OpenVPN en mininet

Seguridad en Redes de Ordenadores

Departamento de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación (GSyC)

Marzo de 2018

Contenidos

1. Introducción al entorno mininet
2. Esquema openVPN, network-network
3. Configuración de openVPN con autenticación PKI
4. Inicio/interrupción del túnel
5. Comunicación a través del túnel
6. Referencias

Contenidos

1 Introducción al entorno mininet

2 Esquema openVPN, network-network

3 Configuración de openVPN con autenticación PKI

4 Inicio/interrupción del túnel

5 Comunicación a través del túnel

6 Referencias

Introducción a mininet

- Mininet es un emulador de red que permite arrancar diferentes topologías que involucren gran cantidad de máquinas con mecanismos ed virtualización ligera.
- Basa su funcionamiento la definición de diferentes *network namespaces* que permiten definir de forma independiente diferentes interfaces de red virtuales en una máquina, con diferentes tablas de encaminamiento. Estas interfaces de red virtuales se conectan entre ellas usando openvSwitch, un software que permite definir el comportamiento de un dispositivo switch al que se conectan las interfaces virtuales de una máquina para reenviarse tráfico entre ellas.
- Las topologías o escenarios de red en Mininet se programan en python.
- Para realizar las prácticas, se os proporcionarán los scripts de python para arrancar el escenario.

Ejemplo de ejecución de mininet

- Ejemplo, arranque de un escenario con 3 hosts (h1, h2, y3) y un switch (s1) al que se conectan las 3 máquinas:

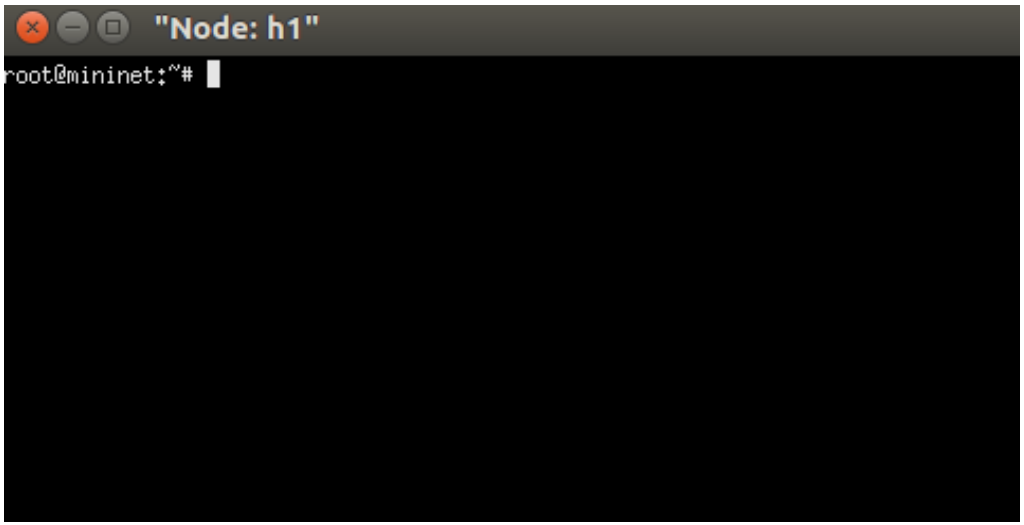
```
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller

*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

Command Line Interface en mininet

- Mininet ofrece una interfaz de línea de comandos (CLI, Command Line Interface) que permite ejecutar comandos propios de mininet y el arranque de terminales asociados a cada una de las máquinas.
- Los terminales de las máquinas nos permitirán ejecutar comandos en las mismas el. Para ello, desde el prompt mininet> podemos lanzar un terminal por ejemplo en h1:

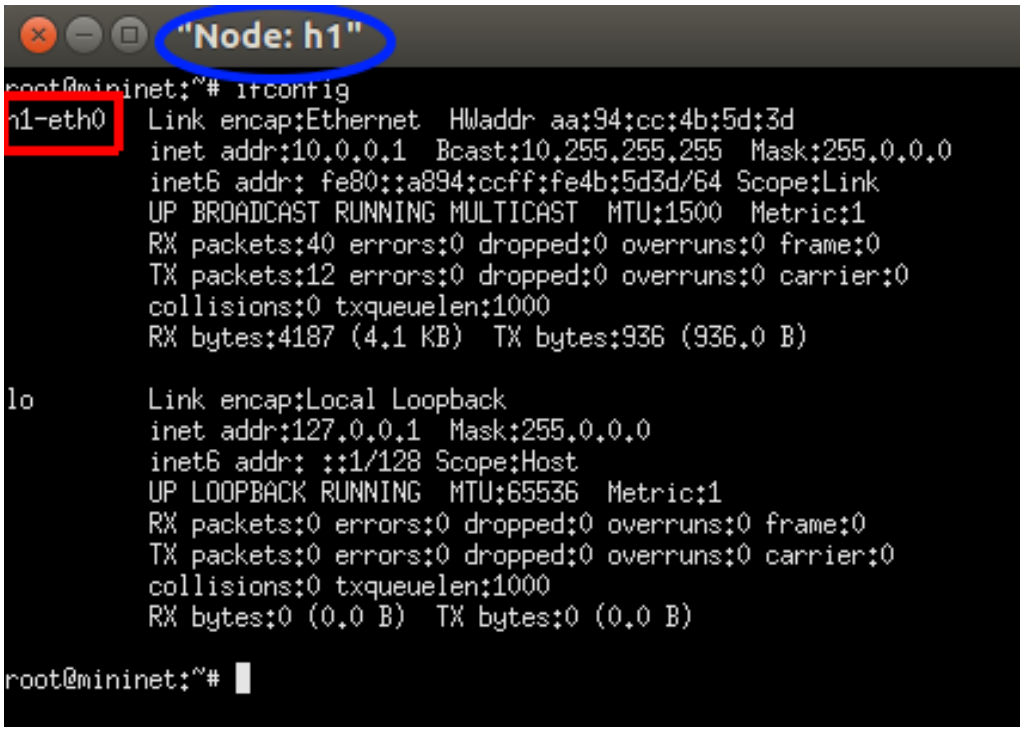
```
mininet> xterm h1
```



- Este comando abrirá una nueva ventana de terminal asociada a la máquina h1 (véase el título de la ventana de terminal).

Terminales de máquinas en mininet

- Al arrancar los terminales de cada máquina, observarás que el sistema de ficheros de cada una de las máquinas es el mismo que el del sistema operativo anfitrión (en nuestro caso el de la máquina virtual ubuntu 16.04 que está instalado en virtualBox).
- El título del terminal indica a qué network namespace pertenece dicho terminal.
- La configuración de la red es independiente para cada máquina y las interfaces creadas se denominan `<nombreMáquina-eth0>`, `<nombreMáquina-eth1>`, etc.



Un único sistema de ficheros en mininet

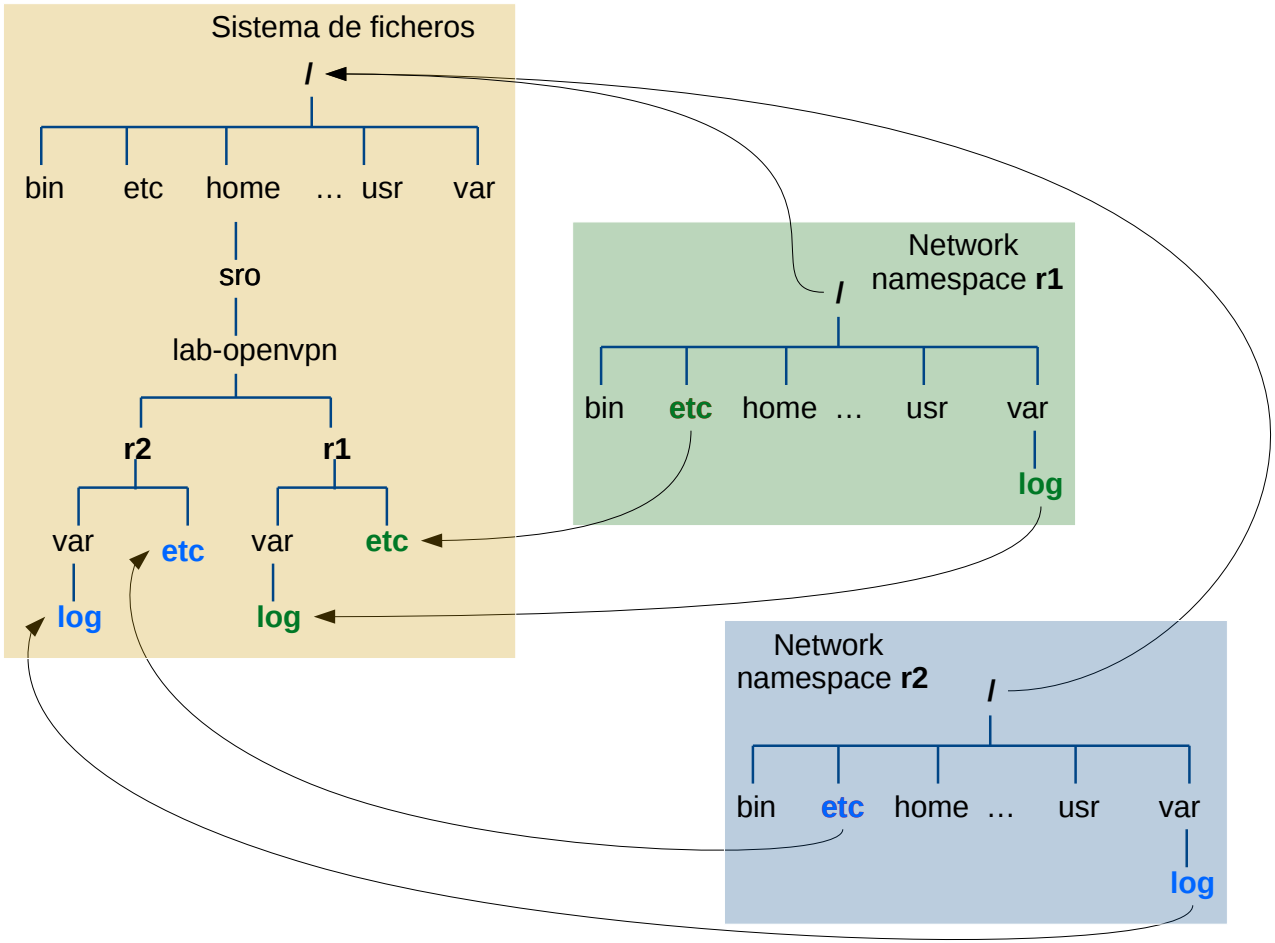
- El sistema de ficheros que ven las máquinas arrancadas en mininet es el mismo que la máquina anfitriona.
- Sin embargo, se puede configurar para que ciertas **carpetas sean de uso privado para cada uno de los network namespaces** creados.
- En particular, se desea que la carpeta `/etc` sea independiente para cada máquina, ya que es allí donde residirán los ficheros de configuración. Mininet permite hacerlo montando dentro de cada network namespace una carpeta diferente que quedará situada dentro de cada máquina en la carpeta `/etc`.

Carpetas independientes para cada network namespace

- En el entorno de prácticas, se ha configurado mininet para que la carpeta `/etc` de cada una de las máquinas en realidad contenga lo que se encuentra dentro de la carpeta del escenario, por ejemplo:

  `~/lab-openvpn/r1/etc`.

- Por tanto, los ficheros de configuración de cada máquina residirán en carpetas diferentes, pero que desde el punto de vista del network namespace creado se verá dentro de la carpeta `/etc`.



Introducción

- OpenVPN se puede usar con TCP o con UDP. El puerto que se suele usar es 1194.
