Trabajo Práctico 3 - IPSec sin 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 consistirá en configurar un túnel Ipsec entre dos routers y analizar el tráfico generado por equipos host conectados a los routers, que actúan como gateways, mediante el analizador de protocolos. Se utilizará para tal fin la distribución de Linux provista por la cátedra y se trabajará con software de virtualización de dispositivos, para lograr que el tráfico generado siempre esté dentro del equipo y no utilize en ningún momento el hardware de red.

Desarrollo

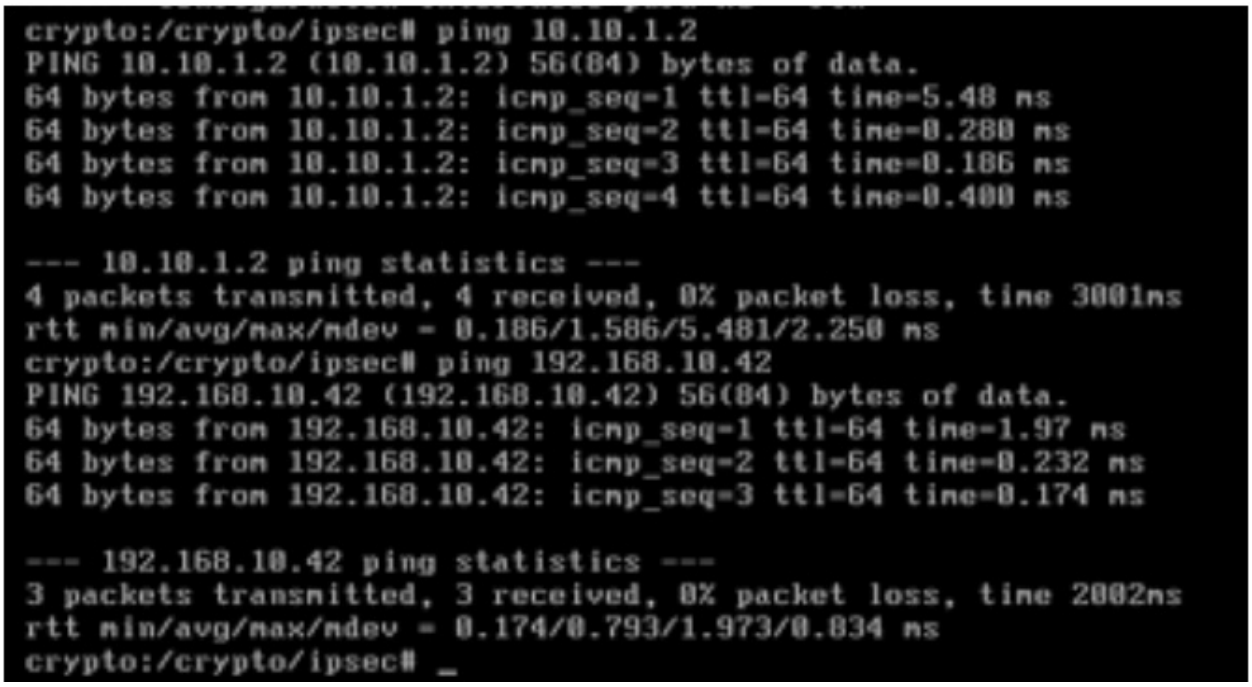
Armado del esquema de red

H1: 10.10.1.2
H2: 10.10.2.2
R1: eth0 192.168.10.41, eth1 10.10.1.1
R2: eth0 192.168.10.42, eth1 10.10.2.1

