Definición de túneles cifrados con OpenVPN

CDA 2022/23

6 de octubre de 2022

Índice

1. Descripción

Ejemplo de uso del software de VPN (*Virtual Private Network*) **openVPN**.

- Definición de un túnel OpenVPN en modo *road-warrior*

Recursos complementarios

- OpenVPN: https://openvpn.net/

2. Entorno de prácticas

2.1. Software de virtualización VIRTUALBOX

En estas prácticas se empleará el software de virtualización VIRTUALBOX para simular los equipos GNU/Linux sobre los que se realizarán las pruebas.

- Página principal: `http://virtualbox.org`
- Más información: `http://es.wikipedia.org/wiki/Virtualbox`

2.2. Imágenes a utilizar

1. Scripts de instalación
   - para GNU/Linux: ejercicio-dmz-openvpn.sh
     `alumno@pc: $ sh ejercicio-dmz-openvpn.sh`
   - para MS windows: ejercicio-dmz-openvpn.ps1
     `Powershell.exe -executionpolicy bypass -file ejercicio-dmz-openvpn.ps1`

   Notas:
   - Se pedirá un identificador (sin espacios) para poder reutilizar las versiones personalizadas de las imágenes creadas (usad por ejemplo el nombre del grupo de prácticas o el login LDAP)
   - En ambos scripts la variable `$DIR_BASE` especifica donde se descargarán las imágenes y se crearán las MVs.
     Por defecto en GNU/Linux será en `$HOME/CDA2223` y en Windows en `C:/CDA2223`.
     Puede modificarse antes de lanzar los scripts para hacer la instalación en otro directorio más conveniente (disco externo, etc)
   - Es posible descargar las imágenes comprimidas manualmente (o intercambiarlas con USB), basta descargar los archivos con extensión `.vdi.zip` de `http://ccia.esei.uvigo.es/docencia/CDA/2223/practicas/` y copiarlos en el directorio anterior (`$DIR_BASE`) para que el script haga el resto.
   - Si no lo hacen desde el script anterior, se pueden arrancar las instancias VIRTUALBOX desde el interfaz gráfico de VirtualBOX o desde la línea de comandos con `VBoxManage startvm <nombre MV>_<id>`
2. Imágenes descargadas
   - **base_cda.vdi** (1,3 GB comprimida, 4,5 GB descomprimida): Imagen genérica (común a todas las MVs) que contiene las herramientas a utilizar
     Contiene un sistema Debian 11 con herramientas gráficas y un entorno gráfico ligero LXDE (*Lighweight X11 Desktop Environment*) [LXDE].
   - **swap1GB.vdi**: Disco de 1 GB formateado como espacio de intercambio (SWAP)
3. Usuarios configurados e inicio en el sistema
   - Usuarios disponibles

login	password
root	purple
usuario	usuario
(con permisos para `sudo`)	

   - Acceso al entorno gráfico una vez logueado (necesario para poder copiar y pegar desde/hacia el anfitrión)