- Los mecanismos de autenticación que utiliza openVPN pueden ser pre-shared-key y PKI.
- Se permiten túneles a nivel de red y a nivel de enlace.

Contenidos

1 Introducción al entorno mininet

2 Esquema openVPN, network-network

3 Configuración de openVPN con autenticación PKI

4 Inicio/interrupción del túnel

5 Comunicación a través del túnel

6 Referencias

Esquema openVPN, network-network

- Este tipo de configuración conecta a través de una red pública (insegura) 2 redes internas, por ejemplo de 2 sucursales de una misma empresa, que pueden tener direccionamiento privado.
- Un extremo será el servidor, por ejemplo `r1`, y el otro extremo actuará de cliente, por ejemplo `r2`.
- Al arrancar el servidor OpenVPN:
  - El servidor esperará las peticiones de los clientes en su **dirección IP de su interfaz pública** 100.0.0.1, puerto 1198 UDP.
  - El servidor configurará una **interfaz tun** asignando a dicha interfaz la primera dirección IP libre del rango privado de la VPN (ejemplo 10.8.0.1) y una ruta para alcanzar el rango de subred de la VPN a través de la interfaz tun.

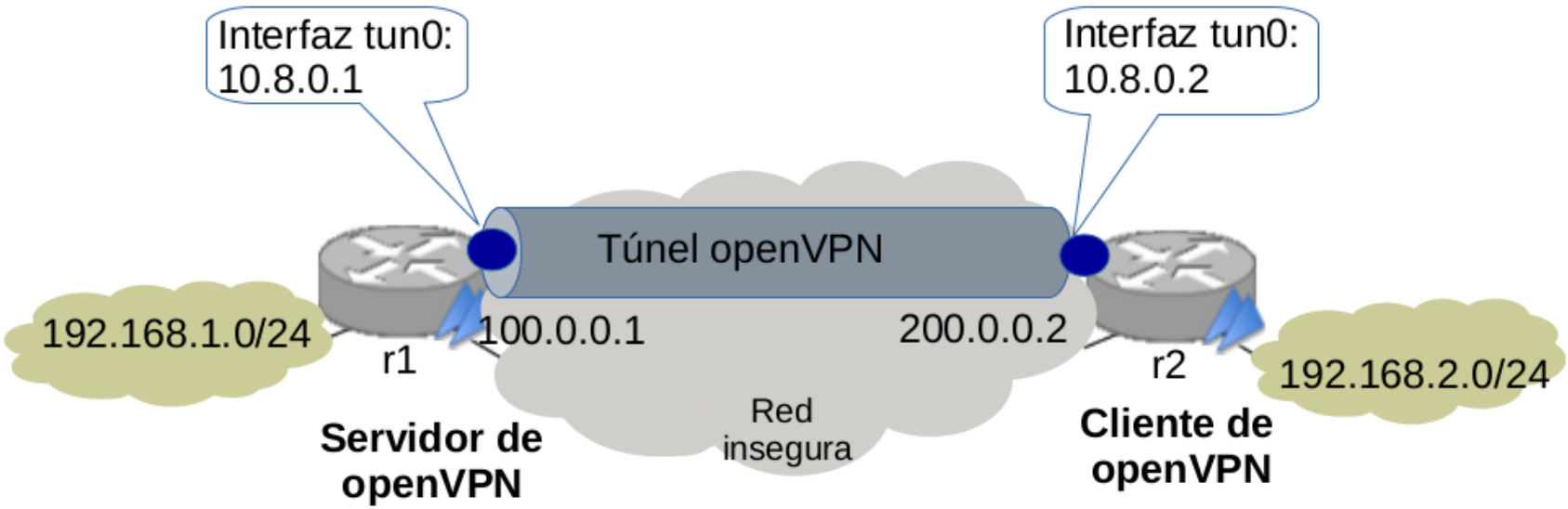
```
Destination     Gateway     Genmask          Iface
10.8.0.0        0.0.0.0     255.255.255.0    tun0
```

Esquema openVPN, network-network: túnel

- Ante la solicitud de un cliente, el servidor configurará una VPN por la que transmitirá los paquetes cifrados según los parámetros acordados en TLS.
  - El servidor asigna una dirección IP del rango privado (10.8.0.0/24) de la VPN al cliente.
  - El cliente configura **interfaz tun** con la dirección IP que le asigna el servidor y una ruta para alcanzar el rango de subred de la VPN a través de la interfaz tun.

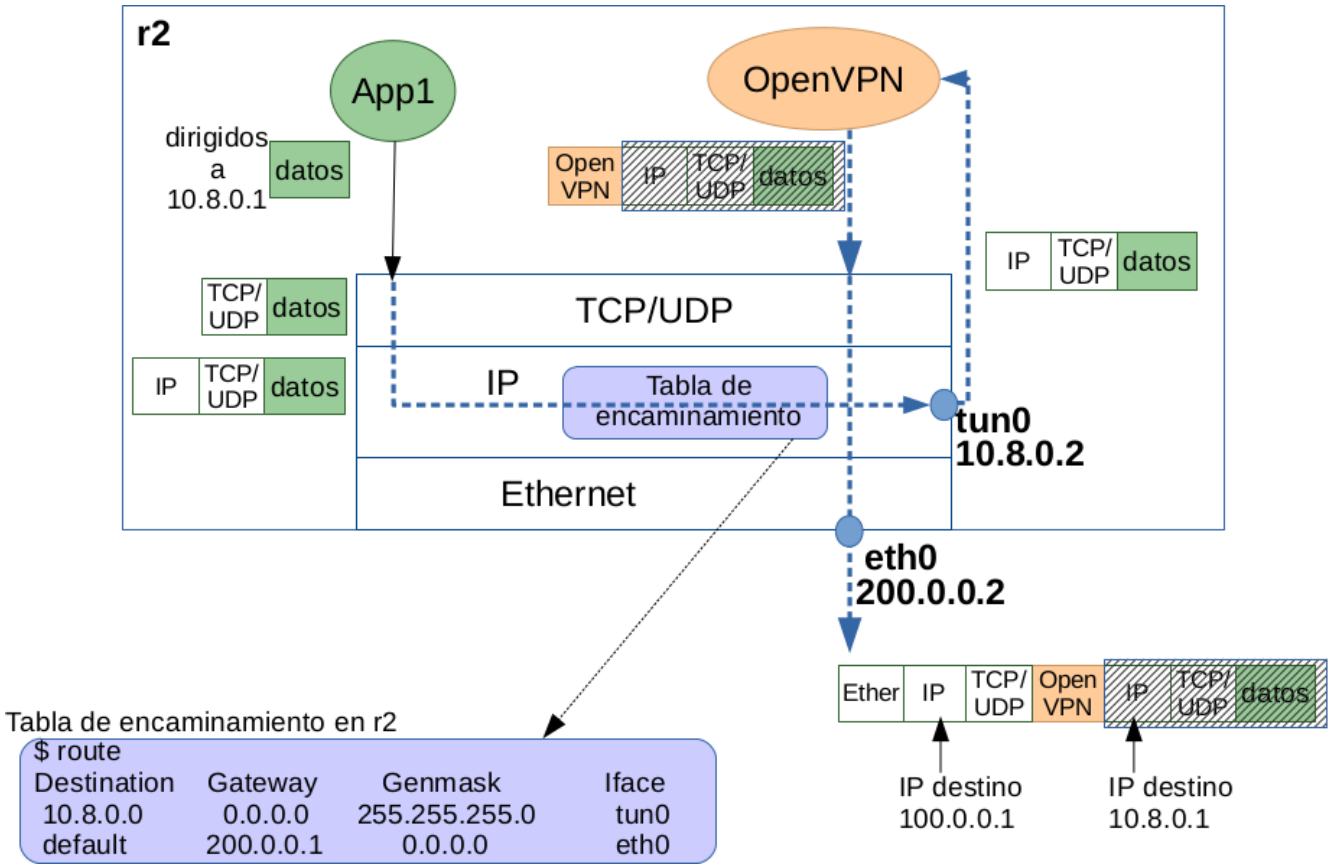
```
Destination     Gateway     Genmask         Iface
10.8.0.0        0.0.0.0     255.255.255.0   tun0
```

- Los paquetes que se transmitan a través de la VPN llevarán **2 cabeceras IP**, la interna con direcciones del rango privado que haya configurado el servidor OpenVPN (en este caso las direcciones del rango 10.8.0.0/24) y en la cabecera externa las direcciones IP públicas (en este caso 100.0.0.1 y 200.0.0.2).



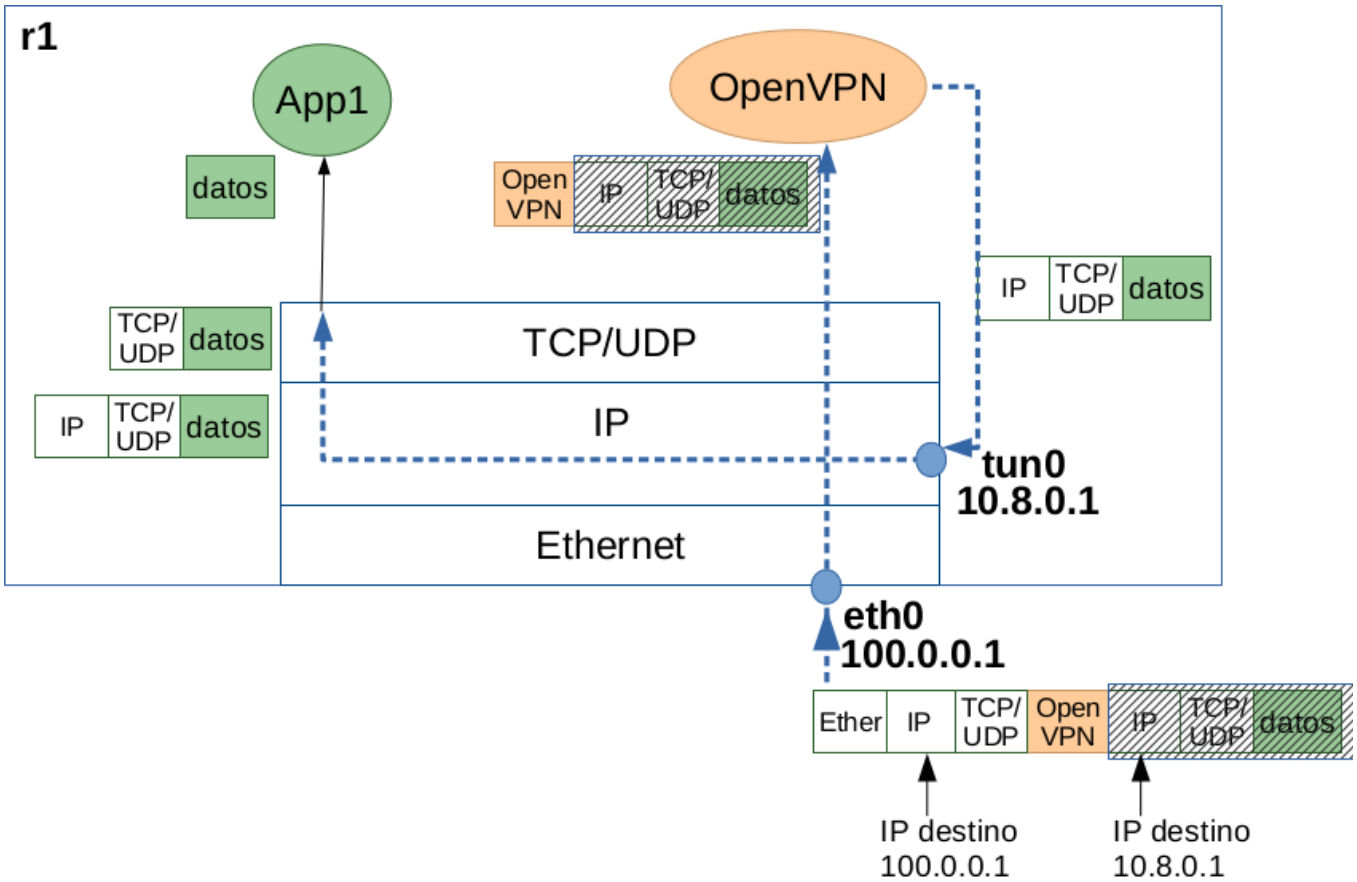
Envío a través de un túnel
Datos generados por una aplicación en el extremo del túnel

- Se utiliza el direccionamiento asignado para la VPN para enviar paquetes a través de dicho túnel.
- Ejemplo: r2 puede enviar datos a través del túnel usando la dirección de r1: 10.8.0.1.



Recepción a través de un túnel
