Installare e configurare OpenVPN: due scenari

Laboratorio di Amministrazione di Sistemi T

Massimiliano Mattetti - Marco Prandini

Installazione (obsoleto per VM Debian 8.7)

- Sulle VM del corso servono i pacchetti software
  - liblzo2
  - libpkcs11-helper
  - openssl-blacklist
  - openvpn
  - openvpn-blacklist
- Normalmente `apt-get install openvpn` scarica tutti i pacchetti e li installa
- Se non c'è accesso a Internet (in Lab)
  - scaricate sull'host il file `openvpn.tgz` dal sito del corso
  - copiatelo sulle VM Client e Router nella home di las
  - entrate sulle VM come root ed eseguite

```
tar xvfz ~las/openvpn.tgz
dpkg -i openvpn/*
```

Static key vs. SSL/TLS

La modalità “static key” di OpenVPN è la più semplice da abilitare:

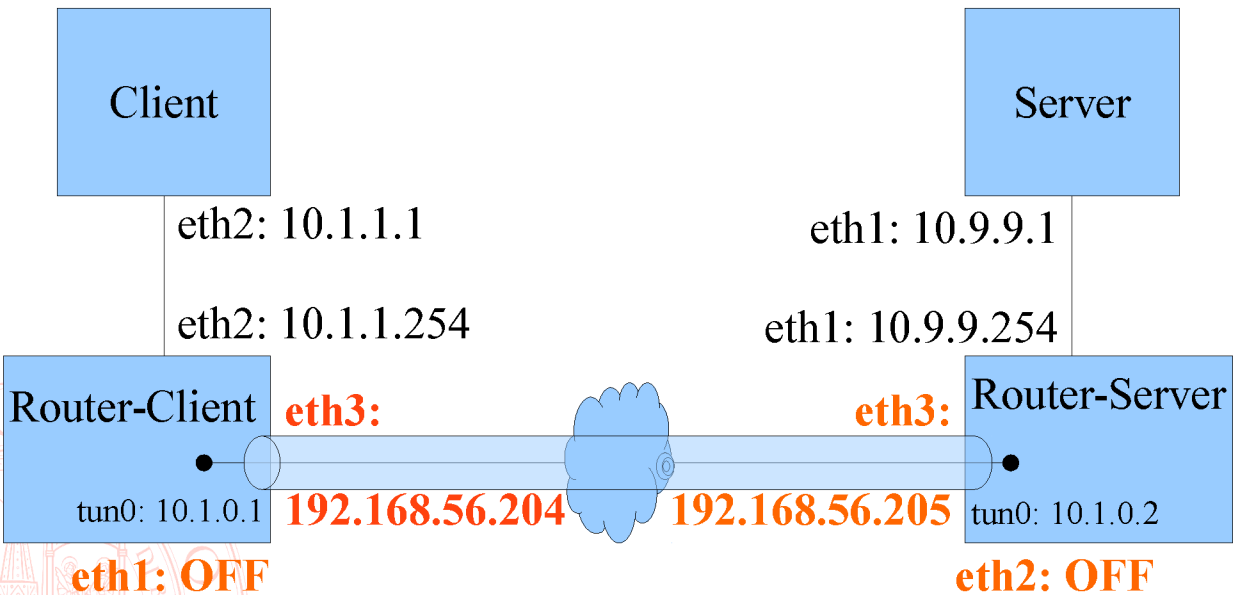
- unica chiave di cifratura simmetrica condivisa fra Client e Server VPN
- non è possibile autenticare gli utenti

La modalità SSL/TLS:

- richiede la generazione di una coppia di chiavi e dei relativi certificati per la mutua autenticazione di Client e Server

Site-to-site: predisposizione delle macchine

- Simuliamo una rete che collega due siti remoti:



Configurazione con chiave condivisa

- Spegniamo la VM Router e la cloniamo cambiando i MAC su una nuova VM Router-Client.
- Accendiamo Router-Client
- Modifichiamo il file `/etc/network/interfaces` per
  - dare a eth3 l'indirizzo 192.168.56.204
  - disabilitare eth1

- Eseguiamo le operazioni che serviranno su entrambe le copie
- Creiamo la chiave condivisa

```
cd /etc/openvpn
sudo openvpn --genkey --secret static.key
sudo chmod 600 /etc/openvpn/static.key
```

Configurazione con chiave condivisa

- Come utente root, creiamo con un editor il file di configurazione

  `/etc/openvpn/server.conf`

  contenente queste direttive:

```
dev tun
local 192.168.56.204
ifconfig 10.1.0.1 10.1.0.2
secret static.key
script-security 3
up ./route.up
verb 3
```