     `root@datos:~# startx`

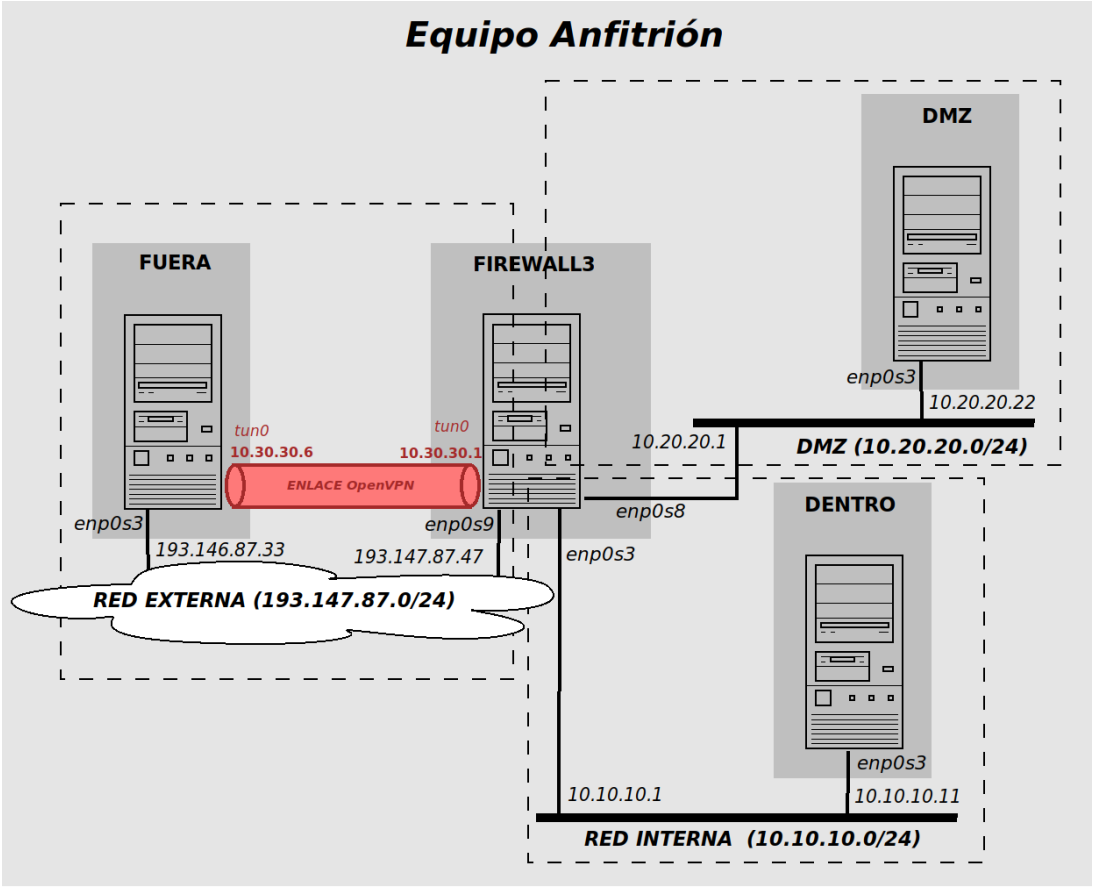
   - Habilitar copiar y pegar desde/hacia el anfitrión en el menú `Dispositivos -> Portapapeles compartido -> bidir`
     de la ventana de la máquina virtual.

2.3. Máquinas virtuales y redes creadas

Una vez ejecutado el script se habrán definido las 3 redes y los 4 equipos virtualizados donde se realizarán los ejercicios:

- Red interna (10.10.10.0 ... 10.10.10.255): máquina **dentro** (enp0s3) + interfaz enp0s3 de **firewall3**
- Red DMZ (10.20.20.0 ... 10.20.20.255): máquina **dmz** (enp0s3) + interfaz enp0s8 de **firewall3**

- Red externa (193.147.87.0 ... 193.147.87.255): máquina **fuera** (enp0s3) + interfaz enp0s9 de **firewall3**



3. Ejercicio: Uso de enlaces cifrados OpenVPN

Se desarrollará un ejercicio de creación de enlaces OpenVPN, donde se creará un enlace cifrado OpenVPN desde un equipo de la red externa y se revisará su integración en el firewall con DMZ configurado con Shorewall.

