



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

86.36 - CRIPTOGRAFÍA Y SEGURIDAD INFORMÁTICA

Tunel IPsec

Alumnos:

DI VITO, Ivan Mariano (95722)

LOCANI, Leandro (84071)

RINALDI, Maximiliano Nahuel (91825)

26 de Junio de 2018

Índice

1. Introducción	2
2. VirtualBox	2
2.1. Creación de máquinas virtuales	2
2.2. Creación de las redes	3
2.3. Asignación de las redes	3
3. Configuración	4
3.1. Main Mode	9
3.2. Quick Mode	9
3.3. Conclusiones de la captura	10
3.4. Descifrado de paquetes	10
4. Certificados	12
4.1. Revocar certificado	15
4.2. Generación de un nuevo certificado	15
4.3. TCPDUMP RSA con/sin certificados	16
4.4. Certificados Apache	17
5. Conclusiones	20
6. Índice de Figuras	21
7. Bibliografía	22

1. Introducción

El presente trabajo práctico consistirá en configurar un tunel Isec entre dos PCs funcionando como routers y analizar, utilizando el analizador de protocolos, el tráfico generado por ambos equipos. Se utilizará para tal fin la distribución de Linux provista por la cátedra. Se contrastará con un caso donde el túnel utilice certificados generados con una autoridad certificanete propia.

2. VirtualBox

Para hacer la pruebas antes mencionadas se trabajará con la plataforma de virtualización VirtualBox.

Se requieren cuatro máquinas virtuales, dos router, dos hosts y 3 redes independientes entre sí acorde al diagrama número 2.1.

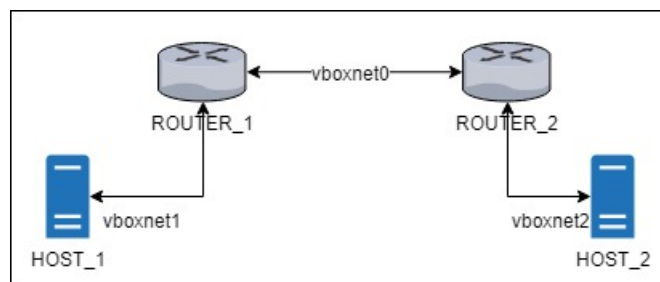


Figura 2.1: Entorno de trabajo virtualbox

2.1. Creación de máquinas virtuales

- hacer click en new
- hacer click en *expert mode*
- asignar un nombre en el campo name(HOST_1, ROUTER_1, ROUTER_2, HOST_2)
- en el campo type poner *Linux*
- en el campo version poner *Debian(32-bit)*
- dejar la cantidad de memoria por defecto, no hay que asignar mucho porque hay que levantar 4
- hacer click en la opcion *do not add virtual disk*
- hacer click en create
- Por ultimo hay que agregar el iso para que bootee para esto hacer click derecho → settings → storage, seleccionar el disco que dice Empty dentro de Controller:Ide y en attributes → Optical Drive hacer click en el icono del cd, en el menu desplegable seleccionar a imagen binary.iso y si no aparece porque es la primera vez hacer click en Chose virtual optical Disk file y buscar el archivo. Por ultimo darle a Ok.

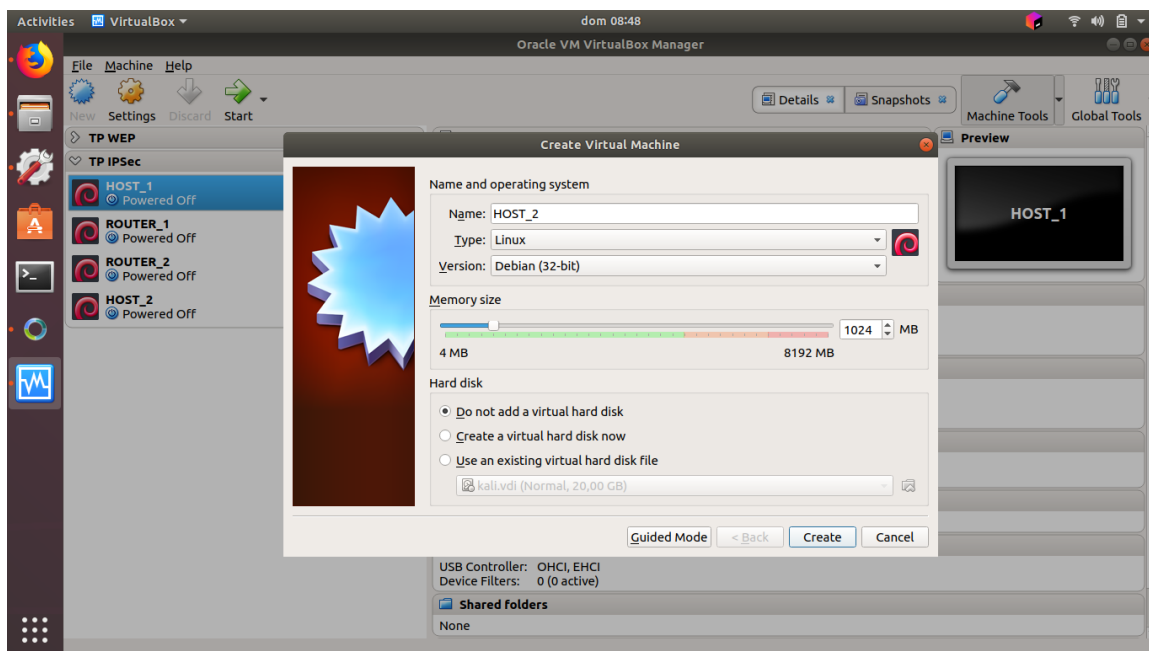


Figura 2.2: Creación máquina virtual

2.2. Creación de las redes

- Ir a file → Host network manager
- hacer click en create 3 veces
- revisar que las 3 redes esten con DHCP server deshabilitado y su direccion/mascara sean: (192.168.56.1/24, 192.168.57.1/24, 192.168.58.1/24)
- Tambien revisar que en propiedades de cada red esté activada la opcion **configure Adapter Manually**

2.3. Asignación de las redes

Para asignar las redes como en el diagrama de la figura 2.1.

- Click derecho en la VM HOST 1 → settings → network → adapter 1
- Revisar que este activado **Enable Network adapter**, en Attached to elegir **Host-only Adapter** y en Name vboxnet1, luego dar a Ok
- Click derecho en la VM ROUTER 1 → settings → network → adapter 1
- Revisar que este activado **Enable Network adapter**, en Attached to elegir **Host-only Adapter** y en Name vboxnet0
- Seleccionar Adapter 2, habilitarlo, en Attached to elegir **Host-only Adapter** y en Name vboxnet1 y luego dar a Ok
- Click derecho en la VM ROUTER 2 → settings → network → adapter 1
- Revisar que este activado **Enable Network adapter**, en Attached to elegir **Host-only Adapter** y en Name vboxnet0
- Seleccionar Adapter 2, habilitarlo, en Attached to elegir **Host-only Adapter** y en Name vboxnet2 y luego dar a Ok
- Click derecho en la VM HOST 2 → settings → network → adapter 1

- Revisar que este activado **Enable Network adapter**, en Attached to elegir **Host-only Adapter** y en Name vboxnet2, luego dar a Ok

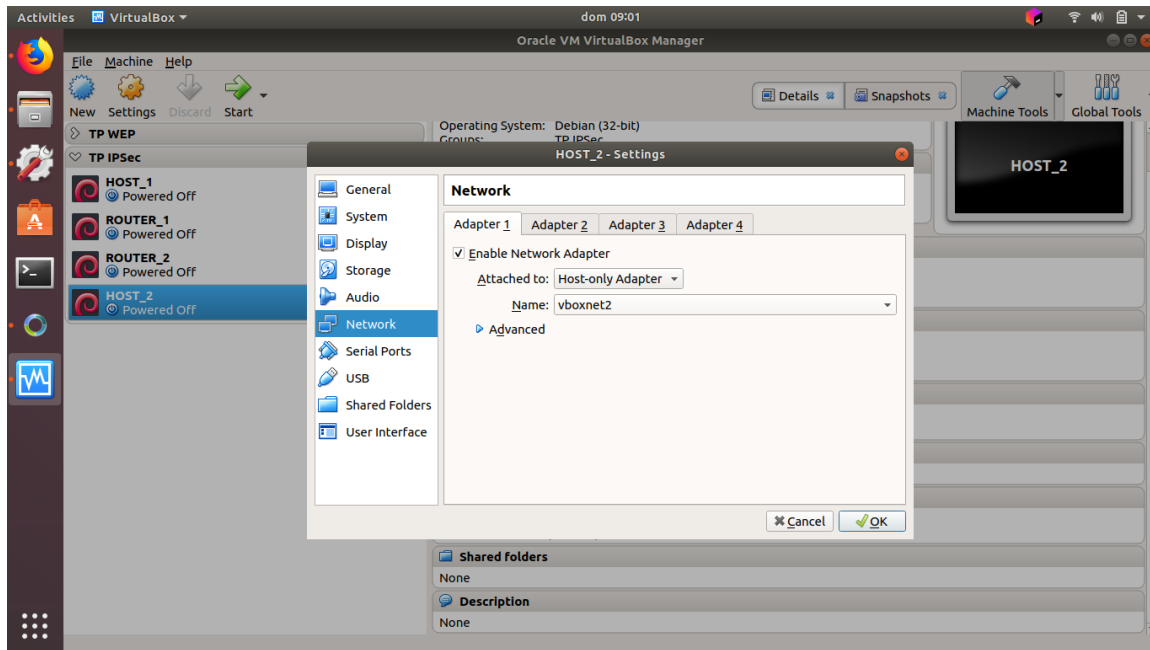


Figura 2.3: Asignación de redes

3. Configuración

Iniciar las cuatro máquinas virtuales haciendoles doble click a cada una, puede demorar unos minutos. Puede iniciarse la interfaz gráfica ejecutando el comando `startx`.

