico entre las distintas redes de nuestra infraestructura.

Comenzaremos por crear una regla NAT (*Network Address Translation*) del *Firewall*; con ella, conseguiremos que la IP pública (la WAN) redireccione donde le indiquemos (traduce direcciones IP privadas a públicas o viceversa).

Podremos hacerlo desde la pestaña “*Firewall*”, “*NAT*”, “*Port Forward*” y *clickamos* en **añadir**.

Edit Redirect Entry

☐ Disabled

☐ Disable this rule

No RDR (NOT)

☐ Disable redirection for traffic matching this rule

This option is rarely needed. Don't use this without thorough knowledge of the implications.

Interface

WAN

Choose which interface this rule applies to. In most cases "WAN" is specified.

Address Family

IPv4

Select the Internet Protocol version this rule applies to.

Protocol

TCP

Choose which protocol this rule should match. In most cases "TCP" is specified.

Source

Display Advanced

Destination

☐ Invert match.

WAN address

Type

Address/mask

Destination port range

Other

9090

Other

9090

From port

Custom

To port

Custom

Specify the port or port range for the destination of the packet for this mapping. The 'to' field may be left empty if only mapping a single port.

Redirect target IP

Single host

192.168.90.225

Type

Address

Enter the internal IP address of the server on which to map the ports. e.g.: 192.168.1.12 for IPv4
In case of IPv6 addresses, it must be from the same "scope",
i.e. it is not possible to redirect from link-local addresses scope (fe80:*) to local scope (::1)

Redirect target port

Other

80

Port

Custom

Specify the port on the machine with the IP address entered above. In case of a port range, specify the beginning port of the range (the end port will be calculated automatically).
This is usually identical to the "From port" above.

Description

servidor web

A description may be entered here for administrative reference (not parsed).

| Rules | | | | | | | | | | |
|--------------------------|-----------|----------|----------------|--------------|---------------|-------------|--------|----------------|-------------|--------------|
| | Interface | Protocol | Source Address | Source Ports | Dest. Address | Dest. Ports | NAT IP | NAT Ports | Description | Actions |
| <input type="checkbox"/> | ✓ | WAN | TCP | * | * | WAN address | 9090 | 192.168.90.225 | 80 (HTTP) | servidor web |

En las redes DMZ no tenemos ninguna regla definida todavía:

Siempre deberíamos tener claro qué tráfico queremos permitir. En esta ocasión vamos a permitir el tráfico 80 (HTTP, servidores web), el tráfico 443 (servidores web seguros) y DNS (que irán a parte, porque utilizan el protocolo TCP/UDP).

PfSense nos permite agrupar varios elementos en uno, utilizando los alias, que es lo que configuraremos a continuación para tener estos 3 elementos listados en el párrafo anterior. Lo haremos desde la pestaña “Firewall”, “Alias”, “Ports” y le daremos al botón de **Añadir**:

Con el alias creado, procedemos a crear alguna regla para la red DMZ. Lo haremos desde la opción “Firewall”, “Rules”, “DMZ” y seleccionaremos el botón **Añadir**:

Como se puede observar, se ha usado ya el alias “webs” para designar que se permita el tráfico de destino hacia los puertos 80 y 443.

Como ahora faltaría permitir la misma regla con el puerto 53, correspondiente a servidores DNS, crearemos otra regla más, simplemente cambiando este dato y el protocolo que utilizan:

Edit Firewall Rule

Action

Pass

Choose what to do with packets that match the criteria specified below.
Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.

Disabled

☐ Disable this rule

Set this option to disable this rule without removing it from the list.

Interface

DMZ

Choose the interface from which packets must come to match this rule.

Address Family

IPv4

Select the Internet Protocol version this rule applies to.

Protocol

TCP/UDP

Choose which IP protocol this rule should match.

Source

Source

☐ Invert match

DMZ net

Source Address

/

Display Advanced

The Source Port Range for a connection is typically random and almost never equal to the destination port. In most cases this setting must remain at its default value, any.

Destination

Destination

☐ Invert match

any

Destination Address

/

Destination Port Range

DNS (53)

From

Custom

DNS (53)

To

Custom

Y ya tendremos disponibles y habilitadas las 2 reglas de la red DMZ:

| Rules (Drag to Change Order) | | | | | | | | | | | |
|------------------------------|-----------|--------------|---------|------|-------------|----------|---------|-------|----------|-------------|---------|
| <input type="checkbox"/> | States | Protocol | Source | Port | Destination | Port | Gateway | Queue | Schedule | Description | Actions |
| <input type="checkbox"/> | ✓ 0 / 0 B | IPv4 TCP | DMZ net | * | * | webs | * | none | | webs | |
| <input type="checkbox"/> | ✓ 0 / 0 B | IPv4 TCP/UDP | DMZ net | * | * | 53 (DNS) | * | none | | webs dns | |

También crearemos una regla más para permitir acceder a los usuarios que accedan a nuestra VPN; desde “Firewall”, “Rules”, “WAN” sólo deberemos establecer los siguientes parámetros:

Destination

Destination

☐ Invert match

This firewall (self)

Destination Address

/

Destination Port Range

(other)

From

1194

Custom

(other)

To

1194

Custom

Specify the destination port or port range for this rule. The “To” field may be left empty if only filtering a single port.

