

Marcos Margulius
Hernán Fernández Brando

Trabajo Práctico N°2 - IPSec

Objetivo del Trabajo Práctico

El presente trabajo práctico consistirá en configurar un tunel Ipsec entre dos routers y analizar, utilizando el analizador de protocolos, el tráfico generado por equipos host conectados a los routers, que actúan como gateways. Se utilizará para tal fin la distribución de Linux provista por la cátedra. Y se trabajará con software de virtualización de dispositivos, con tal de que el tráfico generado siempre esté dentro del equipo y no utilice en ningún momento el hardware de red para que el tráfico no salga.



Figura 1: Esquema de conexionado de red.

Desarrollo del Trabajo Práctico

1- Armado del esquema de Red y Configuración de Equipos H1, H2, R1 y R2.

Inicialmente se procedió al armado de la red de prueba, del mismo modo que el indicado en la Figura 1, y luego se procedió a implementar la siguiente configuración en los dispositivos mediante los scripts provistos dentro del live CD. Se eligió el número de grupo 10 para la configuración de los scripts.

R2:

```
hostname R2
ifconfig eth0 192.168.10.42 netmask 255.255.255.0
ifconfig eth1 10.10.2.1 netmask 255.255.255.0
Archivo /etc/hosts:
o 127.0.0.1 localhost
o 192.168.10.41 R1
o 192.168.10.42 R2
```

La segunda y tercer líneas establecen las direcciones IP y sus máscaras de sub-red asociadas para cada puerto ethernet del router.

```
crypto:/crypto/ipsec# ./R2-1-preparar.sh
>>>>> Configuración Interfaces para R2 - Inicio
>>>>> 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)
>>>>> Configuración Interfaces para R2 - Fin
```

Figura 2: Resultado del script "preparar.sh".

En el archivo `/etc/hosts`, se determinan los nombres de red para cada uno de los dispositivos conocidos por los router en la red. A los fines de este trabajo sólo es necesario que cada router tenga agregados mutuamente sus nombres de host.

Para cada host, un procedimiento análogo es realizado, con la salvedad que también se determina como gateway por defecto al router contiguo. Para H1, se tiene que:

```
H1:
hostname H1
ifconfig eth0 10. NRO_GR.1.2 netmask 255.255.255.0
Agrega ruta a la red 10. NRO_GR.1.1 por medio de R1
Agregar el host H2 al /etc/hosts
```

2- Chequeo de conectividad a nivel Red:

Una vez configurados los 4 equipos, se realizaron 3 distintos chequeos a nivel red, en la red R1-R2 y en cada red Hx-Rx. En la Figura 3 se muestran los resultados de los ping posibles para R1: de R1 a R2 y de R1 a H1.

```
crypto:/crypto/ipsec# ping 10.10.1.2
PING 10.10.1.2 (10.10.1.2) 56(84) bytes of data.
64 bytes from 10.10.1.2: icmp_seq=1 ttl=64 time=5.48 ns
64 bytes from 10.10.1.2: icmp_seq=2 ttl=64 time=0.280 ns
64 bytes from 10.10.1.2: icmp_seq=3 ttl=64 time=0.186 ns
64 bytes from 10.10.1.2: icmp_seq=4 ttl=64 time=0.400 ns

--- 10.10.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 300ms
rtt min/avg/max/mdev = 0.186/1.586/5.481/2.250 ns
crypto:/crypto/ipsec# ping 192.168.10.42
PING 192.168.10.42 (192.168.10.42) 56(84) bytes of data.
64 bytes from 192.168.10.42: icmp_seq=1 ttl=64 time=1.97 ns
64 bytes from 192.168.10.42: icmp_seq=2 ttl=64 time=0.232 ns
64 bytes from 192.168.10.42: icmp_seq=3 ttl=64 time=0.174 ns

--- 192.168.10.42 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 200ms
rtt min/avg/max/mdev = 0.174/0.793/1.973/0.834 ns
crypto:/crypto/ipsec# _
```

Figura 3: Prueba de conectividad en R1, con los dispositivos inmediatamente conectados.