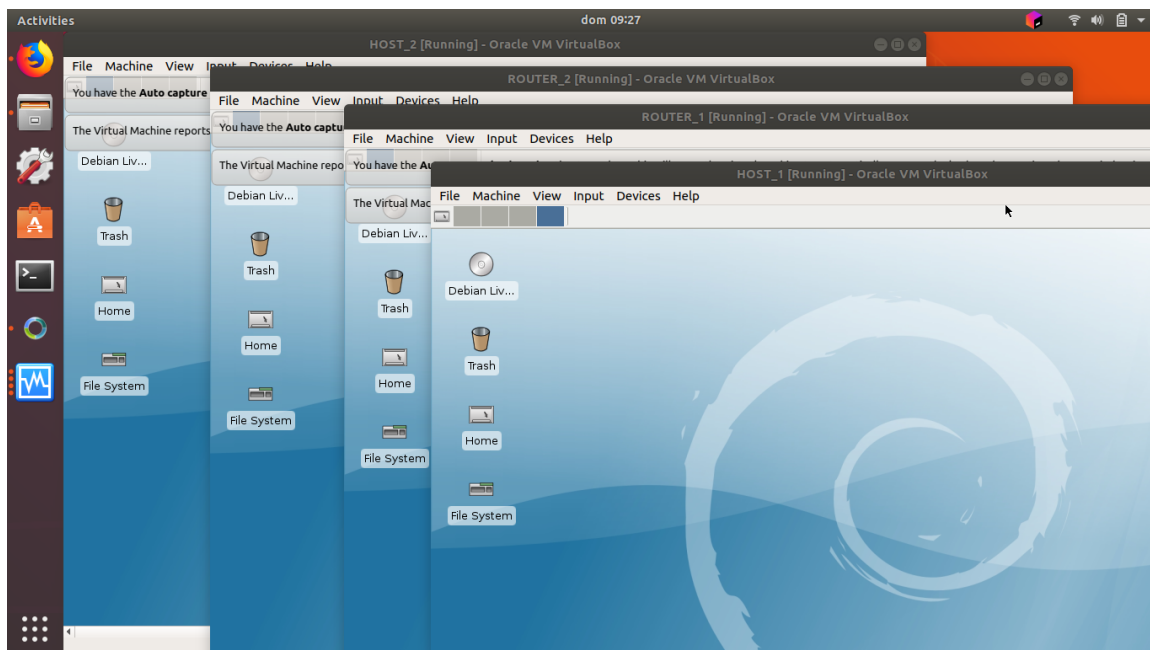


Figura 3.1: VMS Iniciadas

Lo primero que hay que hacer es configurar las maquinas con el archivo **/crypto/conf/config.sh** . Antes de correrlo hay que editarlo con el numero de grupo en nuestro caso el dos(2). Esto mismo debe hacerse en todas las vms.

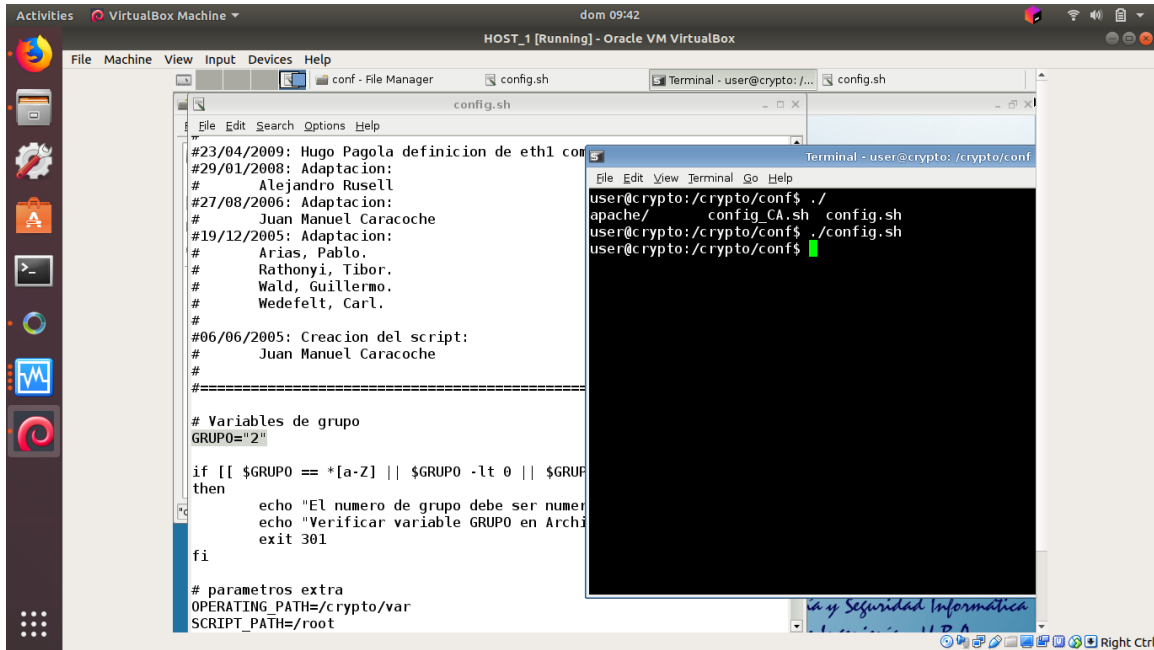


Figura 3.2: Archivo de configuración inicial

Ejecutar los scripts que estan en la carpeta **/crypto/ipsec** utilizando `sudo` ya que se requieren privilegios para hacerlo

```
HOST_1: sudo ./H1-preparar.sh
ROUTER_1: sudo ./R1-1-preparar.sh
ROUTER_2: sudo ./R2-1-preparar.sh
HOST_2: sudo ./H2-preparar.sh
```

Con esto hecho puede hacerse ping desde una vm a todas las demás.

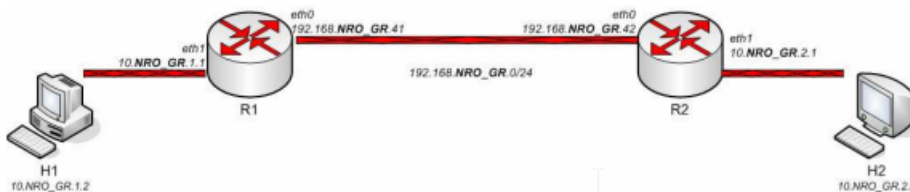


Figura 3.3: Sentidos en que puede ejecutarse el ping

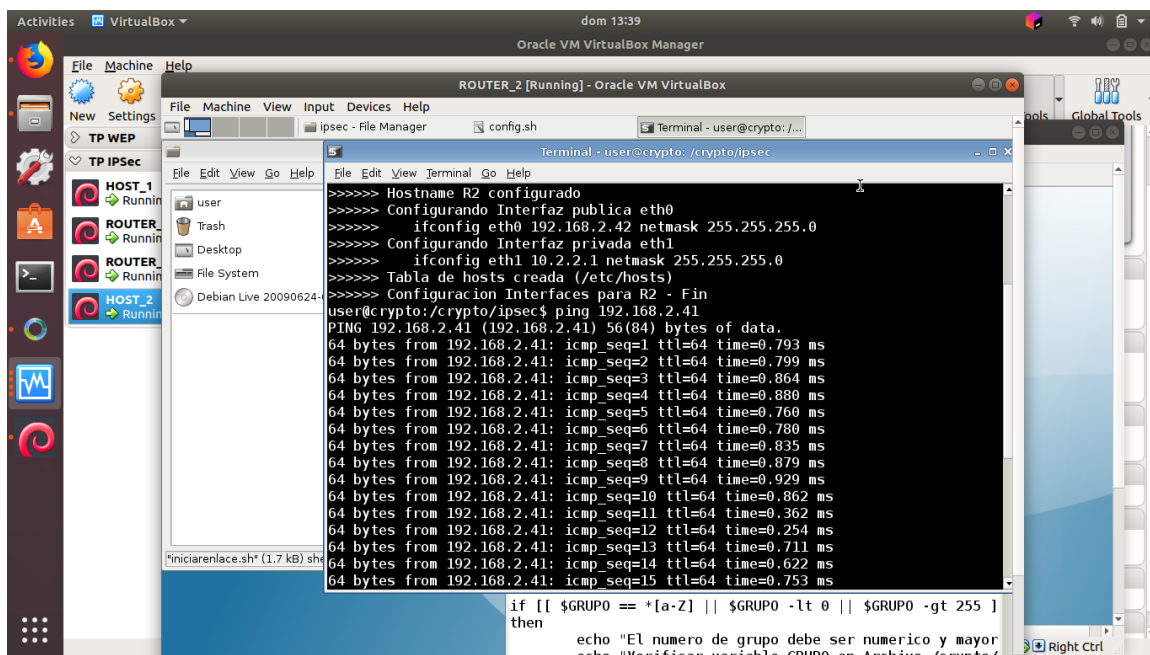


Figura 3.4: Ping de *router₂* a *router₁*

Luego generar las claves IP-sec:

- *ROUTER₁*: `sudo ./R1-2-generarclaves.sh`
- *ROUTER₂*: `sudo ./R2-2-generarclaves.sh`

Cada router tiene que obtener las claves del otro para ello

- *ROUTER₁*: `sudo ./R1-3-obtenerclaveremota.sh`, cuando pregunte si se acepta el host aceptar y cuando pregunte la clave que permite el copiado de archivo ingresar **crypto**
- *ROUTER₂*: `sudo ./R2-3-obtenerclaveremota.sh`, cuando pregunte si se acepta el host aceptar y cuando pregunte la clave que permite el copiado de archivo ingresar **crypto**

Configurar el servicio IP-SEC:

- *ROUTER₁*: `sudo ./R1-4-configurar.sh`
- *ROUTER₂*: `sudo ./R2-4-configurar.sh`

Iniciar el servicio IP-SEC usando desde el *ROUTER_{1/2}* es indistinto cual sea, ya que al considerarse trafico interesante el túnel debería levantarse.

