SAD Tarea 2 Redes Privadas Virtuales

1. VPN de acceso remoto con OpenVPN y certificados x509 (10 puntos)

Configura una conexión VPN de acceso remoto entre dos equipos del cloud:

* + Uno de los dos equipos (el que actuará como servidor) estará conectado a dos redes
  + Para la autenticación de los extremos se usarán obligatoriamente certificados digitales, que se generarán utilizando openssl y se almacenarán en el directorio /etc/openvpn, junto con los parámetros Diffie-Helman y el certificado de la propia Autoridad de Certificación.
  + Se utilizarán direcciones de la red 10.99.99.0/24 para las direcciones virtuales de la VPN. La dirección 10.99.99.1 se asignará al servidor VPN.
  + Los ficheros de configuración del servidor y del cliente se crearán en el directorio

/etc/openvpn de cada máquina, y se llamarán servidor.conf y cliente.conf respectivamente.

* + Tras el establecimiento de la VPN, la máquina cliente debe ser capaz de acceder a una máquina que esté en la otra red a la que está conectado el servidor.

Documenta el proceso detalladamente. A continuación tienes los pasos a seguir para realizar esta configuración:

Instalación y Configuración de un Servidor OpenVPN en Linux

# Ejercicio 1 : OPENSSH

* 1. **Instalación de openssh**

apt-get install openssh-server.

Cambiar entradas en el fichero de configuración del servidor /etc/ssh/sshd\_config y poner: PermitRootLogin yes

PasswordAuthenticacion yes

# Verifica la conexión

[ssh root@10.5.1.10](mailto:sshroot@10.5.1.10)

# Conexiones desatendidas

Para ello utilizaré la autenticación mediante clave pública, para lo cual debo ejecutar como el usuario que

realizará la conexión con el servidor mediante SSH “user1” los siguientes comandos:

ssh-keygen

Este comando generará un par de claves asimétricas RSA (pública y privada) que serán utilizadas en el proceso de autenticación mediante clave pública. Dejo las opciones por defecto.

[ssh-copy-id -i ~/.ssh/id\_rsa.pub **user1**@10.5.1.12](mailto:ssh-copy-id-i~/.ssh/id_rsa.pubuser1@10.5.1.12)

Una vez generadas, debo añadir la clave pública al archivo “authorized\_keys” (lista de claves públicas de

confianza) del usuario del servidor en nombre del cual se ejecutarán los comandos remotamente. Dicha clave será utilizada para verificar la identidad del usuario que trata de acceder mediante el

mecanismo de autenticación por desafío y si su identidad es la correcta le permitirá el acceso sin contraseña.

Para ello utilizo el comando “ssh-copy-id” con la opción “-i” para indicar la ruta donde se encuentra la clave pública que deseo añadir y a continuación indico el usuario al cual añadiré dicha clave “user1 o root” junto con la IP del servidor “10.5.1.12” separados por un símbolo “@”. Tras autenticarme con la contraseña del usuario destino, la clave queda añadida.

# Verifica la conexión

[ssh user1@10.5.1.12](mailto:sshuser1@10.5.1.12)

Si hemos puesto contraseña al añadir la clave privada nos la pedirá, de lo contrario no lo hará.

# Ejercicio 2: Instalación de OpenVPN

* 1. **Instalación del Servidor**

apt-get update

Para actualizar la lista local de repositorios y paquetes disponibles para instalar/actualizar. apt-cache search openvpn

Para comprobar que el paquete “openvpn” está disponible en los repositorios.

apt-get install openvpn

Para instalar el paquete “openvpn”.

# Configuración del servidor

La configuración se realiza mediante un archivo que he de crear, en el cual definiré todos los parámetros necesarios para establecer la conexión punto a punto con el cliente VPN:

cd /etc/openvpn/server

Comienzo cambiando al directorio de trabajo actual al que almacenará todos los archivos necesarios para

configurar el servidor “/etc/openvpn/server”, el cual se crea durante la instalación de OpenVPN.

nano server-vpn.conf

En él creo, mediante el editor “nano”, el archivo de configuración para el servidor al cual llamaré “servervpn.conf”.

dev tun

Para el parámetro “dev” utilizado para seleccionar el dispositivo TUN/TAP para la VPN SSL, indico el valor “tun” (Modo túnel), este emula un dispositivo punto a punto que permite crear túneles virtuales trabajando con el protocolo IP (capa de red).