Datos dirigidos a una aplicación en el extremo del túnel

- En recepción, r1 desencapsula los datos y extrae el datagrama IP original, que va dirigido al propio r1.



Contenidos

1 Introducción al entorno mininet

2 Esquema openVPN, network-network

3 Configuración de openVPN con autenticación PKI

4 Inicio/interrupción del túnel

5 Comunicación a través del túnel

6 Referencias

Contenidos

1. Introducción al entorno mininet
2. Esquema openVPN, network-network
3. Configuración de openVPN con autenticación PKI
   - Creación de certificados
   - Configuración del servidor
   - Configuración del cliente
4. Inicio/interrupción del túnel
5. Comunicación a través del túnel
   - Comunicación entre los extremos del túnel
   - Comunicación del extremo cliente con las subredes internas del servidor
   - Comunicación del extremo servidor con las subredes internas del cliente
   - Comunicación de las subredes internas de un cliente con las subredes internas de otro cliente
6. Referencias

Creación de certificados

- Tanto el lado cliente como el lado servidor autenticarán al otro extremo a través de un certificado que esté firmado por una autoridad de certificación (CA) reconocida por ambos.
- Utilizaremos el estándar X.509 para la creación de certificados.
- En las prácticas crearemos las claves para:
  - CA: clave y certificado autofirmado.
  - cliente: clave y certificado firmado por CA.
  - servidor: clave y certificado firmado por CA.
  - Parámetros Diffie-Hellman para el servidor.

Creación de claves y certificados con las herramientas easy-rsa

- Normalmente las claves se crean en cada una de las entiddades (servidor y clientes) involucradas en la comunicación y la CA firma los certificados de las entidades.
- Para crear estas claves y certificados usaremos las herramientas `easy-rsa` 3.0 que son un recumbrimiento de las herramientas `openssl` para la creación de claves y certificados de forma sencilla, por ejemplo para su uso en openvpn. Las herramientas `easy-rsa` se encuentran disponibles en https://github.com/OpenVPN/easy-rsa.
- Como no dispondremos de máquinas diferentes para crear las claves de cada una de las entidades, tendremos que **generarlas en directorios diferentes para simular la creación en diferentes máquinas**.

Creación de clave y certificado de CA

- Desde una carpeta en la que hay una copia de las herramientas `easy-rsa`, por ejemplo `easy-rsa-CA`, entramos en la carpeta `easyrsa3` y ejecutamos los siguientes comandos:

```
./easyrsa init-pki
./easyrsa build-ca
```

  Nos solicitará que introduzcamos una `Key Passphrase` que se utilizará para almacenar la clave privada de la CA de forma cifrada. Necesitarás recordarla para firmar certificados. El valor de `Common Name` podéis dejar el que se genera por defecto.

  Como resultado se habrá generado:

  - el certificado de la autoridad de certificación: `easy-rsa-CA/pki/ca.crt` (su certificado autofirmado)
  - la clave privada de la autoridad de certificación: `easy-rsa-CA/pki/private/ca.key`.

- Podemos ver el contenido del certificado:

```
openssl x509 -in <nombreDelFicheroConCertificado> -text
```

Creación de clave y solicitud de certificado para el servidor

- Desde una carpeta en la que hay una copia de las herramientas `easy-rsa`, por ejemplo `easy-rsa-servidor`, ejecutamos los siguientes comandos:

  ```
  ./easyrsa init-pki
  ./easyrsa gen-req <nombreMáqServidor>
  ```

  Nos solicitará que introduzcamos una pas phrase que se utilizará para almacenar la clave privada de la máquina de forma cifrada. Necesitarás recordarla para usarla en la autenticación del túnel.

  Como resultado se habrá generado:

  - solicitud de certificado para ser firmada por una autoridad de certificación: `easy-rsa-servidor/pki/reqs/<nombreMáqServidor>.req`
  - la clave privada del servidor: `easy-rsa-servidor/pki/private/<nombreMáqServidor.key`.

- Para ver el contenido de la solicitud de certificado:

  ```
  openssl req -noout -text -in pki/reqs/<nombreMáqServidor>.req
  ```

Firma del certificado del servidor por CA

- Desde la carpeta donde se han generado las claves de CA, se importa el certificado del servidor:

  ```
  ./easyrsa import-req <carpeta/nombreMáqServidor>.req <nombreMáqServidor>
  ```

- La solicitud de certificado se puede comprobar que se ha importado correctamente usando el comando:

  ```
  ./easyrsa show-req <nombreMáqServidor>
  ```

- La CA firma el certificado importado con el `<nombreMáqServidor>`. Te solicitará confirmación para realizar la operación que debes responder con 'yes' y a continuación te solicitará la `Key Passphrase` de la CA para poder usar su clave privada:

  ```
  ./easyrsa sign-req server <nombreMáqServidor>
  ```

  Como resultado se habrá generado:

  - el certificado del servidor firmado por la autoridad de certificación:
    `easy-rsa-CA/pki/issued/<nombreMáqServidor>.crt`

Creación de clave y solicitud de certificado para el cliente

- Desde una carpeta en la que hay una copia de las herramientas `easy-rsa`, por ejemplo `easy-rsa-cliente`, ejecutamos los siguientes comandos:

  ```
  ./easyrsa init-pki
  ./easyrsa gen-req <nombreMáqCliente>
  ```

  Como resultado se habrá generado:

  - solicitud de certificado para ser firmada por una autoridad de certificación:
    `easy-rsa-cliente/pki/req/<nombreMáqCliente>.req`
  - la clave privada del servidor:
    `easy-rsa-cliente/pki/private/<nombreMáquinaCliente.key`.

- Para ver el contenido de la solicitud de certificado:

  ```
  openssl req -noout -text -in pki/req/<nombreMáqCliente>.req
  ```

Firma del certificado del cliente por CA

- Desde la carpeta donde se han generado las claves de CA, se importa el certificado del cliente:

  ```
  ./easyrsa import-req <carpeta/nombreMáqCliente>.req <nombreMáqCliente>
  ```

- La CA firma el certificado importado con el `<nombreMáqCliente>`. Te solicitará confirmación para realizar la operación que debes responder con 'yes' y a continuación te solicitará la `Key Passphrase` de la CA para poder usar su clave privada::

  ```
  ./easyrsa sign-req client <nombreMáqCliente>
  ```

  Como resultado se habrá generado:

  - el certificado del cliente firmado por la autoridad de certificación:
    `easy-rsa-CA/pki/issued/<nombreMáqCliente>.crt`

Generación de los parámetros Diffie-Hellman

- Desde la carpeta en la que hay una copia de las herramientas `easy-rsa` para el servidor, por ejemplo `easy-rsa-servidor`, ejecutamos los siguientes comandos:

  ```
  ./easyrsa gen-dh
  ```

  Como resultado se habrá generado:

  - los parámetros Diffie-Hellman: `easy-rsa-servidor/pki/dh.pem`. El número primo **p** de 2048 bits y el generador **g**.

    El contenido de este fichero se puede visualizar de la siguiente forma:

    ```
    openssl dhparam -inform PEM -in ./pki/dh.pem -noout -text
    ```

Ficheros creados para claves, certificados y DH

- Se han generado los siguientes ficheros importantes:
  - `ca.crt`: certificado raíz autofirmado de la CA.
  - `ca.key`: clave privada de la CA.
  - `dh.pem`: parámetros para el intercambio Diffie-Hellman
  - `nombreMáqServidor.crt`: certificado del servidor firmado por CA
  - `nombreMáqServidor.key`: clave privada del servidor
  - `nombreMáqCliente.crt`: certificado del cliente firmado por CA
  - `nombreMáqCliente.key`: clave privada del cliente.
- Para la configuración de `openvpn` se necesita que cada máquina tenga acceso a los ficheros que ella necesita:
  - La máquina servidor deberá tener su clave privada, su certificado, el certificado de CA y los parámetros Diffie-Hellman.
  - La máquina cliente deberá tener su clave privada, su certificado y el certificado de CA.
- Para el escenario de mininet, cada máquina guardará estos ficheros en la carpeta correspondiente a su nombre de máquina y una vez allí, en `etc/openvpn`.

Contenidos

1. Introducción al entorno mininet
2. Esquema openVPN, network-network
3. Configuración de openVPN con autenticación PKI
   - Creación de certificados
   - Configuración del servidor
   - Configuración del cliente
4. Inicio/interrupción del túnel
5. Comunicación a través del túnel
   - Comunicación entre los extremos del túnel
   - Comunicación del extremo cliente con las subredes internas del servidor
   - Comunicación del extremo servidor con las subredes internas del cliente
   - Comunicación de las subredes internas de un cliente con las subredes internas de otro cliente
6. Referencias

Configuración del servidor (I)

- El servidor se configura a través del fichero `server.conf`.
- Los comentarios en el fichero son las líneas que empiezan por # o por ;
- Nos fijaremos en que están presentes al menos las siguientes líneas (los comentarios del fichero explican su significado):
  1. Puerto donde escuchará el servidor VPN, protocolo y VPN en modo `tun` a nivel 3:

     ```
     port 1194
     proto udp
     dev tun
     ```

  2. Información de los certificados: CA y servidor. Clave privada del servidor y fichero donde se encuentra la configuración para Diffie Hellman:

     ```
     ca /etc/openvpn/ca.crt
     cert /etc/openvpn/<nombreMáqServidor>.crt
     key /etc/openvpn/<nombreMáqServidor>.key
     dh /etc/openvpn/dh.pem
     ```

     Recuerda que la carpeta `/etc` de cada network namespace se corresponde con `/home/sro/lab-openvpn/<máq>/etc`. Los cambios que realices en la carpeta `/etc` de cada máquina quedarán guardados en el sistema de ficheros principal: `/home/sro/lab-openvpn/<máq>/etc`.

Configuración del servidor (II)

- Continuación de la configuración de `server.conf`:
  3. Subred definida para las máquinas que se comunican a través de la VPN. Todos los clientes tendrán una dirección IP dentro de esta subred, por ejemplo 10.8.0.0/24. El servidor tendrá configurada la primera dirección: 10.8.0.1.

     ```
     server 10.8.0.0 255.255.255.0
     ```

  4. El servidor almacenará en el fichero `ipp.txt` la asociación entre las direcciones 10.8.0.0/24 y los clientes.

     ```
     ifconfig-pool-persist /etc/openvpn/ipp.txt
     ```

  5. Se envían mensajes para comprobar si el otro extremo sigue activo, cada 10 segundos. Si no se reciben durante 120 segundos se supone que el extremo está inactivo

     ```
     keepalive 10 120
     ```

  6. Ficheros de log y nivel de depuración.

     ```
     status /var/log/openvpn-status.log

     log /var/log/openvpn.log
     verb 5
     ```

     Recuerda que la carpeta `/var/log` de cada network namespace se corresponde con `/home/sro/lab-openvpn/<máq>/var/log`.

Contenidos

1. Introducción al entorno mininet
2. Esquema openVPN, network-network
3. Configuración de openVPN con autenticación PKI
   - Creación de certificados
   - Configuración del servidor
   - Configuración del cliente
4. Inicio/interrupción del túnel
5. Comunicación a través del túnel
   - Comunicación entre los extremos del túnel
   - Comunicación del extremo cliente con las subredes internas del servidor
   - Comunicación del extremo servidor con las subredes internas del cliente
   - Comunicación de las subredes internas de un cliente con las subredes internas de otro cliente
6. Referencias

Configuración del cliente

- El cliente se configura a través del fichero `client.conf`.
- Nos fijaremos en que están presentes al menos las siguientes líneas, sin comentar:

1. Es un cliente que se va a conectar con el servidor que está configurado en <dirIPServidor> (su dirección IP pública) y puerto 1194 a través de UDP