El puerto 1194 es el que asignamos para la VPN.

Exportar cliente:

Con esta función, los administradores pueden generar archivos de configuración de cliente para OpenVPN de forma rápida y sencilla. Estos archivos contendrán toda la información necesaria para que los clientes puedan conectarse a la red remota utilizando OpenVPN.

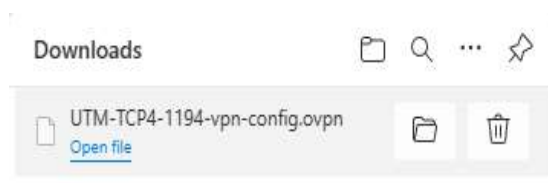
Lo haremos desde la pestaña “VPN”, “OpenVPN”, “Client Export”.

The screenshot shows the 'OpenVPN Server' configuration interface. Under the 'Client Connection Behavior' section, there are three dropdown menus: 'Remote Access Server' set to 'VPN to LAN TCP4:1194', 'Host Name Resolution' set to 'Interface IP Address', and 'Verify Server CN' set to 'Automatic - Use verify-x509-name where possible'. Below these, a note states: 'Optionally verify the server certificate Common Name (CN) when the client connects.'

Realmente, el sistema ya nos deja una configuración por defecto válida, con lo cual no deberemos modificar o especificar ningún campo (prestar atención que los datos de “Remote Access Server” correspondan a los que utilizamos en la creación de la VPN).

Y en la parte inferior de esta misma ventana en la que nos encontramos, clickamos sobre el botón “Most Clients” y nos descargaremos un documento con extensión .ovpn que es el que contendrá toda la configuración que hemos ido estableciendo en todos los pasos anteriores.

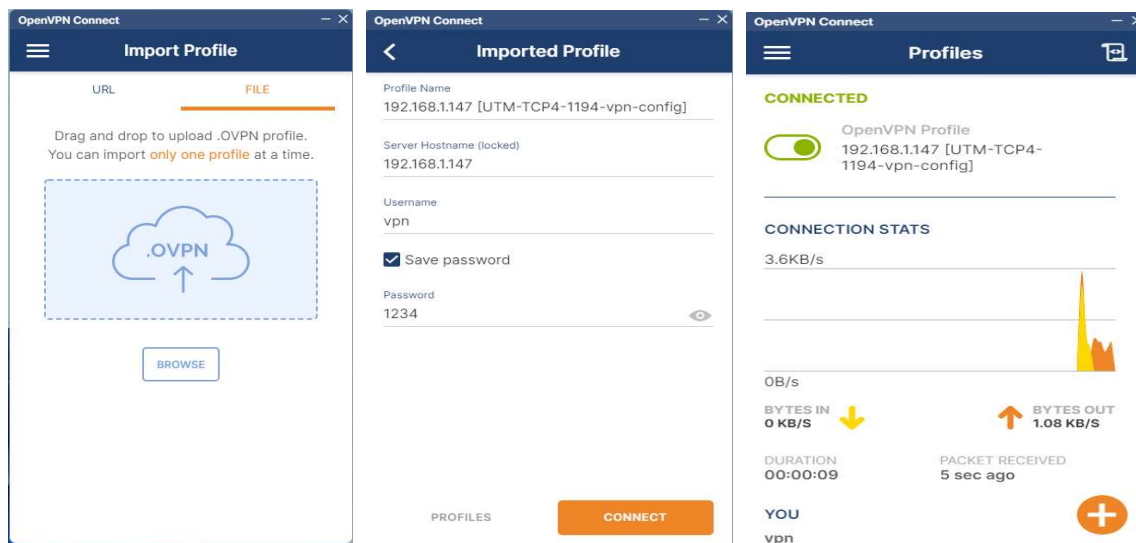
The screenshot shows the 'OpenVPN Clients' interface. It features a table with columns 'User', 'Certificate Name', and 'Export'. The 'User' column contains the value 'vpn'. The 'Export' column lists various download options: 'Inline Configurations' (Most Clients, Android, OpenVPN Connect), 'Bundled Configurations' (Archive, Config File Only), 'Current Windows Installers (2.5.8-1x04)' (64-bit, 32-bit), 'Legacy Windows Installers (2.4.12-1x01)' (10/2016/2019, 7/8/8.1/2012r2), and 'Viscosity (Mac OS X and Windows)' (Viscosity Bundle, Viscosity Inline Config).



Para poder usar el archivo con extensión .ovpn, necesitaremos un cliente VPN. En este caso, se utilizará un software gratuito llamado [VPN Connect](#).

Lo instalaremos en nuestra red LAN, por tanto, en la máquina virtual de Windows 11 (el proceso de instalación es muy sencillo y no requiere de ningún tipo de configuración).

Se nos abrirá una ventana tal que así:



Arrastraremos el archivo .ovpn a la casilla central, se nos permitirá introducir las credencial del usuario creado anteriormente y, si todo está bien configurado, se nos conectará correctamente tal y como aparece en la 3a captura de pantalla.