3.1. Pasos previos (preparación del entorno)

1. PREVIO 1 (ya hecho en las MVs de prácticas). Habilitar el acceso como usuario `root` en el servidor SSH de la máquina **firewall3** [**10.10.10.1, 10.20.20.1, 193.147.87.47**] y reiniciar el servicio

   ```
   firewall3:~# nano /etc/ssh/sshd_config
   ...
   PermitRootLogin yes
   ...

   firewall3:~# systemctl restart sshd
   ```

2. PREVIO 2. Establecer tráfico a través de la máquina **firewall3** [**10.10.10.1, 10.20.20.1, 193.147.87.47**]

   a) **Opción 1:** si se ha retomado la práctica 3 "Definición de zonas desmilitarizadas con Shorewall"

   1) Deshabilitar el filtrado de Shorewall
      `firewall3:~#  shorewall clear`
   2) Habilitar la redirección de tráfico
      `firewall3:~#  echo 1 > /proc/sys/net/ipv4/ip_forward`

b) **Opción 2:** si se ha iniciado la práctica desde cero

1) Establecer la configuración por defecto de NETFILTER/iptables (politica ACCEPT)

```
firewall3:~#  iptables -F
firewall3:~#  iptables -t nat -F

firewall3:~#  iptables -P INPUT ACCEPT
firewall3:~#  iptables -P OUTPUT ACCEPT
firewall3:~#  iptables -P FORWARD ACCEPT
```

2) Habilitar la redirección de tráfico

```
firewall3:~#  echo 1 > /proc/sys/net/ipv4/ip_forward
```

Nota: Para hacer permanente el cambio en la variable del kernel `ip_forward` se puede descomentar la línea `#net.ipv4.ip_forward=1` en el fichero `/etc/sysctl.conf`.

3. **Tarea 1 [escaneo inicial]**: (a incluir en la memoria entregable) Escaneo desde la máquina **fuera** para verificar los servicios accesibles inicialmente

   - desde `fuera`:

```
fuera:~# nmap -T4 10.10.10.11
fuera:~# nmap -T4 10.20.20.22
fuera:~# nmap -T4 193.147.87.47
```

3.2. Parte 1: Creación de un enlace OpenVPN

Se creará un enlace cifrado OpenVPN desde la máquina externa **fuera (193.147.87.33)** a la máquina **firewall3 (193.147.87.47)**.

- OpenVPN establece una conexión TLS/SSL entre el cliente y el servidor
- En ambos extremos del túnel se crea una interfaz de red virtual (dispositivos `tun` o `tap`), que desde el punto de vista del sistema operativo funciona como una tarjeta de red convencional, pudiendo asignársele direcciones IP y participando en las reglas de enrutado y filtrado definidas en cada extremo del tunel.
- En ambos extremos el tráfico dirigido a estas interfaces de red (paquetes IP en dispositivos `tun`, tramas Ethernet en dispositivos `tap`) se encapsula dentro de paquetes TLS/SSL que atraviesan la red pública cifrados y autenticados saliendo por el disposivo (`tun` o `tap`) del otro extremo, donde el tráfico se extrae y continúa su camino

En este ejemplo se usará un esquema TLS/SSL completo, que negocia un conjunto de claves secretas de cifrado y autenticación cada vez que se establece un tunel. Una alternativa más sencilla hubiera sido usar claves secretas preacordadas manualmente.

Usaremos el modo de funcionamiento de OpenVPN *"road warrior"*, donde un servidor OpenVPN crea enlaces cifrados para equipos autorizados situados en redes externas.

- La autenticación se realizará mediante **certificados digitales**, tanto en el servidor como en los equipos cliente.
- A las máquinas que se conecten por VPN se les asignarán direcciones IP del rango **10.30.30.0/24**, donde la máquina **firewall3** (el servidor OpenVPN) tendrá la IP **10.30.30.1**

Certificados y claves necesarias:

- Para el servidor:
  - certificado digital de clave pública de la Autoridad Certificadora (CA) reconocida por ambos participantes: `ca.crt` [certificado raíz autofirmado]
  - clave privada del servidor: `firewall3.cda.net.key`
  - certificado digital de clave pública del servidor: `firewall3.cda.net.crt` (emitido por la CA)

