Trabajo Práctico 4 - IPSec con Certificados

Integrantes:

Madariaga, Eduardo	90824	madariagaedu@gmail.com
Canavari, Agustín	91599	agustincanavari@gmail.cm
López Gallo, Marcelo	92563	marce.lopezgallo@gmail.com

Objetivo del Trabajo Práctico

El presente trabajo práctico tiene por objetivo principal la creación de una entidad certificante para el establecimiento de una conexión con IP Sec, utilizando el conjunto de scripts *easy-rsa* incluídos en la distribución de Linux provista por la cátedra.

Proceso de Implementación

1. Creación de una autoridad certificante
2. Esta entidad emitirá dos certificados, que serán utilizados en una conexión autenticada de ipsec.
3. Se anulará uno de los certificados generando la CRL correspondiente, la conexión no podrá establecerse .
4. Se emitirá un nuevo certificado de servidor para iniciar la conexión correctamente.

Al igual que en el trabajo previo de IPSec sin certificados, la implementación del trabajo se realizará con software de virtualización de dispositivos como se muestra en el esquema de la figura 1. El túnel a implementar estará en la red 192.168.10.0, y la autoridad certificante será R1. Es decir que los certificados de R1 estarán autofirmados.

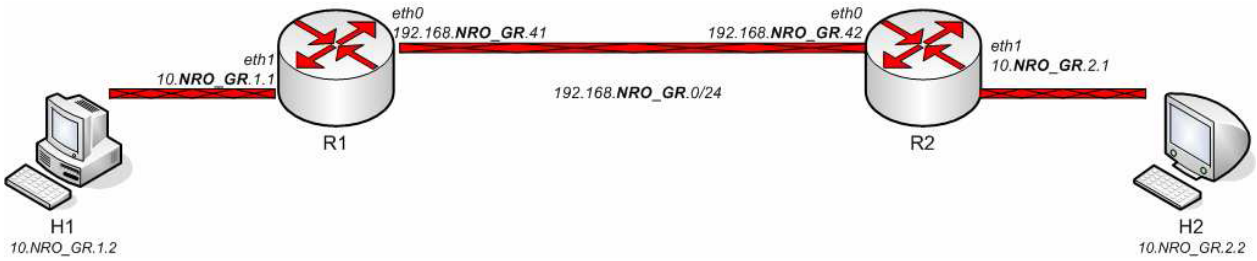

Figura 1: Esquema de conexionado de red.

Desarrollo del Trabajo Práctico

Creación de la Autoridad Certificante.

Inicialmente se procedió al armado de la red de prueba, como se muestra en la Figura 1. A continuación, se completaron las variables del archivo /crypto/conf/config.sh. Luego se generó en R1 la autoridad certificante mediante el script s01-generarCA.sh. que crea un certificado (ca.crt) y clave privada (ca.key) como se muestra en la Figura 2.
Finalmente, se configuró el router 2 para confiar en R1 como autoridad certificante.

```
>>>>>> ////////////////////////////////////////////
>>>>>> //// Generando nueva autoridad certificante
>>>>>> ejecutando /crypto/easy-rsa/build-ca
>>>>>> ////////////////////////////////////////////
Generating a 1024 bit RSA private key
...++++++
.....++++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AR]:
State or Province Name (full name) [BA]:
Locality Name (eg, city) [Buenos Aires]:
Organization Name (eg, company) [6669-Seguridad Redes]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:algo
Email Address [crypto@fi.uba.ar]:
>>>>>> listo.
user@crypto:/crypto/ipsec-CA$ _
```

Figura 2: s01-generarCA.sh.

Emisión de certificados e inicialización de enlace bajo IPSec.

A- Certificados

Primero se generó un par de claves (pública y privada) para cada router para solicitar la certificación de las identidades de R1 y R2. Luego se diferenció el common name de cada dispositivo y para alcanzar la certificación se verificó que coincida la fecha y hora de todos los dispositivos.

