Túneles VPN

Sobre KVM

AUTOR: Josué Monge Corrales

INDICE

1. INTRODUCCIÓN

- Lo que voy a llevar a cabo en este proyecto es intentar implementar túneles vpn sobre máquinas virtualizadas por la aplicación kvm, que se basa en la virtualización por hardware, el motivo que me movió ha realizar este proyecto ha sido que me parecia bastatante interesante no solo montar los distintos túneles VPN si no también el manejar una aplicación que poco a poco se va a ir haciendo más fuerte en el mercado como es kvm.
- El fin del proyecto es tener montados al menos un túnel con Openvpn y otro con Openswan(IPSec), ambos con todas las máquinas necesarias para su funcionamiento virtualizadas por kvm y además bajo el sistema operativo Centos5.4, el cuál aún no habia utilizado y me pareció interesante utilizar en este proyecto de investigación para añadir más cosas que aún no habia tocado aunque esto lo haga algo más complicado.

2. INSTALACIÓN DE KVM

2.1. BREVE INTRODUCIÓN

- KVM viene de Kernel-based Virtual Machine, que en español viene a ser "Máquina virtual basada en linux".
- Es un proyecto de software libre realizado para virtualización de máquinas.
- Para poder instalar kvm necesitas tener una máquina con procesador x86 con soporte Virtualization Technology (virtualización por hardware).
- Kvm utiliza una versión modificada de QEMU.

2.2. INSTALACIÓN DE KVM

- Lo primero que tendremos que hace es activar la virtualización por hardware en la bios si no no se ejecutará bien kvm al instalarlo.
- Para saber si tenemos soporte de virtualización por hardware bastará con ejecutar el siguiente comando:

- egrep '(vmx|svm)' --color=always /proc/cpuinfo

- Este comando nos deberá devolver información sino nos la devuelve es porque no tenemos dicho soporte, nos devolverá una serie de información entre la que encontraremos o bien "vmx" si el soporte es de intel o "svm" si el soporte pertenece a AMD.

- Después, una vez activado el soporte en la bios, ejecutaremos:

- aptitude install kvm qemu libvirt-bin libvirt0 ubuntu-vm-builder bridge-utils
- sudo adduser `id -un` libvirtd → lo ejecutamos como root

- Este comando lo que nos hará será añadir a root al grupo libvirtd.
- reboot

- Ahora lo que haremos después de reiniciar será cargar el modulo o bien de intel o bien de amd dependiendo de nuestra máquina:

- modprobe kvm-intel ò modp-robe kvm-amd
- service libvirtd start
- virsh -c qemu:///system list → *Esto nos mostrará una lista vacía que nos indicará que nuestro kvm está perfectamente instalado.*

2.3. UTILIZACIÓN DE LIBVIRT

2.3.1. VIRT-INSTALL

- Virt-install nos permitirá instalar una máquina virtual diciendole algunas de las características que tendrá nuestra máquina virtual.
- Hay varios parametros que se le pueden pasar al virt-install pero los más usuales serán:

-n nombre_maquina → Será el nombre de nuestra máquina.
-r 256 → Indicaremos la memoria ram virtual que usará nuestra máquina-
-f ruta_disco → Aquí indicaremos la ruta de donde se encuentra su disco duro, el cuál puede ser un fichero, y volúmen lógico o una partición.
-s 5 → Será el tamaño máximo que ocupará su hdd.
-c centos.iso → Aquí le diremos la ruta de la iso que utilizará para instalar el sistema operativo, también podemos indicarle la ruta de un dispositivo como un cdrom físico.
--acelerate → Esto activará la technologia de aceleración del kernel.
--vnc → Esto nos pasará a un guesped por vnc para seguir la instalación.
--noautoconsole → Esto hará que no se ejecute automáticamente en una consola si no que tengamos que ser nosotros el que lo indiquemos.

→ Un ejemplo sería:
virt-install -n prueba -r 256 -f prueba.img ***-s 5 -c centos.iso --accelerate*** **--connect=qemu:///system** **--vnc** ***--noautoconsole***

2.3.2. VIRT-CLONE

- Es una aplicación que nos permite replicar una máquina exactamente igual.
- Los parámetros que utilizamos con esta aplicación son:

-o maquina → *pondremos la máquina original*

-n maquinaclon → *nombre del clon*

-f disco_duro.img → *dirección de volúmen lógico, fichero o partición de donde irá nuestra máquina.*

--connect=qemu:///system → *Hipervisor que vamos a utilizar*

- EJEMPLO:

- virt-clone -o AMY -n FRANSWORTH -f /var/lib/libvirt/images/fransworth.img --connect=quemu:///system

2.3.3. VIRSH

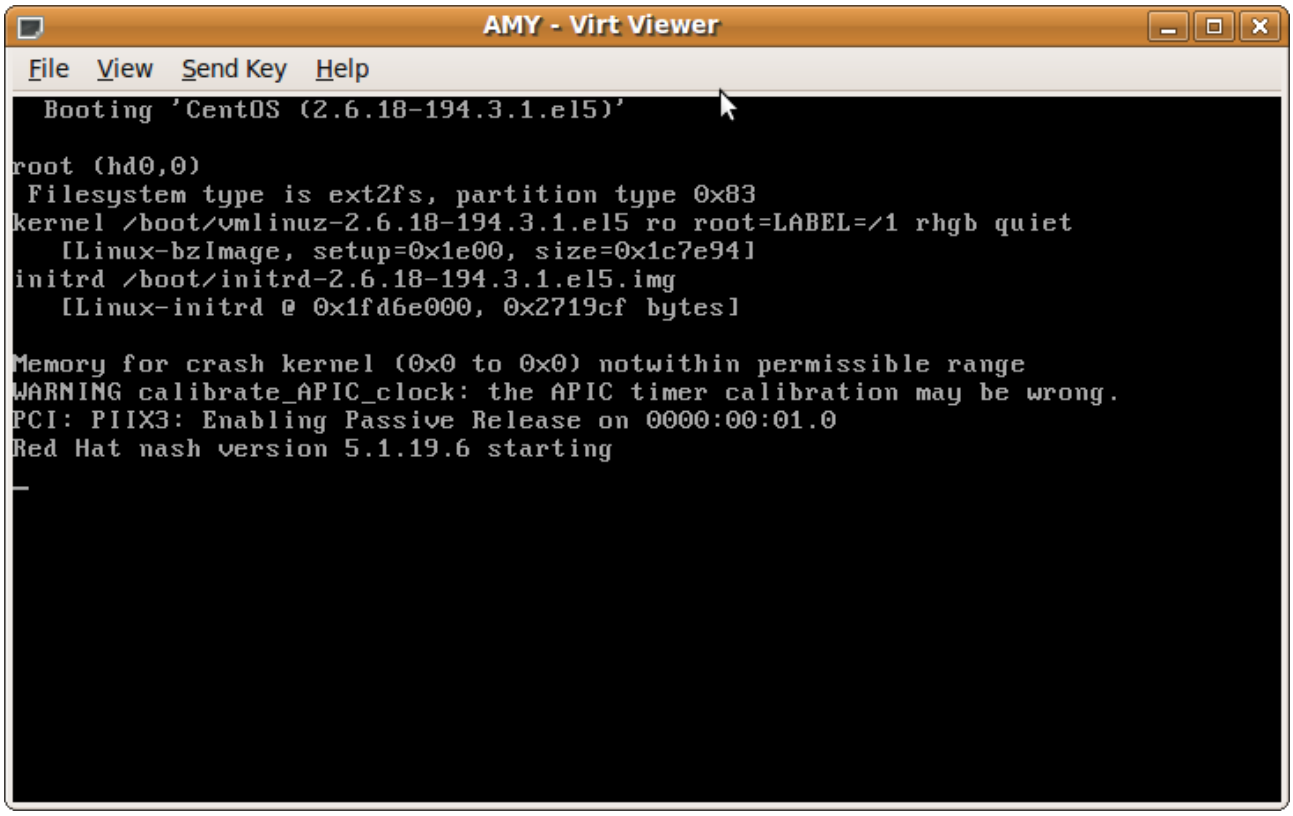
- Es una utilidad que desde linea de comandos nos permitirá interactuar con las máquinas virtuales, para encenderlas, apagarlas, suspenderlas, etc …
- Estos son algunos de los comandos que podremos usar de virsh (para entrar en él solo tendremos que teclear en la consola "#virsh" y darle al intro):

- ***list*** → Nos listará las máquinas que se estan ejecutando.
- ***start nombre_dominio*** → Nos levantará esa máquina.
- ***autostart nombre_dominio*** → Nos lo iniciará automáticamente en al iniciar.
- ***reboot nombre_dominio*** → Nos reinicia la máquina.
- ***save nombre_dominio fichero.state*** → Nos guarda el estado de la máquina.
- ***restore fichero.state*** → Restaura la máquina
- ***shutdown nombre_dominio*** → Nos apaga la máquina.
- ***destroy nombre_dominio*** → Nos apaga la máquina forzada.
- ***attach-disk nombre_dominio /dev/cdrom /media/cdrom*** → ***Nos monta el cdrom en la máquina.***
- Ejemplo:
  - virsh# ***start AMY***

2.3.4. VIRT-VIEWER

- Es una aplicación que nos conecta por vnc a la maquina.
- Para conectarnos solo tendremos que poner:

- virt-viewer nombre_dominio



2.4. INSTALACIÓN DE UNA MÁQUINA VIRTUAL

- Para crear el disco duro de nuestra máquina yo he utilizado una aplicación que trae el propio kvm, yo haré un fichero img pero también podríamos utilizar una partición que tengamos en nuestra máquina o también un volúmen lógico.