Y vamos a crear otra regla que permita el tráfico de cualquier protocolo a través de la VPN. Lo haremos desde la opción “Firewall”, “Rules”, “Open VPN” y le daremos al botón de **añadir**. Dejaremos todos los parámetros por defecto y estableceremos una descripción clara tal como: *pass all 09-04-23* (se recomienda dejar constancia de la fecha en las creaciones de reglas).

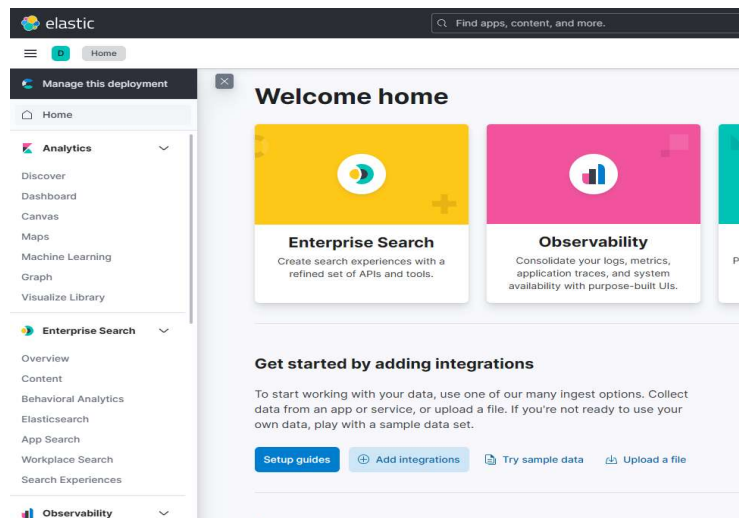
| Rules (Drag to Change Order) | | | | | | | | | | | |
|------------------------------|--------|----------|----------|------|-------------|------|---------|-------|----------|-------------------|---------|
| | States | Protocol | Source | Port | Destination | Port | Gateway | Queue | Schedule | Description | Actions |
| <input type="checkbox"/> | ✓ | 0/0 B | IPv4 TCP | * | * | * | * | * | none | pass all 09-04-23 | |

Instalación de Elasticsearch y Suricata:

Elasticsearch es un motor de búsqueda y análisis de datos distribuido y de código abierto. Se basa en la biblioteca de búsqueda de texto completo Apache Lucene. Es utilizado por muchas empresas y organizaciones para buscar, analizar y visualizar grandes cantidades de datos en tiempo real.

Este servicio lo ejecutaremos desde la nube, así que no será necesario alojarlo en ninguna red de nuestra infraestructura y podremos usarlo desde cualquier dispositivo con conexión a Internet. Utilizaremos una especie de cliente llamado **Elastic Agent**.

Desde su página web oficial (ver link), tan sólo deberemos crearnos una cuenta y seguir los pasos pertinentes. Cuando finalicemos este proceso, tendremos un menú web donde escoger qué hacer, posibles integraciones, registros recolectados, etc.



Integrations

Choose an integration to start collecting and analyzing your data.

Browse integrations **Installed integrations**

All installed

5

Search for integrations

Updates available

0

Only installed Elastic Agent Integrations are displayed.

To learn more about integrations and the Elastic Agent, read our [announcement blog post](#)

Elastic APM

Monitor, detect, and diagnose complex application performance issues.

Elastic Agent

Collect logs and metrics from Elastic Agents.

Fleet Server

Centrally manage Elastic Agents with the Fleet Server integration.

Suricata

Collect logs from Suricata with Elastic Agent.

Synthetics

Technical preview

Suricata es un motor de detección de amenazas de red y sistema de prevención de intrusiones (IPS) de código abierto. Fue desarrollado por la organización sin fines de lucro *Open Information Security Foundation (OISF)* y se basa en la biblioteca *libhttp* para el análisis de protocolos de red. Además, es capaz de inspeccionar el tráfico de red cifrado utilizando SSL/TLS, lo que permite la detección de amenazas en conexiones seguras. También es capaz de integrarse con otros sistemas de seguridad, como *firewalls* y soluciones de seguridad de *endpoints*, para proporcionar una capa adicional de seguridad en toda la infraestructura de red.

Suricata lo instalaremos en una máquina virtual con Kali Linux como Sistema Operativo. Esta red será la DMZ_H, donde integraremos también los Honeypots (ver más adelante).

Para instalarlo, lo haremos en la consola de Kali Linux mediante el comando:

apt install suricata

Ahora deberíamos proceder a instalar el “Agente” de Suricata que lo integre a Elastic. El proceso de instalación o integración se realiza de forma simultánea entre Elastic Cloud (desde el navegador del dispositivo que queramos) y desde la consola de Kali Linux, que es nuestra red DMZ_2. Eso sí, el proceso lo iniciamos desde la misma página web de Elastic, buscando por Suricata y dándole al botón correspondiente:



En el siguiente paso nos darán un *script* para que ejecutemos en la consola de nuestro sistema operativo. Al quererlo instalar en un Kali, seleccionaremos la opción de Linux Tar, aunque también podríamos hacerlo en un MacOS, Windows, etc.

✓

Install Elastic Agent on your host

Select the appropriate platform and run commands to install, enroll, and start Elastic Agent. Reuse commands to set up agents on more than one host. For aarch64, see our [downloads page](#). For additional guidance, see our [installation docs](#).