En la Figura 3 se muestran los resultados de realizar el encriptado (o la firma) de las solicitudes de la identidad de R1 y R2, creando de este modo los certificados.

```
>>>>>> ////////////////////////////////
>>>>>> ////   Firmando los certificados
>>>>>> Ejecutando /crypto/easy-rsa/sign-req r2
Using configuration from /crypto/easy-rsa/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'AR'
stateOrProvinceName   :PRINTABLE:'BA'
localityName          :PRINTABLE:'Buenos Aires'
organizationName      :PRINTABLE:'6669-Seguridad Redes'
commonName            :PRINTABLE:'r2'
emailAddress          :IA5STRING:'crypto@fi.uba.ar'
Certificate is to be certified until May 30 21:36:36 2024 GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated

>>>>>> ////////////////////////////////////////
>>>>>> copiando los certificados y moviendo claves
+ cp /crypto/var/rootCA/r1.crt /crypto/var/routers/r1.pem
+ cp /crypto/var/rootCA/r2.crt /crypto/var/routers/r2.pem
+ mv /crypto/var/rootCA/r1.key /crypto/var/routers
+ mv /crypto/var/rootCA/r2.key /crypto/var/routers
+ set +x
>>>>>> listo.
crypto:/crypto/ipsec-CA#
```

Figura 3: s03-firmaSolicitudes.sh

En la Figura 4 se observan varios archivos en el directorio de la Autoridad Certificante, /crypto/var/rootCA, entre ellos:

- Serial, Contiene una sola línea con el número de serie del próximo certificado a generar. (“01”)
- index.txt, Se genera vacío. Luego se va incorporando el hash y el cn de cada certificado generado.
- Ca.crt y ca.key, el certificado y clave de la autoridad certificante.
- R1.crt, r1.key el certificado autofirmado de r1, y
- R2.crt, r2.key el certificado del router 2.

De los cuales los dos últimos pares de certificados – claves de R1 y R2 son copiados satisfactoriamente al directorio /crypto/var/routers.

```
crypto:/crypto/var/rootCA# ls
01.pem  ca.crt  index.txt       index.txt.attr.old  r1.crt  r2.crt  serial
02.pem  ca.key  index.txt.attr  index.txt.old       r1.csr  r2.csr  serial.old
crypto:/crypto/var/rootCA# cd /crypto/var/routers/
crypto:/crypto/var/routers# ls
r1.key  r1.pem  r2.key  r2.pem
crypto:/crypto/var/routers#
```

Figura 4: archivos contenidos en los directorios de la CA y en el directorio de los routers.

Luego verificamos que la lista de certificados revocados está vacía mediante el script s04-generaCRL. (Figura 5).

```
crypto:/crypto/ipsec-CA# ./s04-generaCRL.sh

>>>>>> ////////////////////////////////////////
>>>>>> Generando lista de certificados revocados
>>>>>> Ejecutando /crypto/easy-rsa/make-crl crl.pem
Using configuration from /crypto/easy-rsa/openssl.cnf
>>>>>> listo.
```

Figura 5: s05-generaCRL.sh

En el paso siguiente, se utilizó el comando grep para recorrer los ficheros de cada certificado y así obtener el nombre de los dueños de los certificados.

```
crypto:/crypto/ipsec-CA# ./s05-getIDs.sh
>>>>>> ////////////////////////////////////
>>>>>> Obteniendo datos de los certificados
       leftid="C=AR, ST=BA, O=6669-Seguridad Redes, CN=r1/emailAddress=crypto@fi.uba.ar"
       rightid="C=AR, ST=BA, O=6669-Seguridad Redes, CN=r2/emailAddress=crypto@fi.uba.ar"
>>>>>> listo.
crypto:/crypto/ipsec-CA#
```

Figura 6: s05-getIDs.sh

B-Configuración enlace IPSec

Primero se configuraron las interfaces del equipo 1 con el comando *ifconfig* y los datos de la autoridad certificante (Figura 7).

```
crypto:/crypto/ipsec-CA# ./s00-configurarInterfacesLocales.sh

>>>>>> ////////////////////////
>>>>>> Configurando Interfaces
>>>>>> Hostname R1 configurado
>>>>>> Configurando Interfaz publica eth0
>>>>>>    ifconfig eth0 192.168.10.41 netmask 255.255.255.0
>>>>>> Configurando Interfaz privada eth1
>>>>>>    ifconfig eth1 10.10.1.1 netmask 255.255.255.0
>>>>>> Tabla de hosts creada (/etc/hosts)
>>>>>> listo.
```