`sudo ./iniciarenlace.sh`

Como el tunel ya se encuentra establecido deberíamos ver una ruta con la red R2H2 en el *ROUTER₁*

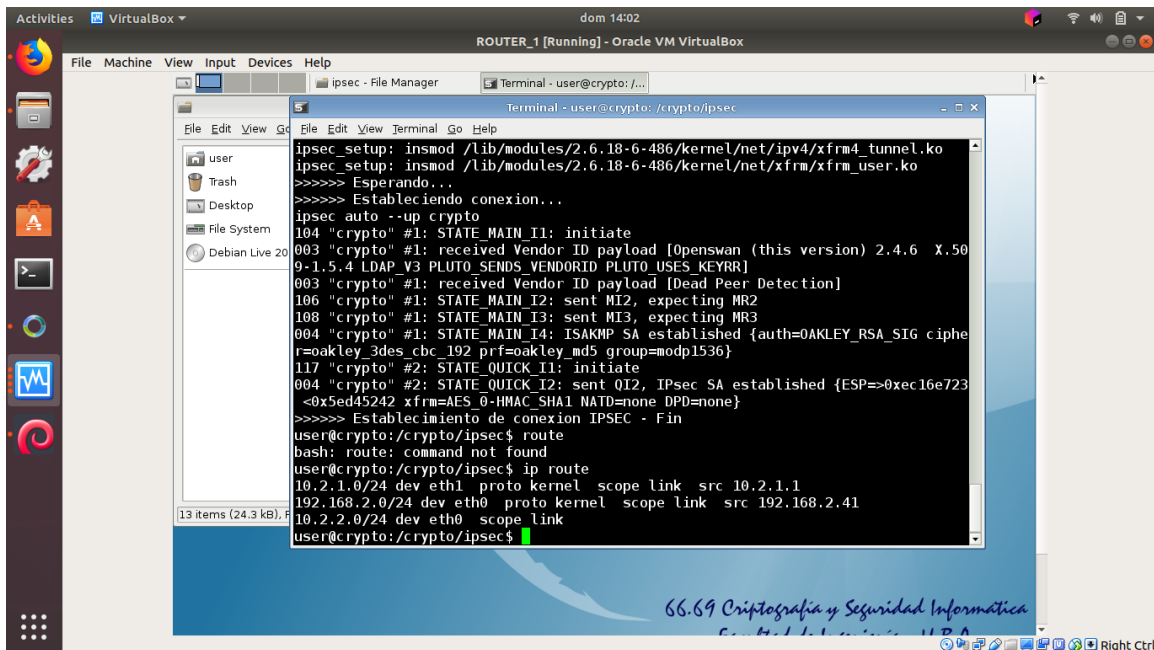


Figura 3.5: Ruteos $ROUTER_1$ con canal establecido

Desde este momento hay visibilidad entre $host_1$ y $host_2$. Haremos capturas de este ping usando wireshark. Para ello sudo wireshark.

Si capturamos en la interfaz $eth1$ del $ROUTER_1$ que está conectada con $HOST_1$, se puede ver los paquetes ICMP del comando PING descryptados.

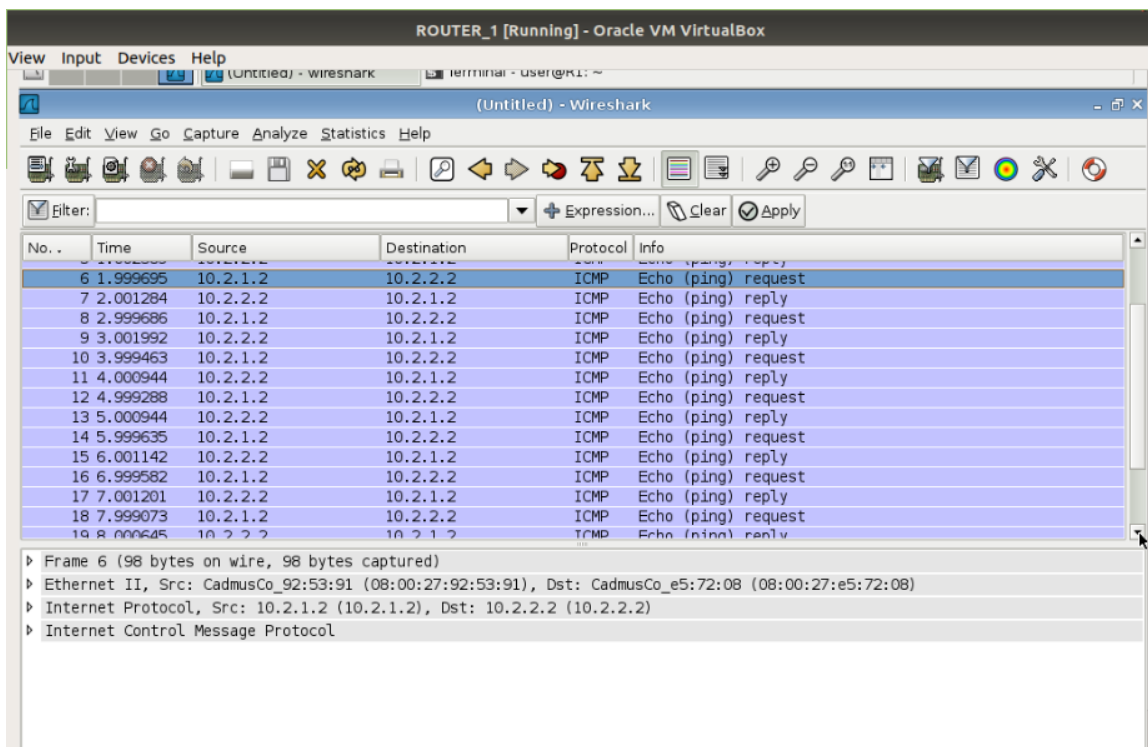


Figura 3.6: $ROUTER_1$ a $HOST_1$ ICMP descryptado

Si capturamos en la interfaz $eth0$ del $ROUTER_1$ que está conectada con $ROUTER_2$, se ven los paquetes ESP entre routers. También puede observarse el paquete ICMP ping reply descryptado que sale de esa interfaz

y va nuevamente a $HOST_1$.

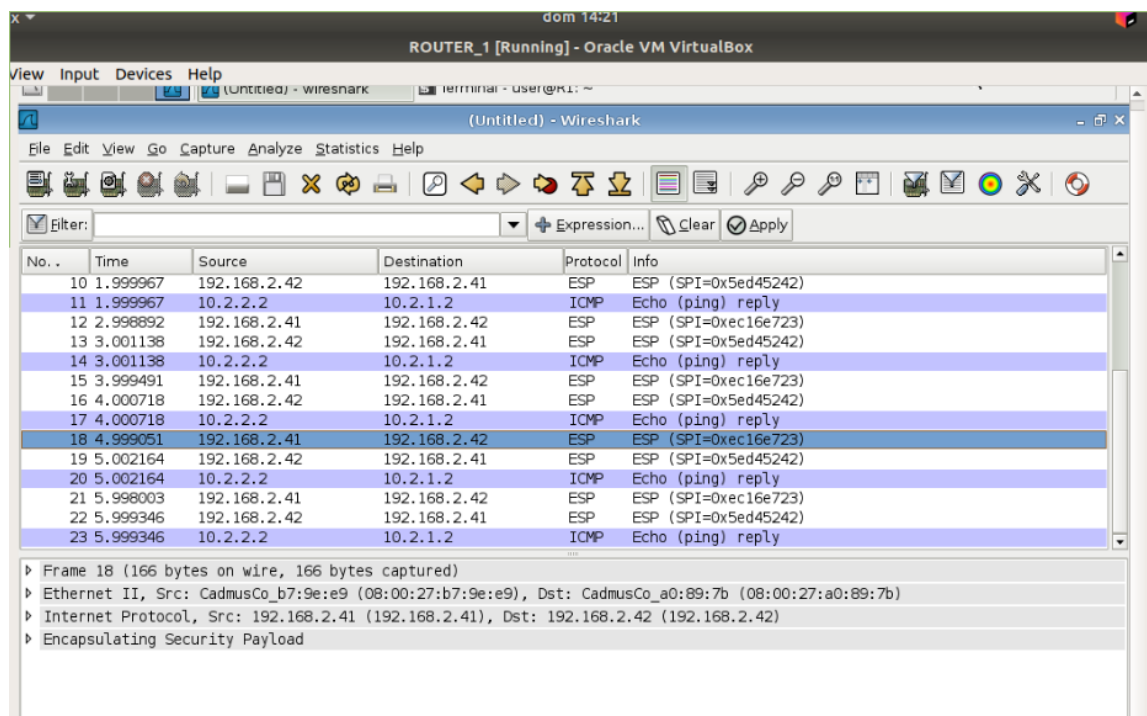


Figura 3.7: $ROUTER_1$ a $ROUTER_2$ ESP cifrado

A continuacion hay que capturar el establecimiento de la conexion para esto primero debemos darla de baja. Desde $ROUTER_2$ ejecutamos lo siguiente, habiendo dejado wireshark levantado en $ROUTER_1$

```
sudo ipsec auto --down crypto
sudo /etc/init.d/ipsec restart
sudo ./iniciarenlace.sh.
```