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

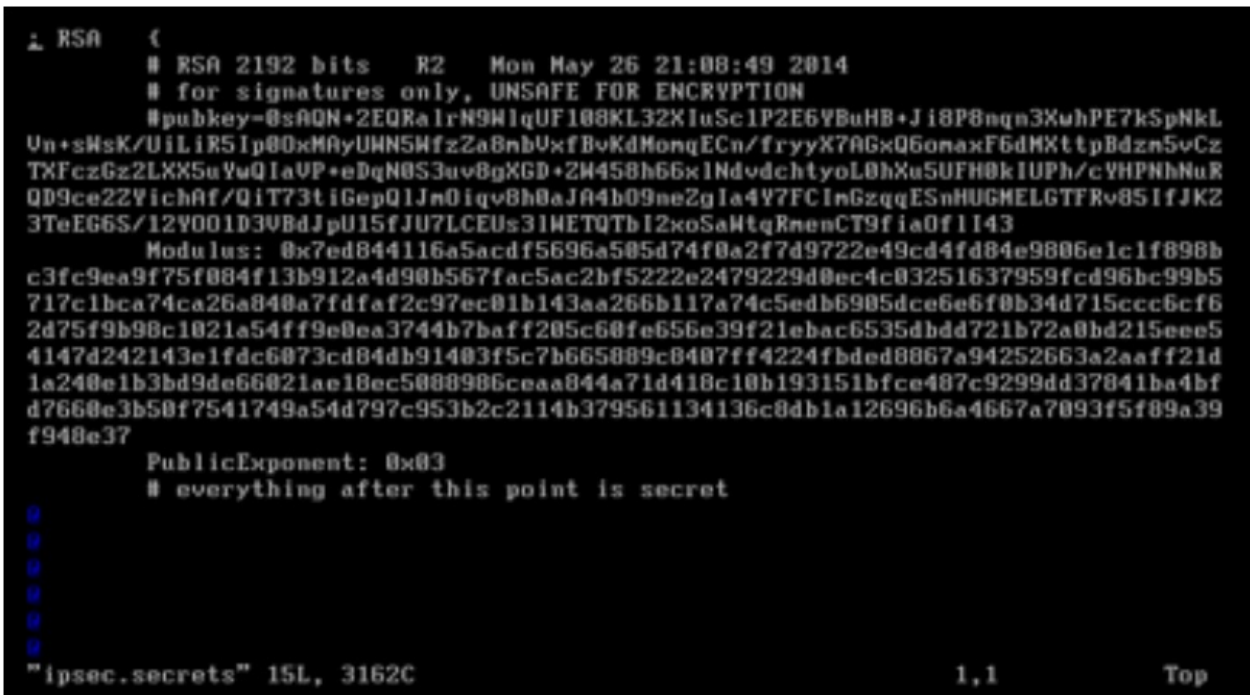
Verificación de la red:

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 siguiente figura se muestran los resultados de los ping posibles para R1: de R1 a R2 y de R1 a H1.



Generación de claves:

Se generó un par de claves (Privada y Pública) para R1 y R2 mediante el algoritmo de RSA para la creación del túnel. La siguiente figura muestra los resultados al aplicarlo en R2:



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 la copia de archivos entre diferentes host, utilizando transferencia de datos por ssh).

```
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:9b: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# _
```

Configuración e iniciación del enlace IPSec y verificación del túnel:

Se inicio el enlace mediante la utilización del script “inciarenlace.sh”