ifconfig 10.0.0.1 10.0.0.2

Mediante el parámetro “ifconfig” configuro las direcciones IP virtuales de los extremos de la conexión VPN. Primero indico la ***IP virtual de la máquina local*** (en este caso el servidor) y a continuación la ***IP virtual de la máquina remota*** que se encontrará en el otro extremo del túnel (en este caso el cliente). Dichas direcciones poseen la particularidad de pertenecer a una subred de máscara 30 (/30).

tls-server

Mediante este parámetro, activo la **validación TLS** indicando que el rol llevado a cabo por la máquina local será el de servidor.

dh /etc/openvpn/server/dh1024.pem

Utilizo el parámetro “dh” para indicar la ruta a los parámetros **Diffie-Hellman** “dh1024.pem” empleados en la creación de la clave de sesión (simétrica) utilizada para **cifrar los datos transmitidos a través del túnel virtual.**

ca /etc/openvpn/server/ca.crt

Utilizo el parámetro “ca” para indicar **la ruta al certificado** “ca.crt” de la “Autoridad de Certificación” (utilizada para firmar tanto el certificado del servidor como el del cliente VPN), el cual se utilizará en ambos extremos del túnel para garantizar la validez del certificado recibido por la máquina remota (su identidad).

cert /etc/openvpn/server/server-vpn.crt

Utilizo el parámetro “cert” para indicar la ruta al certificado “server-vpn.crt” de la máquina local (en este caso el servidor) que le será enviado a la máquina remota como parte del proceso de autenticación (para demostrar que es quien dice ser). Dicho certificado contiene la clave pública del servidor.

key /etc/openvpn/server/server-vpn.key

Utilizo el parámetro “key” para indicar la ruta a la clave privada “server-vpn.key” de la máquina local (en este caso el servidor), la cual es la pareja de la clave pública incluida en el certificado indicado en el parámetro anterior. Ambas serán utilizadas (como ya he indicado) en el proceso de autenticación de los extremos de la conexión VPN.

comp-lzo

Mediante este parámetro, activo la **compresión “LZO”** de los datos transmitidos a través del túnel VPN. keepalive 20 100

Mediante el parámetro “keepalive” se pueden detectar caídas de la conexión con el otro extremo del túnel. Para ello indico en primer lugar, cada cuántos segundos se comprueba si se ha recibido algún paquete del otro extremo (en este caso **20s**) y en el caso de no haber recibido ninguno se **realizará un ping** para comprobar si la máquina remota sigue activa.

Si tras la primera comprobación transcurren los segundos indicados en segundo lugar (en este caso **100s) sin recibir paquetes, se intentará restablecer la conexión.**

log /var/log/server-vpn.log

Utilizo el parámetro “log” para indicar la ruta del archivo de log “server-vpn.log” donde se registrarán los eventos que OpenVPN genere durante el establecimiento de la conexión y el transcurso de la misma.

verb 3

Mediante el parámetro “verb” indico la cantidad de eventos que se registran en el archivo de log, que

permite valores de 0 (ningún evento) a 11 (máximo) y cuyo valor establezco en 3 (recomendado).

Para terminar, guardo los cambios realizados y salgo del editor.

# Certificado CA y claves.

Para generar las clave privada de la Autoridad Certificadora (CA), ejecuto el siguiente comando:

openssl genrsa -out ca.key 2048

El comando “openssl” es utilizado para usar las diversas funciones criptográficas de la biblioteca criptográfica de OpenSSL desde la shell. En este caso la utilizo para crear una clave privada “RSA”, lo cual se indica con la opción “genrsa” y cuyo archivo de salida “-out” será “ca.key”. En último lugar

indico la longitud de la clave, que será de “202048” bits. ls -l

Compruebo que se ha creado el archivo de la clave privada.

cat ca.key

Muestro su contenido para verificar que la clave ha sido creada correctamente.