Figura 7: s00-configurarInterfacesLocales.sh

Luego se copiaron los certificados del router 1, de la autoridad certificante y la lista de certificados revocados (hasta el momento vacía) para que sean utilizados en el establecimiento del enlace IPSec (Figura 8).

```
crypto:/crypto/ipsec-CA# ./s06-instalaCertificados.sh

>>>>>> ////////////////////////////////////////////////
>>>>>> Borrando previos certificados en /etc/ipsec.d/certs
>>>>>> Instalando los certificados al router 1
+ cp -f /crypto/var/rootCA/ca.crt /etc/ipsec.d/cacerts/cacert.pem
+ cp -f /crypto/var/rootCA/crl.pem /etc/ipsec.d/crls
+ cp -f /crypto/var/routers/r1.pem /etc/ipsec.d/certs
+ cp -f /crypto/var/routers/r1.key /etc/ipsec.d/private
+ set +x
>>>>>> listo.
crypto:/crypto/ipsec-CA#
```

Figura 8: s06-instalaCertificados.sh

En la figura 9 se preparan: clave y certificado de R1 y R2, clave y certificado de R1 y R2, clave y certificado de R1 y R2, certificado de la autoridad certificante para que R2 obtenga su certificado remotamente.

```
crypto:/crypto/ipsec-CA# ./s07-preparaArchivo.sh

>>>>>> ////////////////////////////////////////////
>>>>>> Perparando archivo para host remoto, router 2
+ cp -f /crypto/var/rootCA/ca.crt /crypto/var/routers/cacert.pem
+ cp -f /crypto/var/rootCA/crl.pem /crypto/var/routers
+ cp -f /crypto/var/ids /crypto/var/routers
+ zip /crypto/var/forr2.zip -r /crypto/var/routers
  adding: crypto/var/routers/ (stored 0%)
  adding: crypto/var/routers/ids (deflated 43%)
  adding: crypto/var/routers/crl.pem (deflated 22%)
  adding: crypto/var/routers/cacert.pem (deflated 36%)
  adding: crypto/var/routers/r2.key (deflated 21%)
  adding: crypto/var/routers/r1.key (deflated 22%)
  adding: crypto/var/routers/r2.pem (deflated 45%)
  adding: crypto/var/routers/r1.pem (deflated 45%)
+ set +x
>>>>>> Borrando directorio
>>>>>> listo.
crypto:/crypto/ipsec-CA#
```

Figura 9: s07-preparaArchivo.sh

En la figura 10 se ve el resultado de la configuración de las interfaces para R2

```
crypto:/crypto# cd ipsec-CA/
crypto:/crypto/ipsec-CA# ./s08r-configurarInterfacesRemotas.sh

>>>>>> ////////////////////////
>>>>>> Configurando Interfaces
>>>>>> Hostname R2 configurado
>>>>>> Configurando Interfaz publica eth0
>>>>>>     ifconfig eth0 192.168.10.42 netmask 255.255.255.0
>>>>>> Configurando Interfaz privada eth1
>>>>>>     ifconfig eth1 10.10.2.1 netmask 255.255.255.0
>>>>>> Tabla de hosts creada (/etc/hosts)
>>>>>> Terminando procesos IPSEC
NET: Unregistered protocol family 15
ipsec_setup: Stopping Openswan IPsec...
>>>>>> listo.
```

Figura 10: s08r-configurarInterfacesRemotas.sh

Una vez configurada la interfaz de R2, se realiza la copia del archivo zip con los certificados, y los instala en las ubicaciones correspondientes. Para ello, se utilizó el comando SecureCopy (scp, que permite que puedan copiarse archivos entre diferentes host, utilizando transferencia de datos por ssh). Ver Figura 11.