- parámetros para intercambio de clave Diffie-Hellam: `dh.pem`

- Para cada uno de los clientes que se conecten con OpenVPN:
  - certificado digital de calve pública de la Autoridad Certificadora reconocida por ambos participantes: `ca.crt` [certificado raíz autofirmado]
  - clave privada del cliente: `fuera.key`
  - certificado digital de clave pública del cliente: `fuera.crt` (emitido por la CA)

3.2.1. Creación de la CA y de los certificados de servidor y clientes

Los certificados digitales necesarios para configurar las conexiones TLS/SSL pueden obtenerse de Autoridades Certificadoras externas, tanto de pago como gratuitas (como Let's Encrypt, `https://letsencrypt.org/es/`), y reconocidos por defecto en las diferentes aplicaciones (navegadores, etc).

En nuestro ejemplo crearemos nuestra propia Autoridad Certificadora (CA) de uso interno (cuyos certificados serán reconocidos únicamente en las aplicaciones y servicios de nuestra organización).

La distribución de OpenVPN incluye un conjunto de scripts para implantar una CA básica. Estos scripts usan internamente el comando `openssl` para las operaciones de creación de pares de claves (pública y privada) y firma de certificados.

Ver detalles en `https://wiki.debian.org/OpenVPN#Init_easy-rsa`.

1. Crear la "autoridad certificadora" (CA) en el firewall

   (Opcional) Editar los parámetros de nuestra CA y los metadatos del los certificados a generar (no es imprescindible hacerlo)

   ```
   firewall3:~# cd /usr/share/easy-rsa
   firewall3:/usr/share/easy-rsa# cp vars.example vars
   firewall3:/usr/share/easy-rsa# nano vars

   ...
   set_var EASYRSA_REQ_COUNTRY    "ES"
   set_var EASYRSA_REQ_PROVINCE   "Ourense"
   set_var EASYRSA_REQ_CITY       "Ourense"
   set_var EASYRSA_REQ_ORG        "ESEI"
   set_var EASYRSA_REQ_EMAIL      "cda@cda.net"
   set_var EASYRSA_REQ_OU         "CDA"
   ...
   ```

   Inicializar la CA (en el directorio de configuración de OpenVPN, puede hacerse en cualquier otro lugar)

   ```
   firewall3:~# cd /etc/openvpn/
   firewall3:/etc/openvpn# /usr/share/easy-rsa/easyrsa init-pki
   ```

   - Inicializa la CA, creando el directorio `/etc/openvpn/pki` con los subdirectorios `issued` (certificados digitales emitidos con las claves públicas generadas) y `private` (claves privadas generadas).

   Generar el par de claves de la CA

   ```
   firewall3:/etc/openvpn# /usr/share/easy-rsa/easyrsa build-ca nopass
   ```

   - Crea la clave privada (en `pki/private/ca.key`) y la clave pública (en `pki/ca.crt`, como certificado raíz autofirmado) de la CA.
   - Cuando pida el valor `Common Name`, indicar `CA prueba`
   - Con la opción `nopass` se omite proteger la clave privada de la CA con una *passphrase*.

2. Crear el certificado del equipo "servidor" OpenVPN

```
firewall3:/etc/openvpn# /usr/share/easy-rsa/easyrsa build-server-full firewall3.cda.net nopass
```

- Debe indicarse como parámetro el nombre DNS completo del servidor OpenVPN (o su dirección IP). Este parámetro es IMPORTANTE porque se usa para establecer el valor `Common Name` que identica al equipo propietario del certificado. El cliente OpenVPN comprueba este valor durante la fase de establecimiento de conexión y debe coincidir con el nombre del equipo al que se conecta para evitar conexiones con servidores falsos.
- Crea la clave privada (en `pki/private/firewall3.cda.net.key`) y la clave pública (en `pki/issued/firewall3.cda.`
como certificado firmado por la CA) del servidor OpenVPN.
- Con la opción `nopass` se omite proteger la clave privada del servidor con una *passphrase*. Dado que OpenVPN se iniciará desde un script de arranque en `/etc/init.d/`, se omite esta protección para simplificar la configuración y que este script de inicio no se bloquee.