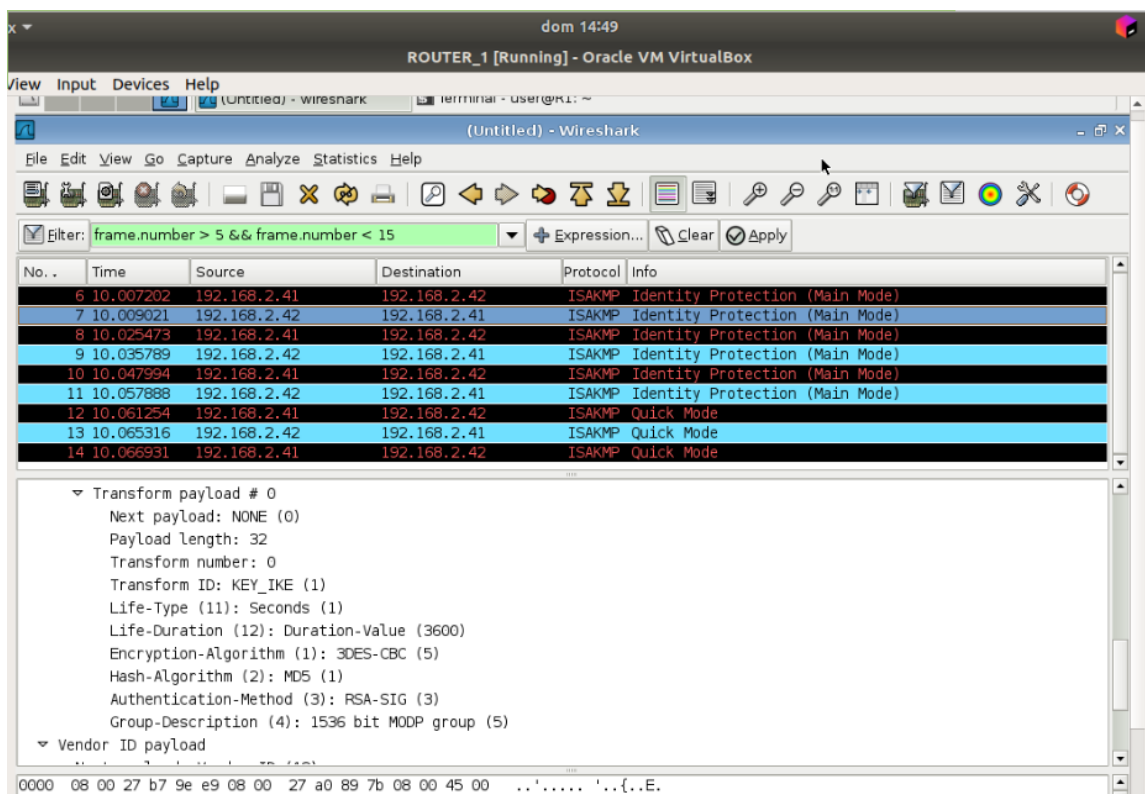


Figura 3.8: *ROUTER₁* a *ROUTER₂* ESP cifrado

Los primeros seis paquetes son la primera fase de establecimiento del tunel IP-SEC por el protocolo ISAKMP, los que dicen **Main mode**. Los siguientes tres son la segunda fase o **Quick mode**. Hay tres mas de Quick mode mas abajo que son los simetricos para la SA que va en el otro sentido.

3.1. Main Mode

Utilizado para establecer un canal seguro entre ambas partes y autenticar. Los pasos son:

- Acuerdo de los algoritmos a utilizar, el primer mensaje es una lista de propuestas, y el segundo es una respuesta seleccionando alguna de las proposiciones.
- Se intercambia (en ambos sentidos) una clave generada localmente para establecer la clave de sesión mediante el método de Diffie-Hellman, y además se intercambia un nonce para evitar ataques del tipo replay.
- Finalmente, una vez ya establecida la clave de sesión, se intercambian dos mensajes para autenticar a ambos usuarios

3.2. Quick Mode

Una vez que establecieron una ISAKMP SA, se utiliza *quick mode*. Los mensajes intercambiados son los siguientes, los cuales de ahora en más serán todos cifrados.

- Propuesta de una nueva SA, y el SPI establecido para los mensajes entrantes
- Respuesta del SPI que debe usar para los mensajes salientes
- Confirmación

3.3. Conclusiones de la captura

Se ejecutan dos fases, en la primera se genera un canal cifrado a través de la cual ambos terminadores pueden negociar la fase dos, intercambian parámetros para generar una clave de sesión vía Diffie-Hellman. En la segunda fase se llega a un acuerdo de varios parámetros, que tráfico puede pasar por la VPN y como debe ser cifrado/autenticado el mismo. A este acuerdo se lo llama SA, por sus siglas en inglés Security Association (Asociación de Seguridad). El cifrado en este caso es 3DES-CBC. Como hay redes no públicas se requiere usar modo túnel que es lo que efectivamente se está haciendo.

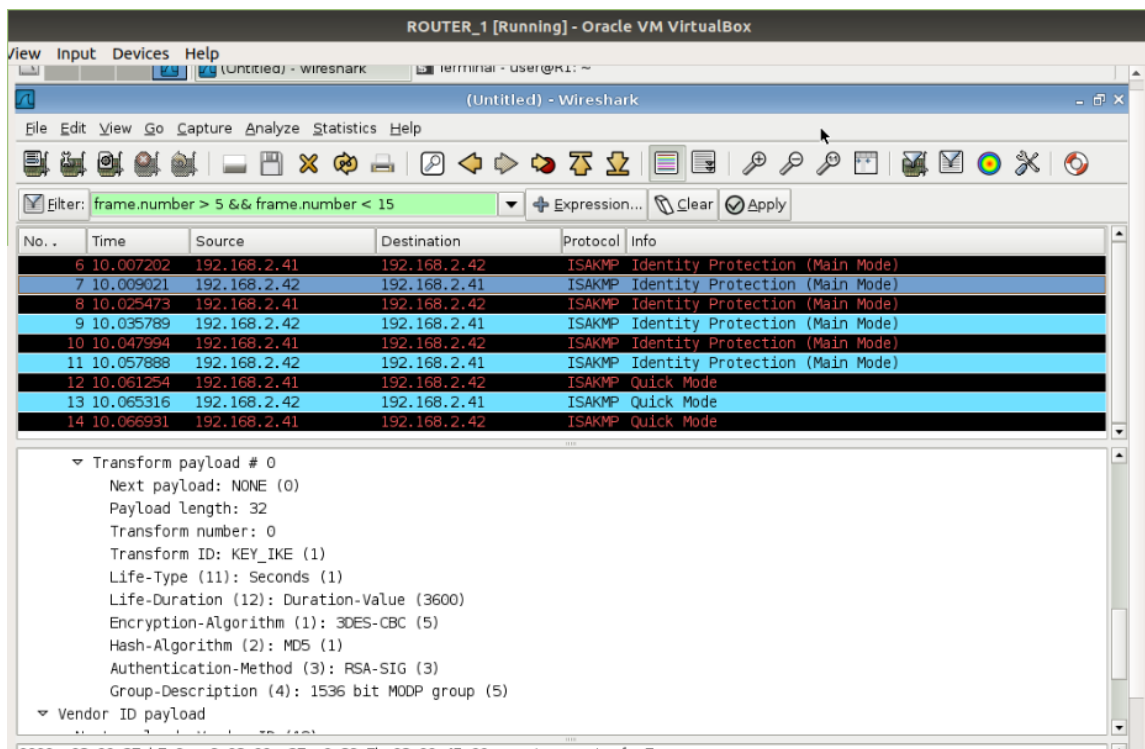


Figura 3.9: Captura Wireshark: Algoritmo de cifrado

3.4. Descifrado de paquetes

Para descifrar el contenido de un paquete que viaja entre el $HOST_1$ y el $HOST_2$ con el tunel levantado, se requieren los parámetros se generaron para el mismo.

Utilizando el comando `setkey -D` podemos conseguirlos y luego usarlos en Wireshark, para ello debemos ir a **Edit** → **Preferences** → **Protocols** → **ESP**.

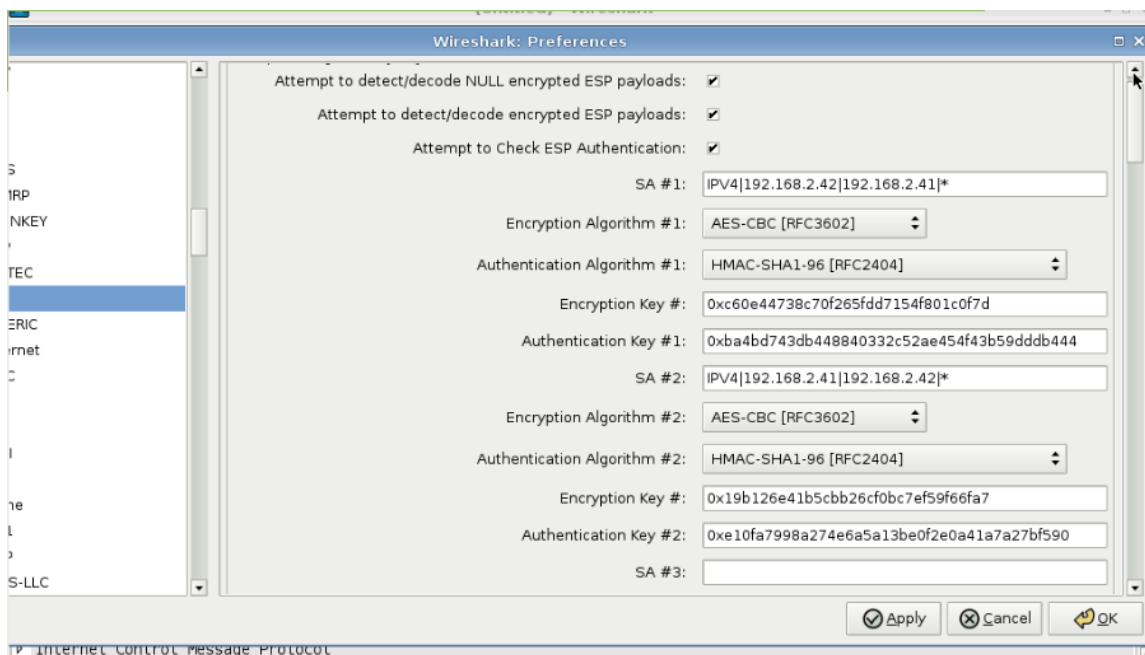


Figura 3.10: Configuración Wireshark: Parámetros para descifrar

Ahora podemos ver el contenido de los paquetes descifrado, aunque puede observarse la capa ESP, lo que nos garantiza que no viajó como un *payload* plano.