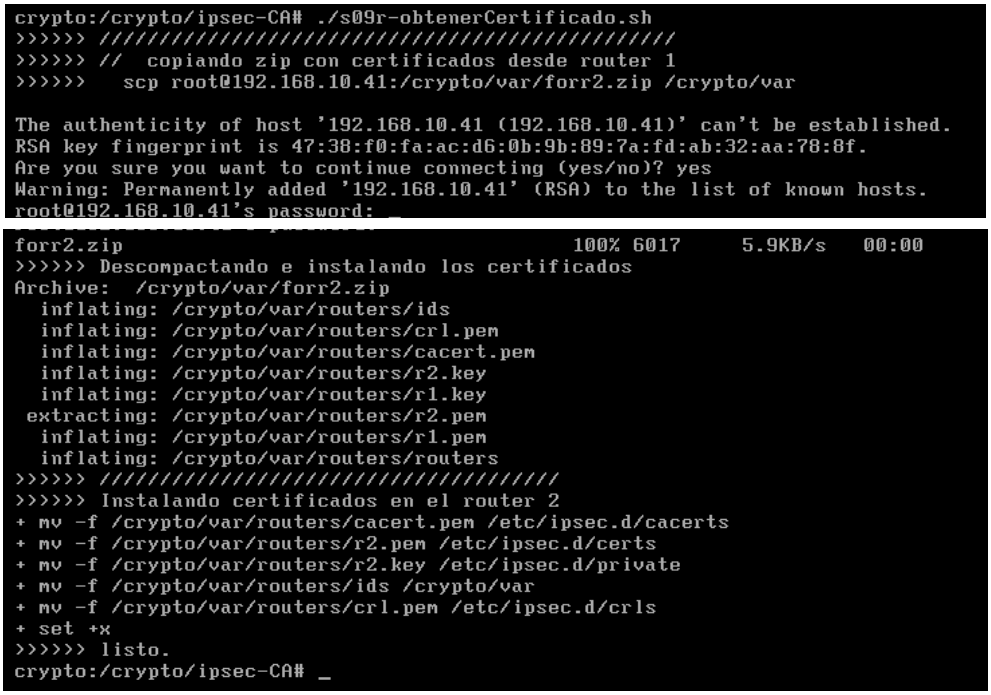

Figura 11: s09-obtenerCertificado.sh

Una vez configuradas las interfaces e instalados los certificados en ambos equipos, el paso siguiente es configurar el enlace IPSec, creando y completando el fichero /etc/ipsec.conf, en cada uno de los equipos (R1 y R2). Para ello, se ejecutó el script 10 (Figura 12). Analizando el código de dicho script, se nota que el empleo de certificados está dado por los campos leftrsasigkey y rightrsasigkey donde se introdujo el valor %cert, el cual indica que las claves públicas utilizadas se cargan desde certificados (cuya ubicación está dada por leftcert y rightcert, respectivamente).

Figura 12: s10c-generarIPsecConf.sh

Por último, con el script s11c-generarIPsecSec.sh (Figura 13), se genera el archivo donde se guarda la clave necesaria para decodificar la clave privada de cada respectivo equipo.

```
crypto:/crypto/ipsec-CA# ./s11c-generarIPsecSec.sh
>>>>>> Configurando Interfaces
>>>>>> usando archivo: /etc/ipsec.secrets
Ingrese la clave del certificado correspondiente al host R2: 1234
>>>>>> Generando archivo de clave de IPsec /etc/ipsec.secrets con clave 1234
>>>>>> listo.
crypto:/crypto/ipsec-CA# _
```

Figura 13: s11c-generarIPsecSec.sh

Una vez realizado esto, se puede iniciar el servicio IPSec en ambos routers, con el comando

/etc/init.d/ipsec start

Y el túnel se levanta desde alguno de los routers con el comando:

ipsec auto --up crypto

```
crypto:/crypto/ipsec-CA# cd /etc/init.d/
crypto:/etc/init.d# ./ipsec auto --up crypto
Usage: ipsec setup {--start|--stop|--restart|--status}
crypto:/etc/init.d# ./ipsec start
Initializing IPsec netlink socket
ipsec_setup: Starting Openswan IPsec U2.4.6/K2.6.18-6-486...
ipsec_setup: insmod /lib/modules/2.6.18-6-486/kernel/net/ipv4/ah4.ko
ipsec_setup: insmod /lib/modules/2.6.18-6-486/kernel/net/ipv4/esp4.ko
ipsec_setup: insmod /lib/modules/2.6.18-6-486/kernel/net/ipv4/ipcomp.ko
ipsec_setup: insmod /lib/modules/2.6.18-6-486/kernel/net/ipv4/tunnel4.ko
ipsec_setup: insmod /lib/modules/2.6.18-6-486/kernel/net/ipv4/xfrm4_tunnel.ko
ipsec_setup: insmod /lib/modules/2.6.18-6-486/kernel/net/xfrm/xfrm_user.ko
crypto:/etc/init.d# _
```

Figura 14: Ipsec start

Anulación de uno de los certificados.