Linux Tar

Mac

Windows

RPM

DEB

Kubernetes

```
$ProgressPreference = 'SilentlyContinue'
Invoke-WebRequest -Uri https://artifacts.elastic.co/downloads/beats/elastic-agent/elastic-ag
Expand-Archive .\elastic-agent-8.7.0-windows-x86_64.zip -DestinationPath .
cd elastic-agent-8.7.0-windows-x86_64
.\elastic-agent.exe install --url=https://810dcc28cf00404186ead48d3327ef1c.fleet.europe-west
```

Copied

2

Confirm agent enrollment

Listening for agent...

Se quedará en suspensión, esperando la instalación pertinente.

Ejecutaremos el comando copiado en el portapapeles en la consola de Kali:

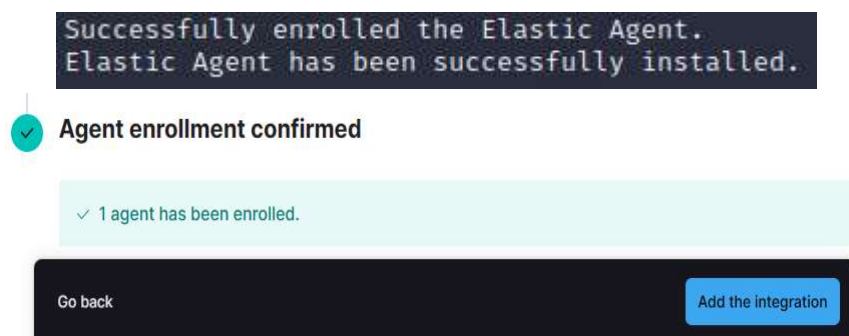
```
(kali@kali)-[~/suricata-6.0.10]
└─$ curl -L -O https://artifacts.elastic.co/downloads/beats/elastic-agent/elastic-agent-8.7.0-linux-x86_64.tar.gz
└─$ tar xzvf elastic-agent-8.7.0-linux-x86_64.tar.gz
└─$ cd elastic-agent-8.7.0-linux-x86_64
└─$ sudo ./elastic-agent install --url=https://810dcc28cf00404186ead48d3327ef1c.fleet.europe-west1.gcp.cloud.es.io:443 --enrollment-token=TF9UbmFJY0ItLUtIVlR2dmpyNGU6czVERGppZ2JUSC1iaDlwCUpMdENLdw==
```

El proceso de instalación se mostrará del siguiente modo:

```
kali@kali: ~/suricata-6.0.10
File Actions Edit View Help

.disabled
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/monitors.d/
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/monitors.d/sample.htt
p.yml.disabled
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/monitors.d/sample.icm
p.yml.disabled
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/monitors.d/sample.tcp
.yml.disabled
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/osquery-extension.ext
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/osquerybeat
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/osquerybeat.reference
.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/osquerybeat.spec.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/osquerybeat.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/osqueryd
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/packetbeat
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/packetbeat.reference
.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/packetbeat.spec.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/packetbeat.yml
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/pf-host-agent
elastic-agent-8.7.0-linux-x86_64/data/elastic-agent-fc4a15/components/pf-host-agent.spec.yml
elastic-agent-8.7.0-linux-x86_64/elastic-agent
[sudo] password for kali:
Elastic Agent will be installed at /opt/Elastic/Agent and will run as a service. Do you want
to continue? [Y/n]:Y
```

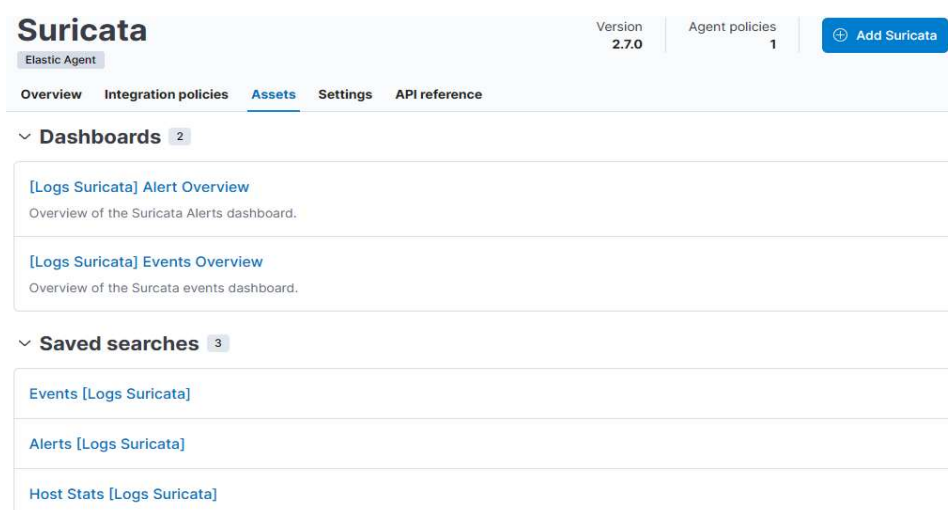
Y cuando finalice, obtendremos por los dos lados el mismo mensaje de confirmación: “*enrolled*”.