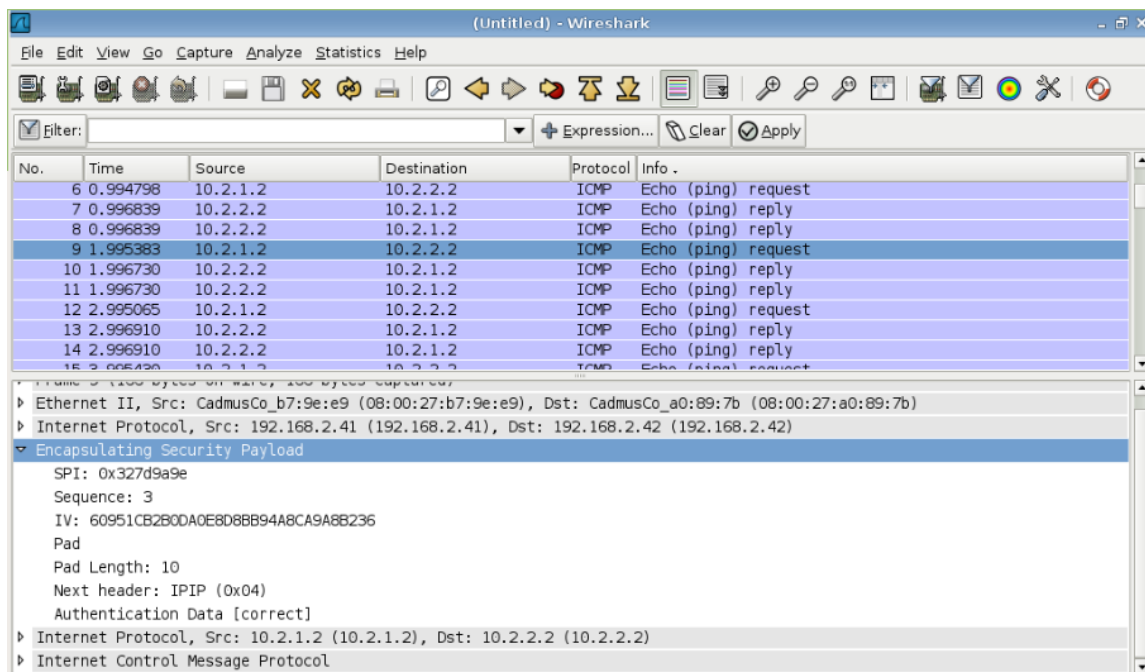
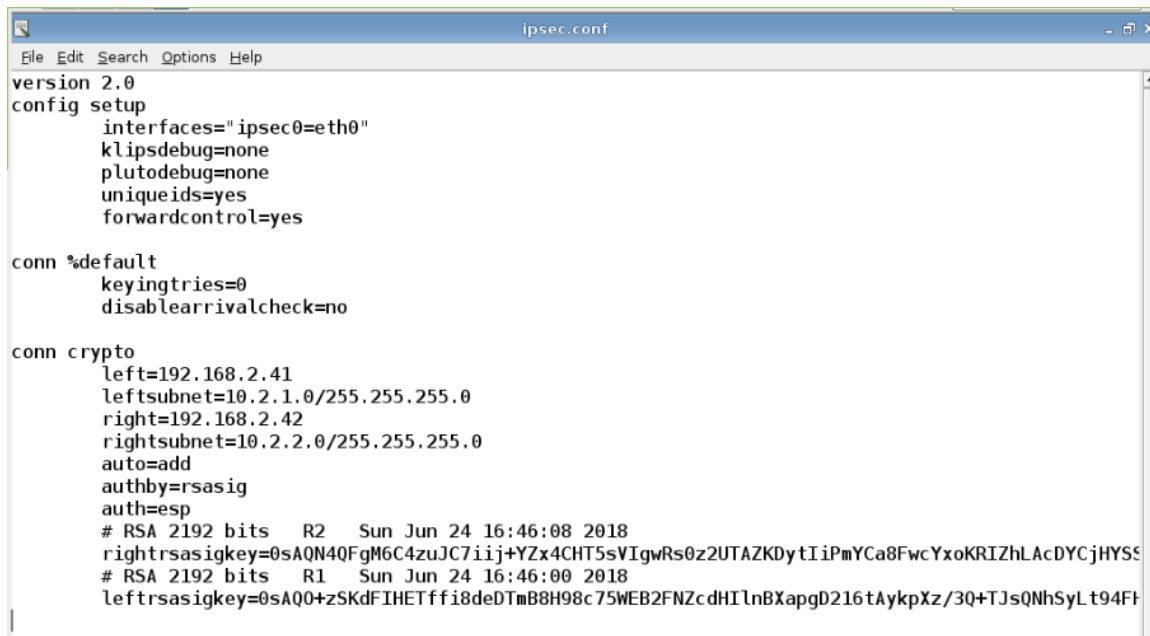


Figura 3.11: Capturar ICMP ESP descifrada por Wireshark

Al reiniciar la conexión el contenido de los paquetes deja de ser visible ya que hubo una renegociación de SA en la nueva conexión y nuestros parámetros de cifrado/descifrado dejan de ser válidos.

4. Certificados

Archivos de configuración sin certificado.

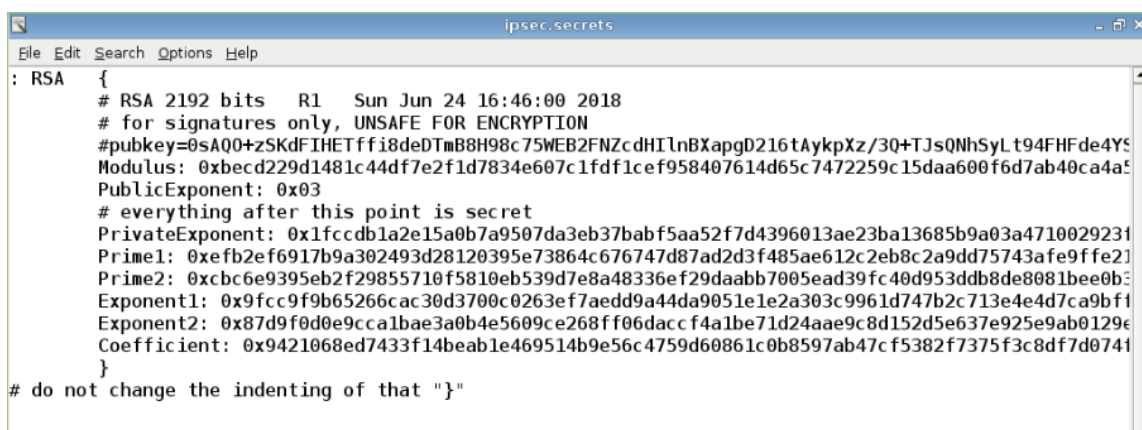


```
version 2.0
config setup
    interfaces="ipsec0=eth0"
    klipsdebug=none
    plutodebug=none
    uniqueids=yes
    forwardcontrol=yes

conn %default
    keyingtries=0
    disablearrivalcheck=no

conn crypto
    left=192.168.2.41
    leftsubnet=10.2.1.0/255.255.255.0
    right=192.168.2.42
    rightsubnet=10.2.2.0/255.255.255.0
    auto=add
    authby=rsasig
    auth=esp
    # RSA 2192 bits   R2   Sun Jun 24 16:46:08 2018
    rightrsasigkey=0sAQN4QFgM6C4zuJC7iij+YZx4CHT5sVIgwRs0z2UTAZKDYtIiPmYCa8FwcYxoKRIZhLAcDYCjHYSS
    # RSA 2192 bits   R1   Sun Jun 24 16:46:00 2018
    leftrsasigkey=0sAQ0+zSKdFIHETffi8deDTmB8H98c75WEB2FNZcdHILnBXapgd216tAykpXz/3Q+TJsQNhSyL t94FH
```

Figura 4.1: ipsec.conf sin certificado



```
: RSA {
    # RSA 2192 bits   R1   Sun Jun 24 16:46:00 2018
    # for signatures only, UNSAFE FOR ENCRYPTION
    #pubkey=0sAQ0+zSKdFIHETffi8deDTmB8H98c75WEB2FNZcdHILnBXapgd216tAykpXz/3Q+TJsQNhSyL t94FHfde4YS
    Modulus: 0xbecd229d1481c44df7e2f1d7834e607c1fdf1cef958407614d65c7472259c15daa600f6d7ab40ca4a5
    PublicExponent: 0x03
    # everything after this point is secret
    PrivateExponent: 0x1fccdb1a2e15a0b7a9507da3eb37babf5aa52f7d4396013ae23ba13685b9a03a4710029231
    Prime1: 0xefb2ef6917b9a302493d28120395e73864c676747d87ad2d3f485ae612c2eb8c2a9dd75743afe9ffe21
    Prime2: 0xcabc6e9395eb2f29855710f5810eb539d7e8a48336ef29daabb7005ead39fc40d953ddb8de8081bee0b3
    Exponent1: 0x9fcc9f9b65266cac30d3700c0263ef7aedd9a44da9051e1e2a303c9961d747b2c713e4e4d7ca9bfbf
    Exponent2: 0x87d9f0d0e9cca1bae3a0b4e5609ce268ff06dacc f4a1be71d24aae9c8d152d5e637e925e9ab0129e
    Coefficient: 0x9421068ed7433f14beable469514b9e56c4759d60861c0b8597ab47cf5382f7375f3c8df7d074f
}
# do not change the indenting of that "}"
```

Figura 4.2: ipsec.secrets sin certificado

Para generar los certificados, en el $ROUTER_1$ ejecutamos los scripts. Siempre que solicitó una clave usamos **password** y cuando solicitó una cuenta de correo usamos **grupo2@fi.uba.ar** y en unidad dentro de la organizacion **certificados**.

```
#Para generar la CA
s01-generarCA.sh
```

```
# Para generar las solicitudes, ejecutamos lo siguiente
# cuando pide nombre de la organizacion ponemos R1 y R2 respectivamente.
sudo s02-generaSolicitudes.sh
```

```
# Para firmar las solicitudes
sudo s03-firmaSolicitudes.sh
```

```
# Generamos una lista de certificados revocados (CRL) vacía.
sudo s04-generaCRL.sh
```

```
# Para obtener los datos para los archivos de configuracion de ipsec
s05-getIDs.sh
```

Con todos estos pasos ya realizados se puede configurar IPSec con certificados. Empezamos con *ROUTER₁* que es donde se generaron.

```
# Para configurar las interfaces de red.
sudo s00-configurarInterfacesLocales.sh
```

```
# Para instalar los certificados
sudo s06-instalaCertificados.sh
```

```
# Para preparar los archivos requeridos por el $ROUTER_2$ para ipsec
sudo s07-preparaArchivo.sh
```