2- Generación de claves IPSec:

Como primer paso para la creación del tunel, con el comando `newhostkey` se genera un par de claves pública y privada por RSA tanto para R1 como para R2, a continuación, en la Figura 4, se muestran los resultados de ejecutar el script para la generación en R2.

```
ipsec newhostkey --output /etc/ipsec.secrets --hostname R1
ipsec showhostkey --left > /tmp/left.key
```

```
crypto:/crypto/ipsec# ./R2-2-generarclaves.sh
>>>>> Generacion de claves en R2 - Inicio
>>>>> usando archivo: /etc/ipsec.secrets
>>>>> Generacion de claves en R2 - Fin
crypto:/crypto/ipsec# _
```

Figura 4: Resultado de la ejecución del script "`generarclaves.sh`".

```

RSA {
# RSA 2192 bits R2 Mon May 26 21:08:49 2014
# for signatures only, UNSAFE FOR ENCRYPTION
#pubkey=0sAQN+2EQRa1rN9MlqUF108KL32XluSc1P2E6YBuHB+Ji8P8nqn3XwhPE7kSpNkL
Un+sMsK/UilIR5Ip00xMAYUWN5Wfz2a8nbUxfBvKdMonqECn/fryyX7AGxQ6onaxF6dMXttpBdzn5vCz
TXFczGz2LXX5uVuQlaVP+eDqN0S3uv8gXGD+2W458b66x1NdvdchtyoL0hXu5UFH0kIUPh/cVHPNhNuR
QD9ce2Z2YichAf/QiT73tiGepQ1Jn0iqv8h0aJA4b09ne2gla4Y7FCInGzqqESnHUGMELGTRv85IfJKZ
3TeEG6S/12V001D3V8dJpU15fJU7LCEUs31WETQTb12xoSaMtqRmenCT9fiaOf1143
Modulus: 0x7ed844116a5acdf5696a505d74f0a2f7d9722e49cd4fd84e9806e1c1f898b
c3fc9ea9f75f084f13b912a4d90b567fac5ac2bf5222e2479229d0ec4c03251637959fcd96bc99b5
717c1bca74ca26a840a7fdaf2c97ec01b143aa266b117a74c5edb6905dce6e6f0b34d715ccc6cf6
2d75f9b98c1021a54ff9e0ea3744b7baff205c60fe656e39f21ebac6535dbdd721b72a0bd215eee5
4147d242143e1fdc6073cd04db91403f5c7b665889c0407ff4224fbded8867a94252663a2aaff21d
1a240e1b3bd9de66021ae18ec5080906ceaa844a71d418c10b193151bfce407c9299dd37041ba4bf
d7660e3b50f7541749a54d797c953b2c2114b379561134136c0db1a12696b6a4667a7093f5f09a39
f948e37
PublicExponent: 0x03
# everything after this point is secret
0
0
0
0
0
0
"ipsec.secrets" 15L, 3162C
1,1 Top

```

Figura 5: Visualización de clave pública recientemente generada.

3- Obtención de claves IPSec del equipo remoto:

Una vez generadas las claves, se procedió al copiado de las claves del otro equipo (en cada uno de los Routers), mediante el comando SecureCopy (scp, que permite que puedan copiarse archivos entre diferentes host, utilizando transferencia de datos por ssh). Ver Figura 6.

```

crypto:/crypto/ipsec# ./R2-3-obtenerclaveremota.sh
>>>>> Copiado de clave IPSEC de R1 - Inicio
The authenticity of host 'r1 (192.168.10.41)' can't be established.
RSA key fingerprint is 47:38:f0:fa:ac:d6:0b:89:7a:fd:ab:32:aa:78:8f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'r1,192.168.10.41' (RSA) to the list of known hosts.
root@r1's password:
left.key 100% 435 0.4KB/s 00:00
>>>>> Clave IPSEC de R1 copiada
>>>>> Copiado de clave IPSEC de R1 - Fin
crypto:/crypto/ipsec# _

```

Figura 6: Copiado de claves públicas.

4- Configuración e Iniciación del enlace IPSec:

Para la iniciación se ejecutó el siguiente comando, mediante el script provisto (iniciarenlace.sh) (ver Figura 7):

```
ipsec auto --up crypto
```

5- Verificación del Tunel:

Hasta el momento ambos terminales carecían de conectividad en la capa 3. Una vez iniciado el enlace, la conectividad comenzó a existir, y esto se pudo ver reflejado con el comando route, donde se observó que la tabla de ruteo se actualizó con la información de las redes privadas correspondientes al otro host.