Por último, desde el navegador web donde está el cliente de Elastic, confirmaremos las siguientes opciones de configuración por defecto:



Y ya habríamos terminado el proceso de instalación. Podemos ver, incluso, que ya ha extraído algo de información desde el primer momento.



Honeypots:

Un **honeypot** es un sistema de seguridad informática que se utiliza para detectar, recopilar y analizar los intentos de intrusión en una red o sistema informático. Se configuran para simular un sistema vulnerable o un servicio que podría ser un objetivo para los atacantes, con el fin de atraerlos y registrar sus acciones (recopilación de información: técnicas de ataque, herramientas utilizadas, direcciones de IP de origen, etc.).

A continuación usaremos un **honeypot** llamado **Cowrie** que simula un **servidor SSH** (*Cowrie simula un servidor SSH en un entorno seguro, capturando información sobre los intentos de conexión y los comandos enviados por los atacantes. Además, también puede registrar cualquier intento de explotar vulnerabilidades conocidas en el sistema emulado*).

Lo instalaremos en Kali Linux (a través de la consola de comandos) y mediante Docker. Los pasos que seguiremos para tal fin se listan a continuación:

- **sudo -s** (si queremos estar con permisos de administrador)
- **apt update**
- **apt install docker.io** (para instalar Docker, en caso de que no lo tengas ya)
- **systemctl enable docker --now** (para habilitar el servicio Docker de manera continua)
- **docker ps** (nos situamos en el interior de la carpeta de Docker)
- **docker run -p 2222:2222 cowrie/cowrie** (levantamos un contenedor de SSH)

Tras esto, Cowrie se quedará a la escucha.

Desde otra ventana diferente de la consola, podemos simular un intento de intrusión mediante el comando: `ssh root@127.0.0.1 -p2222` (especificamos el mismo puerto en el que hemos lanzado Cowrie)

Lo que hagamos en la 2a ventana, lo irá recogiendo el **honeypot** y podremos ver qué intentos de

conexión ha efectuado el presunto atacante, la contraseña que ha escrito para *loguearse*, los comandos que ejecuta, etc.

```
kali@kali: ~  
File Actions Edit View Help  
└─$ ssh root@127.0.0.1 -p2222  
The authenticity of host '[127.0.0.1]:2222 ([127.0.0.1]:2222)' can't be established.  
ED25519 key fingerprint is SHA256:19PT00zENzbzVz0yYR9te7IQrbkRe7wWprMKu5GXP+E.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '[127.0.0.1]:2222' (ED25519) to the list of known hosts.  
root@127.0.0.1's password:  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
root@svr04:~# ls  
root@svr04:~# echo hola hola  
hola hola  
root@svr04:~# echo hola >hola  
root@svr04:~# ls  
hola  
root@svr04:~# cat hola  
hola
```









Podemos ver que lo que “pintamos” en la ventana de arriba, va quedando registrado y mostrado en la de abajo...

```
root@kali: /  
File Actions Edit View Help  
g_auth b'password'  
2023-04-10T14:59:43+0000 [HoneyPotSSHTransport,0,172.17.0.1] Could not read etc/userdb.txt,  
default database activated  
2023-04-10T14:59:43+0000 [HoneyPotSSHTransport,0,172.17.0.1] login attempt [b'root'/b'kali']  
succeeded  
2023-04-10T14:59:43+0000 [HoneyPotSSHTransport,0,172.17.0.1] Initialized emulated server as  
architecture: linux-x64-lsb  
2023-04-10T14:59:43+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' authen  
ticated with b'password'  
2023-04-10T14:59:43+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service  
b'ssh-connection'  
2023-04-10T14:59:43+0000 [cowrie.ssh.connection.CowrieSSHConnection#debug] got channel b'ses  
sion' request  
2023-04-10T14:59:43+0000 [cowrie.ssh.session.HoneyPotSSHSession#info] channel open  
2023-04-10T14:59:43+0000 [cowrie.ssh.connection.CowrieSSHConnection#debug] got global b'no-m  
ore-sessions@openssh.com' request  
2023-04-10T14:59:43+0000 [twisted.conch.ssh.session#info] Handling pty request: b'xterm-256c  
olor' (28, 92, 0, 0)  
2023-04-10T14:59:43+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPot  
SSHTransport,0,172.17.0.1] Terminal Size: 92 28  
2023-04-10T14:59:43+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPot  
SSHTransport,0,172.17.0.1] request_env: LANG=en_US.UTF-8  
2023-04-10T14:59:43+0000 [twisted.conch.ssh.session#info] Getting shell  
2023-04-10T15:00:29+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: ls  
2023-04-10T15:00:29+0000 [HoneyPotSSHTransport,0,172.17.0.1] Command found: ls  
2023-04-10T15:00:49+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: echo hola hola  
2023-04-10T15:00:49+0000 [HoneyPotSSHTransport,0,172.17.0.1] Command found: echo hola hola  
2023-04-10T15:02:08+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: echo hola >hola  
2023-04-10T15:02:08+0000 [HoneyPotSSHTransport,0,172.17.0.1] Command found: echo hola > hola  
2023-04-10T15:02:10+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: ls  
2023-04-10T15:02:10+0000 [HoneyPotSSHTransport,0,172.17.0.1] Command found: ls  
2023-04-10T15:02:12+0000 [HoneyPotSSHTransport,0,172.17.0.1] CMD: cat hola  
2023-04-10T15:02:12+0000 [HoneyPotSSHTransport,0,172.17.0.1] Command found: cat hola
```

Creación de nuevas reglas:

Como vamos a utilizar el *honeypot* Cowrie, que emula un servidor SSH (normalmente alojado en el puerto 22), vamos a crear una regla por la cual cada intento de conexión a nuestra red por el puerto 22, se redirija al 2222 de nuestra DMZ_H, que es donde lo tendremos en ejecución.