3. Crear el certificado del equipo "cliente" OpenVPN

```
firewall3:/etc/openvpn# /usr/share/easy-rsa/easyrsa build-client-full fuera nopass
```

- En este caso, no es necesario indicar como parámetro el nombre DNS completo del cliente OpenVPN (o su dirección IP), dado que el servidor sólo valida que el lado cliente tenga un certificado emitido por la CA reconocida.
- Crea la clave privada (en `pki/private/fuera.key`) y la clave pública (en `pki/issued/fuera.crt`, como certificado firmado por la CA) del cliente OpenVPN.
- Con la opción `nopass` se omite proteger la clave privada del cliente con una *passphrase*. Dado que OpenVPN se iniciará desde un script de arranque en `/etc/init.d/`, se omite esta protección para simplificar la configuración y que este script de inicio no se bloquee.

4. Crear los parámetros del algoritmo de intercambio de claves Diffie-Hellman necesarios en el lado servidor y usados para la negociación de claves secretas durante el establecimiento de la conexión TLS/SSL (ver `https://es.wikipedia.org/wiki/Diffie-Hellman`). (suele tardar bastante)

```
firewall3:/etc/openvpn# /usr/share/easy-rsa/easyrsa gen-dh
```

- Crea el fichero con los parámetros (un número primo "grande" y un "generador") en `pki/dh.pem`.

5. Se puede comprobar el contenido de los certificados creados (opcional)

```
firewall3:/etc/openvpn# /usr/share/easy-rsa/easyrsa show-ca
firewall3:/etc/openvpn# /usr/share/easy-rsa/easyrsa show-cert firewall3.cda.net
firewall3:/etc/openvpn# /usr/share/easy-rsa/easyrsa show-cert fuera
```

Misma información usando directamente el comando `openssl`

```
firewall3:/etc/openvpn# openssl x509 --text --in pki/ca.crt
firewall3:/etc/openvpn# openssl x509 --text --in pki/issued/firewall3.cda.net.crt
firewall3:/etc/openvpn# openssl x509 --text --in pki/issued/fuera.crt
```

3.2.2. Configuración y creación del enlace OpenVPN

1. Configuración del servidor: en la máquina **firewall3**, directorio `/etc/openvpn/server`.

- Crear una clave secreta para la autenticación HMAC (*hash-based message authentication code*) de los paquetes SSL

```
firewall3:~# cd /etc/openvpn
firewall3:/etc/openvpn# cd server/

firewall3:/etc/openvpn/server# openvpn --genkey secret ta.key
```

- Crear el fichero de configuración del servidor:
Se usará como base el ejemplo disponible en `/usr/share/doc/openvpn/examples/sample-config-files/`

```
firewall3:/etc/openvpn/server# cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf .
```

Editar los parámetros concretos para nuestros túneles VPN:

```
firewall3:/etc/openvpn# nano server.conf
ó
firewall3:/etc/openvpn# mousepad server.conf &
```

Parámetros destacados (con "→" se señalan los cambios efectuados para nuestro ejemplo):

```
     port 1194     ## puerto por defecto del servidor OpenVPN
     proto udp     ## protocolo por defecto del servidor OpenVPN
     dev tun       ## tipo de dispositivo de red virtual (= tarjeta de red "software") a través
                   ##    del cual se accederá al túnel cifrado establecido
     ...
->   ca   /etc/openvpn/pki/ca.crt                              ## parametros de cifrado
->   cert /etc/openvpn/pki/issued/firewall3.cda.net.crt
->   key  /etc/openvpn/pki/private/firewall3.cda.net.key
     ...
->   dh   /etc/openvpn/pki/dh.pem
     ...
->   server 10.30.30.0 255.255.255.0          ## rango de direcciones a asignar a los clientes
                                              ##    OpenVPN que se vayan conectando
     ...
->   push "route 10.10.10.0 255.255.255.0"    ## configuración de las rutas a establecer (''empujar'') en los
->   push "route 10.20.20.0 255.255.255.0"    ##    clientes para las conexiones cifradas que se vayan creando
                                              ##    en nuestro caso son las rutas hacia las 2 redes (interna
                                              ##    y dmz) gestionadas por firewall3
     ...
->   tls-auth /etc/openvpn/server/ta.key 0
     ...
```