- Crearemos un disco duro para una máquina a la cuál vamos a llamar "prueba", para ello el comando que vamos a pasar es:

- kvm-img create -f qcow2 prueba.img 4G

- *Con esto nos creará un archivo de imagen de 4g y con el nombre de "prueba.img", aunque el archivo limipio al principio no nos ocupará 4G sino que nos ocupará según vayamos instalando y llenando la máquina.*

- Ahora lo que haremos será arrancar el disco duro con un CDRom para instalarlo, también pondremos el ejemplo para una imagen ISO:

- kvm -cdrom /dev/cdrom prueba.img

- kvm -cdrom /centos.iso prueba.img

- Después de haber instalado ya el sistema operativo o mientras termina de instalar crearemos el archivo xml que va a hacer del cual vamos a definir nuestra máquina y le pasaremos la información del nombre que va a tener, donde se encuentra el disco duro, si va tener cdrom, etc... Esto sería un ejemplo de archivo xml, este en concreto será el que utilizemos para nuestra máquina llamada "prueba":

```
<domain type='kvm'>
 <name>prueba</name>
 <uuid>73815423-9b95-fc1d-97ac-60d9037c3ed3</uuid>
 <memory>524288</memory>
 <currentMemory>524288</currentMemory>
 <vcpu>1</vcpu>
 <os>
  <type arch='i686' machine='pc'>hvm</type>
  <boot dev='hd'/>
 </os>
 <features>
  <acpi/>
```

```
  <apic/>
  <pae/>
 </features>
 <clock offset='utc'/>
 <on_poweroff>destroy</on_poweroff>
 <on_reboot>restart</on_reboot>
 <on_crash>restart</on_crash>
 <devices>
  <emulator>/usr/bin/kvm</emulator>
  <disk type='block' device='cdrom'>
   <target dev='hdc' bus='ide'/>
   <readonly/>
  </disk>
  <disk type='file' device='disk'>
   <source file='/var/lib/libvirt/images/prueba.img'/>
   <target dev='hda' bus='virtio'/>
  </disk>
 <interface type='network'>
   <mac address='54:52:00:24:ac:c9'/>
   <source network='intranet'/>
  </interface>
<serial type='pty'>
   <source path='/dev/pts/2'/>
   <target port='0'/>
  </serial>
  <console type='pty' tty='/dev/pts/2'>
   <source path='/dev/pts/2'/>
   <target port='0'/>
  </console>
  <input type='mouse' bus='ps2'/>
  <graphics type='vnc' port='-1' autoport='yes' keymap='es'/>
  <sound model='es1370'/>
 </devices>
```

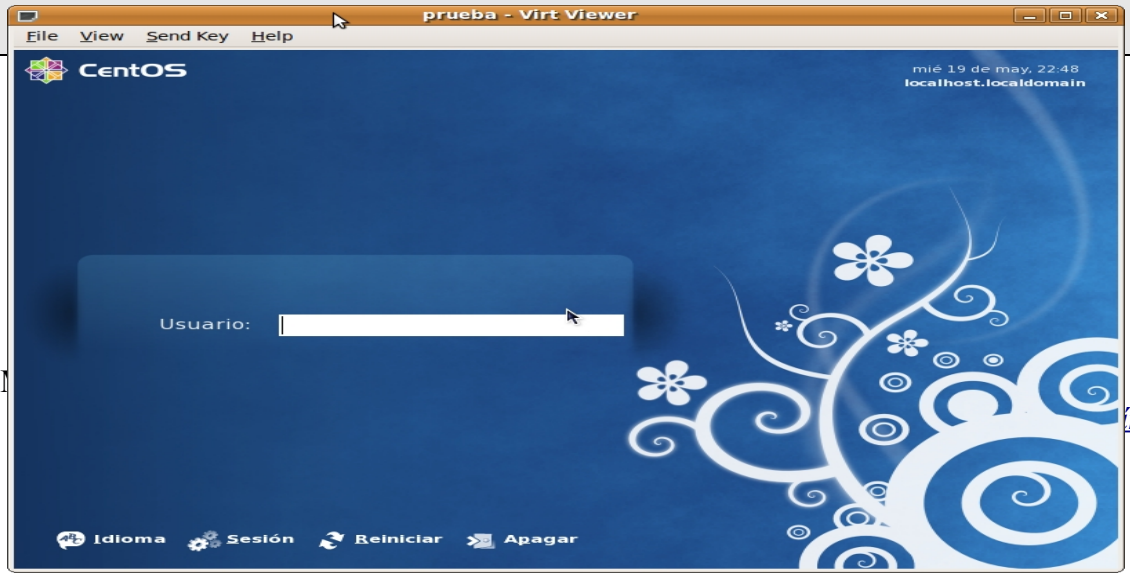
```
</domain>
```

- Estas son las especificaciones que le hemos pasado en nuestro archivo:
    - dominio de tipo “kvm”
    - nombre del dominio “prueba”
    - 524 de ram.
    - Le hemos puesto una unidad de cdrom
    - Especificamos la ruta del disco duro.
    - le decimos que los graficos iran por vnc y que la codificación del teclado será la de españa.
- Ahora definiremos el dominio en kvm con “virsh”, y a partir de aquí ya podremos utilizar nuestra máquina:

    - ***virsh define /etc/libvirt/qemu/prueba.img***
    - *Para verificar que nos ha creado nuestra máquina bien, solo tendremos que hacer un:*
        - ***virsh list --all***
        - *Nos enseñará algo como esto:*

```
root@portatil-ubuntu:/etc/libvirt/qemu# virsh list --all
Connecting to uri: qemu:///system
 Id Nombre                 Estado
----------------------------------
  1 AMY                    ejecutándose
  - prueba                 callar
```

    - *Ahora levantaremos la máquina con el siguiente comando:*
        - ***virsh start prueba***
    - *Se iniciará la máquina y ya podremos entrar a través de vnc con virt-viewer, con el siguiente comando:*
        - ***-virt-viewer prueba &***



3. INSTALACIÓN DE OPENVPN SERVER

- Lo primero que haremos será instalar el repositorio DAG en nuestro centos, para ello nos iremos a la página: http://dag.wieers.com/rpm/packages/rpmforge-release/
- Nos descargaremos el paquete llamado: "rpmforge-release.0.3.6.el5.rf.i385.rpm"
- Haremos los siguientes comandos:

```
- rpm -Uvh  rpmforge-release.0.3.6.el5.rf.i385.rpm
- yum -y update
- yum -y install openvpn
```

3.1. GENERANDO LOS CERTIFICADOS

- Nos iremos a la carpeta: "***/usr/share/doc/openvpn-2.0.9/*****"**
- Ahora copiaremos la carpeta "easy-rsa":

```
- cp -R /usr/share/doc/openvpn-2.0.9/easy-rsa /etc/openvpn
- cp /usr/share/doc/openvpn-2.0.9/sample-config-files/server.conf
  /etc/openvpn
```

3.1.1. CERTIFICADO DE AUTORIDAD CERTIFICADORA(CA)

```
- cd /etc/openvpn/easy-rsa
- . ./vars
- . ./clean-all
- . ./build-ca
```

- *Dejaremos todas las preguntas por defecto excepto la del nombre que le pondremo "openvpn-ca".*
- *Ya tendremos nuestro certificado de la autoridad certificadora.*

3.1.2. CERTIFICADO PARA EL SERVIDOR

- cd /etc/openvpn/easy-rsa

- ./build-key-server server

- *Dejaremos todas las preguntas por defecto excepto el nombre que le pondremos "servidor".*

- Nos pedirá también una contraseña yo le pondré "root".

- Este apartado además nos hará otras dos preguntas a las que responderemos que si "y".

3.1.3. CERTIFICADOS PARA LOS CLIENTES

- *Para este caso crearemos certificados para dos clientes, a los cuales llamaremos "cliente1" y "cliente2".*

- cd /etc/openvpn/easy-rsa

- . ./build-key-pass cliente1

- *Este certificado se puede crear igual pero con el script "build-key" la diferencia es que con este no tendríamos que meter contraseña y con el que nosotros hemos utilizado sí.*

- *Dejaremos todas las preguntas por defecto excepto la del nombre que pondremos "cliente1", y la contraseña que le pondremos "usuario".*

- ***. ./build-key-pass cliente2***

- Para este cliente haremos las misma operación pero poniendole de nombre "cliente2" y de contraseña le dejaremos la misma ("usuario").

3.2. CONFIGURACIÓN DEL SERVIDOR OPENVPN

- Nos iremos al fichero **" *~~/etc/openvpn/server.conf~~* " :**

- *# Puerto por el que va a trabajar*
  - ***port 1194***

- *# TCP or UDP server?(trabajara por udp)*
  - ***proto udp***

- *# "dev tun" will create a routed IP tunnel, "dev tap" will create an ethernet tunnel.*
  - ***dev tun***

- *#certificados necesarios*
  - ***ca /etc/openvpn/ca.crt***
  - ***cert /etc/openvpn/server.crt***
  - ***key /etc/openvpn/server.key***

- *# Parametro Diffie hellman*
  - ***dh /etc/openvpn/dh1024.pem***

- *##Direcciones que le asignara a los clientes y el servidor siempre será el .1*
  - ***server 10.10.10.0 255.255.255.0***

- *#Esto es por si el openvpn se reinicia que mantenga las direcciones asignadas anteriormente.*
  - ***ifconfig-pool-persist ipp.txt***

- *## Esto se lo añadimos para que tenga conexión con la red local*

- push "route 172.16.0.0 255.255.255.0"

- #Hará ping cada 10 segundos y esperará como maximo 120seg la respuesta

- keepalive 10 120

- *# Es el compresor de archivos que utiliza OpenVpn*

- comp-lzo

- *# En caso de reinicio guardaría la sesión y no la perdería*

- persist-key

- persist-tun

- *# fichero log del estado*

- status openvpn-status.log

- *# fichero log*

- log /var/log/openvpn.log

- # Set the appropriate level of log

 # file verbosity.

#

 # 0 is silent, except for fatal errors

 # 4 is reasonable for general usage

 # 5 and 6 can help to debug connection problems

 # 9 is extremely verbose

#Nivel de lo que se va a guardar en el fichero log, ahora mismo lo tenemos en cosas razonables o que nos interese ver.