Lo haremos desde la pestaña “*Firewall*”, “*NAT*”, “*Port Forward*”.

| Rules | | | | | | | | | | | |
|--------------------------|---|-----------|----------|----------------|--------------|---------------|-------------|----------------|-----------|--------------|---|
| <input type="checkbox"/> | | Interface | Protocol | Source Address | Source Ports | Dest. Address | Dest. Ports | NAT IP | NAT Ports | Description | Actions |
| <input type="checkbox"/> | <input checked="" type="checkbox"/>  | WAN | TCP | * | * | WAN address | 9090 | 192.168.90.225 | 80 (HTTP) | servidor web |    |
| <input type="checkbox"/> | <input checked="" type="checkbox"/>  | WAN | TCP | * | * | WAN address | 22 (SSH) | DMZ_H address | 2222 | ssh cowrie |    |

En DMZ_H implantaremos 2 reglas también:

Una que permita a los posibles atacantes, desde cualquier origen y con destino a DMZ_H, el tráfico o conexión al puerto 2222.

Y otra que permita que los registros que vayan recopilando el honeypot y Suricata puedan moverse de la DMZ_H a la WAN

| Rules (Drag to Change Order) | | | | | | | | | | | |
|------------------------------|-------------------------------------|----------|----------|-----------|-------------|-----------|---------|-------|----------|-------------|---|
| <input type="checkbox"/> | States | Protocol | Source | Port | Destination | Port | Gateway | Queue | Schedule | Description | Actions |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | 0 / 0 B | IPv4 TCP | DMZ_H net | * | WAN net | * | * | none | |    |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | 0 / 0 B | IPv4 TCP | * | * | DMZ_H net | 2222 | * | none | |    |

Por último, deberemos añadir una regla de forma manual en Suricata; lo realizaremos desde la consola de Kali Linux.

Nos situaremos en la carpeta de Suricata y deberemos encontrar dentro de ella el fichero “*rules*”. Desde él, crearemos un archivo llamado ***suricata.rules*** con la regla que queramos establecer:

```
(root@kali)-[/etc/suricata/rules]
# ls
app-layer-events.rules  files.rules             modbus-events.rules     smtp-events.rules
decoder-events.rules    http2-events.rules      mqtt-events.rules       ssh-events.rules
dhcp-events.rules       http-events.rules       nfs-events.rules        stream-events.rules
dnp3-events.rules       ipsec-events.rules      ntp-events.rules        tls-events.rules
dns-events.rules        kerberos-events.rules   smb-events.rules
```

```
root@kali: /etc/suricata/rules
File Actions Edit View Help
GNU nano 6.3 suricata.rules *
alert tcp any any -> any 2222 (msg:"ssh"; sid:100; priority:1;)
```

Esta regla permite el registro de las conexiones entrantes a través del puerto SSH, que, como habíamos establecido anteriormente, serán redireccionados por el puerto 2222.

Pruebas:

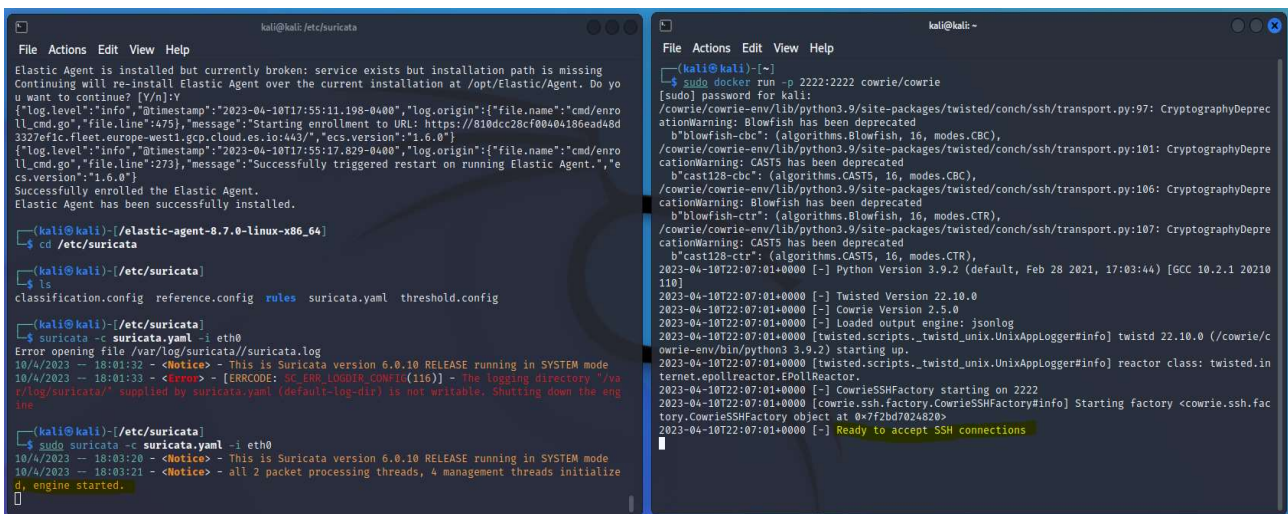
Podemos empezar a realizar alguna prueba, especialmente para ver si el Elastic registra algún evento o alerta en concreto.

