ALUMNES: Arturo Aragón Hidalgo, Jesika Jiménez Prado

GRUP: |5012

DOCUMENTACIÓ PRÀCTICA 3 - SESSIÓ 3 - VPN

—----------------------------------------------------------------------

\*\*\***VPN con Clave Estática**\*\*\*

Objetivo:

Para esta primera parte, utilizaremos NAT y red interna, para que puedan comunicarse las máquinas y tengan acceso a internet.

Establecer una conexión VPN entre un servidor y un cliente utilizando una clave estática compartida (sin certificados ni CA).

Asignamos las IPs en la red local:

Servidor: 10.4.10.4

Cliente: 10.4.10.5

Estas son las IPs locales que usamos para comunicarnos entre las dos máquinas antes de establecer la VPN.

Generamos la clave estática en el servidor:

Ingresamos al directorio de configuración de OpenVPN:

cd /etc/openvpn

Ejecutamos el siguiente comando para generar la clave:

openvpn --genkey secret static.key

Este archivo static.key será usado por ambos equipos para cifrar la conexión VPN.

Copiamos la clave al cliente:

Desde el servidor, enviamos la clave al cliente usando scp:

scp static.key entel@10.4.10.5:/home/entel

Esto copia el archivo al directorio personal del usuario entel en el cliente.

En el cliente, como superusuario, movemos la clave al directorio de OpenVPN:

mv /home/entel/static.key /etc/openvpn/

Configuramos el servidor

Creamos el archivo /etc/openvpn/tun0.conf con el siguiente contenido:

dev tun0  
 ifconfig 10.9.8.1 10.9.8.2  
 secret /etc/openvpn/static.key

dev tun0: crea una interfaz virtual llamada tun0 para el túnel VPN.

ifconfig 10.9.8.1 10.9.8.2: asigna las IPs de la VPN, el servidor usará la primera y el cliente la segunda.

secret: indica la ubicación del archivo de clave compartida.

Configuramos el cliente:

Creamos el archivo /etc/openvpn/tun0.conf con este contenido:

dev tun0  
 ifconfig 10.9.8.2 10.9.8.1  
 secret /etc/openvpn/static.key

Aquí invertimos las IPs: el cliente toma la segunda dirección (10.9.8.2) y el servidor la primera (10.9.8.1), para que coincidan desde ambos lados.

Verificamos el firewall:

Si la VPN se utiliza dentro de una red con NAT, normalmente no necesitamos hacer cambios adicionales, ya que OpenVPN usa el puerto UDP 1194 por defecto, y suele venir abierto en muchas configuraciones internas.

**\*\*\*PRUEBAS DE VERIFICACIÓN\*\*\***

**pruebas\_client\_clave.txt:**Las pruebas consistieron en verificar la conectividad y el correcto funcionamiento de una interfaz de túnel VPN (tun1) configurada en un cliente OpenVPN. Se comprobó la configuración de las interfaces de red y se realizaron pings entre las direcciones IP del túnel (10.9.8.2 y 10.9.8.1) para asegurar la comunicación bidireccional sin pérdida de paquetes. Además, se capturaron y analizaron los paquetes ICMP con tcpdump, confirmando que el tráfico de ping atravesaba correctamente el túnel VPN, evidenciando así la operatividad y estabilidad de la conexión segura establecida.

**pruebas\_servidor\_clave.txt:**Las pruebas realizadas consistieron en la configuración y verificación de una interfaz VPN tunelada usando OpenVPN con clave estática. Se comprobó la correcta asignación de direcciones IP en las interfaces de red, la conexión del túnel tun1 y la comunicación entre los extremos mediante comandos ping específicos, confirmando la respuesta exitosa y sin pérdida de paquetes. Además, se capturó el tráfico con tcpdump para analizar los paquetes ICMP intercambiados, asegurando que la VPN cifrada funcionara correctamente y que los paquetes circulaban a través del túnel seguro.

**\*\*\*ARCHIVOS DE CONFIGURACIÓN\*\*\***

**tun1\_cliente.conf**

**tun1\_server.conf**

**VPN CON CERTIFICADOS**

Configuración OpenVPN con Certificados

Preparación Inicial

Se activa el IP Forwarding para permitir que el tráfico pueda enrutar desde la VPN a otras red. Esto es esencial para que los clientes conectados a la VPN puedan acceder a otras red, como Internet o una red LAN.