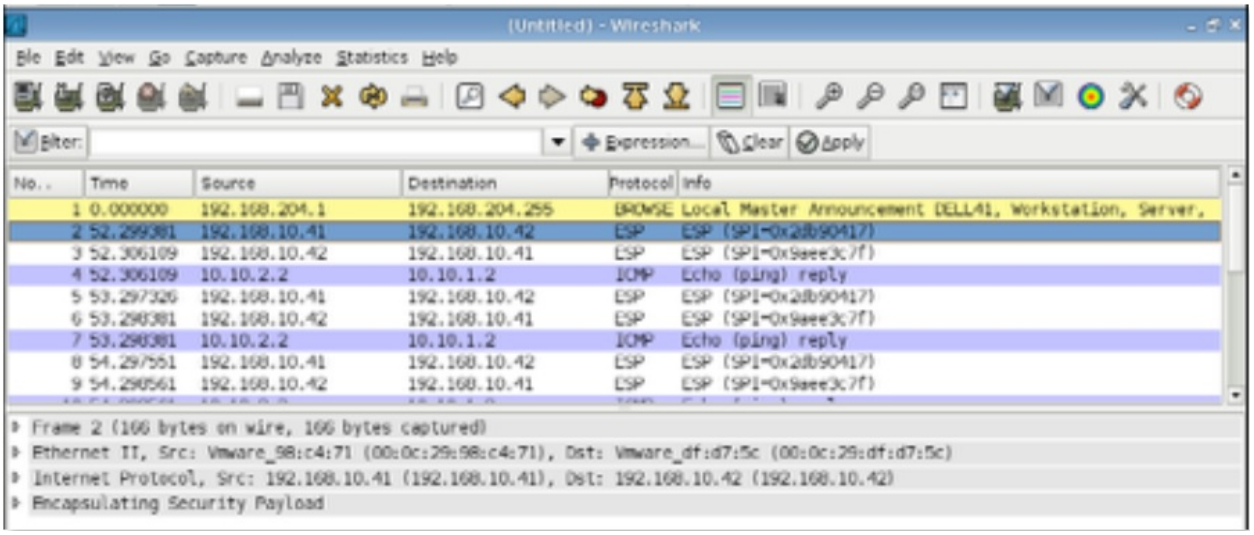
```
/etc/init.d/ipsec restart
NET: Unregistered protocol family 15
ipsec_setup: Stopping Openswan IPsec...
NET: Registered protocol family 15
Initializing IPsec netlink socket
ipsec_setup: Starting Openswan IPsec 2.4.6...
ipsec_setup: insmod /lib/modules/2.6.18-6-486/kernel/net/key/af_key.ko
ipsec_setup: 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# _
```

Una vez iniciado el enlace comenzó a haber conectividad entre ambos terminales. Se realizó un ping para verificar la conexión

```
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: icnp_seq=1 ttl=62 time=17.8 ms
64 bytes from 10.10.1.2: icnp_seq=2 ttl=62 time=0.941 ms
64 bytes from 10.10.1.2: icnp_seq=3 ttl=62 time=0.508 ms
64 bytes from 10.10.1.2: icnp_seq=4 ttl=62 time=0.631 ms
64 bytes from 10.10.1.2: icnp_seq=5 ttl=62 time=0.518 ms
64 bytes from 10.10.1.2: icnp_seq=6 ttl=62 time=0.589 ms
^C64 bytes from 10.10.1.2: icnp_seq=7 ttl=62 time=0.495 ms
64 bytes from 10.10.1.2: icnp_seq=8 ttl=62 time=0.569 ms
64 bytes from 10.10.1.2: icnp_seq=9 ttl=62 time=0.511 ms
64 bytes from 10.10.1.2: icnp_seq=10 ttl=62 time=0.555 ms
```

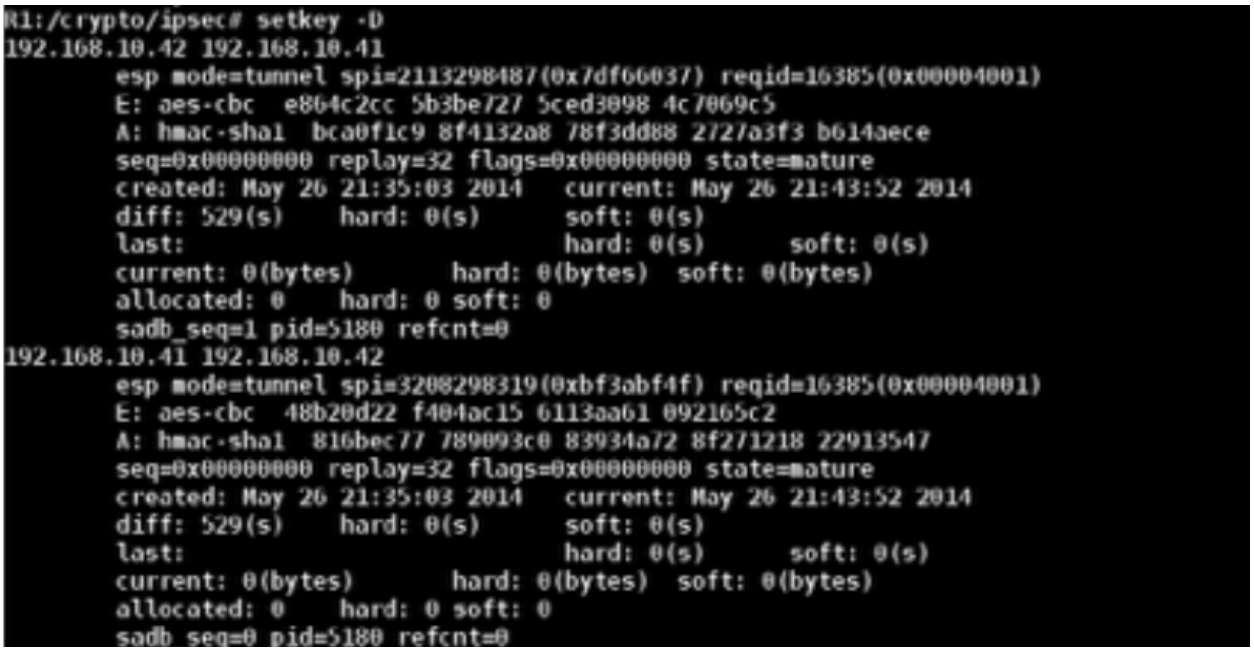
Captura del protocolo:

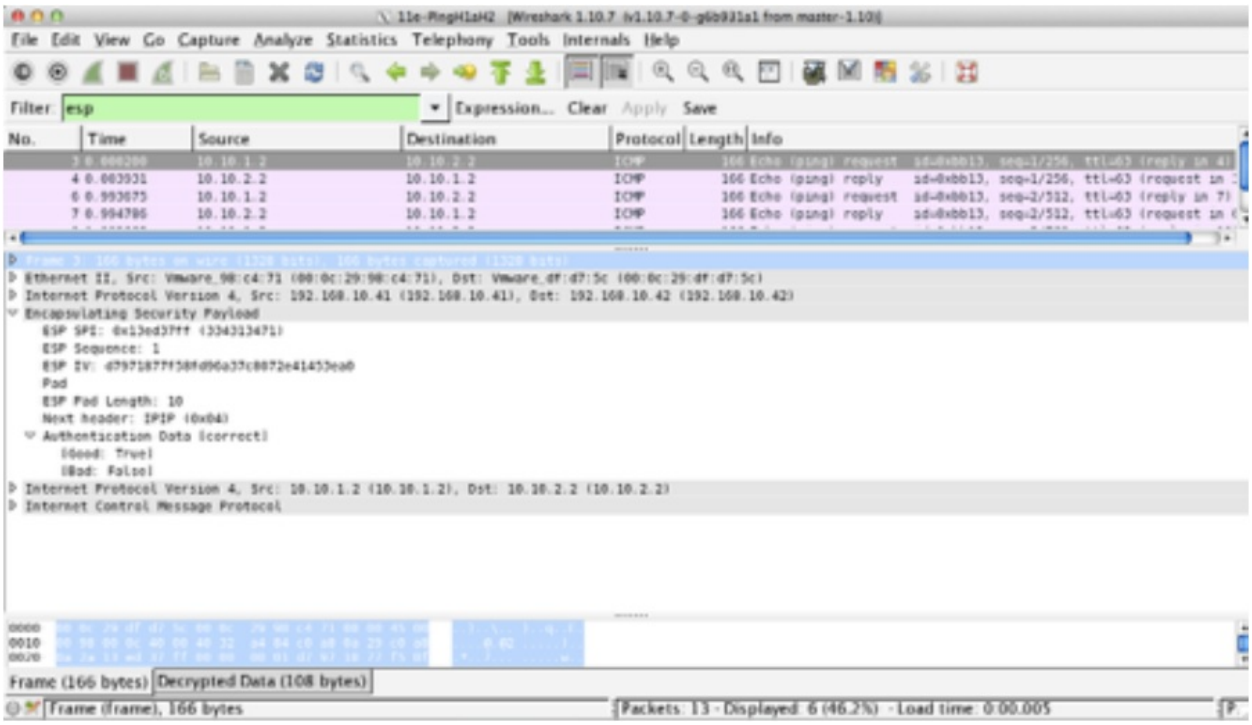
En la próxima figura, una captura del programa wireshark que muestra los paquetes transmitidos con el túnel habilitado y realizando una solicitud ping desde uno de los hosts.
Como puede verificarse, la información está encriptada y solo pueden visualizarse las cabeceras ESP de los paquetes.