A modo de resumen:

- Tenemos el PfSense como red WAN; está conectado directamente a Internet.
- Tenemos el Windows 11 con una VPN habilitada (aunque para hacer la prueba no es necesario) como red LAN; aquí tenemos conexión a Internet.
- La red DMZ no tiene uso real en nuestra infraestructura porque se ha optado por usar el servicio de Elastic a través de la nube y no hemos implantado ninguna máquina virtual finalmente;
- La red DMZ_H ejecutará, a través de un Kali Linux, Suricata y el Honeypot; desde esta red no tenemos conexión a Internet ni a la red LAN, aunque sí a la WAN para poder guardar los logs o registros que se van recogiendo.

Vamos a ejecutar Suricata y Cowrie al mismo tiempo:

- **suricata** : `sudo suricata -c suricata.yaml -i eth0`
- **cowrie** : `sudo docker run -p 2222:2222 cowrie/cowrie`



The image shows two terminal windows side-by-side. The left window is titled 'kali@kali:~/etc/suricata' and shows the process of installing the Elastic Agent and then Suricata. It includes commands like 'cd /etc/suricata', 'ls', and 'suricata -c suricata.yaml -i eth0'. The right window is titled 'kali@kali:~' and shows the execution of 'sudo docker run -p 2222:2222 cowrie/cowrie'. It displays the Docker container's output, including the Cowrie version (2.5.0) and the Twisted version (22.10.0), and confirms that the SSH factory is ready to accept connections.

Y desde cualquier máquina que no esté dentro de nuestra infraestructura de red, lanzaremos un intento de conexión a nuestra red WAN por el puerto 22, al igual que hemos realizado anteriormente.

```
C:\Users\Gerard>ssh root@192.168.1.147
The authenticity of host '192.168.1.147 (192.168.1.147)' can't be established.
ECDSA key fingerprint is SHA256:/aDhh9FhgT/EK3glwEKEXQwINQJlIf/J14qZTG04ook.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.147' (ECDSA) to the list of known hosts.
root@192.168.1.147's password:
```

La IP 192.168.1.147 es la del PfSense.

Observaremos como, nuevamente, en nuestro *honeypot* se siguen registrando los intentos de conexión y los comandos que se ejecutan.

Y, teóricamente, estos eventos estarán generando alertas, que podremos ver desde el Dashboard de nuestro Elastic.

En este punto me doy cuenta que Elastic no está registrando ningun alerta...

Procedo a revisar todo:

- Compruebo qué redes tienen conexión a Internet y cuales no.
- Elimino el agente de Elastic de Kali y vuelvo a realizar la integración de Suricata con Elastic.
- Reviso todas las reglas definidas en PfSense y realizo cambios (hago un any any en la LAN y añado bloqueo de la DMZ_H a la LAN, cambio los puertos 2222 al rango 2221-2223, añado alguna IP específica en vez de usar los campos prefdefinidos como “DMZ net”, etc.). Al final, termino con esta configuración de reglas exactamente:

NAT:

| Rules | | | | | | | | | | | | |
|--------------------------|-------------------------------------|--|-----------|----------|----------------|--------------|---------------|-------------|----------------|-----------|--------------|---------|
| <input type="checkbox"/> | | | Interface | Protocol | Source Address | Source Ports | Dest. Address | Dest. Ports | NAT IP | NAT Ports | Description | Actions |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | | WAN | TCP | * | * | WAN address | 9090 | 192.168.90.225 | 80 (HTTP) | servidor web | |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | | WAN | TCP | * | * | WAN address | 22 (SSH) | 192.168.50.100 | 2222 | ssh cowrie | |

WAN:

| Rules (Drag to Change Order) | | | | | | | | | | | |
|------------------------------|------------|----------|---------|------|----------------|----------------|---------|-------|----------|------------------------|---------|
| <input type="checkbox"/> | States | Protocol | Source | Port | Destination | Port | Gateway | Queue | Schedule | Description | Actions |
| <input type="checkbox"/> | 0 / 0 B | IPv4 TCP | * | * | 192.168.90.225 | 80 (HTTP) | * | none | | NAT servidor web | |
| <input type="checkbox"/> | 0 / 0 B | IPv4 TCP | * | * | This Firewall | 1194 (OpenVPN) | * | none | | VPN to LAN | |
| <input type="checkbox"/> | 0 / 14 KiB | IPv4 TCP | * | * | 192.168.50.100 | 2221 - 2223 | * | none | | NAT ssh cowrie correct | |
| <input type="checkbox"/> | 0 / 0 B | IPv4 TCP | WAN net | * | DMZ_H net | 2221 - 2223 | * | none | | | |