Editamos el archivo /etc/sysctl.conf y descomentamos la línea:

net.ipv4.ip\_forward=1

Luego aplicamos los cambios con el comando:

sysctl -p

Después, instalamos el paquete de OpenVPN que es el software que utilizaremos para crear la VPN:

apt install openvpn

Generación de Certificados (Servidor)

2.1 Preparar entorno EasyRSA

EasyRSA es una herramienta que permite crear la infraestructura de clave pública (PKI), necesaria para generar certificados. Copiamos el entorno base de EasyRSA y accedemos a su directorio:

cp -r /usr/share/easy-rsa /etc/openvpn/

cd /etc/openvpn/easy-rsa

Editamos el archivo vars.example para establecer la información por defecto de los certificados:

nano vars.example

Al final del archivo añadimos las siguientes líneas con las datos de la organización:

export KEY\_COUNTRY="SPAIN"

export KEY\_PROVINCE="BCN"

export KEY\_CITY="BCN"

export KEY\_ORG="seax"

export KEY\_EMAIL="admin@example.com"

export KEY\_OU="OpenVPN"

2.2 Generar PKI y Certificados

Inicializamos la PKI, generamos la CA (autoridad certificadora), la solicitud y certificado del servidor, los parámetros Diffie-Hellman (para intercambio seguro de claves), y una clave TLS para autenticación extra:

./easyrsa init-pki

./easyrsa build-ca nopass

./easyrsa gen-req server nopass

./easyrsa sign-req server server

./easyrsa gen-dh

openvpn --genkey secret ta.key

2.3 Copiar archivos clave

Copiamos los archivos necesarios para el servidor VPN en el directorio principal de OpenVPN:

cp {ta.key,ca.crt,server.crt,server.key,dh.pem} /etc/openvpn/

Generar Certificado Cliente (Servidor)

Creamos una solicitud y certificado para el cliente. Se utiliza el mismo sistema para generar certificados únicos por cada cliente.

./easyrsa gen-req cliente nopass

./easyrsa sign-req cliente cliente

Creamos un directorio para almacenar los certificados del cliente:

mkdir /etc/openvpn/client

Copiamos los certificados del cliente en este directorio:

cp {pki/es.crt,pki/issued/client.crt,pki/private/client.key} /etc/openvpn/client/

Configurar Servidor OpenVPN

4.1 Crear el archivo de configuración /etc/openvpn/server.conf con las siguientes opciones:

modo server

puerto 1194

proto udp

dev tun

topology subnet

es /etc/openvpn/es.crt

cierto /etc/openvpn/server.crt

key /etc/openvpn/server.key

dh /etc/openvpn/dh.pem

push "route 10.4.1.0 255.255.255.0"

tls-auth /etc/openvpn/ta.key 0

server 10.4.1.0 255.255.255.0

ifconfig 10.4.1.1 10.4.1.2

persist-key

persist-tun

keepalive 10 120

cipher AES-256-CBC

data-ciphers AES-256-GCM:AES-256-CBC

user nobody

group nogroup

status openvpn-status.log

verb 3

Se utiliza la subred 10.4.1.0/24 para la VPN. El servidor tendrá la IP 10.4.1.1 y el cliente 10.4.1.2. El protocolo se UDP puede ser más eficiente para VPNs. TLS-auth mejora la seguridad protegiendo contra ataques tipo DoS o escaneo de puertos.

4.2 Iniciar el servicio de OpenVPN en el servidor:

systemctl start openvpn@server

systemctl status openvpn@server

CONFIGURAR CLIENTE VPN

5.1 Crear el archivo de configuración /etc/openvpn/client.conf con las siguientes opciones:

cliente

dev tun

proto udp

remote 10.4.10.4 1194

resolv-retry infinite

nobind

user nobody

group nogroup

persist-key

persist-tun

es /etc/openvpn/es.crt

cierto /etc/openvpn/client.crt

key /etc/openvpn/client.key

remote-cert-tls server

tls-auth /etc/openvpn/ta.key 1

cipher AES-256-CBC

data-ciphers AES-256-GCM:AES-256-CBC

verb 3

La IP 10.4.10.4 es la IP del servidor VPN vista desde el cliente. Puede ser una IP interna de una red local o IP pública si el servidor está expuesto en Internet.