2. Configuración de los clientes: en la máquina **fuera (193.147.87.33)**, directorio `/etc/openvpn/client`

   - Copiar (mediante copia segura sobre SSH con `scp`) las claves/certificados necesarios al directorio `/etc/openvpn/clien`

     ```
     fuera:# cd  /etc/openvpn
     fuera:/etc/openvpn# cd client
     fuera:/etc/openvpn/client# scp root@firewall3.cda.net:/etc/openvpn/pki/{ca.crt,issued/fuera.crt,private/fuera.key} .
     ```

     Importante: Es necesario haber habilitado el *login* como `root` en la configuración del servidor SSH (ver PREVIO 1)

   - Copiar (mediante copia segura sobre SSH con `scp`) la clave secreta de autenticación de paquetes HMAC

     ```
     fuera:/etc/openvpn/client# scp root@firewall3.cda.net:/etc/openvpn/server/ta.key .
     ```

   - Crear el fichero de configuración del cliente
     Se usará como base el ejemplo disponible en `/usr/share/doc/openvpn/examples/sample-config-files/`

     ```
     fuera:/etc/openvpn/client# cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf  .
     ```

     Editar los parámetros concretos para nuestros túneles VPN

     ```
     fuera:/etc/openvpn/client# nano client.conf
     ```

     Parámetros destacados (con "→" se señalan los cambios efectuados para nuestro ejemplo):

     ```
         client      ## indica que es la configuración para un cliente

         dev tun     ## tipo de dispositivo de red virtual (= tarjeta de red "software") a través
                     ##     del cual se accederá al túnel cifrado establecido con el servidor

     ->  remote firewall3.cda.net 1194   ## dirección IP y puerto de escucha del servidor OpenVPN
                                  ##     con el que se establecerá el túnel cifrado

         ...
     ->  ca   /etc/openvpn/client/ca.crt    ## parametros de cifrado
     ->  cert /etc/openvpn/client/fuera.crt
     ->  key  /etc/openvpn/client/fuera.key
         ...
     ->  tls-auth /etc/openvpn/client/ta.key 1
     ```

3. Crear el túnel OpenVPN

   Importante: antes de iniciar el túnel asegurar que en **firewall3** está activado el *IP forwading* y desactivadas las reglas `iptables` de Shorewall (ver PREVIO 2)

- Iniciar OpenVPN en servidor (**firewall3**), ejecutar:

  ```
  firewall3:/etc/openvpn/server# systemctl restart openvpn-server@server
  ```

- Iniciar OpenVPN en cliente (**fuera**), ejecutar:

  ```
  fuera:/etc/openvpn/client# systemctl restart openvpn-client@client
  ```

RESULTADO: En ambos extremos del túnel cifrado se crea un interfaz de red "virtual" `/dev/tun0` por el que se accede al enlace cifrado que conforma la red privada virtual.