```
client
dev tun
remote <dirIPServidor> 1194
proto udp
```

2. Información de los certficados: CA y servidor. Clave privada del servidor y fichero donde se encuentra la configuración para Diffie Hellman:

```
ca /etc/openvpn/ca.crt
cert /etc/openvpn/<nombreMáqCliente>.crt
key /etc/openvpn/<nombreMáqCliente>.key
```

Recuerda que la carpeta `/etc` de cada network namespace se corresponde con `/home/sro/lab-openvpn/<máq>/etc`. Los cambios que realices en la carpeta `/etc` de cada máquina quedarán guardados en el sistema de ficheros principal: `/home/sro/lab-openvpn/<máq>/etc`.

Contenidos

1 Introducción al entorno mininet

2 Esquema openVPN, network-network

3 Configuración de openVPN con autenticación PKI

4 Inicio/interrupción del túnel

5 Comunicación a través del túnel

6 Referencias

Inicio/interrupción del túnel

- En el lado servidor es necesario ejecutar en el terminal del servidor:

```
openvpn /etc/openvpn/server.conf
```

- En el lado cliente es necesario ejecutar en el terminal del cliente:

```
openvpn /etc/openvpn/client.conf
```

- Para modificar la configuración, será necesario interrumpir la ejecución y después volver a iniciarlo. Para interrumpir la ejecución desde el terminal de la máquina en la que se desea interrumpir se pulsa `Ctrl+C`.

Contenidos

1 Introducción al entorno mininet

2 Esquema openVPN, network-network

3 Configuración de openVPN con autenticación PKI

4 Inicio/interrupción del túnel

5 Comunicación a través del túnel

6 Referencias

Contenidos

1. Introducción al entorno mininet
2. Esquema openVPN, network-network
3. Configuración de openVPN con autenticación PKI
   - Creación de certificados
   - Configuración del servidor
   - Configuración del cliente
4. Inicio/interrupción del túnel
5. Comunicación a través del túnel
   - Comunicación entre los extremos del túnel
   - Comunicación del extremo cliente con las subredes internas del servidor
   - Comunicación del extremo servidor con las subredes internas del cliente
   - Comunicación de las subredes internas de un cliente con las subredes internas de otro cliente
6. Referencias

Comunicación entre los extremos del túnel

- Una vez iniciado el túnel, puedes observar como en cada extremo se ha configurado una interfaz `tun0` a la que se le asignado una dirección IP local dentro de la subred 10.8.0.0/24.
- El servidor OpenVPN tendrá asignada la primera dirección IP de ese rango, 10.8.0.1.
- Usaremos la **topología de subred** para definir las interfaces de los clientes en openVPN, lo que significa que cada vez que se conecte un cliente, el servidor le asignará una dirección IP del rango definido 10.8.0.0/24.
- La tabla de encaminamiento cambia tanto en el cliente como en el servidor ya que ahora, las direcciones IP correspondientes a la subred del túnel, 10.8.0.0/24, se alcanzarán a través de la interfaz `tun0`.

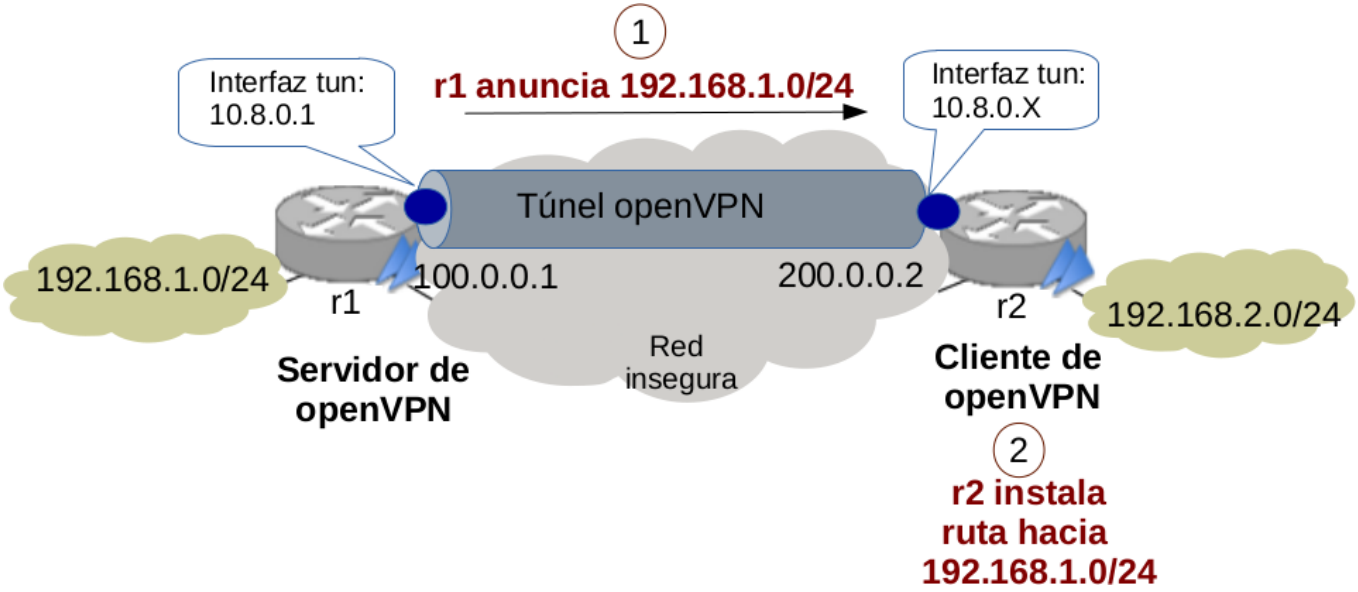
```
Destination     Gateway     Genmask          Iface
10.8.0.0        0.0.0.0     255.255.255.0    tun0
```

- El servidor guardará en el fichero `ipp.txt` la asociación entre nombre de máquina y dirección IP asignada para el túnel (dentro del rango configurado, por ejemplo 10.8.0.0/24).

Contenidos

1. Introducción al entorno mininet
2. Esquema openVPN, network-network
3. Configuración de openVPN con autenticación PKI
   - Creación de certificados
   - Configuración del servidor
   - Configuración del cliente
4. Inicio/interrupción del túnel
5. Comunicación a través del túnel
   - Comunicación entre los extremos del túnel
   - Comunicación del extremo cliente con las subredes internas del servidor
   - Comunicación del extremo servidor con las subredes internas del cliente
   - Comunicación de las subredes internas de un cliente con las subredes internas de otro cliente
6. Referencias

El servidor anuncia sus subredes internas (192.168.1.0/24)



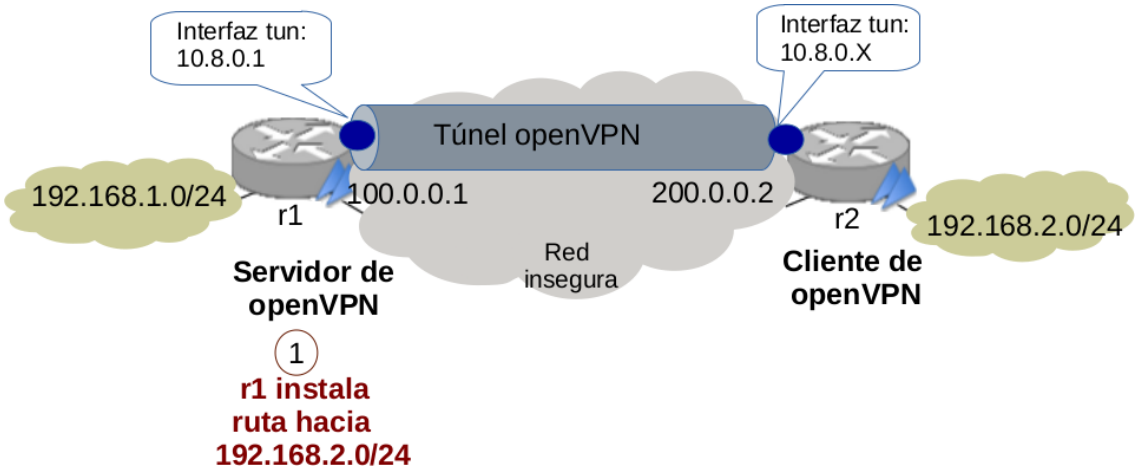
- Con la VPN que se ha creado, `r2` puede enviar paquetes a `r1` (10.8.0.1) utilizando el túnel. Sin embargo, si `r2` quisiera enviar paquetes a las subredes que se encuentran detrás de `r1` (192.168.1.0/24) no podría hacerlo.
- Para que el servidor anuncie rutas a los clientes a través del túnel, el servidor debe incluir en su fichero de configuración `server.conf` la siguiente información que incluye la subred interna que se desea alcanzar a través del túnel y máscara de la misma:

```
push "route 192.168.1.0 255.255.255.0"
```

Contenidos

1. Introducción al entorno mininet
2. Esquema openVPN, network-network
3. Configuración de openVPN con autenticación PKI
   - Creación de certificados
   - Configuración del servidor
   - Configuración del cliente
4. Inicio/interrupción del túnel
5. Comunicación a través del túnel
   - Comunicación entre los extremos del túnel
   - Comunicación del extremo cliente con las subredes internas del servidor
   - Comunicación del extremo servidor con las subredes internas del cliente
   - Comunicación de las subredes internas de un cliente con las subredes internas de otro cliente
6. Referencias

Rutas instaladas en el lado servidor para alcanzar las subredes internas del cliente (192.168.2.0/24)



- Con la VPN que se ha creado, `r1` puede enviar paquetes a `r2` (10.8.0.X) utilizando el túnel. Sin embargo, si `r1` quisiera enviar paquetes a las subredes que se encuentran detrás de `r2` (192.168.2.0/24) no podría hacerlo.
- El servidor debe incluir en su fichero `server.conf`:

```
client-config-dir /etc/openvpn/ccd
route 192.168.2.0 255.255.255.0
```

- Además el servidor debe crear en la carpeta `/etc/openvpn/ccd` un fichero con el nombre del cliente (`r2`). El contenido del fichero debe ser el siguiente para que el servidor openVPN encamine el tráfico a dicha subred a través del cliente `r2`:

```
iroute 192.168.2.0 255.255.255.0
```

- Parece que la línea `route` e `iroute` son redundantes, sin embargo, son necesarias ya que `route` controla la tabla de encaminamiento de la máquina a través de la interfaz `tun0` e `iroute` controla el encaminamiento desde el servidor openVPN a los clientes.

Contenidos

1. Introducción al entorno mininet
2. Esquema openVPN, network-network
3. Configuración de openVPN con autenticación PKI
   - Creación de certificados
   - Configuración del servidor
   - Configuración del cliente
4. Inicio/interrupción del túnel
5. Comunicación a través del túnel
   - Comunicación entre los extremos del túnel
   - Comunicación del extremo cliente con las subredes internas del servidor
   - Comunicación del extremo servidor con las subredes internas del cliente
   - Comunicación de las subredes internas de un cliente con las subredes internas de otro cliente
6. Referencias

Encaminamiento entre subredes de clientes

- Para permitir que las subredes internas de un cliente sean alcanzables desde otro cliente a través del servidor openVPN, es necesario definir en el fichero `server.conf`:

```
client-to-client
push "route 192.168.2.0 255.255.255.0"
```

- De esta forma el servidor anunciará las subredes 192.168.2.0/24 a otros clientes que se conecten a este mismo servidor.

Contenidos

1 Introducción al entorno mininet

2 Esquema openVPN, network-network

3 Configuración de openVPN con autenticación PKI

4 Inicio/interrupción del túnel

5 Comunicación a través del túnel

6 Referencias

Referencias

- Ejemplos de configuración: https://openvpn.net/index.php/open-source/documentation/examples.html