En esta etapa se ejecutó en R1 el script que revoca el certificado local e instala la nueva lista de certificados revocados (Figura 15). Luego se actualiza la nueva lista de certificados revocados (CRL) en R2 (Figura 16). Nuevamente, para realizar esta actualización se utiliza el comando secure copy (scp).

```
crypto:/crypto/ipsec-CA# ./srs0-revocarR1.sh
>>>>>> Revocando certificado del router 1
Using configuration from /crypto/easy-rsa/openssl.cnf
Revoking Certificate 01.
Data Base Updated
Using configuration from /crypto/easy-rsa/openssl.cnf
r1.crt: /C=AR/ST=BA/O=6669-Seguridad Redes/CN=r1/emailAddress=crypto@fi.uba.ar
error 23 at 0 depth lookup:certificate revoked
>>>>>> Instalando nueva lista de certificados revocados en el equipo local
crypto:/crypto/ipsec-CA# _
```

Figura 15: srs0-revocarR1.sh en R1

```
crypto:/crypto/ipsec-CA# ./srs1r-instalarCRLnuevo.sh
>>>>>> Instalando nueva lista de certificados revocados en el equipo local
root@192.168.10.41's password:
crl.pem                                       100%  503     0.5KB/s   00:00
crypto:/crypto/ipsec-CA# _
```

Figura 16: srs1r-instalarCRLnuevo.sh en R2

Ahora, a modo de test, se reinició el túnel e intentó levantar con el certificado revocado en R1, con lo que se verificó que surgió un error de clave: El router 2 está rechazando el certificado.

```
crypto:/crypto/ipsec-CA# /etc/init.d/ipsec restart
NET: Unregistered protocol family 15
ipsec_setup: Stopping Openswan IPsec...
NET: Registered protocol family 15
Initializing IPsec netlink socket
ipsec_setup: Starting Openswan IPsec 2.4.6...
ipsec_setup: insmod /lib/modules/2.6.18-6-486/kernel/net/key/af_key.ko
ipsec_setup: insmod /lib/modules/2.6.18-6-486/kernel/net/ipv4/xfrm4_tunnel.ko
ipsec_setup: insmod /lib/modules/2.6.18-6-486/kernel/net/xfrm/xfrm_user.ko
crypto:/crypto/ipsec-CA# ipsec auto --up crypto
104 "crypto" #2: STATE_MAIN_I1: initiate
003 "crypto" #2: received Vendor ID payload [Openswan (this version) 2.4.6  X.50
9-1.5.4 LDAP_V3 PLUTO_SENDS_VENDORID PLUTO_USES_KEYRR]
003 "crypto" #2: received Vendor ID payload [Dead Peer Detection]
106 "crypto" #2: STATE_MAIN_I2: sent MI2, expecting MR2
108 "crypto" #2: STATE_MAIN_I3: sent MI3, expecting MR3
003 "crypto" #2: no RSA public key known for 'C=AR, ST=BA, O=6669-Seguridad Rede
s, CN=r1, E=crypto@fi.uba.ar'
217 "crypto" #2: STATE_MAIN_I3: INVALID_KEY_INFORMATION
_
```

Figura 17: Intento de establecimiento del tunel con el certificado de R1 revocado.

Emisión de un nuevo certificado

Luego se generó el nuevo certificado y se levantó la conexión mediante el túnel, como lo indica la Figura 18 a y b.

```
crypto:/crypto/ipsec-CA# ./srs2-generarNuevoCertificado.sh
>>>>>> Generando un nuevo certificado. Ingresar los mismos datos que en el certi
ficado previo (y mismo password)
Generating a 1024 bit RSA private key
.......++++++
.................++++++
writing new private key to 'r1n.key'
Enter PEM pass phrase:_
```

Figura 18a: Generación de nuevo certificado de R1 y establecimiento del tunel.