- Un interfaz *tun* que simula un dispositivo de red a nivel IP, pero en lugar de enviar los paquetes IP dentro de tramas Ethernet sobre un cable de red, los encapsula dentro de los paquetes de la conexión TLS establecida
  - En nuestro caso se trata de una conexión TLS al puerto 1194 UDP de la máquina **firewall3**
- El enlace OpenVPN definirá la red **10.30.30.0/24**
  - El servidor tendrá la dir. IP **10.30.30.1**
  - A los clientes se les asignarán direcciones a partir de **10.30.30.6**
  - El *gateway* (puerta de enlace) de los clientes conectado por VPN será **10.30.30.5**, que reenvía a **10.30.30.1**

  Se puede comprobar la configuración de red en ambos extremos con el comando `ip address`
- En este caso las rutas hacia las dos "redes internas" (red dmz y red interna) se 'inyectan" en el cliente VPN al crear el túnel (comprobar con el comando `ip route` en la máquina **fuera**)
  - La ruta por defecto de los equipos internos usa como *gateway* a **firewall3** que a su vez conoce la ruta hacia las máquinas clientes VPN
  - Por ello, en este caso concreto no es necesario indicar rutas adicionales para que los equipos **dentro** y **dmz** respondan y se comuniquen con los clientes OpenVPN
- En este momento, para el equipo firewall3 tendremos 4 redes
  - 10.10.10.0/24: red interna en el interfaz *enp0s3*
  - 10.20.20.0/24: red dmz en el interfaz *enp0s8*
  - 10.30.30.0/24: equipos externos conectados sobre VPN en el interfaz "virtual" *tun0*
  - red externa en el interfaz *enp0s9*

4. **Tarea 2** [**Comprobar el túnel creado**]

   Comprobar el acceso desde la máquina (**fuera**) a las 2 redes internas detrás de **firewall3**, que inicialmente no eran accesibles.

   - Desde fuera:

     ```
     fuera:~# nmap -T4 10.10.10.11     [escaneo de dentro]
     fuera:~# nmap -T4 10.20.20.22     [escaneo de dmz]
     ```

   - Otra opción: hacer una conexión `ssh` + comprobar con el comando `who` quien está conectado y desde dónde

     ```
     fuera:~# ssh  usuario@10.10.10.11     [con contraseña usuario]
     fuera:~# ssh  usuario@10.20.20.22     [con contraseña usuario]
     ```

3.3. Parte 2: Integración del enlace OpenVPN con Shorewall

Shorewall prevee la posibilidad de dar soporte a conexiones VPN. Veremos como integrar nuestro túnel openVPN en Shorewall

3.3.1. Preparación de Shorewall

1. **Opción 1:** si se ha retomado la práctica 3 "Definición de zonas desmilitarizadas con Shorewall"

   a) Se partirá de la configuración de Shorewall ya existente.

2. **Opción 2:** si se ha iniciado la práctica desde cero

   a) Completar los pasos 1 a 7 de la sección 4.4 de la práctica 3 "Definición de zonas desmilitarizadas con Shorewall"

3.3.2. Pasos a seguir

1. Crear una nueva zona (`road`) para los clientes conectado con OpenVPN en el fichero `/etc/shorewall/zones`

```
firewall3:/etc/shorewall# mousepad zones &

################################################################################
#ZONE   TYPE    OPTIONS                 IN                      OUT
#                                       OPTIONS                 OPTIONS
fw      firewall
net     ipv4
loc     ipv4
dmz     ipv4
road    ipv4
```

   Nota: otra opción más directa sería vincular en el fichero `interfaces` la tarjeta de red *tun0* a la zona `loc`

   - De ese modo, todo el tráfico que llegará al firewall mediante los túneles OpenVPN se consideraría como perteneciente a la zona `loc` (red interna).

2. Asociar el interfaz *tun0* a la zona `road` en el fichero `/etc/shorewall/interfaces`

```
 firewall3:/etc/shorewall# mousepad interfaces &

################################################################################
?FORMAT 2
################################################################################
#ZONE   INTERFACE         OPTIONS
net     enp0s9
loc     enp0s3
dmz     enp0s8
road    tun+
```

3. Definir las políticas y reglas que afectan a los clientes OpenVPN

   Haremos que los equipos conectados por openVPN (zona `road`) tengas las mismas restricciones/privilegios que los de la red interna (zona `loc`).