Luego se realizó un ping para finalizar la comprobación de este enlace (Figura 8).

```

/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
>>>>> Esperando...
>>>>> Estableciendo conexion...
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.50
9-1.5.4 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->0x9aee3c7f
<0x2db90417 xfrm=AES_0-HMAC_SHA1 NATD=none DPD=none}
>>>>> Establecimiento de conexion IPSEC - Fin
crypto:/crypto/ipsec# _

```

Figura 7: Iniciado del enlace IPsec.

```

crypto:/crypto/ipsec# route
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
10.10.2.0        *                255.255.255.0   U      0      0      0 eth0
10.10.1.0        R2               255.255.255.0   UG     0      0      0 eth0
crypto:/crypto/ipsec# ping 10.10.1.2
PING 10.10.1.2 (10.10.1.2) 56(84) bytes of data:
64 bytes from 10.10.1.2: icmp_seq=1 ttl=62 time=17.8 ms
64 bytes from 10.10.1.2: icmp_seq=2 ttl=62 time=0.941 ms
64 bytes from 10.10.1.2: icmp_seq=3 ttl=62 time=0.508 ms
64 bytes from 10.10.1.2: icmp_seq=4 ttl=62 time=0.631 ms
64 bytes from 10.10.1.2: icmp_seq=5 ttl=62 time=0.518 ms
64 bytes from 10.10.1.2: icmp_seq=6 ttl=62 time=0.509 ms
^C64 bytes from 10.10.1.2: icmp_seq=7 ttl=62 time=0.495 ms
64 bytes from 10.10.1.2: icmp_seq=8 ttl=62 time=0.569 ms
64 bytes from 10.10.1.2: icmp_seq=9 ttl=62 time=0.511 ms
64 bytes from 10.10.1.2: icmp_seq=10 ttl=62 time=0.555 ms

```

Figura 8: Verificación del tunnel mediante la visualización de la tabla de ruteo y comando ping.

6- Captura del protocolo:

En la Figura 10, puede verse el resultado de una captura de los paquetes transmitidos, con el tunel habilitado y realizando una solicitud ping desde uno de los hosts.

Como puede verificarse, la información está encriptada y solo pueden visualizarse las cabeceras ESP de los paquetes.

Los paquetes ICMP que se ven, se debe a la manera en que Wireshark efectúa la captura de los mensajes en estas maquinas virtuales puesto que utilizamos adaptadores virtuales, no obstante, la información transmitida está efectivamente encriptada en su totalidad.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.204.1	192.168.204.255	BROADCAST	Local Master Announcement DELL41, Workstation, Server,
2	52.299381	192.168.10.41	192.168.10.42	ESP	ESP (SPI=0x2b90417)
3	52.306109	192.168.10.42	192.168.10.41	ESP	ESP (SPI=0x9eee3c7f)
4	52.306109	10.10.2.2	10.10.1.2	ICMP	Echo (ping) reply
5	53.297326	192.168.10.41	192.168.10.42	ESP	ESP (SPI=0x2b90417)
6	53.298381	192.168.10.42	192.168.10.41	ESP	ESP (SPI=0x9eee3c7f)
7	53.298381	10.10.2.2	10.10.1.2	ICMP	Echo (ping) reply
8	54.297551	192.168.10.41	192.168.10.42	ESP	ESP (SPI=0x2b90417)
9	54.298561	192.168.10.42	192.168.10.41	ESP	ESP (SPI=0x9eee3c7f)

Frame 2 (166 bytes on wire (1328 bytes captured) on interface 0: Capture on eth0

- Ethernet II, Src: Vmware_98:c4:71 (00:0c:29:98:c4:71), Dst: Vmware_df:d7:5c (00:0c:29:df:d7:5c)
- Internet Protocol, Src: 192.168.10.41 (192.168.10.41), Dst: 192.168.10.42 (192.168.10.42)
- Encapsulating Security Payload

Figura 9: Captura de la comunicación encriptada.

7- Desencriptado del tráfico transmitido:

Para realizar el desencriptado, se utilizó el comando Setkey -D (Figura 10), el cual muestra todos los parámetros utilizados tanto para el encriptado como para la autenticación en ambos sentidos de la comunicación. Notese que en un sentido de la comunicación, se utiliza una clave simétrica AES-CBC, mientras que en el otro sentido, la clave es totalmente distinta. Esto lo que nos muestra es que para hacer un ataque como lo es el man in the middle, se debería vulnerar la seguridad tanto en un sentido como en el otro la comunicación.