- verb 4

- *# Para que actue como servidor*

- tls-server

- Ahora reiniciaremos el servicio con:

- ***service openvpn restart***

- Podemos observar si hacemos un "*ifconfig*" que nos ha creado una nueva interfaz:

```
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
-00
          inet addr:10.10.10.1  P-t-P:10.10.10.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

3.3. CONFIGURACIÓN DEL CLIENTE OPENVPN

- Ahora lo primero que haremos será pasar los certificados para el "cliente1" desde el servidor hasta el cliente y para ello emplearemos "scp":

- ***scp user@192.168.10.157:/etc/openvpn/easy-rsa/keys/cliente1.* /etc/openvpn/***
- ***scp user@192.168.10.157:/etc/openvpn/easy-rsa/keys/ca.cert /etc/openvpn/***
  - *Nos pedirá la contraseña para el usuario "user" y una vez introducida se pasarán hacia el cliente.*

- Una vez tenemos nuestros certificados en el cliente pasaremos a configurar su archivo de configuración:

- ***# nano /etc/openvpn/cliente.conf***
  - *# Dispositivo de tunel*
    - ***dev tun***
  - *# Direccion del servidor*
    - ***remote 192.168.10.157***
  - *# Aceptar directivas del extremo remoto*
    - ***pull***
  - *# Hacer de cliente*
    - ***tls-client***
  - *# certificados necesarios*
    - ***ca /etc/openvpn/ca.crt***

- cert /etc/openvpn/cliente1.crt

- key /etc/openvpn/cliente1.key

- # Hará ping cada 10 segundos y esperará como maximo 120seg la respuesta

- keepalive 10 120

- # Es el compresor de archivos que utiliza OpenVpn

- comp-lzo

- # Nivel de lo que se va a guardar en el fichero log, ahora mismo lo tenemos en cosas razonables o que nos interese ver.

- verb 4

- Ahora reiniciaremos el servicio opnvpn:

- service openvpn restart

- Si todo ha ido como debe, ahora nos haremos un "ifconfig" y nos debe crear un tunel con una dirección dentro del rango "10.10.10.0/24", además cuando vayamos a buscar alguna dirección del rango "172.16.0.0/24", deberá salir por la "10.10.10.1", que es el servidor vpn, para comprobar todo esto iremos haciendo las siguientes pruebas:

ifconfig

- #

```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 54:52:00:24:AC:D7
          inet addr:192.168.50.69  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: fe80::5652:ff:fe24:acd7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:84 errors:0 dropped:0 overruns:0 frame:0
          TX packets:131 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8763 (8.5 KiB)  TX bytes:17326 (16.9 KiB)
          Interrupt:11 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2542 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2542 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5592776 (5.3 MiB)  TX bytes:5592776 (5.3 MiB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.10.10.6  P-t-P:10.10.10.5  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

- *Aquí podemos ver como nos a levantado una interfaz llamada "tun0" y podemos ver como tenemos la ip "10.10.10.6".*

- *Ahora veremos si nos ha cambiado la tabla de enrutamiento, y nos a añadido además de la entrada para la red "10.10.10.0/24" , la de la red "172.16.0.0/24" con salida por una ip de la red "10.10.10.0/24" .*

- ***# route***

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.10.1      10.10.10.5      255.255.255.255 UGH   0      0        0 tun0
10.10.10.5      *               255.255.255.255 UH    0      0        0 tun0
192.168.50.0    *               255.255.255.0   U     0      0        0 eth0
172.16.0.0      10.10.10.5      255.255.255.0   UG    0      0        0 tun0
169.254.0.0     *               255.255.0.0     U     0      0        0 eth0
default         192.168.50.1    0.0.0.0         UG    0      0        0 eth0
```

- *Ahora probaremos que tenemos conexión con alguna máquina de la red interna de la vpn, osea, con "172.16.0.0/24" , para ello realizaremos un ping a "prueba(172.16.0.6)":*

```
[root@localhost ~]# ping -c 2 172.16.0.6
PING 172.16.0.6 (172.16.0.6) 56(84) bytes of data.
64 bytes from 172.16.0.6: icmp_seq=1 ttl=63 time=3.88 ms
64 bytes from 172.16.0.6: icmp_seq=2 ttl=63 time=3.08 ms

--- 172.16.0.6 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
```

- *Y por últimos probaremos que realmente está saliendo por donde debe de salir para llegar hasta la máquina "prueba" y para ello realizaremos un tracert para ver el camino que sigue el paquete:*

.

