

# Definición de túneles cifrados con OpenVPN

---

## Tabla de contenidos

---

- Instalación del entorno de prácticas
- Preparación del entorno
- Escaneando la red desde FUERA
- Creación de la CA y las claves
- Configuración del servidor OpenVPN
- Creación del túnel OpenVPN
- Integración con Shorewall
- Escaneando la red de nuevo
- Curiosidades
- Opinión personal
- Bibliografía

## Instalación del entorno de prácticas

---

Iniciamos el autoinstalador para Linux

```
curl -o- \
  http://ccia.esei.uvigo.es/docencia/CDA/1819/practicas//ejercicio-dmz-openvpn.sh \
  | bash -
```

Seguir cualquier indicación mostrada en [esta página](#) en caso de duda o error.

## Preparación del entorno

---

Lo primero será habilitar la redirección del tráfico en la máquina **FIREWALL3** con el comando `echo 1 > /proc/sys/net/ipv4/ip_forward | cat /proc/sys/net/ipv4/ip_forward`

## Escaneando la red desde FUERA

---

---

Procederemos a escanear la red desde FUERA:

```
nmap -T4 193.147.87.47 # FIREWALL3
nmap -T4 10.10.10.11   # DENTRO
nmap -T4 10.20.20.22   # DMZ
```

En mi caso usaré el comando `nmap -T4 193.147.87.47 10.10.10.11 10.20.20.22` para escanear las tres a la vez. El resultado será el siguiente:

```
Starting Nmap 7.40 ( https://nmap.org ) at 2018-11-17 13:10 CET
Nmap scan report for firewall3.cda.net (193.147.87.47)
Host is up (0.00038s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
79/tcp    open  finger
80/tcp    open  http
110/tcp   open  pop3
143/tcp   open  imap
3260/tcp  open  iscsi
3306/tcp  open  mysql
MAC Address: 08:00:27:66:66:66 (Oracle VirtualBox virtual NIC)
Nmap done: 3 IP addresses (1 host up) scanned in 5.25 seconds
```

Como podemos observar, sólo están visibles los servicios del FIREWALL3. El resto de las máquinas aparecen como desconectadas.

## Creación de la CA y las claves

---

Editamos el archivo de configuración de los CA con `nano /usr/share/easy-rsa/vars` introduciendo los datos de nuestra red:

```
export KEY_COUNTRY="ES"
export KEY_PROVINCE="Ourense"
export KEY_CITY="Ourense"
export KEY_ORG="CDA"
export KEY_EMAIL="cda@cda.net"
```

Ahora generaremos las claves dentro de la carpeta `/usr/share/easy-rsa/vars` ( `cd /usr/share/easy-rsa/vars` ) usando los siguientes comandos:

```
cp openssl-1.0.0.cnf openssl.cnf
```

```
source vars
./clean-all
./build-ca
```

Dejamos todas las opciones predeterminadas menos para "COMMON\_NAME:", en el cual deberemos poner "CA\_pruebas".

Generaremos la clave del servidor usando el comando `./build-key-server firewall3` y dejando todas las opciones por defecto salvo para "COMMON\_NAME:", en la cual pondremos "firewall3.cda.net".

Crearemos también los parámetros de intercambios de claves con:

```
./build-dh
```

Por último, creamos el certificado del cliente dejando todas las opciones por defecto con el comando:

```
./build-key fuera
```

-  Es importante firmar los certificados `Sign the certificate? [y/n]:y`.

## Configuración del servidor OpenVPN

---

Trabajaremos en esta ocasión dentro de la carpeta `/etc/openvpn` ( `cd /etc/openvpn` )

Copiamos las claves y certificados generados a nuestro directorio actual de trabajo:

```
cp /usr/share/easy-rsa/keys/{ca.crt,firewall3.crt,firewall3.key,dh2048.pem} .
```

Creamos una clave secreta para la autenticación HMAC/SSL usando

```
openvpn --genkey --secret ta.key
```

Configuraremos el servidor usando la plantilla que ofrece OpenVPN:

```
cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz .
gunzip server.conf.gz
```

Para ello, editamos el fichero de configuración con `nano server.conf` modificando o añadiendo estos valores:

```
port 1194 # Puerto
proto udp # Protocolo
dev tun # Tipo de Red Virtual
ca /etc/openvpn/ca.crt # Autoridad Certificadora
cert /etc/openvpn/firewall3.crt # Certificado del Servidor
key /etc/openvpn/firewall3.key # Clave del Servidor
dh /etc/openvpn/dh2048.pem # Intercambio de claves
server 10.30.30.0 255.255.255.0 # Rango de direcciones asignadas
push "route 10.10.10.0 255.255.255.0" # Redirección hacia DENTRO
push "route 10.20.20.0 255.255.255.0" # Redirección hacia la DMZ
tls-auth /etc/openvpn/ta.key 0 # Clave secreta HMAC
```