Para configurar *ROUTER₂* debemos hacer lo siguiente desde el mismo

```
# Para configurar las interfaces de red
sudo s08r-configurarInterfacesRemotas.sh
```

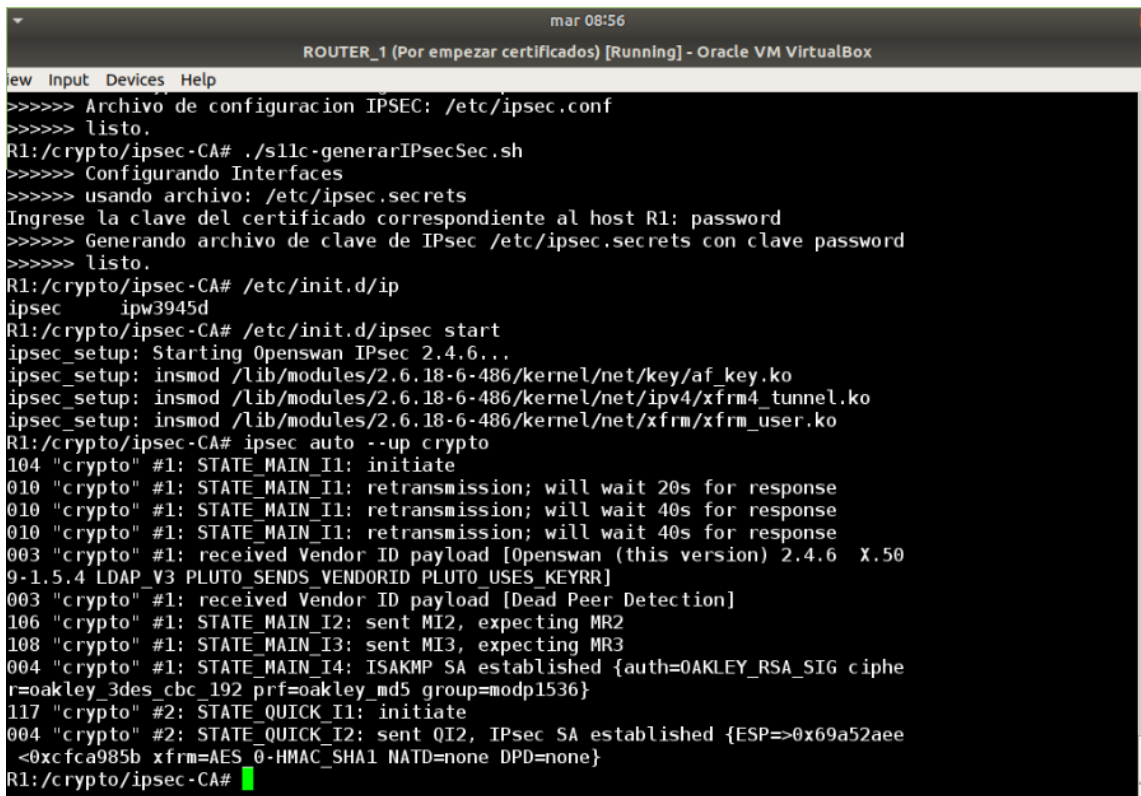
```
# Para copiar los certificados desde el otro router e instalarlos, usando clave crypto
sudo s09r-obtenerCertificado.sh
```

Ejecutar en ambos router.

```
# Para generar respectivamente ipsec.conf
sudo s10c-generarIPsecConf.sh
```

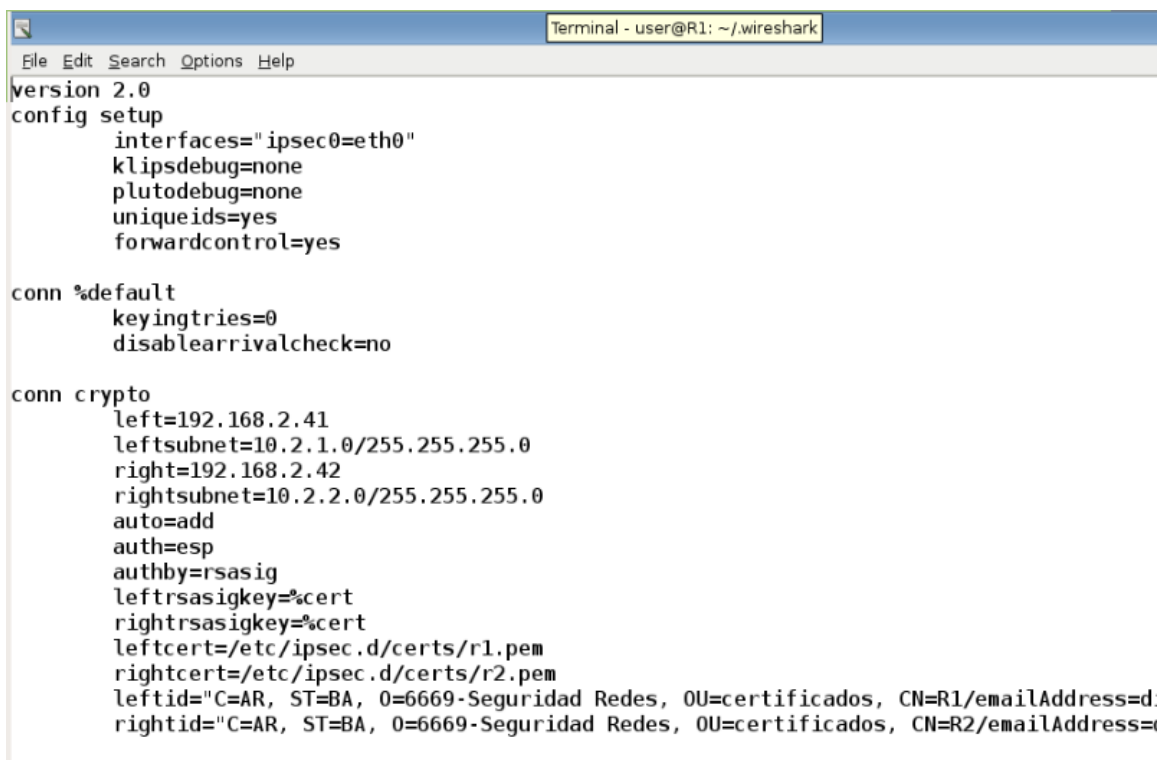
```
# Para generar respectivamente ipsec.secret
# Nos requerirá la clave utilizada inicialmente (password)
sudo s11c-generarIPsecSec.sh
```

Para levantar el servicio de ambos routers debe ejecutarse el comando `/etc/init.d/ipsec start` y solo uno de los dos router debe iniciar el canal con el comando `ipsec auto --up crypto`



```
mar 08:56
ROUTER_1 (Por empezar certificados) [Running] - Oracle VM VirtualBox
new Input Devices Help
>>>>> Archivo de configuracion IPSEC: /etc/ipsec.conf
>>>>> listo.
R1:/crypto/ipsec-CA# ./sllc-generarIPsecSec.sh
>>>>> Configurando Interfaces
>>>>> usando archivo: /etc/ipsec.secrets
Ingrese la clave del certificado correspondiente al host R1: password
>>>>> Generando archivo de clave de IPsec /etc/ipsec.secrets con clave password
>>>>> listo.
R1:/crypto/ipsec-CA# /etc/init.d/ipsec
ipsec ipw3945d
R1:/crypto/ipsec-CA# /etc/init.d/ipsec start
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
R1:/crypto/ipsec-CA# ipsec auto --up crypto
104 "crypto" #1: STATE_MAIN_I1: initiate
010 "crypto" #1: STATE_MAIN_I1: retransmission; will wait 20s for response
010 "crypto" #1: STATE_MAIN_I1: retransmission; will wait 40s for response
010 "crypto" #1: STATE_MAIN_I1: retransmission; will wait 40s for response
003 "crypto" #1: received Vendor ID payload [Openswan (this version) 2.4.6 X.509-1.5.4 LDAP_V3 PLUTO SENDS_VENDORID PLUTO USES_KEYRR]
003 "crypto" #1: received Vendor ID payload [Dead Peer Detection]
106 "crypto" #1: STATE_MAIN_I2: sent MI2, expecting MR2
108 "crypto" #1: STATE_MAIN_I3: sent MI3, expecting MR3
004 "crypto" #1: STATE_MAIN_I4: ISAKMP SA established {auth=OAKLEY_RSA_SIG ciphe
r=oakley_3des_cbc_192 prf=oakley_md5 group=modp1536}
117 "crypto" #2: STATE_QUICK_I1: initiate
004 "crypto" #2: STATE_QUICK_I2: sent QI2, IPsec SA established {ESP=>0x69a52aee
<0xcfa985b xfrm=AES_0-HMAC_SHA1 NATD=none DPD=none}
R1:/crypto/ipsec-CA#
```

Figura 4.3: Ejecución comandos para certificado *ROUTER₁*



```
Terminal - user@R1: ~/wireshark
File Edit Search Options Help
version 2.0
config setup
    interfaces="ipsec0=eth0"
    klipsdebug=none
    plutodebug=none
    uniqueids=yes
    forwardcontrol=yes

conn %default
    keyingtries=0
    disablearrivalcheck=no

conn crypto
    left=192.168.2.41
    leftsubnet=10.2.1.0/255.255.255.0
    right=192.168.2.42
    rightsubnet=10.2.2.0/255.255.255.0
    auto=add
    auth=esp
    authby=rsasig
    lefttrsasigkey=%cert
    righttrsasigkey=%cert
    leftcert=/etc/ipsec.d/certs/r1.pem
    rightcert=/etc/ipsec.d/certs/r2.pem
    leftid="C=AR, ST=BA, O=6669-Seguridad Redes, OU=certificados, CN=R1/emailAddress=d.
    rightid="C=AR, ST=BA, O=6669-Seguridad Redes, OU=certificados, CN=R2/emailAddress=d.
```

Figura 4.4: ipsec.secrets con certificado

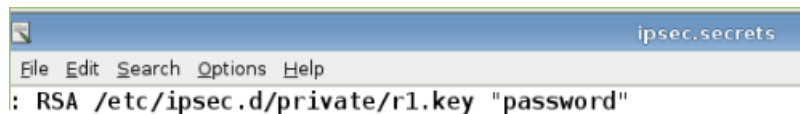


Figura 4.5: ipsec.secrets con certificado

4.1. Revocar certificado

Para revocar el certificado del $ROUTER_1$ ejecutamos `srs0-revocarR1.sh` que nos genera un nuevo CRL con este certificado revocado.