Para generar el certificado de la Autoridad Certificadora (CA), ejecuto el siguiente comando:

openssl req -new -x509 -key ca.key -out ca.crt -days 1825

Donde (utilizando nuevamente “openssl”) indico que se cree una solicitud de certificado “req” nueva “new”, la cual será autofirmada “-x509” para la clave privada “-key” que se almacena en el archivo “ca.key” (creado previamente). El archivo de salida “-out” que almacenará el certificado será “ca.crt” y el certificado tendrá una validez de “1825” días “-days” (5 años). Para terminar, proporciono los datos solicitados los cuales serán incluidos en el certificado.

ls -l

Compruebo que se ha creado el archivo del certificado.

cat ca.crt

Muestro su contenido para verificar que el certificado ha sido creado correctamente.

# Parámetros Diffie-Hellman “dh1024.pem” empleados en la creación de la clave de sesión (simétrica) utilizada para cifrar los datos transmitidos a través del túnel virtual. Cliente Instalación de OpenVPN en la Máquina Cliente Utilizando Validación TLS

**2 b.0 Instalación cliente**

Para realizar la instalación, ejecuto (como “root”) los siguientes comandos en el cliente:

apt-get update

Para actualizar la lista local de repositorios y paquetes disponibles para instalar/actualizar.

apt-cache search openvpn

Para comprobar que el paquete “openvpn” está disponible en los repositorios.

apt-get install openvpn

Para instalar el paquete “openvpn”.

# b.1) Configuración

* 1. Configuración de OpenVPN en la Máquina Cliente

La configuración se realiza mediante un archivo que he de crear, en el cual definiré todos los parámetros necesarios para establecer la conexión punto a punto con el servidor VPN:

cd /etc/openvpn/client

Comienzo cambiando el directorio de trabajo actual al que almacenará todos los archivos necesarios para

configurar el cliente “/etc/openvpn/client”, el cual se crea durante la instalación de OpenVPN.

nano client-vpn.conf

En él creo, mediante el editor “nano”, el archivo de configuración para el cliente al cual llamaré “clientvpn.conf”.

En dicho archivo configuro los siguientes parámetros:

dev tun

Para el parámetro “dev” utilizado para seleccionar el **dispositivo TUN/TAP** para la VPN SSL, índico el valor “tun” (Modo túnel), este emula un dispositivo punto a punto que permite crear túneles virtuales trabajando con el protocolo IP (capa de red).

ifconfig 10.0.0.2 10.0.0.1

Mediante el parámetro “ifconfig” configuro las direcciones IP virtuales de los extremos de la conexión VPN. Primero indico la IP virtual de la máquina local (en este caso el cliente) y a continuación la IP virtual de la máquina remota que se encontrará en el otro extremo del túnel (en este caso el servidor). Dichas direcciones poseen la particularidad de pertenecer a una subred de máscara 30 (/30).

tls-client

Mediante este parámetro, activo la validación TLS indicando que el rol llevado a cabo por la máquina local será el de cliente.

remote 10.5.1.12

Mediante el parámetro “remote” indico la dirección del adaptador de red del servidor al cual debe conectarse “10.5.1.12”.

ca /etc/openvpn/client/ca.crt

Utilizo el parámetro “ca” para indicar la ruta al certificado “ca.crt” de la “Autoridad de Certificación” (utilizada para firmar tanto el certificado del servidor como el del cliente VPN), el cual se utilizará en ambos extremos del túnel para garantizar la validez del certificado recibido por la máquina remota (su identidad).

cert /etc/openvpn/client/client-vpn.crt

Utilizo el parámetro “cert” para indicar la ruta al certificado “client-vpn.crt” de la máquina local (en este caso el cliente) que le será enviado a la máquina remota como parte del proceso de autenticación (para demostrar que es quien dice ser). Dicho certificado contiene la **clave pública del cliente.**

key /etc/openvpn/client/client-vpn.key

Utilizo el parámetro “key” para indicar la ruta a la clave privada “client-vpn.key” de la máquina local (en este caso el cliente), la cual es la pareja de la clave pública incluida en el certificado indicado en el parámetro anterior. Ambas serán utilizadas (como ya he indicado) en el proceso de autenticación de los extremos de la conexión VPN.

comp-lzo

Mediante este parámetro, activo la compresión “LZO” de los datos transmitidos a través del túnel VPN.

keepalive 20 100

Mediante el parámetro “keepalive” se pueden detectar caídas de la conexión con el otro extremo del túnel. Para ello indico en primer lugar, cada cuantos segundos se comprueba si se ha recibido algún paquete del otro extremo (en este caso 10s) y en el caso de no haber recibido ninguno se realizará un ping para comprobar si la máquina remota sigue activa. Si tras la primera comprobación transcurren los segundos indicados en segundo lugar (en este caso 100s) sin recibir paquetes, se intentará restablecer la conexión.