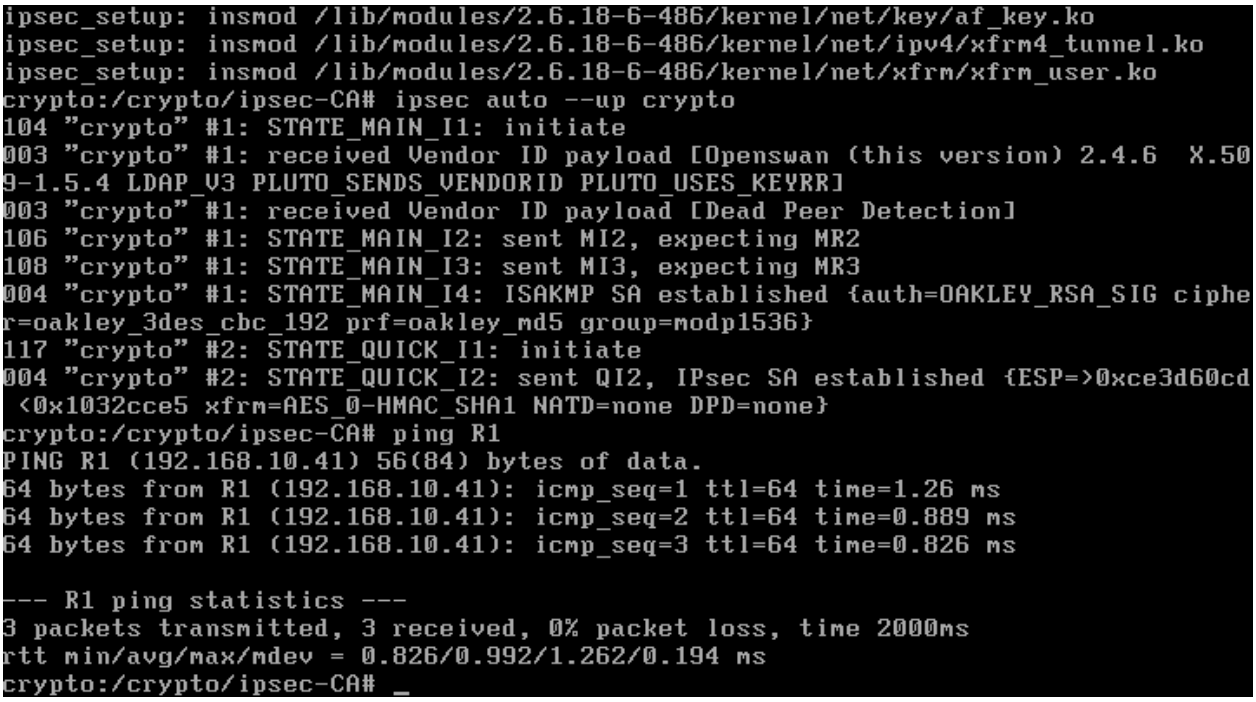

Figura 18b: Generación de nuevo certificado de R1 y establecimiento del tunel.

Conclusión

A lo largo del trabajo práctico, se estudió cómo establecer un canal seguro entre dos puntos mediante el protocolo IPSec, con y sin certificados. Se establecieron procedimientos para generar claves privada y pública mediante RSA y utilizarlas para crear un canal seguro generando claves IPSec y transportándolas mediante un comando seguro de punto a punto. La ventaja que se puede deducir respecto al canal generado sin el uso de certificados, es que se puede controlar qué usuario puede transmitir/recibir información desde un tercero (autoridad certificante). Esto es no solo útil a la hora de detectar posibles usuarios maliciosos y denegarles acceso rápidamente, sino que también a la hora de controlar tráficos de mayor volumen donde no es necesario que los usuarios estén intercambiando claves continuamente entre ellos.
En cuanto a la implementación del trabajo práctico, nos encontramos con algunos problemas ya que nunca habíamos utilizado virtualizadores con características para configurar redes. La configuración de los parámetros de cada pc requirió de bastantes pruebas de ping hasta lograr la interconexión. Por otro lado, fue necesario asignar más memoria RAM al router virtual que capturaba el tráfico de paquetes ya que éste requirió el uso de entorno gráfico para el programa Wireshark.