## Configuración del cliente OpenVPN

---

En la máquina cliente (**FUERA** en nuestro caso), debemos copiar de la manera más cómoda y segura para nosotros los archivos `ca.crt`, `firewall3.crt`, `firewall3.key`, `dh2048.pem` y `ta.key` dentro de la carpeta `/etc/openvpn`.

```
scp root@firewall3.cda.net:/usr/share/easy-rsa/keys/{ca.crt,fuera.crt,fuera.key} \
/etc/openvpn/
scp root@firewall3.cda.net:/etc/openvpn/ta.key /etc/openvpn/
```

Configuraremos el cliente de la misma manera que el servidor: copiando la plantilla con el comando `cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/openvpn` y editándolo con `nano /etc/openvpn/client.conf` de tal forma que tengamos definidos los siguientes parámetros:

```
client # Indica que somos clientes
proto udp # Protocolo
dev tun # Tipo de Red Virtual
remote 193.147.87.47 1194 # Conexión con el servidor
ca /etc/openvpn/ca.crt # Autoridad Certificadora
cert /etc/openvpn/fuera.crt # Certificado del Cliente
key /etc/openvpn/fuera.key # Clave del Cliente
tls-auth /etc/openvpn/ta.key 1 # Clave secreta HMAC
```

## Creación del túnel OpenVPN

---

Iniciaremos el servicio OpenVPN con el siguiente comando para obtener los logs y poder realizar un seguimiento de los posibles errores:

```
# FIREWALL3 (Servidor)
```

```
openvpn --config /etc/openvpn/server.conf
# FUERA (Cliente)
openvpn --config /etc/openvpn/client.conf
```

Podemos comprobar que todo ha salido de manera correcta ejecutando el comando `ip r` en **FIREWALL3** y comprobando que se ha creado la interfaz `tun0` con la IP **10.30.30.1**:

```
default via 193.147.87.1 dev enp0s9
10.10.10.0/24 dev enp0s3 proto kernel scope link src 10.10.10.1
10.20.20.0/24 dev enp0s8 proto kernel scope link src 10.20.20.1
10.30.30.0/24 via 10.30.30.2 dev tun0
--
10.30.30.2 dev tun0 proto kernel scope link src 10.30.30.1
--
193.147.87.0/24 dev enp0s9 proto kernel scope link src 193.147.87.47
```

Del mismo modo, comprobaremos la conexión al túnel en la máquina **FUERA** con el comando `ip r`:

```
default via 193.147.87.1 dev enp0s3
--
10.10.10.0/24 via 10.30.30.5 dev tun0
10.20.20.0/24 via 10.30.30.5 dev tun0
10.30.30.1 via 10.30.30.5 dev tun0
--
10.30.30.5 dev tun0 proto kernel scope link src 10.30.30.6
--
193.147.87.0/24 dev enp0s3 proto kernel scope link src 193.147.87.33
```

Vemos además que tiene acceso a las subredes de la DMZ e interna.

Haremos un escaneo de puertos para ver si esto es realmente cierto:

```
# nmap -T4 10.10.10.11 10.20.20.22
Starting Nmap 7.40 ( https://nmap.org ) at 2018-11-17 17:22 CET
Nmap scan report for 10.10.10.11
Host is up (0.0023s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
79/tcp    open  finger
80/tcp    open  http
110/tcp   open  pop3
143/tcp   open  imap
3260/tcp  open  iscsi
3306/tcp  open  mysql
Nmap scan report for 10.20.20.22
Host is up (0.0026s latency).
```

```
Not shown: 990 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
79/tcp    open  finger
80/tcp    open  http
110/tcp   open  pop3
143/tcp   open  imap
3260/tcp  open  iscsi
3306/tcp  open  mysql
Nmap done: 2 IP addresses (2 hosts up) scanned in 51.28 seconds
```

En este caso sí que podemos acceder a los equipos de la red interna.

## Integración con Shorewall

Teniendo en cuenta que se ha levantado un servicio Shorewall siguiendo [la guía anterior](#), crearemos una nueva zona llamada **road** para los clientes de la VPN editando el fichero `/etc/shorewall/zones` y añadiendo lo siguiente:

```
road  ipv4
```

Asociamos la interfaz `tun0` a la zona **road** añadiendo la siguiente línea en el fichero `/etc/shorewall/interfaces`:

```
road  tun+
```

Habilitamos el tráfico desde el túnel VPN a la red interna añadiendo la siguiente línea al fichero `/etc/shorewall/policy` antes de la regla `all all REJECT info`:

```
road      loc      ACCEPT
```