Desde el $ROUTER_2$ debemos actualizar el CRL recientemente generado, para ello ejecutamos `srs1r-instalarCRLnuevo.sh` nos requerirá la clave de la pc **crypto**.

Al intentar iniciar la conexión desde cualquiera de los dos router con el comando `ipsec auto --up crypto` obtendremos el código de error **INVALID_KEY_INFORMATION**.

```

ROUTER_2 (por empezar certificados) [Running] - Oracle VM VirtualBox
View Input Devices Help
S 0-HMAC_SHA1 NATD=none DPD=none}
R2:/crypto/ipsec-CA# /etc/init.d/ipsec restart
ipsec_setup: Stopping Openswan IPsec...
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
R2:/crypto/ipsec-CA# ipsec auto --down crypto
R2:/crypto/ipsec-CA# ./srs1r-instalarCRLnuevo.sh
>>>>> Instalando nueva lista de certificados revocados en el equipo local
root@192.168.2.41's password:
crl.pem                               100% 560      0.6KB/s   00:00
R2:/crypto/ipsec-CA# ipsec auto --down crypto
R2:/crypto/ipsec-CA# /etc/init.d/ipsec restart
ipsec_setup: Stopping Openswan IPsec...
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
R2:/crypto/ipsec-CA# ipsec auto --down crypto
R2:/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.509-1.5.4 LDAP_V3 PLUT
0 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 Redes, OU=certificados,
CN=R1, E=divitoivan@gmail.com'
217 "crypto" #2: STATE_MAIN_I3: INVALID_KEY_INFORMATION

```

Figura 4.6: Instalación CRL actualizado y error de invalid key

4.2. Generación de un nuevo certificado

Generamos un nuevo certificado válido para el $ROUTER_1$ con el comando `srs2-generarNuevoCertificado.sh`, reiniciamos el servicio de ipsec en ambos router y podemos observar que el tunel se inicia con normalidad.


```
ROUTER_2 (por empezar certificados) [Running] - Oracle VM VirtualBox
View Input Devices Help
108 "crypto" #1: STATE_MAIN_I3: sent MI3, expecting MR3
003 "crypto" #1: no RSA public key known for 'C=AR, ST=BA, O=6669-Seguridad Redes, OU=certificados, CN=R1, E=divitoivan@gmail.com'
217 "crypto" #1: STATE_MAIN_I3: INVALID_KEY_INFORMATION
R2:/crypto/ipsec-CA# /etc/init.d/ipsec restart
ipsec_setup: Stopping Openswan IPsec...
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
R2:/crypto/ipsec-CA# ipsec auto --down crypto
R2:/crypto/ipsec-CA# ipsec auto --down crypto
R2:/crypto/ipsec-CA# /etc/init.d/ipsec restart
ipsec_setup: Stopping Openswan IPsec...
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
R2:/crypto/ipsec-CA# ipsec auto --up crypto
104 "crypto" #1: STATE_MAIN_I1: initiate
003 "crypto" #1: received Vendor ID payload [Openswan (this version) 2.4.6 X.509-1.5.4 LDAP_V3 PLUT
0 SENDS_VENDORID PLUTO_USES_KEYRR]
003 "crypto" #1: received Vendor ID payload [Dead Peer Detection]
106 "crypto" #1: STATE_MAIN_I2: sent MI2, expecting MR2
108 "crypto" #1: STATE_MAIN_I3: sent MI3, expecting MR3
004 "crypto" #1: STATE_MAIN_I4: ISAKMP SA established {auth=0AKLEY_RSA_SIG cipher=oakley_3des_cbc_19
2 prf=oakley_md5 group=modp1536}
117 "crypto" #2: STATE_QUICK_I1: initiate
004 "crypto" #2: STATE_QUICK_I2: sent QI2, IPsec SA established {ESP=>0xff754e38 <0x49c9e9da xfrm=AE
S 0-HMAC_SHA1 NATD=none DPD=none}
R2:/crypto/ipsec-CA#
```

Figura 4.7: Canal activo con el nuevo certificado

Como puede verse en las figuras 4.1, 4.2, 4.4 y 4.5 el archivo ipsec.conf tiene en el caso sin certificado las claves publicas a utilizar ipsec.secret las claves privadas asociadas.

Mientras que en el caso con certificados ipsec.conf tiene una referencia a la ubicación de los certificados e informacion de los mismos, y el archivo ipsec.secret tiene la ubicación de la clave privada y la clave para descifrarla.

4.3. TCPDUMP RSA con/sin certificados

Lo diferencia más importante a observar en trabajar con/sin certificados se encuentra en el proceso de negociación fase uno, ya que luego de establecer la SA es todo igual. Puede verse en las figuras 4.8 y 4.9 como el el primer paquete cifrado (el cuarto de los del protocolo ISAKMP Main Mode), en el caso sin certificados que tiene un tamaño de 256 bytes, y en el caso con certificados es de 1248 bytes. La diferencia se da en que en el segundo caso se incluye el certificado cifrado.