   - Fichero `/etc/shorewall/policy`

     Habilitar el acceso sin restricciones a la zona interna (`loc`) desde los equipos que lleguen a través del túnel OpenVPN (zona `road`)

```
firewall3:/etc/shorewall# mousepad policy &

################################################################################
#SOURCE         DEST            POLICY          LOG LEVEL       LIMIT:BURST
loc             all             DROP
net             all             DROP
dmz             all             DROP

road            loc             ACCEPT

# THE FOLLOWING POLICY MUST BE LAST
all             all             REJECT          info
```

   - Fichero `/etc/shorewall/rules`

     Replicar las entradas con origen en la zona `loc`, cambiando su campo origen de `loc` a `road`.

     Nota: esto es una simplificación para acelerar el desarrollo del ejemplo. En un entorno real, puede no ser necesario/razonable que los equipos de los usuarios "itinerantes" se equiparen en cuanto a restricciones de acceso con los equipos internos (especialmente si el único mecanismo de autenticación es el uso exclusivo de certificados digitales de clientes).

```
firewall3:/etc/shorewall# mousepad rules &

ACCEPT          road            net               tcp     80,443
ACCEPT          road            net               tcp     22

ACCEPT          road            dmz:10.20.20.22   tcp     80,443
ACCEPT          road            dmz:10.20.20.22   tcp     25,110
ACCEPT          road            dmz               tcp     22

DNS(ACCEPT)     road            net

ACCEPT          road            fw              tcp     22
```

4. Dar de alta el túnel OpenVPN `/etc/shorewall/tunnels`

```
firewall3:/etc/shorewall# mousepad tunnels &

#TYPE                  ZONE    GATEWAY         GATEWAY-ZONE
openvpnserver:1194     net     0.0.0.0/0
```

5. Comprobar la configuración del firewall y el funcionamiento del túnel OpenVPN
   - Recompilar y arrancar el cortafuegos generado por Shorewall con las nuevas configuraciones

     ```
     firewall3~# shorewall start
     ```

   - Reiniciar el servidor OpenVPN en `firewall3`

     `firewall3:~# systemctl restart openvpn-server@server`
   - Reiniciar el cliente OpenVPN en `fuera`

     `fuera:~# systemctl restart openvpn-client@client`
   - **Tarea 3 [Comprobar integración con Shorewall]:** Repetir las comprobaciones realizadas en el punto *(4)* del apartado 3.2.2 y documentar los resultados obtenidos.
     - En concreto, con NMAP se puede comprobar que desde el equipo `fuera` se tiene acceso a los mismos servicios de las redes interna y DMZ que en el caso de equipos de la red interna.

       ```
       fuera~# nmap -T4 10.10.10.11
       fuera~# nmap -T4 10.20.20.22
       ```

4. Documentación a entregar

Esquema propuesto

- Descripción **breve** del ejercicio realizado
- Detallar la situación inicial del la red del ejemplo (escaneos de **Tarea 1** [**escaneo inicial**])
- Detallar las comprobaciones realizadas en el punto *(4)* del apartado 3.2.2 y documentar los resultados obtenidos después de aplicar la configuración inicial de OpenVPN (**Tarea 2** [**comprobar túnel creado**])
- Detallar las comprobaciones realizadas en el punto *(5)* del apartado 3.3.2 y documentar los resultados obtenidos después de integrar el enlace con Shorewall (**Tarea 3** [**comprobar integración shorewall**])
- **Tarea 4.** Describir cómo es el flujo de paquetes (por dónde pasan, "cómo" pasan, "qué" sucede en cada ubicación, etc) que tiene lugar en las pruebas realizadas desde la máquina **fuera** en la **Tarea 3** y cómo les afectan las reglas de filtrado y enrutado establecidas en la máquina **firewall3**.
- Conclusiones: detallar los problemas encontrados, posibles mejoras o alternativas, impresiones sobre la idoneidad de las herramientas, etc

Entrega: MOOVI

Fecha límite: 13/11/2022