Configurazione con chiave condivisa

- Come utente root, creiamo con un editor il file `/etc/openvpn/route.up`
  contenente:

```
#!/bin/bash
route add -net 10.9.9.0 netmask 255.255.255.0 gw 10.1.0.2
```

  e lo rendiamo eseguibile con
  `sudo chmod +x /etc/openvpn/route.up`

- Shutdown della macchina

Clonazione e personalizzazione

- Spegniamo la VM Router-Client e la cloniamo su Router-Server cambiando i MAC
- Accendiamo Router-Server
- Modifichiamo
  - il file `/etc/network/interfaces` per
    - dare a eth3 l'indirizzo 192.168.56.205
    - riattivare eth1
    - disabilitare eth2
  - il file `/etc/openvpn/server.conf`
    - lo rinominiamo `client.conf`
    - sostituiamo la keyword `local` con `remote`
    - invertiamo gli indirizzi di `ifconfig`
  - il file `/etc/openvpn/route.up`:

```
route add -net 10.1.1.0 netmask 255.255.255.0 gw 10.1.0.1
```

Avvio e test

- Riavviamo Router-Server
- Avviamo Router-Client
- Avviamo il servizio su entrambe le macchine con

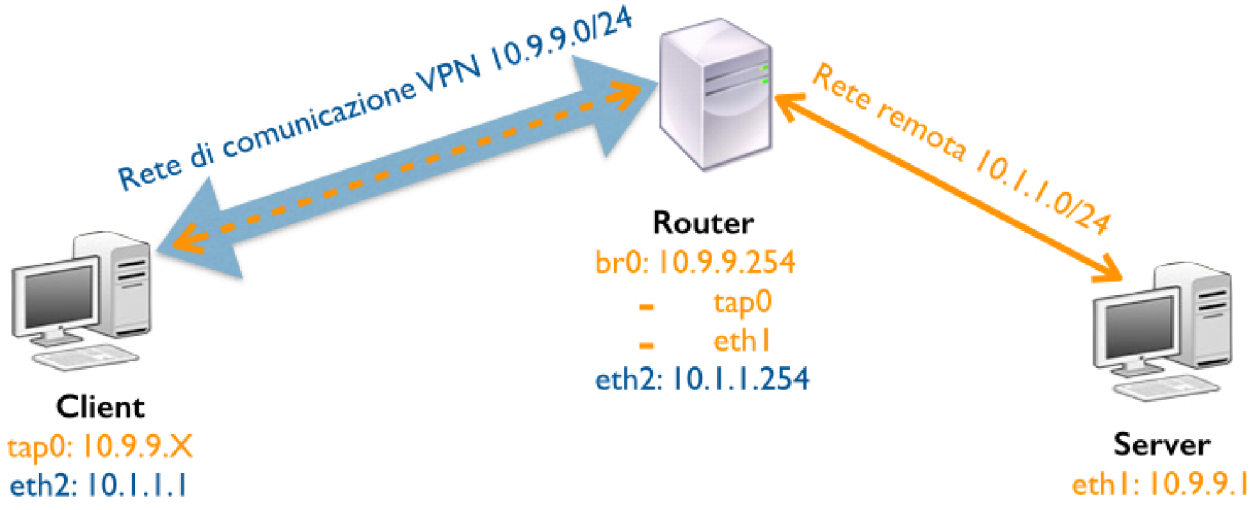
  `sudo systemctl start openvpn`

  - Nota: non riparte automaticamente al boot a meno che non si dia anche il comando

    `sudo systemctl enable openvpn`
- Test vari:
  - ping
  - traceroute
  - tcpdump/wireshark sulle diverse interfacce (reali e virtuali)

Road Warrior

- Viene così definita la configurazione di un client su rete pubblica che vuole accedere alla rete aziendale



Road Warrior bridged vs. routed

Per consentire la comunicazione tra il Client VPN e gli host della rete remota vi sono due possibili strade:

- configurare la tabella di routing del Server VPN per instradare i pacchetti da e verso la rete del Client
- configurare un bridge ethernet per connettere l'interfaccia VPN del Server con l'interfaccia ethernet connessa alla rete locale
  - questa soluzione consente al client l'uso di protocolli basati su LAN broadcast (discovery di servizi ed enumerazione di risorse)
  - l'assegnamento di un ip della rete aziendale semplifica la configurazione di servizi e firewall
- Nel seguito verrà descritto come configurare una connessione VPN tra le macchine virtuali Client e Router utilizzando la modalità SSL/TSL e il bridging delle interfacce

Configurazione bridge su Router (1)

- Installare il pacchetto bridge-utils:
  - avendo accesso a Internet:

    ```
    sudo apt-get install bridge-utils
    ```
  - dal Lab:
    - scaricare dal sito del corso sull'host il pacchetto e copiarlo su Router
    - installarlo con `sudo dpkg -i bridge-utils_1.4-5_i386.deb`
- Modificare il file `/etc/network/interfaces` cambiando la configurazione di eth1 e aggiungendo quella del bridge br0 (vedi slide successiva)
- Riavviare il servizio di networking

  ```
  sudo systemctl restart networking
  ```
- Verificare la corretta configurazione del bridge con il comando

  ```
  sudo brctl show
  ```