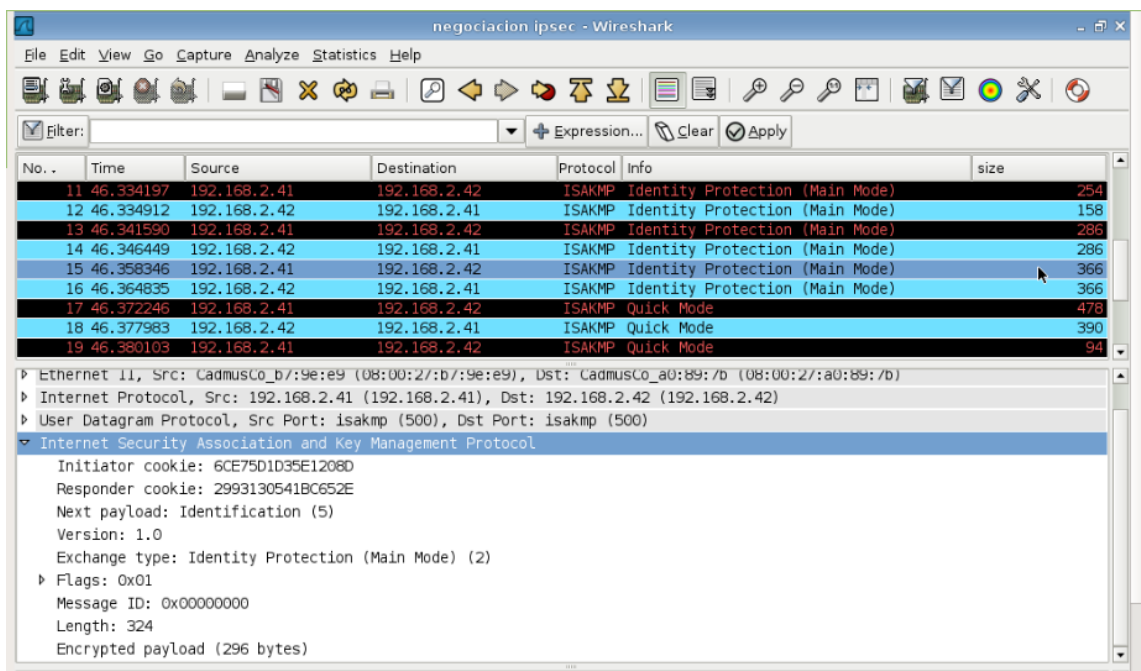


Figura 4.8: Negociación IPsec sin certificado

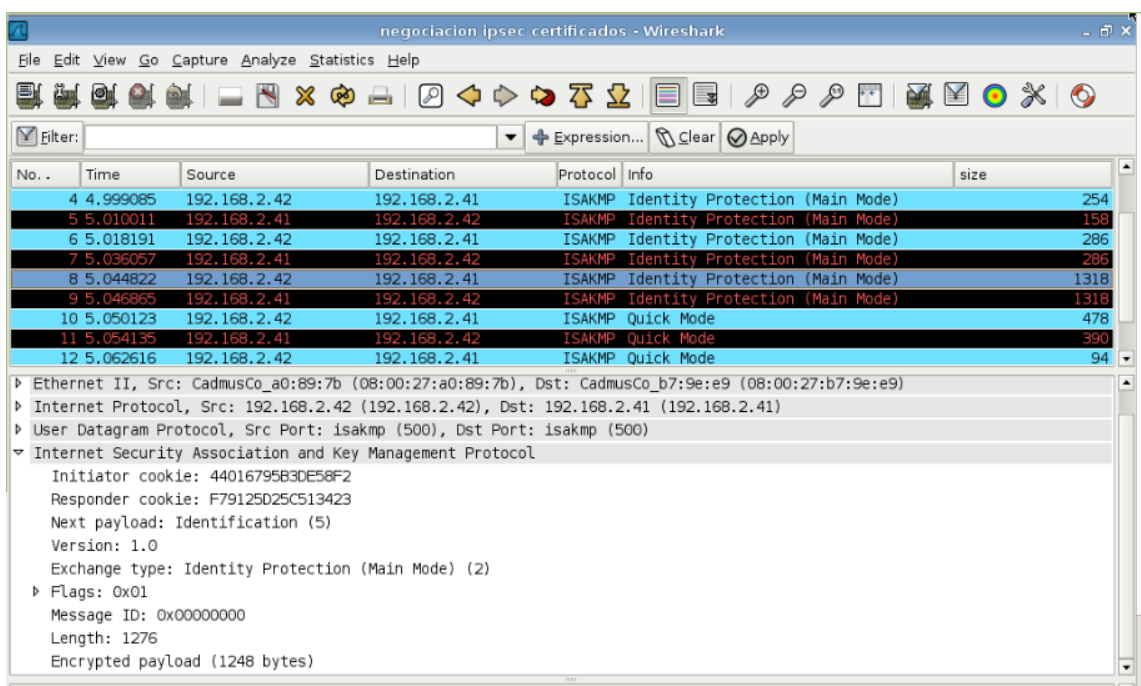


Figura 4.9: Negociación IPsec con certificado

4.4. Certificados Apache

```
R1:/crypto/apache-ssl# ./01-generar_CA.sh
>>>>> Borrando posibles previas autoridades certificadoras
>>>>> Generando nueva autoridad certificante
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]:AR
State or Province Name (full name) [BA]:BA
Locality Name (eg, city) [Buenos Aires]:Buenos Aires
Organization Name (eg, company) [66.69-Criptografia]:66.69-Criptografia
Organizational Unit Name (eg, section) []:certificados
Common Name (eg, your name or your server's hostname) []:R1
Email Address []:grupo2@fi.uba.ar
>>>>> listo.

R1:/crypto/apache-ssl# ./02-generar_solicitudes.sh
>>>>> Removiendo previos certificados
>>>>> Generando peticiones de firma para servidor
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to 'ssl-server.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]:AR
State or Province Name (full name) [BA]:BA
Locality Name (eg, city) [Buenos Aires]:Buenos Aires
Organization Name (eg, company) [66.69-Criptografia]:66.69-Criptografia
Organizational Unit Name (eg, section) []:certificados
Common Name (eg, your name or your server's hostname) []:R1
Email Address []:grupo2@fi.uba.ar

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
>>>>> listo.

R1:/crypto/apache-ssl# ./03-firmar_certificados.sh
>>>>> Firmando los certificados
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:'66.69-Criptografia'
organizationalUnitName:PRINTABLE:'certificados'
commonName            :PRINTABLE:'R1'
emailAddress          :IA5STRING:'grupo2@fi.uba.ar'
Certificate is to be certified until Jun 26 13:35:23 2028 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
```

```
>>>>> listo.
```

```
R1:/crypto/apache-ssl# ./04-instalar_certificados.sh
```

```
>>>>> Borrando previos certificados
```

```
>>>>> Instalando los certificados del servidor ssl-server
```

```
>>>>> listo.
```

```
R1:/crypto/apache-ssl# ./05-configurarApache.sh
```

```
>>>>> //////////////////////////////////////
```

```
>>>>> Configurando puerto 443 del apache en /etc/apache2/ports.conf
```

```
>>>>> Agregando el puerto 443
```

```
>>>>> //////////////////////////////////////
```

```
>>>>> Habilitando el modulo mod_ssl
```

```
a2enmod ssl
```

```
Module ssl installed; run /etc/init.d/apache2 force-reload to enable.
```

```
>>>>> Agregando la definicion del sitio al apache
```

```
cp -f data/ssl-site.conf /etc/apache2/sites-available/ssl-site
```

```
>>>>> Las paginas del sitio SSL estan en /crypto/var/www
```

```
>>>>> Habilitando el sitio al apache
```

```
a2ensite ssl-site
```

```
Site ssl-site installed; run /etc/init.d/apache2 reload to enable.
```

```
>>>>> Configurando ssl-server como 127.0.1.2 en archivo hosts
```

```
>>>>> Listo
```

```
R1:/crypto/apache-ssl# ./06-reiniciar_apache.sh
```

```
Reiniciando apache2
```

```
Forcing reload of web server (apache2)... waiting .
```

```
Listo
```

Al iniciar apache podemos ver el siguiente error en el **error.log**

```
[warn] RSA server certificate CommonName (CN) 'R1' does NOT match server name!?
```

Se soluciona con agregar la directiva **ServerName** en el archivo del sitio ssl y reiniciar el servicio.

Luego al intentar conectarnos localmente nos sale el siguiente error

```
R1:/crypto/apache-ssl/data# wget -O /dev/null https://R1
```

```
--13:41:36-- https://R1/
```

```
=> '/dev/null'
```

```
Resolving R1... 127.0.0.1
```

```
Connecting to R1|127.0.0.1|:443... connected.
```

```
OpenSSL: error:140770FC:SSL routines:SSL23_GET_SERVER_HELLO:unknown protocol
```

```
Unable to establish SSL connection.
```

Para solucionar este inconveniente debemos editar el archivo donde esta configurado el virtualhost con ssl y cambiar ssl-server por el CN generado en nuestro caso R1

```
# Resolver error de "OpenSSL: error:140770FC:SSL
routines:SSL23_GET_SERVER_HELLO:unknown protocol"
R1:/etc/apache2/sites-enabled# head -10 /etc/apache2/sites-enabled/ssl-site
#<VirtualHost ssl-server:443>
<VirtualHost R1:443>

# Configuración general
    DocumentRoot "/crypto/var/www"
    ServerAdmin grupo2@fi.uba.ar

# Resolver Error de nombre de servidor invalido
    ServerName R1
```

Si bien ahora ya podemos conectarnos con el servidor aún sigue sin ser considerado un sitio de confianza ya que no conoce nuestra CA. Para ello debemos instalar el certificado raíz en el almacén de certificados.

The owner of R1 has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

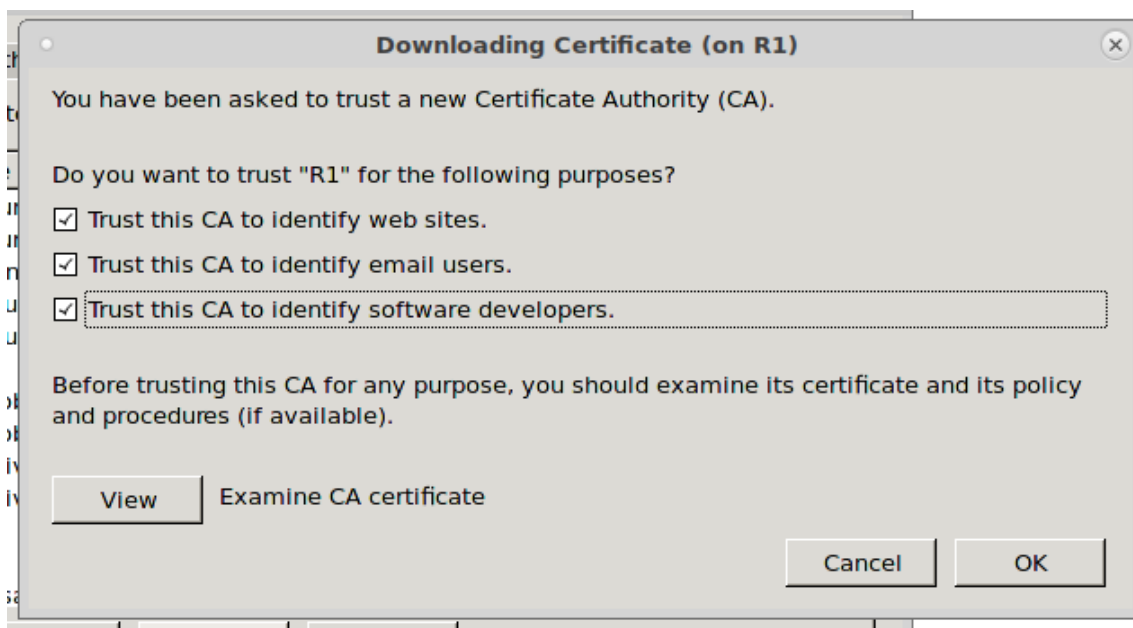


Figura 4.10: Instalar root Certificate en Firefox

5. Conclusiones

Durante el armado de esta maqueta aprendimos los pasos necesarios para montar un tunel IPsec que permite mejorar la seguridad en la comunicacion a travez de una red publica. Pudimos comprobar que gracias a los mecanismos que se usan durante la negociacion de las SA que no es posible interpretar los mensajes enviados sin conocer las claves de sesion establecidas. Tambien pudimos comprobar como funciona el sistema de certificados tanto para la autentificacion de los clientes de dicho tunel como para configurar un servidores apache. Esto facilita el intercambio de claves y agrega una capa mas de seguridad al estar las claves publicas firmadas por una CA.

Usar certificados también soluciona el problema de la autenticación y con ello el ataque man in the middle se hace mucho mas dificil. Es decir, tendría que poder también falsificar la cadena de certificados de algún modo.

Tambien pudimos revocar un certificado, lo cual sirve como ejemplo, para el caso de que las claves privadas hallan perdido su secreto.

6. Índice de Figuras

Índice de figuras

2.1. Entorno de trabajo virtualbox	2
2.2. Creación máquina virtual	3
2.3. Asignación de redes	4
3.1. VMS Iniciadas	4
3.2. Archivo de configuración inicial	5
3.3. Sentidos en que puede ejecutarse el ping	5
3.4. Ping de <i>router</i> ₂ a <i>router</i> ₁	6
3.5. Ruteos <i>ROUTER</i> ₁ con canal establecido	7
3.6. <i>ROUTER</i> ₁ a <i>HOST</i> ₁ ICMP descriptado	7
3.7. <i>ROUTER</i> ₁ a <i>ROUTER</i> ₂ ESP cifrado	8
3.8. <i>ROUTER</i> ₁ a <i>ROUTER</i> ₂ ESP cifrado	9
3.9. Captura Wireshark: Algoritmo de cifrado	10
3.10. Configuración Wireshark: Parámetros para descifrar	11
3.11. Capturar ICMP ESP descifrada por Wireshark	11
4.1. ipsec.conf sin certificado	12
4.2. ipsec.secrets sin certificado	12
4.3. Ejecución comandos para certificado <i>ROUTER</i> ₁	14
4.4. ipsec.secrets con certificado	14
4.5. ipsec.secrets con certificado	15
4.6. Instalación CRL actualizado y error de invalid key	15
4.7. Canal activo con el nuevo certificado	16
4.8. Negociación IPSec sin certificado	17
4.9. Negociación IPSec con certificado	17
4.10. Instalar root Certificate en Firefox	20

7. Bibliografía

Referencias

- [1] IPSec VPN Negotiations

https://www.watchguard.com/help/docs/fireware/12/es-419/Content/es-419/mvpn/general/ipsec_vpn_negotiations_c.html

- [2] 3 Common Causes of Unknown SSL Protocol Errors with cURL

<http://blog.techstacks.com/2010/03/3-common-causes-of-unknown-ssl-protocol-errors-with-curl.html>