5.2 Copiar certificados desde servidor al cliente

Se debe transferir manualmente los archivos necesarios al cliente mediante scp y después moverlos a su ubicación definitiva:

scp /etc/openvpn/client/es.crt entel@IP\_CLI:/home

scp /etc/openvpn/client/client.crt entel@IP\_CLI:/home/entel

scp /etc/openvpn/client/client.key entel@IP\_CLI:/home/entel

scp /etc/openvpn/ta.key entel@IP\_CLI:/home/entel

Una vez copiados en el cliente, movemos los archivos al directorio correspondiente:

cd /hombre/entel

sudo cp es.crt /etc/openvpn/

sudo cp client.crt /etc/openvpn/

sudo cp client.key /etc/openvpn/

sudo cp ta.key /etc/openvpn/

5.3 Iniciar servicio en el cliente

Una vez configurado el archivo client.conf y copiados los certificados, iniciamos el cliente OpenVPN:

systemctl start openvpn@client

Verificamos que el servicio esté activo:

systemctl status openvpn@client

Verificar Conexión

En el cliente, comprobamos que la interfaz de red virtual tun0 está activa y que tenga una IP en el rango de la VPN:

ip a

Debe aparecer una interfaz tun0 con una IP en el rango 10.4.1.x, asignada por el servidor.

En el servidor, monitoreamos el log de OpenVPN para verificar que haya conexiones entrantes desde los clientes:

tail -f /var/log/openvpn/openvpn.log

Notas sobre configuraciones y decisiones tomadas:

UDP y puerto 1194: UDP es más rápido y eficiente para túneles VPN porque no requiere el overhead de control de errores que ya maneja OpenVPN por sí mismo. El puerto 1194 es el puerto estándar de OpenVPN.

Subred 10.4.1.0/24: Se usa para asignar direcciones IP a los clientes VPN. Se eligió este rango privado por ser fácil de identificar y evitar conflictos con redes comunes como 192.168.x.x.

tls-auth y ta.key: Añade una capa de seguridad para evitar que usuarios no autorizados inicien conexión siquiera. Protege contra escaneo de puertos y ataques de denegación de servicio.

AES-256-CBC y AES-256-GCM: Se eligieron por ser algoritmos de cifrado fuertes y ampliamente compatibles. GCM es más moderno y eficiente, pero se incluye CBC por compatibilidad.

user nobody y group nogroup: Se ejecuta el proceso con permisos mínimos por motivos de seguridad.

El firewall que hemos pensado en este caso es el siguiente:

#!/usr/sbin/nft -f

flush ruleset

# --- Definición de Variables ---

define EXT\_IF = "eth0" # Interfaz externa (ajustar según tu sistema)

define VPN\_IF = "tun0" # Interfaz VPN

define VPN\_PORT = 1194 # Puerto OpenVPN

define SSH\_PORT = 22 # Puerto SSH

define LOCAL\_NET = 10.4.1.0/24 # Red VPN

# --- Tablas y Cadenas Base ---

table inet firewall {

chain input {

type filter hook input priority 0; policy drop;

# Conexiones establecidas y relacionadas

ct state established,related accept

# Loopback

iifname "lo" accept

# ICMP (Ping)

icmp type echo-request accept

# OpenVPN

udp dport $VPN\_PORT accept

# Logging de paquetes rechazados

log prefix "Input-Dropped: " counter drop

}

chain forward {

type filter hook forward priority 0; policy drop;

# Permitir tráfico VPN <-> Internet

iifname $VPN\_IF oifname $EXT\_IF accept

iifname $EXT\_IF oifname $VPN\_IF ct state established,related accept

# Logging de paquetes rechazados

log prefix "Forward-Dropped: " counter drop

}

chain output {

type filter hook output priority 0; policy accept;

}

# --- NAT para Clientes VPN ---

chain postrouting {

type nat hook postrouting priority srcnat; policy accept;

# SNAT para tráfico de clientes VPN

ip saddr $LOCAL\_NET oifname $EXT\_IF masquerade

}

}

**\*\*\*PRUEBAS DE VERIFICACIÓN\*\*\***

**1. pruebas\_funcionamiento\_server\_certificado.txt**

Pruebas de conectividad y operación del servidor VPN:

Verificación de interfaces de red (ip a), mostrando la interfaz VPN tun0 activa con IP 10.4.1.1.

Test de conectividad con el cliente (ping -I tun0 10.4.1.2), confirmando latencia baja y 0% pérdida de paquetes.

Captura de tráfico VPN (tcpdump -i tun0) que muestra intercambio constante de paquetes ICMP.

Estado del servicio OpenVPN (systemctl status), confirmando conexión activa con el cliente client\_seax y asignación de IP 10.4.1.2.

**2. route\_client.txt**

Verificación de rutas en el cliente:

Comprobación de ruta hacia el servidor VPN (ip route get 10.4.1.1), confirmando que el tráfico usa la interfaz tun0 con IP origen 10.4.1.2.

**3. prueba\_servidor\_tcpdump.tx**t

Monitoreo de tráfico VPN externo:

Captura de tráfico UDP en el puerto 1194 (tcpdump -i enp0s3), detectando paquetes entre 10.4.10.4 (servidor) y 10.4.10.5 (cliente).