```
[root@localhost ~]# tracert 172.16.0.6
traceroute to 172.16.0.6 (172.16.0.6), 30 hops max, 40 byte packets
 1  10.10.10.1 (10.10.10.1)  37.997 ms  99.004 ms  99.037 ms
 2  172.16.0.6 (172.16.0.6)  98.770 ms  98.858 ms  98.882 ms
```

- Y con esto ya tendríamos nuestra Servidor y nuestro Cliente VPN funcionando la modalidad de vpn que hemos utilizado se llama "ROAD WARRIORS" Para realizar esta parte del proyecto el esquema de red que hemos utilizado con KVM es el siguiente:

4. TUNEL IPSEC LAN A LAN UTILIZANDO OPENSWAN

4.1. INTRODUCCIÓN

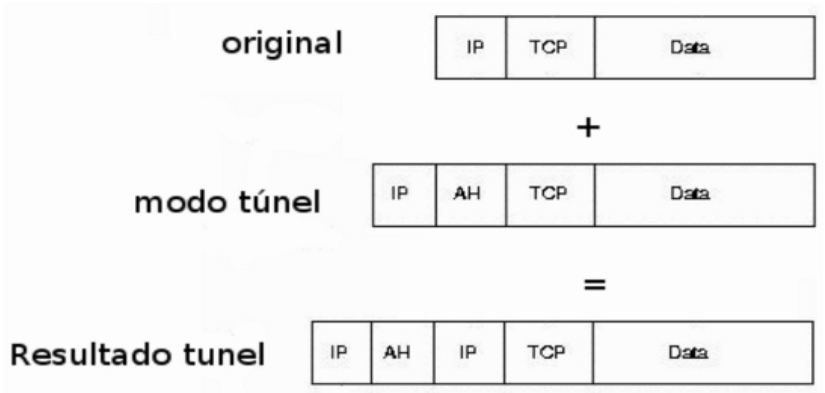
- IPSec es un estándar que se encarga de dar seguridad a la capa IP y a todos los protocolos superiores del nivel de red como pueden ser TCP o UDP.
- Las siglas IPSec vienen de Protocolo de Seguridad de Internet (Internet Protocol Security).
- Se puede integrar en la versión actual de IPV4 y ya viene por defecto integrado con IPV6.

4.2. MODOS DE FUNCIONAMIENTO

- IPSec puede trabajar de dos modos:

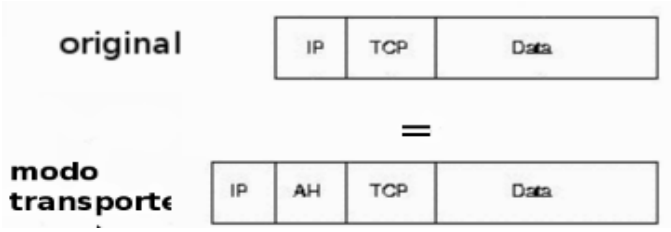
  - **Modo de Túnel:**

    - Este modo se utiliza cuando algunas de las dos extremos es una máquina que hace de gateway o puente de una red. (Este será nuestro caso)
    - Además el contenido del datagrama AH o ESP se encapsula en un nuevo datagrama ip en el que incluye el original.



  - **Modo de Transporte:**

    - Se utiliza normalmente para las conexiones host a host.
    - En este modo los datos que vienen de la capa de transporte se encapsulan direcatemente en un datagram AH o ESP.



4.3. METODOS DE AUTENTICACIÓN

- Tiene tres tipos de autenticación:

  - **Por Clave Compartida:**

    - Esta metodo de autenticación se basa en que ambos extremos conocen una cadena de caracteres que solo conocen ellos dos y a la hora de hacer la comunicación ambos demuestran que saben la cadena por medio del hash pero sin mostrar nunca el contenido, de este modo un atacante no sabría cual es la cadena de caracteres aunque sniffen paquetes.

  - **Por Firmas digitales RSA:**

    - Este metodo de autenticación consta de un par de llaves que tiene que generar aleatoriamente cada extremo y después se tendrían que intercambiar la clave pública cada uno de los extremos para así conocerse al tramitar la conexión.

  - **Por Certificados X.509:**

    - Se basa en la autenticación por certificados digitales, los certificados contendrán los datos de su propietario y su clave pública, el certificado debe ser avalado por una Autoridad Certificadora(CA) la cuál a firmado dicho certificado digitalmente.

    - Funciona por que el cliente le manda una serie de caracteres que el servidor cifrará con su propia clave para desencriptarla, si el servidor puede desencriptarlo sin problemas se dará luz verde a la conexión y de lo contrario se rechazará la conexión.

4.4. NEGOCIACIÓN DEL TÚNEL IPSEC

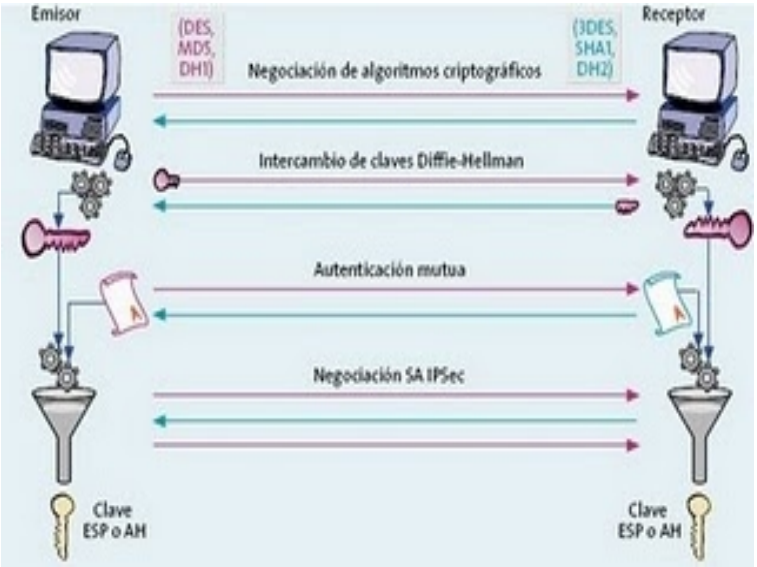
- La negociación se lleva a cabo en dos fases:

  - **Fase 1:**

    - En esta fase se establece un canal seguro y autenticado por el que va a viajar la información.

  - **Fase 2:**

    - En esta fase los dos extremos se pondrán de acuerdo sobre el protocolo que van a utilizar (AH o ESP) y los algoritmos de autenticación que van a emplear para intercambiarse información.



4.5. INSTALACIÓN DE OPENSWAN EN CENTOS5.4

- Para instalar openswan lo podemos instalar de dos maneras, desde los repositorios directamente o desde la pagina oficial de openswan (www.openswan.org/download)

- Para su instalación por repositorios solo tendremos que ejecutar desde una consola:

  - ***# yum -y install openswan***

    - *Los archivos más importantes que nos creará serán dos archivos:*

      - */etc/ipsec.secrets*

      - */etc/ipsec.conf*

    - *También nos creará una carpeta que es:*

      - */etc/ipsec.d/*

4.5.1. HERRAMIENTAS DE OPENSWAN

- Estas son las herramientas que openswan nos incorpora al instalarlo:

  - **ipsec auto --add *conexión***

    - Esta herramienta la utilizaremos cada vez que añadamos una nueva conexión en el archivo "ipsec.conf" para agregarla al demonio pluto, así él la reconocerá cada vez que queramos levantar esta conexión.

  - **ipsec auto --up *conexión***

    - Con esta herramienta levantaremos la conexión especificada.

  - **ipsec auto --delete *conexión***

    - Esta herramienta tiraremos la conexión especificada y que anteriormente se habría negociado.

  - **ipsec auto --status**

    - Nos mostrará el estado actual de todas las conexiones levantadas.

  - **ipsec auto --rereadsecrets**

    - Esta herramienta solo la utilizaremos cuando hayamos hecho alguna modificación en el archivo "ipsec.secrets" para decirle a pluto que la vuelva a leer y cargar correctamente, si no hacemos este paso cuando modifiquemos este fichero pluto seguirá utilizando la clave anterior.

  - **ipsec auto --rereadcacerts**

    - Esta herramienta su utilización es por el mismo motivo que la anterior pero con la diferencia que esta herramienta lo que nos volvería a leer sería el certificado CA que normalmente se encuentra en "/etc/ipsec.d/cacerts/".

4.6. CONFIGURACIÓN TUNEL IPSEC LAN A LAN CON PSK

- Empezaremos por configurar el archivo "/etc/ipsec.secrets" el cuál quedará de la siguiente manera tanto en el cliente como en el servidor:

- : PSK "palabra clave"

-*En ambos sitios debe quedar exactamente igual.*

- Ahora pasaremos al archivo "/etc/ipsec.conf" del servidor(**farnsworth,192.168.10.158**):

```
config setup
        # interfaces fisicas y virtuales, en nuestro caso cogera la que tenga de gateway
        interfaces=%defaultroute
        # For Red Hat Enterprise Linux and Fedora, leave
        protostack=netkey

conn %default
# Aqui ira lo que queramos que sea comun a todas las conexiones

conn kiff
      # Para empezar la conexion automaticamente
        auto=start

        # autenticacion mediante clave PSK
        authby=secret

        # algoritmos de cifrado y autenticacion
        esp=3des-md5

        # algoritmos de cifrado y autenticacion(fase1)
        ike=3des-md5-modp1024

        # Direccion IP local
        left=192.168.10.158
```

```
        # Subred local
        leftsubnet=172.16.0.0/24

        # Direccion IP remota
        right=192.168.10.60

        # Subred remota
        rightsubnet=192.168.20.0/24
```

- El siguiente paso que haremos será registrarlo en el demonio pluto tanto la nueva configuración, como la PSK:

```
- # ipsec auto --rereadsecrets
- # ipsec auto --add kiff
```

- Ahora nos iremos al cliente (kiff, 192.168.10.60) y pondremos el archivo "ipsec.secrets" tal y como dijimos arriba, por tanto pasamos al archivo "ipsec.conf" que quedaría de la siguiente manera:

```
config setup
        # interfaces fisicas y virtuales, en nuestro caso cogera la que tenga de gateway
        interfaces=%defaultroute
        # For Red Hat Enterprise Linux and Fedora, leave
        protostack=netkey

conn %default
# Aqui ira lo que queramos que sea comun a todas las conexiones

conn farnsworth
      # Para empezar la conexion automaticamente
        auto=start

        # autenticacion mediante clave PSK
        authby=secret

        # algoritmos de cifrado y autenticacion
        esp=3des-md5
```

 # algoritmos de cifrado y autenticacion(fase1)

ike=3des-md5-modp1024

 # Direccion IP local

left=192.168.10.60

 # Subred local

leftsubnet=192.168.20.0/24

 # Direccion IP remota

right=192.168.10.158

 # Subred remota

rightsubnet=172.16.0.0/24

- La diferencia con respecto al servidor ahora será que ademas de registrar en pluto la PSK y la conexión farnsworth, también levantaremos el túnel:

- ***# ipsec auto --rereadsecrets***
- ***# ipsec auto --add farnsworth***
- ***# ipsec auto --up farnsworth***

- Y listo!, ya tendríamos nuestro túnel ipsec listo para usar, si queremos ver como ha hecho la conexión no tendríamos más que abrir el log "/var/log/secure.log"

4.7. CONFIGURACIÓN DE TÚNEL IPSEC LAN A LAN CON RSA

- Para este túnel en vez de utilizar clave compartida(PSK) como antes, utilizaremos certificados generados desde las propias máquinas, para generarlos utilizaremos una herramienta de ipsec y la utilizaremos de la siguiente manera:

- ***ipsec rsasigkey –verbose 1024 –configdir /etc/pki/nssdb/ >keys***

  -Esto nos creará en el directorio donde lo hayamos ejecutado un archivo llamado keys el cuál tendrá esta pinta mas o menos:

 # RSA 1024 bits localhost.localdomain Wed Jun 16 23:53:36 2010

 # for signatures only, UNSAFE FOR ENCRYPTION

#pubkey=0sAQOvkL9Ps3Dp7bswEvmvjNKUx2AveUKuweDTrvnufbfa+GWu8/cMtf/VOAGRSKZGu58+KlAoRpxRdW00EVI3DHGxzk5BDlXZNeaYeLHcFhuCtEjrrLbf6a12bwPvWOC7CcIs/LmTO41QCIJ6QsjFljgjv84w/jX5i7WJYxQ7TRNxxQ==

Modulus:
0xaf90bf4fb370e9edbb3012f9af8cd294c7602f7942aec1e0d3aef9ee7db7daf865aef3f70cb5ffd538019148a646bb9f3e2a5028469c51756d341152370c71b1ce4e410e55d935e69878b1dc161b82b448ebacb6dfe9ad766f03ef58e0bb09c22cfcb9933b8d5008827a42c8c5963823bfce30fe35f98bb58963143b4d1371c5

PublicExponent: 0x03

 # everything after this point is CKA_ID in hex format when using NSS

PrivateExponent: 0xa72c650170218fcc20810b2402532f01837cf337

Prime1: 0xa72c650170218fcc20810b2402532f01837cf337

Prime2: 0xa72c650170218fcc20810b2402532f01837cf337

Exponent1: 0xa72c650170218fcc20810b2402532f01837cf337

Exponent2: 0xa72c650170218fcc20810b2402532f01837cf337

Coefficient: 0xa72c650170218fcc20810b2402532f01837cf337

CKAIDNSS: 0xa72c650170218fcc20810b2402532f01837cf337

- Tendremos este fichero pero con la diferencia que las claves serán más largas y distintas, no las pongo largas por ahorrar espacios.

*- Una vez generado este fichero configuraremos el archivo **"/etc/ipsec.secrets"** de la siguiente manera:*

: RSA {

Modulus:
0xaf90bf4fb370e9edbb3012f9af8cd294c7602f7942aec1e0d3aef9ee7db7daf865aef3f70cb5ffd538019148a646bb9f3e2a5028469c51756d341152370c71b1ce4e410e55d935e69878b1dc161b82b448ebacb6dfe9ad766f03ef58e0bb09c22cfcb9933b8d5008827a42c8c5963823bfce30fe35f98bb58963143b4d1371c5

PublicExponent: 0x03

 # everything after this point is CKA_ID in hex format when using NSS

PrivateExponent: 0xa72c650170218fcc20810b2402532f01837cf337

Prime1: 0xa72c650170218fcc20810b2402532f01837cf337

Prime2: 0xa72c650170218fcc20810b2402532f01837cf337

Exponent1: 0xa72c650170218fcc20810b2402532f01837cf337

Exponent2: 0xa72c650170218fcc20810b2402532f01837cf337

Coefficient: 0xa72c650170218fcc20810b2402532f01837cf337