log /var/log/client-vpn.log

Utilizo el parámetro “log” para indicar la ruta del archivo de log “client-vpn.log” donde se registrarán los

eventos que OpenVPN genere durante el establecimiento de la conexión y el transcurso de la misma.

verb 3

Mediante el parámetro “verb” indico la **cantidad de eventos que se registran en el archivo de log,** que permite valores de 0 (ningún evento) a 11 (máximo) y cuyo valor establezco en 3 (recomendado).

Para terminar, guardo los cambios realizados y salgo del editor.

# 2 b.2) Certificado y claves en cliente

* 1. Creación de Clave Privada y Certificado para la Máquina Cliente

Para generar la clave privada de la Máquina Cliente OpenVPN, ejecuto el siguiente comando en el servidor:

openssl genrsa -out /etc/openvpn/client/client-vpn.key 2048

Al igual que para la “Máquina Servidor”, utilizo “openssl” para crear la clave privada “RSA”, lo cual se indica con la opción “genrsa” y cuyo archivo de salida “-out” será “client-vpn.key” que se ubicará en “/etc/openvpn/client”. En último lugar indico la longitud de la clave, que será de “2048” bits.

ls -l /etc/openvpn/client

Compruebo que se ha creado el archivo de la clave privada.

cat /etc/openvpn/client/client-vpn.key

Muestro su contenido para verificar que la clave ha sido creada correctamente.

Para generar el certificado de la Máquina Cliente OpenVPN, comienzo generando la solicitud de certificado ejecutando el siguiente comando:

openssl req -new -key /etc/openvpn/client/client-vpn.key -out /etc/openvpn/client/client- vpn.p10

Donde (utilizando “openssl”) indico que se cree una solicitud de certificado “req” nueva “-new”, para la clave privada “-key” que se almacena en el archivo “client-vpn.key” ubicada en “/etc/openvpn/client” cuya salida “-out” será el archivo llamado “client-vpn.p10” que se ubicará en “/etc/openvpn/client”. Para terminar, proporciono los datos solicitados (incluida una “passphrase”) los cuales serán incluidos en la solicitud de certificado.

ls -l /etc/openvpn/client

Compruebo que se ha creado el archivo de solicitud de certificado.

cat /etc/openvpn/client/client-vpn.p10

Muestro su contenido para verificar que la solicitud de certificado ha sido creada correctamente. Ahora debo firmar la solicitud de certificado ejecutando el siguiente comando:

openssl x509 -req -in /etc/openvpn/client/client-vpn.p10 -out /etc/openvpn/client/client- vpn.crt -CA ca.crt -CAkey ca.key -CAcreateserial -days 365

Donde (utilizando “openssl”) indico que deseo realizar una firma “x509” de una solicitud de certificado

“-req”, indicando el archivo de entrada “-in” que almacena dicha solicitud “client-vpn.p10” ubicado en “/etc/openvpn/client” y el archivo de salida “-out” que almacenará el certificado firmado “clientvpn.crt” que se ubicará en “/etc/openvpn/client”. A continuación, indico el certificado “-CA” de la “Autoridad Certificadora” (CA) que se utilizará para la firma “ca.crt”, seguida de su clave privada “CAkey” con la cual se firmará la solicitud “ca.key”. Para terminar, indico que se cree el archivo de número de serie de la CA si no existe “-CAcreateserial” y le otorgo una validez al certificado “-days” de “365” días (1 año).

ls -l /etc/openvpn/client

Compruebo que se ha creado el archivo del certificado.

cat /etc/openvpn/client/client-vpn.crt

Muestro su contenido para verificar que el certificado ha sido creado correctamente.

# 2 b.3) Ubicación correcta de los archivos en la máquina cliente

Copiar de Manera Segura los Archivos Necesarios para la Máquina Cliente a través de shared. Ahora, para copiar al cliente los archivos que necesita del servidor, hago lo siguiente:

cp */mnt/shared/*ca.crt /etc/openvpn/client

Copio el certificado de la “Autoridad Certificadora” (CA) creado anteriormente al directorio donde he

creado los archivos para el cliente.

chmod o+r /etc/openvpn/client/\*

Modifico los permisos de los archivos para que puedan ser leídos por usuarios distintos del propietario.

ls -l /etc/openvpn/client

Compruebo que los archivos y los permisos son correctos.

apt-get install openssh-server

Instalo el servidor SSH mediante el paquete “openssh-server”.

scp user1@10.5.1.10:/etc/openvpn/client/\* .