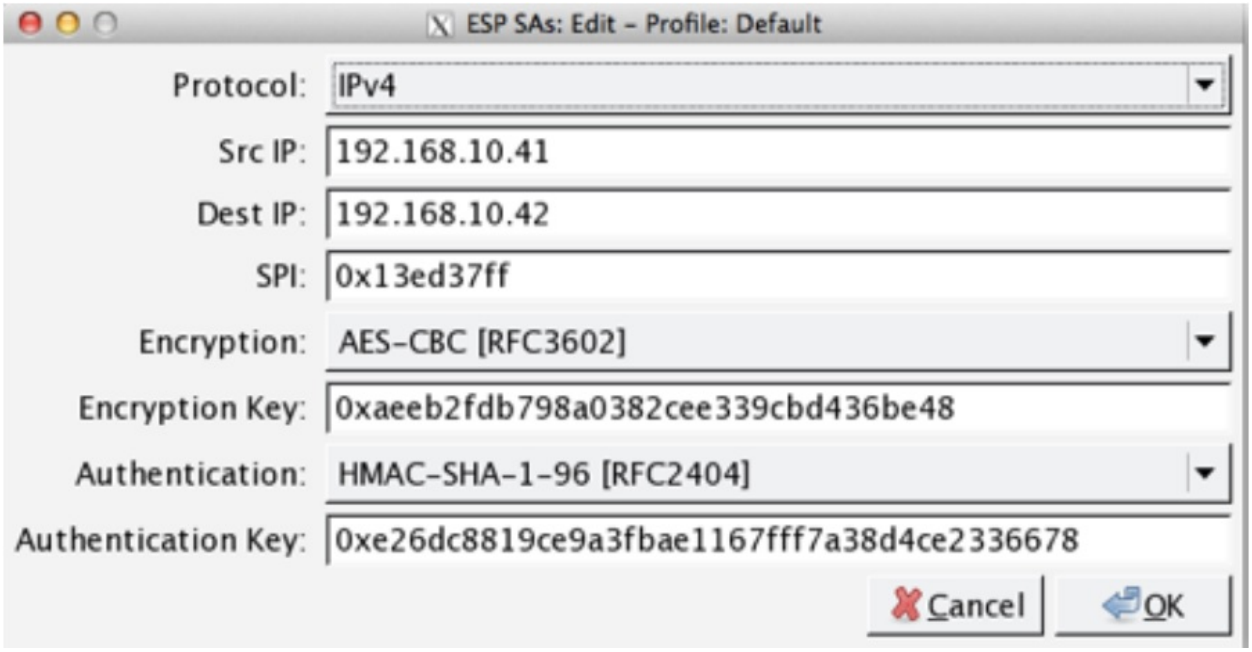


Desencriptado del tráfico transmitido:

Para realizar el desencriptado, se utilizó el comando Setkey -D, el cual muestra todos los parámetros utilizados tanto para el encriptado como para la autenticación en ambos sentidos de la comunicación. Nótese 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 obliga a que un ataque del tipo man in the middle deba 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 y como era de esperarse, se observó toda la información transmitida y desencriptada, aprovechando que Wireshark da la posibilidad de ingresar estos datos para analizar la seguridad de un protocolo.




File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: esp
Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
Encapsulating Security Payload
ESP SPI: 0x13ed37ff (334313471)
ESP Sequence: 1
Pad
ESP Pad Length: 10
Next header: IPIP (0x04)
Authentication Data (correct)
Frame (166 bytes)
Decrypted Data (108 bytes)
Frame (frame), 166 bytes
Packets: 13 · Displayed: 6 (46.2%) · Load time: 0:00.005


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