```
CKAIDNSS: 0xa72c650170218fcc20810b2402532f01837cf337
        }
```

- Ahora modificaremos el archivo **"/etc/ipsec.conf"** el cuál tendrá una diferencia ya que necesitaremos la clave publica generada por el otro extremo, en este caso el cliente ***"kiff",*** Podremos pasarlo por medio de un pendrive, disco, scp, etc..., nosotros lo haremos por un scp:

```
- # scp 192.168.10.60:/etc/ipsec.d/keys2 /etc/ipsec.d/clavef
  # root@192.168.10.60's password: *****
  keys2                                100% 1049     1.0KB/s   00:00
  # cat /etc/ipsec.d/clavef
#pubkey=0sAQO4naOCkHFQOmyucFvVwqyQJ3oex8CgKKGpUEeZuSt/dBUlb1pIa0Dkoek+8Kma7e8IANPcmj2uawOZA0Acb9xSIiUhXyT6QIcz2g/C70elCt15tEyhs9U4avVu0ZEUJlTNHCfhyEhPYUaq+bjQx95wqBHqOuyfWMLrMESkU65NRw==
```

- Una vez tenemos la clave pública del otro extremo ya podemos configurar el archivo **"/etc/ipsec.conf"** y quedará de la siguiente manera:

```
config setup
        # interfaces fisicas y virtuales, en nuestro caso cogera la que tenga de gateway
        interfaces=%defaultroute
        # For Red Hat Enterprise Linux and Fedora, leave
        protostack=netkey

conn %default
# Aqui ira lo que queramos que sea comun a todas las conexiones

conn kiff
      # Para empezar la conexion automaticamente
        auto=start

        # autenticacion mediante clave RSA
        authby=rsasig

        # algoritmos de cifrado y autenticacion
        esp=3des-md5
```

```
# algoritmos de cifrado y autenticacion(fase1)
ike=3des-md5-modp1024

# Direccion IP local
left=192.168.10.158

# Subred local
leftsubnet=172.16.0.0/24

# Direccion IP remota
right=192.168.10.60

# Subred remota
rightsubnet=192.168.20.0/24

# Clave publica de kiff

rightrsasigkey=0sAQO4naOCkHFQOmyucFvVwqyQJ3oex8CgKKGpUEeZuSt/dBUlb1pIa0Dkoek+8Kma7e8IANPcmj2uawOZA0Acb9xSIiUhXyT6QIcz2g/C70elCt15tEyhs9U4avVu0ZEUJlTNHCfhyEhPYUaq+bjQx95wqBHqOuyfWMLrMESkU65NRw==
```

- Tendremos que ejecutar los siguientes comandos:

```
- # ipsec auto --rereadsecrets
- # ipsec auto --add kiff
```

- Ahora pasaremos a configurar el cliente(***kiff, 192.168.10.60***), para conseguir la clave pública del servidor haremos la misma estrategia:

```
- # scp 192.168.10.158:/etc/ipsec.d/keys2 /etc/ipsec.d/clavek
  # root@192.168.10.158's password: *****
  keys2                          100% 1049     1.0KB/s   00:00
  # cat /etc/ipsec.d/clavek
#pubkey=0sAQOvkL9Ps3Dp7bswEvmvjNKUx2AveUKuweDTrvnufbfa+GWu8/cMtf/VOAGRSKZGu58+KlAoRpxRdW00EVI3DHGxzk5BDlXZNeaYeLHcFhuCtEjrrLbf6a12bwPvWOC7CcIs/LmTO41QCIJ6QsjFljgjv84w/jX5i7WJYxQ7TRNxxQ==
```

- Vamos a configurar el archivo ***"/etc/ipsec.secrets"*** de kiff:

```
: RSA {
        Modulus:
0xb89da3829071503a6cae705bd5c2ac90277a1ec7c0a028a1a9504799b92b7f7415256f5a486b40
e4a1e93ef0a99aedef0800d3dc9a3dae6b039903401c6fdc522225215f24fa408733da0fc2ef47a50a
dd79b44ca1b3d5386af56ed191142654cd1c27e1c8484f6146aaf9b8d0c7de70a811ea3aec9f58c2e
b3044a453ae4d47
        PublicExponent: 0x03
        # everything after this point is CKA_ID in hex format when using NSS
        PrivateExponent: 0x9310c35833a5f7a28b6bcb39161845a51ed3a0ba
        Prime1: 0x9310c35833a5f7a28b6bcb39161845a51ed3a0ba
        Prime2: 0x9310c35833a5f7a28b6bcb39161845a51ed3a0ba
        Exponent1: 0x9310c35833a5f7a28b6bcb39161845a51ed3a0ba
        Exponent2: 0x9310c35833a5f7a28b6bcb39161845a51ed3a0ba
        Coefficient: 0x9310c35833a5f7a28b6bcb39161845a51ed3a0ba
        CKAIDNSS: 0x9310c35833a5f7a28b6bcb39161845a51ed3a0ba
        }
```

- Acontinuación tocaremos el el fichero "***/etc/ipsec.conf***":

```
config setup
        # interfaces fisicas y virtuales, en nuestro caso cogera la que tenga de gateway
        interfaces=%defaultroute
        # For Red Hat Enterprise Linux and Fedora, leave
        protostack=netkey

conn %default
# Aqui ira lo que queramos que sea comun a todas las conexiones

conn farnsworth
      # Para empezar la conexion automaticamente
        auto=start

        # autenticacion mediante clave PSK
        authby=secret
```

 # algoritmos de cifrado y autenticacion

esp=3des-md5

 # algoritmos de cifrado y autenticacion(fase1)

ike=3des-md5-modp1024

 # Direccion IP local

left=192.168.10.60

 # Subred local

leftsubnet=192.168.20.0/24

 # Direccion IP remota

right=192.168.10.158

 # Subred remota

rightsubnet=172.16.0.0/24

 # Clave publica de kiff

rightrsasigkey=*0sAQOvkL9Ps3Dp7bswEvmvjNKUx2AveUKuweDTrvnufbfa+GWu8/cMtf/VOAGRSKZGu58+KlAoRpxRdW00EVI3DHGxzk5BDlXZNeaYeLHcFhuCtEjrrLbf6a12bwPvWOC7CcIs/LmTO41QCIJ6QsjFljgjv84w/jX5i7WJYxQ7TRNxxQ==*

- Y para terminar cargaremos tanto la nueva conexión, como el nuevo certificado, además levantaremos la conexión:

- ***# ipsec auto --rereadsecrets***
- ***# ipsec auto --add farnsworth***
- ***# ipsec auto --up farnsworth***

- Para comprobar que efectivamente está funcionando haremos tanto en el cliente como en el servidor lo siguiente(esto sirve para comprobar cualquiera de las conexiones, ya sea con PSK(estas capturas son con PSK) o RSA ya que el resultado es practicamente el mismo):

- ***# ipsec auto --status***
  - *Nos imprimirá en pantalla algo como esto:*

```
000 "kiff": 172.16.0.0/24===192.168.10.158<192.168.10.158>[+S=C]...192.168.10.60<192.168.10.60>[+
S=C]===192.168.20.0/24; erouted; eroute owner: #6
000 "kiff":     myip=unset; hisip=unset;
000 "kiff":   ike_life: 3600s; ipsec_life: 28800s; rekey_margin: 540s; rekey_fuzz: 100%; keyingtr
ies: 0
000 "kiff":   policy: PSK+ENCRYPT+TUNNEL+PFS+UP+IKEv2ALLOW+lKOD+rKOD; prio: 24,24; interface: eth
0;
000 "kiff":   newest ISAKMP SA: #5; newest IPsec SA: #6;
000 "kiff":   IKE algorithms wanted: 3DES_CBC(5)_000-MD5(1)-MODP1024(2); flags=-strict
000 "kiff":   IKE algorithms found:  3DES_CBC(5)_192-MD5(1)_128-2,
000 "kiff":   IKE algorithm newest: 3DES_CBC_192-MD5-MODP1024
000 "kiff":   ESP algorithms wanted: 3DES(3)_000-MD5(1); flags=-strict
000 "kiff":   ESP algorithms loaded: 3DES(3)_192-MD5(1)_128
000 "kiff":   ESP algorithm newest: 3DES_0-HMAC_MD5; pfsgroup=<Phase1>
000
000 #6: "kiff":500 STATE_QUICK_R2 (IPsec SA established); EVENT_SA_REPLACE in 28527s; newest IPSE
C; eroute owner; isakmp#5; idle; import:not set
000 #6: "kiff" esp.12afdc4f@192.168.10.60 esp.b1bc9616@192.168.10.158 tun.0@192.168.10.60 tun.0@1
92.168.10.158 ref=0 refhim=4294901761
000 #5: "kiff":500 STATE_MAIN_R3 (sent MR3, ISAKMP SA established); EVENT_SA_REPLACE in 3327s; ne
west ISAKMP; lastdpd=-1s(seq in:0 out:0); idle; import:not set
```

- Y el cliente:

```
000 "farnsworth": 192.168.20.0/24===192.168.10.60<192.168.10.60>[+S=C]...192.168.10.158<192.16
8.10.158>[+S=C]===172.16.0.0/24; erouted; eroute owner: #3
000 "farnsworth":     myip=unset; hisip=unset;
000 "farnsworth":   ike_life: 3600s; ipsec_life: 28800s; rekey_margin: 540s; rekey_fuzz: 100%;
 keyingtries: 0
000 "farnsworth":   policy: PSK+ENCRYPT+TUNNEL+PFS+UP+IKEv2ALLOW+lKOD+rKOD; prio: 24,24; inter
face: eth0;
000 "farnsworth":   newest ISAKMP SA: #2; newest IPsec SA: #3;
000 "farnsworth":   IKE algorithms wanted: 3DES_CBC(5)_000-MD5(1)-MODP1024(2); flags=-strict
000 "farnsworth":   IKE algorithms found:  3DES_CBC(5)_192-MD5(1)_128-2,
000 "farnsworth":   IKE algorithm newest: 3DES_CBC_192-MD5-MODP1024
000 "farnsworth":   ESP algorithms wanted: 3DES(3)_000-MD5(1); flags=-strict
000 "farnsworth":   ESP algorithms loaded: 3DES(3)_192-MD5(1)_128
000 "farnsworth":   ESP algorithm newest: 3DES_0-HMAC_MD5; pfsgroup=<Phase1>
000
000 #4: "farnsworth":500 STATE_MAIN_I3 (sent MI3, expecting MR3); EVENT_RETRANSMIT in 35s; las
tdpd=-1s(seq in:0 out:0); idle; import:admin initiate
000 #4: pending Phase 2 for "farnsworth" replacing #0
000 #3: "farnsworth":500 STATE_QUICK_R2 (IPsec SA established); EVENT_SA_REPLACE in 28439s; ne
west IPSEC; eroute owner; isakmp#2; idle; import:not set
000 #3: "farnsworth" esp.3cb8a391@192.168.10.158 esp.f93b87e2@192.168.10.60 tun.0@192.168.10.1
58 tun.0@192.168.10.60 ref=0 refhim=4294901761
000 #2: "farnsworth":500 STATE_MAIN_R3 (sent MR3, ISAKMP SA established); EVENT_SA_REPLACE in
3239s; newest ISAKMP; lastdpd=-1s(seq in:0 out:0); idle; import:not set
000
[root@kiff ~]#
```

4.8. INSTALACIÓN Y CONFIGURACIÓN DE OPENSWAN EN UBUNTU 9.04

- Son pocas las diferencias entre ubuntu y centos pero voy a redactar ambas porque por mínima que sean son diferencias a tener en cuenta.
- Para instalar openswan haremos:

- ***# aptitude install openswan***
  - *Una de las grandes diferencias que vamos a encontrar es la estructura de la carpeta "ipsec.d" ya que la versión de los repositorios de ubuntu incluye las siguientes carpetas:*
  - ***#ls -l /etc/ipsec.d/***
    - ***drwxr-xr-x 2 root root 4096 2009-10-06 17:17 aacerts***
    - ***drwxr-xr-x 2 root root 4096 2009-10-06 17:17 cacerts***
    - ***drwxr-xr-x 2 root root 4096 2009-10-06 17:17 certs***
    - ***drwxr-xr-x 2 root root 4096 2009-10-06 17:17 crls***
    - ***drwxr-xr-x 2 root root 4096 2010-06-14 09:21 examples***
    - ***drwxr-xr-x 2 root root 4096 2009-10-06 17:17 ocspcerts***
    - ***drwxr-xr-x 2 root root 4096 2010-06-14 09:21 policies***
    - ***drwx------ 2 root root 4096 2009-10-06 17:17 private***
  - *Además también nos creará los archivos "/etc/ipsec.conf" y "/etc/ipsec.secrets"*

4.8.1. CONFIGURACIÓN PARA LAN A LAN POR PSK

- Para hacer esta configuración lo primero que haremos será activar el bit de forward a 1 para que puedan reenviar los paquetes a las máquinas situadas dentro de la subred interna, tanto en el cliente como en el servidor:

- # echo 1 >/proc/sys/net/ipv4/ip_forward

- Ahora modificaremos el archivo "/etc/ipsec.secret"

- # nano /etc/ipsec.secret

- y deberá quedar así:

- : PSK "CLAVE"

- Ahora pondremos los parámetros de configuración para el servidor, que en este caso se llamara "***nibbler***" y tendrá la ip "***192.168.10.158***" así que abriremos "***/etc/ipsec.conf***":

- # nano /etc/ipsec.conf

- *Quedando de la siguiente manera:*

```
# basic configuration
config setup
        interfaces=%defaultroute
          # Add connections here
conn fry
        # iniciar automaticamente
        auto=start
        # autenticacion por clave compartida
        authby=secret
        # algoritmos de cifrado y autenticacion(fase2)
        esp=3des-md5
        # algoritmos de cifrado y autenticacion(fase1)
        ike=3des-md5-modp1024
        # Direccion ip local
        left=192.168.10.158
        # Direccion red interna
        leftsubnet=172.16.0.0/24
        # Direccion ip remota
```

```
right=192.168.10.60
# Direccion red remota
rightsubnet=192.168.20.0/24
```

- Ahora para que nos lea tanto el archivo secrets como el de configuración deberemos ejecutar los siguientes comandos:

```
- /etc/init.d/ipsec start
- ipsec auto --rereadsecret
- ipsec auto --add fry
```

- Acontinuación pasaremos a configurar el otro extremo ya que este extremo ya lo tenemos listo y en esperando a que le otro extremo levante la conexión. El otro extremo se llamará ***"fry"*** y tendrá de ip ***"192.168.10.60".***

- Comenzaremos por configurearle el archivo "ipsec.secrets":

```
- # nano /etc/ipsec.secrets
- Quedará así:
        - : PSK "CLAVE"
```

- Lo siguiente que haremos será configurar el archivo "ipse.conf":

```
- # nano /etc/ipsec.conf
- Quedará así:
# basic configuration
config setup
    interfaces=%defaultroute
# Add connections here
conn %default
conn nibbler
    # iniciar automatico
    auto=start
    # atenticacion por clave compartida
    authby=secret
    # algoritmos de autenticacion y cifrado fase2
    esp=3des-md5
    # algoritmos de cifrado y autenticacion fase1
    ike=3des-md5-modp1024
    # ip local
```

left=192.168.10.60

 # subred local

leftsubnet=192.168.20.0/24

 # direccion remota

right=192.168.10.158

 # subred remota

rightsubnet=172.16.0.0/24

- Lo siguiente que haremos será ejecutar los siguientes comandos:

- /etc/init.d/ipsec start

- ipsec auto --rereadsecret

- ipsec auto --add nibbler

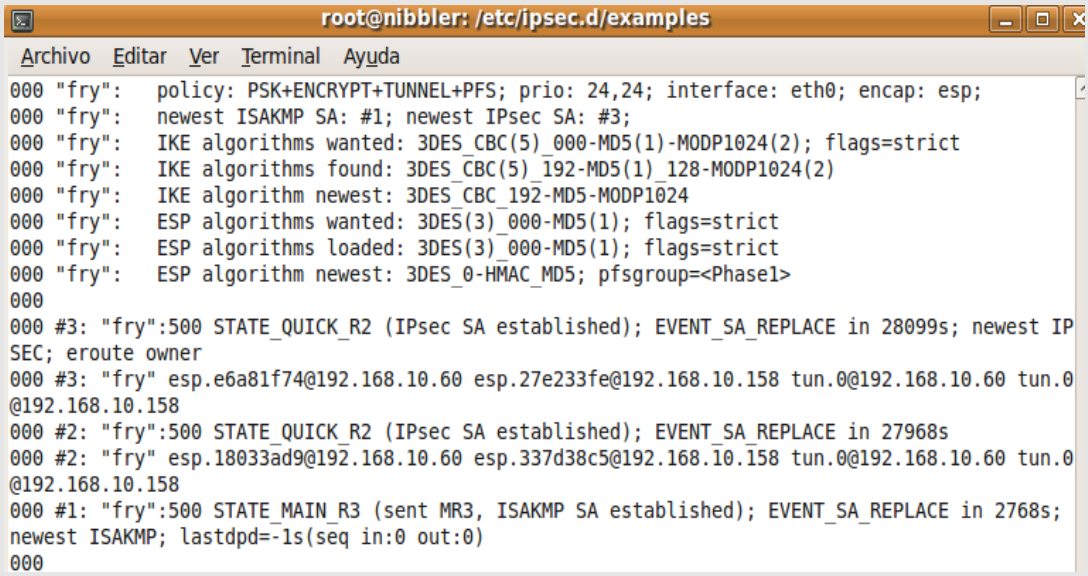
- ipsec auto --up nibbler

- Saliendo como resultado en pantalla:

```
root@fry:/etc# nano ipsec.conf
root@fry:/etc# ipsec auto --add nibbler
root@fry:/etc# ipsec auto --up nibbler
104 "nibbler" #1: STATE_MAIN_I1: initiate
003 "nibbler" #1: received Vendor ID payload [Openswan (this version) 2.4.12  LD
AP_V3 PLUTO_SENDS_VENDORID PLUTO_USES_KEYRR]
003 "nibbler" #1: received Vendor ID payload [Dead Peer Detection]
003 "nibbler" #1: received Vendor ID payload [RFC 3947] method set to=109
106 "nibbler" #1: STATE_MAIN_I2: sent MI2, expecting MR2
003 "nibbler" #1: NAT-Traversal: Result using RFC 3947 (NAT-Traversal): no NAT d
etected
108 "nibbler" #1: STATE_MAIN_I3: sent MI3, expecting MR3
004 "nibbler" #1: STATE_MAIN_I4: ISAKMP SA established {auth=OAKLEY_PRESHARED_KE
Y cipher=oakley_3des_cbc_192 prf=oakley_md5 group=modp1024}
117 "nibbler" #2: STATE_QUICK_I1: initiate
004 "nibbler" #2: STATE_QUICK_I2: sent QI2, IPsec SA established {ESP=>0x337d38c
5 <0x18033ad9 xfrm=3DES_0-HMAC_MD5 NATD=none DPD=none}
root@fry:/etc# ipsec auto --up nibbler
117 "nibbler" #3: STATE_QUICK_I1: initiate
004 "nibbler" #3: STATE_QUICK_I2: sent QI2, IPsec SA established {ESP=>0x27e233fe <0xe6a8
1f74 xfrm=3DES_0-HMAC_MD5 NATD=none DPD=none}
root@fry:/etc#
```

- Nos dirigimos hacia el otro extremo ***("nibbler")***, y realizaremos lo siguiente:

- # ipsec auto --status



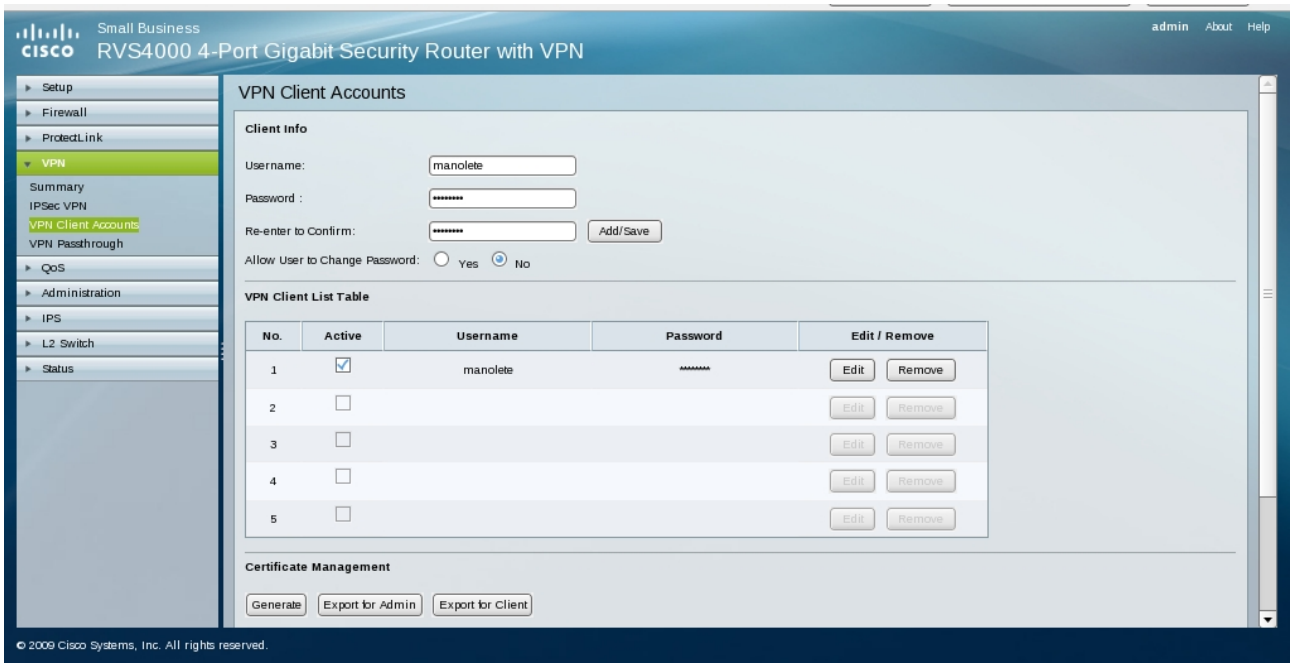
- # route