Ahora desde el cliente, utilizando el comando “scp” (secure copy) basado en SSH, accedo al equipo con IP “10.5.1.10” (IP del Servidor) utilizando el usuario “user1”, separando usuario y equipo mediante un símbolo “@” e indico la ruta de los archivos a copiar “/etc/openvpn/client/\*” (origen) separada por “:”, como destino indico el directorio actual “.”.

Dado que es la primera vez que accedo al servidor desde este equipo mediante SSH, me solicita añadir el host a la lista de hosts de confianza del cliente (lo acepto) y una vez autenticado mediante la contraseña del usuario “user1” se realiza la copia de los archivos a través de la red de manera segura (cifrando los datos copiados).

ls -l

Compruebo que los archivos necesarios han sido correctamente copiados.

Por último, compruebo que cuento con todos los archivos necesarios para establecer la conexión VPN desde la Máquina Cliente OpenVPN.

# 2 c) Comprobaciones

**2 c1) Comprobaciones en el servidor**

* 1. Comprobaciones desde la Máquina Servidor

Para realizar las comprobaciones comienzo levantando el servicio VPN en el servidor ejecutando:

openvpn /etc/openvpn/server/servervpn.conf

El comando “openvpn” iniciará el demonio y utilizará para ello los parámetros definidos en el archivo de configuración indicado “server-vpn.conf” (quedando a la espera de conexiones por parte de clientes VPN).

Desde el cliente establezco la conexión VPN con el servidor ejecutando:

openvpn /etc/openvpn/client/client-vpn.conf

Utilizará para ello los parámetros definidos en el archivo de configuración indicado “client-vpn.conf”.

Para realizar las comprobaciones, ejecuto los siguientes comandos:

Ifconfig

Mediante el comando “ifconfig”, verifico que se ha creado la interfaz virtual del túnel VPN “tun0”.

route

Mediante el comando “route”, muestro la tabla de rutas del servidor, donde se encuentra la ruta para el túnel VPN “tun0”.

tail /var/log/server-vpn.log

Mediante el comando “tail” muestro las últimas líneas del log “server-vpn.log” que en su última línea

indica que el túnel VPN se ha establecido correctamente.

ping -c 4 10.0.0.2

Para finalizar, realizo un ping al cliente VPN a través del túnel cuya IP virtual es “10.0.0.2”.

# c2) Comprobaciones en el cliente

* 1. Comprobaciones desde la Máquina Cliente

Para realizar las comprobaciones, ejecuto los siguientes comandos:

Ifconfig

Mediante el comando “ifconfig”, verifico que se ha creado la interfaz virtual del túnel VPN “tun0”.

route

Mediante el comando “route”, muestro la tabla de rutas del cliente, donde se encuentra la ruta para el túnel VPN “tun0”.

tail /var/log/client-vpn.log

Mediante el comando “tail” muestro las últimas líneas del log “client-vpn.log” que en su última línea indica

que el túnel VPN se ha establecido correctamente.

ping -c4 10.0.0.1

Para finalizar, realizo un ping al servidor VPN a través del túnel cuya IP virtual es “10.0.0.1”.

# 2 d) Prueba de concepto

**2 d1) Instalo servidor web en el servidor**

Para la última prueba debo ejecutar los siguientes comandos en el servidor:

apt-get install apache2

Instalo el servidor web “Apache” mediante el paquete “apache2”.

cd /var/www/html

Cambio el directorio de trabajo a la raíz del servidor web.

wget <https://web2.iesmiguelherrero.com/?p=1367>> index2.html

Mediante la herramienta “wget”, descargo en él la página de prueba.

ls -l

Por último, compruebo que se ha descargado correctamente.

cat *index2*.html

Comprobando también el código HTML que la forma.

Ahora, mediante el navegador del cliente, accedo a través del túnel VPN a la página “index2.html” alojada

en el servidor VPN. Para ello indico la IP virtual del servidor VPN (10.0.0.1) y el nombre de la página.