LAN:

| Rules (Drag to Change Order) | | | | | | | | | | | |
|-------------------------------------|--------------|----------|---------|------|-------------|------|---------|-------|----------|------------------------------------|---------|
| <input type="checkbox"/> | States | Protocol | Source | Port | Destination | Port | Gateway | Queue | Schedule | Description | Actions |
| <input checked="" type="checkbox"/> | 0 / 5.57 MiB | * | * | * | LAN Address | 80 | * | * | | Anti-Lockout Rule | |
| <input type="checkbox"/> | 9 / 1.09 GiB | IPv4 * | LAN net | * | * | * | * | none | | Default allow LAN to any rule | |
| <input type="checkbox"/> | 0 / 0 B | IPv6 * | LAN net | * | * | * | * | none | | Default allow LAN IPv6 to any rule | |
| <input type="checkbox"/> | 0 / 0 B | IPv4 TCP | * | * | * | * | * | none | | | |

DMZ (aunque no interviene):

| Rules (Drag to Change Order) | | | | | | | | | | | |
|------------------------------|--------|----------|--------------|---------|-------------|------|----------|-------|----------|-------------|---|
| <input type="checkbox"/> | States | Protocol | Source | Port | Destination | Port | Gateway | Queue | Schedule | Description | Actions |
| <input type="checkbox"/> | ✓ | 0/0 B | IPv4 TCP | DMZ net | * | * | webs | * | none | webs |     |
| <input type="checkbox"/> | ✓ | 0/0 B | IPv4 TCP/UDP | DMZ net | * | * | 53 (DNS) | * | none | webs dns |     |

DMZ_H:

| Rules (Drag to Change Order) | | | | | | | | | | | |
|------------------------------|--------|----------|----------|-----------|-------------|-----------|-------------|-------|----------|-------------|---|
| <input type="checkbox"/> | States | Protocol | Source | Port | Destination | Port | Gateway | Queue | Schedule | Description | Actions |
| <input type="checkbox"/> | ✓ | 0/0 B | IPv4 TCP | DMZ_H net | * | WAN net | * | none | | |     |
| <input type="checkbox"/> | ✓ | 0/0 B | IPv4 TCP | * | * | DMZ_H net | 2221 - 2223 | none | | |     |
| <input type="checkbox"/> | ✗ | 0/0 B | IPv4 TCP | DMZ net | * | LAN net | * | none | | |     |

Y reinicio las máquinas virtuales.

Lo primero que compruebo es la pestaña de *Fleet* de Elastic:

Fleet

Centralized management for Elastic Agents.

[Agents](#) [Agent policies](#) [Enrollment tokens](#) [Data streams](#) [Settings](#)

🕒 Agent activity

Add Fleet Server

Add agent

Status 4

Tags 0

Agent policy 5

Upgrade available

Showing 2 agents

● Healthy 2

● Unhealthy 0

● Updating 0

● Offline 0

| <input type="checkbox"/> | Status | Host | Agent policy | CPU ① | Memory ① | Last activity | Version | Actions |
|--------------------------|---------|--------------|--------------------------------------|-------|----------|---------------|---------|---------|
| <input type="checkbox"/> | Healthy | kali | Agent policy 3 rev. 2 | N/A ① | N/A ① | 5 minutes ago | 8.7.0 | ... |
| <input type="checkbox"/> | Healthy | 3b2531e4d13c | Elastic Cloud agent policy rev. 5 | N/A ① | N/A ① | 8 seconds ago | 8.7.0 | ... |

Rows per page: 20

< 1 >

Anteriormente me salía Kali en OFF, como si no estuviera activo; desconozco aún la razón de ello, pero tras aplicar todos los cambios y hacer los reinicios de las máquinas, finalmente sale con la etiqueta “*Healthy*”.

Vuelvo entonces a realizar la prueba - pruebo rápidamente desde la conexión al *honeypot* un *ls* y un *ifconfig* y obtengo finalmente en “*Dashboard*” los registros mediante los cuales aparecen alertas por el login y la ejecución de los 2 comandos indicados:

| Alerts (Logs Suricata) | | | | | | | | | |
|--------------------------|-----------------------------|-----------|-------------------|---------------|-------------|----------------|------------------|------------------------------|-------------------------------|
| Columns 1 field sorted | | | | | | | | | |
| | @timestamp | host.name | suricata.event_id | source.ip | source.port | destination.ip | destination.port | source.geo.country_iso_co... | destination.geo.country_is... |
| <input type="checkbox"/> | Apr 11, 2023 @ 00:33:44.578 | - | 1248822742388189 | 192.168.1.139 | 7,673 | 192.168.50.100 | 2,222 | - | - |
| <input type="checkbox"/> | Apr 11, 2023 @ 00:21:36.244 | - | 38661348448730 | 192.168.1.139 | 7,586 | 192.168.50.100 | 2,222 | - | - |

Se comprueba, así, que el *honeypot* transmite la información a Suricata y ésta, a su vez, a Elastic, comenzando a registrar los comportamientos anómalos (aunque en esta práctica, en concreto, y por haberse usado sólo un *honeypot* específico, únicamente se registren dichos comportamientos anómalos relacionado con el servicio SSH).

La infraestructura de red funciona correctamente.