Y daremos los mismos privilegios que tiene **loc** a **road** modificando el archivo `/etc/shorewall/rules`:

```
# Acceso ssh desde el túnel al firewall y a la DMZ
SSH(ACCEPT) road  $FW
SSH(ACCEPT) road  dmz
# Acceso web y ssh desde el túnel hacia la red externa
ACCEPT road  net      tcp 22
ACCEPT road  net      tcp 80,443
# Acceso web, ssh y dns desde el túnel a la DMZ
ACCEPT road  dmz:10.20.20.22 tcp 80,443
```

```
ACCEPT road dmz:10.20.20.22 tcp 25,110
# Acceso a internet desde el túnel
DNS(ACCEPT) road net
```

A mayores, deberemos dar de alta el túnel VPN en el archivo `/etc/shorewall/tunnels` añadiendo la siguiente línea:

```
openvpnserver:1194 net 0.0.0.0/0
```

Ahora solamente falta levantar el servicio Shorewall y reiniciar el túnel VPN:

```
# FIREWALL3
rm /etc/shorewall/stoppedrules
shorewall start
systemctl restart openvpn@server.service
# FUERA
systemctl restart openvpn@client.service
```

## Escaneando la Red de nuevo

---

Volvemos a realizar un escaneo desde **FUERA** para comprobar si el firewall está levantado:

```
Starting Nmap 7.40 ( https://nmap.org ) at 2018-11-17 17:54 CET
Nmap scan report for 10.10.10.11
Host is up (0.0020s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
79/tcp    open  finger
80/tcp    open  http
110/tcp   open  pop3
143/tcp   open  imap
3260/tcp  open  iscsi
3306/tcp  open  mysql
Nmap scan report for 10.20.20.22
Host is up (0.0014s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
110/tcp   open  pop3
Nmap done: 2 IP addresses (2 hosts up) scanned in 49.62 seconds
```

Como podemos observar, no existe ninguna restricción hacia la máquina interna ( 10.10.10.11 ) dado que "estamos" en esa red, pero el firewall sí que actúa hacia la DMZ.

## Curiosidades

---

Una cosa curiosa es el tema de los paquetes a través de la red.

Si hacemos un `traceroute` desde **FUERA** al servidor web de la **DMZ**, obtenemos lo siguiente:

```
traceroute to 10.20.20.22 (10.20.20.22), 30 hops max, 80 byte packets
 1  10.30.30.1  3.005 ms  2.736 ms  2.681 ms
 2  10.30.30.1  3.288 ms  3.202 ms  6.382 ms
```

La IP mostrada no es la del FIREWALL3, como en el ejemplo de la DMZ, sino la del FIREWALL3 dentro del túnel (lógicamente). Aún así podemos observar el rebote que éste genera.

Si intentamos hacer un PING al servidor de la **DMZ**, no podremos obtener respuesta:

```
# ping 10.20.20.22
PING 10.20.20.22 (10.20.20.22) 56(84) bytes of data.
From 10.30.30.1 icmp_seq=1 Destination Host Unreachable
From 10.30.30.1 icmp_seq=2 Destination Host Unreachable
^C
--- 10.20.20.22 ping statistics ---
121 packets transmitted, 0 received, +121 errors, 100% packet loss, time 120289ms
```

Pero sí que podremos conectarnos mediante el servidor SSH y veremos que nuestra IP ha sido enmascarada mostrando la del túnel:

```
# ssh usuario@10.20.20.22
usuario@dmz:~$ who
usuario pts/0          2018-11-17 18:02 (10.30.30.6)
```

## Opinión Personal

---

En general pienso que las soluciones detalladas dan muy buen resultado con el fin de proteger la red.

Como en todo, siempre existen agujeros de seguridad, por ejemplo:

En la versión instalada en la máquinas virtuales ( `OpenVPN 2.4.0` ), un atacante sin autenticar






podría denegar el servicio de todos los usuarios de la VPN explotando el error **CVE-2017-7478**, el cual [no es difícil de replicar](#).

OpenVPN es un servicio que ha de estar actualizándose de manera constante dada la enorme cantidad de fallas de seguridad de las que gozan sus versiones, por lo que supone un esfuerzo realmente alto mantener una red segura.

En este caso, la herramienta de seguridad es la que, si eres descuidado, te va a provocar una gran brecha y suponer peligro. Esto, claramente, no es así si se toman las medidas oportunas y se está al tanto de las nuevas fallas que salen cada día.

## Bibliografía

---

-  <https://github.com/Student-Puma/HomeLab>
-  [https://lihuen.linti.unlp.edu.ar/index.php/Configurando\\_Redes\\_Privadas\\_Virtuales\\_con\\_OpenVPN](https://lihuen.linti.unlp.edu.ar/index.php/Configurando_Redes_Privadas_Virtuales_con_OpenVPN)
-  <http://ccia.esei.uvigo.es/docencia/CDA/1819/practicas/ejercicio-openvpn/ejercicio-openvpn.html>