Configurazione bridge su Router (2)

```
...
auto eth1
iface eth1 inet manual
  up ip link set $IFACE up promisc on
  down ip link set $IFACE down promisc off

auto br0
iface br0 inet static
  address 10.9.9.254
  netmask 255.255.255.0
  # network interfaces on which to enable the bridge
  bridge_ports eth1
  # optional configurations if the machine is a VM
  bridge_fd 9       ##  forward delay time
  bridge_hello 2    ##  hello time
  bridge_maxage 12 ## maximum message age
  bridge_stp off    ##  spanning tree protocol
…
```

Generazione dei certificati (1)

- L'installazione di OpenVPN porta sul sistema come dipendenza anche una serie di script chiamati "easy-rsa" per la creazione dei certificati.
- Per mantenere separate le operazioni di OpenVPN da eventuali altre attività coi certificati, si crei una copia di questi script nella directory /etc/openvpn

```
sudo cp -r /usr/share/easy-rsa /etc/openvpn/
sudo chown -R $USER /etc/openvpn/easy-rsa/
```

Generazione dei certificati (2)

- Nel file `/etc/openvpn/easy-rsa/vars` è necessario inserire i dati di default dell'ente a cui viene rilasciato il certificato (i valori predefiniti si trovano verso la fine del file):

```
export KEY_COUNTRY="IT"
export KEY_PROVINCE="BO"
export KEY_CITY="Bologna"
export KEY_ORG="Unibo"
export KEY_EMAIL="info@example.com"
```

Generazione dei certificati (3)

- Creare i certificati con i seguenti comandi:

```
cd /etc/openvpn/easy-rsa/
source vars
./clean-all
```
(solo la prima volta, fa pulizia di tutte le chiavi)

- Creazione dei certificati e delle chiavi per la CA:

```
./build-ca
```

- Creazione dei certificati e delle chiavi per il Server OpenVPN :

```
./build-key-server server
```

- Creazione dei parametri crittografici di Diffie-Hellman:

```
./build-dh
```

Generazione dei certificati (4)

- Creare i certificati per il Client:

```
cd /etc/openvpn/easy-rsa/
./build-key client
```

- Linkare la directory contenete i certificati in **/etc/openvpn**

```
cd /etc/openvpn/
sudo ln -s easy-rsa/keys keys
```

Configurazione di OpenVPN sul Router (1)

- Creare gli script per connettere e disconnettere l'interfaccia tap dal bridge

/etc/openvpn/up.sh

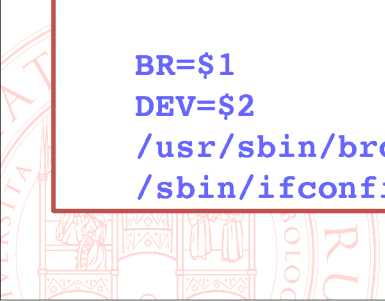
```
#!/bin/sh

BR=$1
DEV=$2
MTU=$3
/sbin/ifconfig $DEV mtu $MTU promisc up
/usr/sbin/brctl addif $BR $DEV
```

/etc/openvpn/down.sh

```
#!/bin/sh

BR=$1
DEV=$2
/usr/sbin/brctl delif $BR $DEV
/sbin/ifconfig $DEV down
```



Configurazione di OpenVPN sul Router (2)

- Rendere eseguibili i due script:

```
sudo chmod 755 /etc/openvpn/down.sh
sudo chmod 755 /etc/openvpn/up.sh
```

- Creare i file di configurazione `/etc/openvpn/server.conf`

  scaricandolo dal sito del corso

- Riavviare il servizio OpenVPN

```
sudo systemctl restart openvpn
```

Configurazione del Client (1)

- Installare OpenVPN sul Client:

```
sudo apt-get install openvpn
```

- Creare la directory che ospiterà i certificati:

```
sudo mkdir /etc/openvpn/keys
sudo chown -R $USER /etc/openvpn/keys
```

- Copiare i certificati della CA e del Client creati in precedenza sul Router:

```
scp las@192.168.56.202:/etc/openvpn/keys/ca.crt
/etc/openvpn/keys/
scp las@192.168.56.202:/etc/openvpn/keys/client.crt
/etc/openvpn/keys/
scp las@192.168.56.202:/etc/openvpn/keys/client.key
/etc/openvpn/keys/
```

Configurazione del Client (2)

- Creare il file /etc/openvpn/client.conf
  scaricandolo dal sito del corso
- Riavviare il servizio OpenVPN
  ```
  sudo systemctl restart openvpn
  ```
- Test vari:
  - ping
  - traceroute
  - tcpdump/wireshark sulle diverse interfacce (reali e virtuali)