Esta información de claves se introdujo posteriormente en la configuración del protocolo ESP en Wireshark (Figura 11), y como era de esperarse, se observó toda la información transmitida y desencriptada, puesto que Wireshark da la posibilidad de ingresar estos datos para analizar la seguridad de un protocolo (Figura 12).

```

R1:/crypto/ipsec# setkey -D
192.168.10.42 192.168.10.41
  esp mode=tunnel spi=2113298487(0x7df66037) reqid=16385(0x00004001)
  E: aes-cbc e864c2cc 5b3be727 5ced3098 4c7069c5
  A: hmac-sha1 bca0f1c9 8f4132a8 78f3dd88 2727a3f3 b614aee
  seq=0x00000000 replay=32 flags=0x00000000 state=mature
  created: May 26 21:35:03 2014    current: May 26 21:43:52 2014
  diff: 529(s)    hard: 0(s)      soft: 0(s)
  last:          hard: 0(s)      soft: 0(s)
  current: 0(bytes)    hard: 0(bytes) soft: 0(bytes)
  allocated: 0    hard: 0 soft: 0
  sadb_seq=1 pid=5180 refcnt=0
192.168.10.41 192.168.10.42
  esp mode=tunnel spi=3208298319(0xbfb3abf4f) reqid=16385(0x00004001)
  E: aes-cbc 48b20d22 f404ac15 6113aa61 092165c2
  A: hmac-sha1 816bec77 789093c0 83934a72 8f271218 22913547
  seq=0x00000000 replay=32 flags=0x00000000 state=mature
  created: May 26 21:35:03 2014    current: May 26 21:43:52 2014
  diff: 529(s)    hard: 0(s)      soft: 0(s)
  last:          hard: 0(s)      soft: 0(s)
  current: 0(bytes)    hard: 0(bytes) soft: 0(bytes)
  allocated: 0    hard: 0 soft: 0
  sadb_seq=0 pid=5180 refcnt=0

```

Figura 10: Resultado del comando Setkey -D.