Análisis del archivo de captura (servidor\_captura\_tcpdump.pcap), mostrando intercambio constante de paquetes de 108 bytes (tráfico de control OpenVPN).

**4. pruebas\_funcionamiento\_cliente\_certificado.txt**

Pruebas de operación del cliente VPN:

Verificación de interfaces (ip a), confirmando la interfaz tun0 activa con IP 10.4.1.2.

Test de conectividad con el servidor (ping -I tun0 10.4.1.1), con latencia estable y 0% pérdida.

Estado del servicio OpenVPN cliente (systemctl status), validando inicialización correcta y uso de cifrado AES-256-GCM.

**5. route\_server.txt**

Verificación de rutas en el servidor VPN:

Comprobación de ruta hacia el cliente VPN (ip route get 10.4.1.2), confirmando que el tráfico se dirige a través de la interfaz tun0 con IP origen 10.4.1.1. Esto valida que la ruta hacia la IP del cliente (asignada por el servidor VPN) está correctamente configurada en la tabla de enrutamiento del servidor.

**\*\*\*ARCHIVOS DE CONFIGURACION\*\*\***

**client\_archivos.txt**

Validación de configuración y certificados del cliente:

Listado de archivos en /etc/openvpn/ (certificados y claves).

Exhibición de contenidos:

Certificado CA (ca.crt).

Configuración del cliente (client.conf): conexión UDP a 10.4.10.4:1194 con cifrado AES-256.

Certificado (client.crt) y clave privada (client.key) del cliente.

Estado del servicio OpenVPN cliente (systemctl status), mostrando túnel activo con IP 10.4.10.2 y conexión establecida.

**client\_conf.txt**

Revisión de configuración del cliente:

Visualización del archivo de configuración (client.conf), detallando parámetros como protocolo UDP, certificados, y opciones de cifrado (AES-256-GCM/CBC).

**server\_archivos.txt**

Validación de archivos de configuración y certificados del servidor:

Exhibición de contenidos críticos:

Certificado de la CA (ca.crt), idéntico al usado por el cliente.

Parámetros Diffie-Hellman (dh.pem) para el intercambio de claves.

Certificado del servidor (server.crt), emitido por la CA "seax" con SAN DNS:seax y uso extendido para autenticación de servidor TLS.

Clave privada del servidor (server.key).

Listado completo de archivos en /etc/openvpn/, mostrando todos los componentes necesarios: certificados, claves, configuración (server.conf), y archivo de autenticación TLS (ta.key).

**USUARI WINDOWS (Opcional):**

Antes de configurar nada, debemos obtener los archivos en nuestra máquina Host con windows los archivos que generamos en la Máquina Virtual donde configuramos el VPN server con OpenVPN:

* ca.crt
* client.crt
* client.key
* ta.key

Configuración:

Primero descargaremos OpenVPN en nuestra máquina host con Windows (<https://openvpn.net/community-downloads/>). Más concretamente la versión OpenVPN 2.6.14.

Una vez instalado, marcamos la opción EasyRSA 3 Certificate Management Scripts para poder gestionar los certificados.

Crearemos un fichero de configuración en una ruta similar a esta:  
  
C:\Program Files\OpenVPN\config\client\

Y aquí copiaremos los ficheros que mencionamos antes (ca.crt, client.crt, client.key, ta.key)

Y en la misma ruta crearemos un fichero client.ovpn, con el siguiente contenido:  
  
client

dev tun

proto udp

remote 10.4.10.4 # IP del servidor Debian

resolv-retry infinite

nobind

persist-key

persist-tun

ca ca.crt

cert client.crt

key client.key

remote-cert-tls server

tls-auth ta.key 1

cipher AES-256-CBC

verb 3

Abrimos la GUI de OpenVPN como administrador y seleccionamos la opción de “Conectar” en el perfil client.

Para comprobar que funciona correctamente deberíamos ver el siguiente mensaje:  
  
Initialization Sequence Completed

Verificación  
  
Al hacer ipconfig en un terminal cmd o PowerShell deberíamos ver una nueva interfaz como la siguiente:  
  
TAP-Windows Adapter

con una IP del rango 10.4.1.0/24

También podemos ver que con un ping a la misma dirección funciona:  
  
ping 10.4.10.4

[añadir ping]

**Bibliografia:**

<https://openvpn.net>

<https://www.netzmafia.de/skripten/server/openvpn.html>

<https://openvpn.net/community-resources/ethernet-bridging/>

<https://help.ubuntu.com/community/OpenVPN>