```
root@nibbler:/etc/ipsec.d/examples# route
Tabla de rutas IP del núcleo
Destino         Pasarela        Genmask         Indic Métric Ref    Uso Interfaz
192.168.20.0    *               255.255.255.0   U     0      0        0 eth0
172.16.0.0      *               255.255.255.0   U     1      0        0 eth2
192.168.10.0    *               255.255.255.0   U     1      0        0 eth0
link-local      *               255.255.0.0     U     1000   0        0 eth2
default         portatil-ubuntu 0.0.0.0         UG    0      0        0 eth0
root@nibbler:/etc/ipsec.d/examples#
```

5. CONFIGURACIÓN DE UNA VPN UTILIZANDO CISCO

5.1. CONFIGURACIÓN DE ROUTER CISCO RVS4000

- La instalación de esta VPN es sencilla ya que nosotros tampoco necesitabamos una instalación de gran complejidad, aunque este router te permite hacer desde túneles muy sencillos como el nuestro a túneles más complejos en los cuáles debes de especificar hasta el más mínimo detalle, así que en este apartado voy a explicar una instalación sencilla ya que por el tiempo que tenía para abarcar el proyecto no me daba para explicarlo y abarcarlo todo.
- Lo primero será entrar a la administración del router, por defecto, la ip es: ***192.168.1.1***, el usuario es: ***admin*** y la contraseña: ***admin.***
- Una vez dentro lo primero que haremos será crearnos un usuario para el cliente, para hacerlo entraremose en los menús:
  VPN – VPN Client Accounts , Para esta conexión nos crearemos un usuario llamado "***manolete"*** y de contraseña "***manolete***", lo escribiremos en el pequeño formulario que nos aparece pidiendo nombre y contraseña, una vez lo hayamos escrito pulsaremos sobre "***Add/Save***" y observaremos que el usuario se nos a creado y se nos ha puesto en la tabla de abajo.



- Ahora pulsaremos sobre el botón de la parte inferior que pone "***Generate***" esperaremos unos segundos, y esto nos habrá generado un nuevo certificado para nuestro router, y si nos fijamos justamente abajo nos dirá cuando se generó el último certificado.

- Ahora pulsaremos sobre el botón **"*export for client*"** , esto nos exportará el certificado que debemos dar al cliente, asi que lo descargaremos y lo guardaremos para entregarselo a él.

- Una vez echo esto y habiendo ya echo las demás configuraciones generales para cualquier router como pueden ser: cambiar la ip de la lan, desactivar dhcp, etc... ya tendremos nuestro cisco preparado y esperando una conexión.

- Tal y como habia dicho antes esta instalación es bastante sencilla como habeis comprobado ya que con solo ya tenemos el cisco preparado para realizar una conexión.

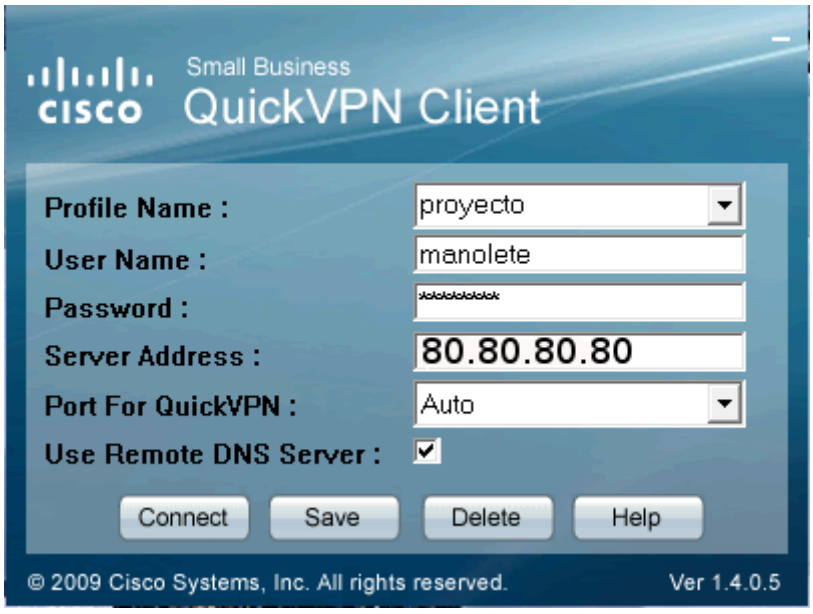
5.2. CONFIGURACIÓN DEL CLIENTE

- Para preparar al cliente también es bastante sencillo ya que el mismo cisco te facilita un CD donde viene un software llamado: **QuickVPN** , que será el que utilizemos para realizar la conexión al servidor.
- Para instalarlo bastará con introducir el cd buscar el ejecutable para la instalación de QuickVPN, procederemos a su instalación en un cliente windows que sería todo por defecto, osea, siguiente siguiente siguiente... como el 80% de las aplicaciones para windows.
- Nos creará un acceso directo en el escritorio:

- Antes de abrirlo para conectarnos deberemos copiar el certificado anteriormente exportado desde el cisco, y lo pegaremos en la ruta: "***C:/Archivos de programas/Cisco Small Business/QuickVPN/"***

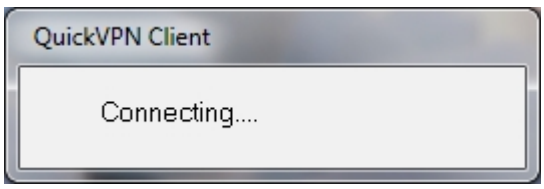
- Una vez lo copiemos ya nos podremos conectar, para ello abriremos la aplicación y rellenaremos los datos requeridos:

  - **Profile:** proyecto → Podrá ser el que queramos.
  - **Username:** manolete → usuario dado de alta en el cisco.
  - **Password:** ******** → contraseña que le pusimos al usuario(“**manolete”**)
  - **Server Address:** 80.80.80.80 → ip del servidor
  - **Port for QuickVPN:** auto → el puerto que va a utilizar para conectarse como no tenemos ninguna preferencia dejaremos que se conecte por el que él localice.
  - **Use Remote DNS Server** → Lo activaremos si queremos utilizar el S.dns remoto, si no lo desactivamos.

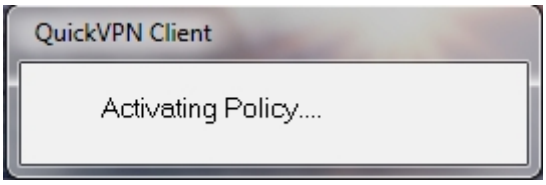


- Ahora pulsaremos “***Save”*** si queremos que nos recuerde la información, esto no lo recordará todo menos la contraseña por eso es seguro tenerlo guardado, una vez guardemos los datos pulsaremos en “***Connect*** “ y nos irá pasando por distintos pasos:

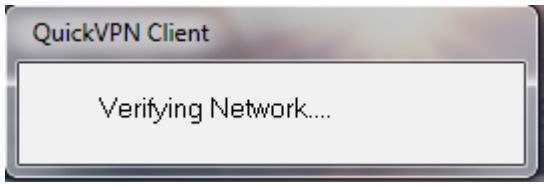
  1. En el primer paso mira que el servidor este levantado, que poseemos certificado para la conexión, y comprobará que el usuario y contraseña estan correctos.



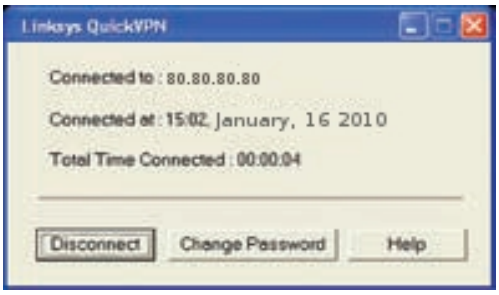
2. En el segundo paso comprobará que nuestra ip no dará conflictos con las de la red del otro extremo ya que estas deben estar en distinto rango, de lo contrario fallará, una vez comprobado que no haya conflictos nos asignará una ip virtual.



3. En el siguiente paso lo que comprueba es que tenemos ping a la subred remota sin problemas.



4. Una vez pasados estos pasos nos saldrá un cuadrado parecido a este en el cuál indica la ipa a la que se ha conectado, cuando se ha conectado y el tiempo que lleva conectado, también como podemos observar en la imagen tendremos la opción de cambiar la contraseña al usuario desde la misma aplicación sin necesidad de entrar al router para cambiarla.



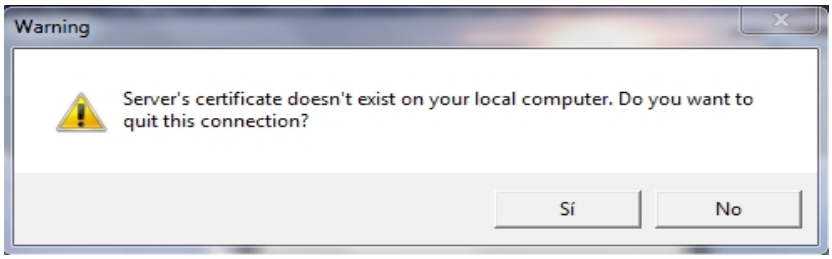
- Podremos ver el estado sin necesidad de tener que abrir esta ventana ya que en la parte inferior derecha de la pantalla a la izquierda de la del reloj, los iconos serán los siguientes:

- Si está desconectado: 

- Si está conectado: 

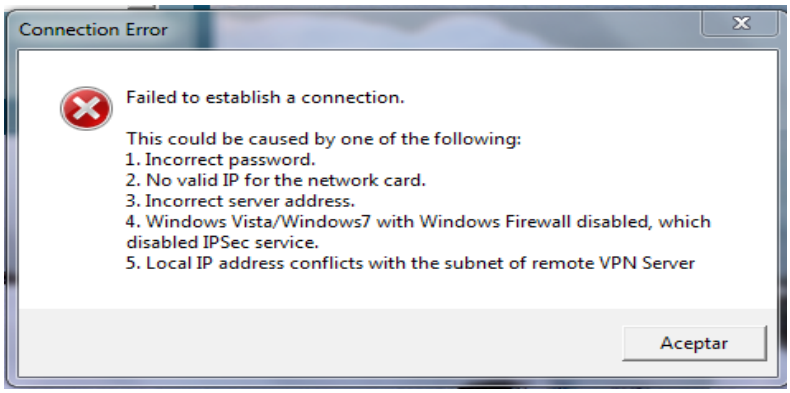
5.3. POSIBLES ERRORES DURANTE LA CONEXIÓN

- El primer error será este:



- Este error no impedirá que nos podamos conectar, pero esto nos saldrá cuando el vaya a comprobar si tenemos certificado y no lo encuentre o bien el que encuentra no es correcto. Si le respondemos que "NO" seguirá con la conexión y si lo demás está correcto conectará, pero el problema será que los datos no viajarán cifrados.

- El segundo error que vamos a ver nos saldrá mientras conecta o comprueba el usuario y es este:



- Como podemos leer en el mismo cuadro este error podrá ser producido porque o bien hayamos introducido el nombre de usuario mal o la contraseña, también lo puede provocar si la dirección que hemos indicado no es la correcta o tiene algún tipo de problema a la hora de buscar el router, si utilizamos windows vista o 7 tendremos que tener el cortafuegos activado(todo lo contrario que con xp) ya que en este es donde se activa el servicio IPSec o como nombramos en uno de los puntos anterior, este error lo podría causar que nuestro rango ip sea el mismo que el de la subnet remota y eso daría conflicto.

- Otro error puede ser que nos diga que la puerta remota no responde, y este error lo lanza cuando el durante el proceso de "***verifying network***" Comprueba los ping a la red remota y por cualquier motivo estos no estan llegando correctamente. Este error será común sobre todo en los windows vista y 7, ya que para que estos funcionen correctamente el cisco será el que tenga que estar en el otro lado directamente conectado a internet, no valdrá que tenga otro router delante, como en algunas ocaciones que el router de la compañía este delante es obligatorio deberemos de configurarlos como routers transparentes así será como si no estuviesen y el cisco será el que esté directamente conectado, desapareciendo así este problema.

6. CONCLUSIÓN

- Sobre KVM cuando empecé me creía que era un proyecto que aún estaba muy verde, pero a medida que he ido avanzando en el proyecto me me ha quedado un poco más que lo que realmente esta verde es la administración gráfica de libvirt, aunque la administración por consola la verdad que va muy bien ya que a raíz que fuí administrando kvm solo desde consola no ha vuelto a fallar.

- OpenVPN me ha parecido muy buena aplicación ya que su instalación y administración es bastante sencilla y funciona perfectamente, después trae herramientas propias para generar certificados y demás, en general era una aplicación que no habia utilizado nunca y me ha gustado bastante.

- OpenSwan(IPSec) me ha costado más trabajo sacarlo adelante aunque al final ha salido, su administración me parace algo más compleja que la de openvpn, aunque también me ha parecido una aplicación que abarca mucho más y me ha parecido más segura que openvpn gracias a su abaníco de opciones para elegir modos de conexión y de autenticación, así que una de las cosas que más me ha gustado de openswan es que puede ser todo lo complejo o sencillo que tú quieras porque no hay que seguir ninguna plantilla ni ningunos parametros fijos, cada una de las conexiones son distintas según la necesidad de cada una, entonces es algo que me ha gustado mucho que es una aplicación muy flexible.

- Además Este proyecto en lo personal me ha dado mucha confianza en mí mismo ya que practicamente haces el proyecto tú, buscando soluciones por tí mismo o en foros o preguntando y cuando ya lo ves todo terminado y funcionando te da mucha satisfacción. Al principio de empezar el proyecto creo que todos pensamos que no valía para nada ya que si habiamos demostrado durante el curso lo que sabíamos a que venía ahora un proyecto, pero después te das cuentas de que te aporta mucho en lo personal sobre todo en lo moral ya que te anima mucho ver funcionando algo que has montado tú "solo".

7. BIBLIOGRAFÍA

- www.linux-kvm.org
- https://help.ubuntu.com/9.10/serverguide/C/libvirt.html
- http://www.howtoforge.com/virtualization-with-kvm-on-ubuntu-9.04
- http://blog.jorgeivanmeza.com/2010/01/administrar-maquinas-virtuales-con-kvm-desde-linea-de-comando-en-ubuntu-9-10/
- http://sw-libre.blogspot.com/2010/04/openvpn-configuracion-de-vpn-basada-en.html
- http://www.ecualug.org/?q=2007/02/06/comos/centos/c_mo_instalar_y_configurar_openvpn
- **Libro titulado:** Redes Privadas Virtuales de Javier Andrés Alonso de la editorial RA-MA
- Además de algun que otro foro buscado en Google que no tengo sus referencias.