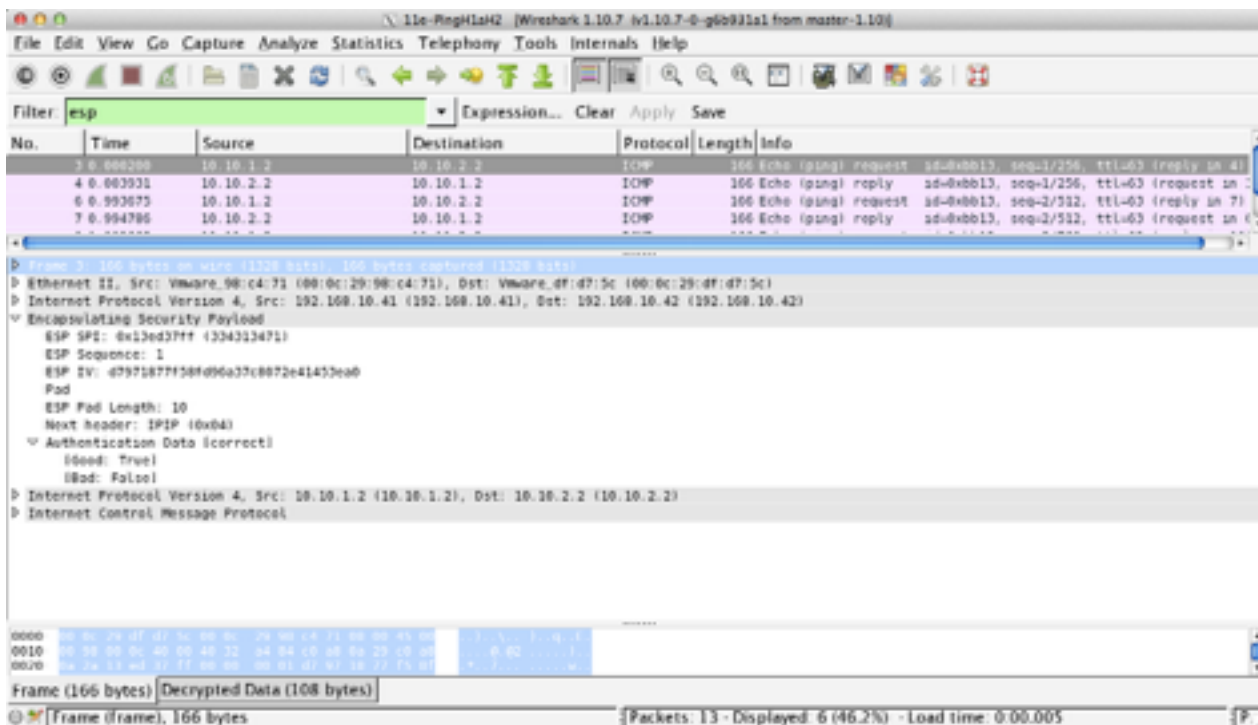
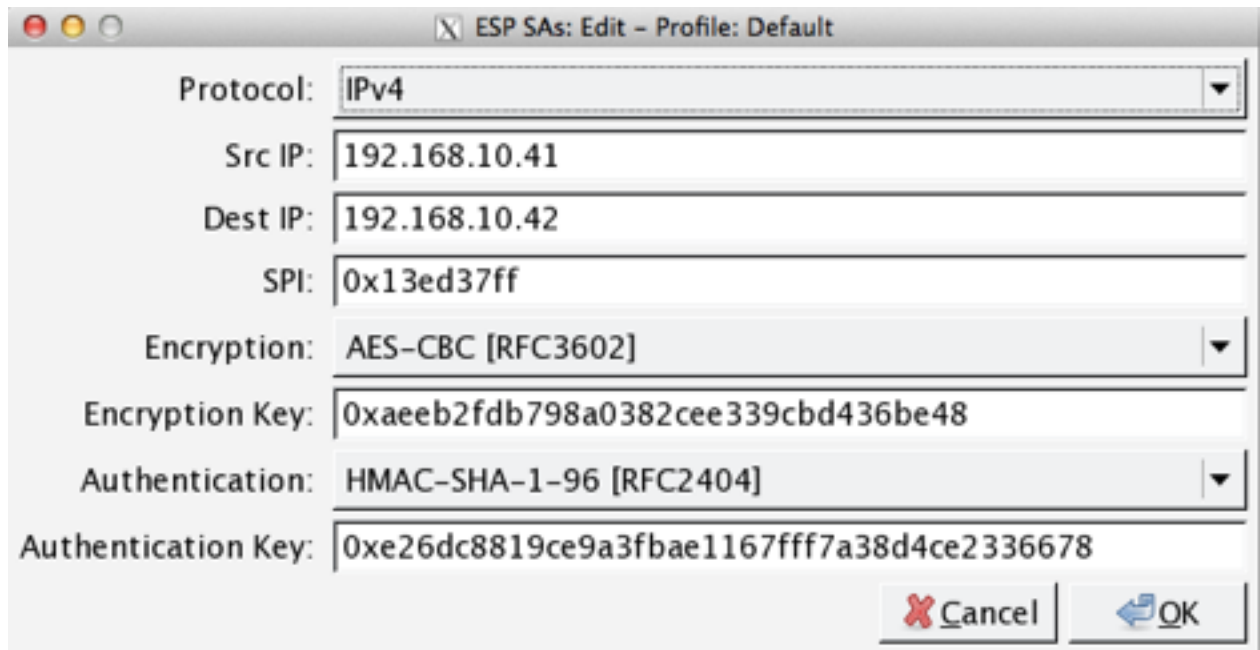


Figura 11: ESP descriptado. Notese que ahora se puede ver IPv4 junto con ICMP que en no se veía (Figura 9)



ESP SAs: Edit - Profile: Default

Protocol: IPv4

Src IP: 192.168.10.41

Dest IP: 192.168.10.42

SPI: 0x13ed37ff

Encryption: AES-CBC [RFC3602]

Encryption Key: 0xaeeb2fdb798a0382cee339cbd436be48

Authentication: HMAC-SHA-1-96 [RFC2404]

Authentication Key: 0xe26dc8819ce9a3fbae1167fff7a38d4ce2336678

Cancel OK

Figura 12: Captura de ingreso de datos de encriptación a Wireshark. Obsérvese que los datos están ingresados en notación hexadecimal