



Facultad de Ingeniería Universidad de Buenos Aires

66.44 Instrumentos Electrónicos

Trabajo Práctico N°1: Puntas de osciloscopios

Integrantes:

Padrón	Nombre	Email
91227	Soler, José Francisco	francisco._tw@hotmail.com

Índice

1. Objetivo	3
2. Desarrollo	4
2.1. Punta de prueba de alta impedancia	4
2.2. Punta de prueba de baja impedancia	4
2.3. Punta de prueba de corriente	4
2.4. Modos de funcionamiento	5
2.4.1. Modo transporte	5
2.4.2. Modo túnel	5
2.5. Protocolo AH	6
2.6. Protocolo ESP	7
3. Desarrollo	9
3.1. Configuración de las máquinas virtuales	9
3.2. Configuración de adaptadores	9
3.2.1. Configuración del adaptador de red de H1	9
3.2.2. Configuración del adaptador de red de H2	10
3.2.3. Configuración del adaptador de red de R1	10
3.2.4. Configuración del adaptador de red de R2	11
3.3. Configuración de redes y VMs	11
3.3.1. Configuración de R1	13
3.3.2. Configuración de R2	14
3.3.3. Configuración de H1	14
3.3.4. Configuración de H2	15
3.4. Verificación de la comunicación	15
3.5. Configuración túnel IPSEC	16
3.5.1. Generación de claves IPSEC	16
3.5.2. Obtención de la claves IPSEC	17
3.5.3. Configuración de IPSEC	17
3.5.4. Inicialización del enlace IPSEC	18
3.6. Verificación del túnel	18
3.7. Análisis del tráfico	19
3.8. Descriptación del tráfico	21
4. Conclusiones	23

1. Objetivo

El objetivo del presente trabajo práctico es determinar el comportamiento y fiabilidad de 3 tipos de puntas de medición, de alta impedancia, de baja impedancia y de corriente.

2. Desarrollo

Para llevar a cabo las mediciones, se utilizan los siguientes instrumentos:

- Un generador de señales con la capacidad de realizar un barrido en frecuencias.
- Un osciloscopio con la capacidad de cambiar a alta o baja su impedancia de entrada.
- Las puntas de prueba.
- Un cable que interconecta el generador con el osciloscopio, el cual, se comporta como una línea de transmisión.

2.1. Punta de prueba de alta impedancia

2.2. Punta de prueba de baja impedancia

2.3. Punta de prueba de corriente

Para este escenario, al cable utilizado se lo tuvo que romper para que entre dentro de las puntas de corriente, esto logra un cambio de la impedancia en el mismo por la disminución de la sección (generando así reflexiones y distorsiones en la señal).

IPsec (Internet Protocol security) es un conjunto de protocolos cuya función es asegurar las comunicaciones sobre el Protocolo de Internet (IP) autenticando y/o cifrando cada paquete IP en un flujo de datos. IPsec también incluye protocolos para el establecimiento de claves de cifrado.

Los protocolos de IPsec actúan en la capa de red (capa 3). Otros protocolos de seguridad para Internet, como SSL, TLS y SSH operan de la capa de transporte (capas OSI 4 a 7) hacia arriba. Esto hace que IPsec sea más flexible, ya que puede ser utilizado para proteger protocolos de la capa 4, incluyendo TCP y UDP. Otra ventaja es que para que una aplicación pueda usar IPsec no hay que hacer ningún cambio, mientras que para usar SSL y otros protocolos de niveles superiores, las aplicaciones tienen que modificar su código.

IPsec está implementado por un conjunto de protocolos criptográficos para:

- Asegurar el flujo de paquetes.
- Garantizar la autenticación mutua.
- Establecer parámetros criptográficos.

La arquitectura de seguridad IP utiliza el concepto de asociación de seguridad (SA) como base para construir funciones de seguridad en IP. Una asociación de seguridad es simplemente el paquete de algoritmos y parámetros (tales como las claves) que se está usando para cifrar y autenticar un flujo particular en una dirección. Por lo tanto, en el tráfico normal bidireccional, los flujos son asegurados por un par de asociaciones de seguridad. La decisión final de los algoritmos de cifrado y autenticación (de una lista definida) le corresponde al administrador de IPsec.

Para decidir qué protección se va a proporcionar a un paquete saliente, IPsec

utiliza el índice de parámetro de seguridad (SPI), un índice a la base de datos de asociaciones de seguridad (SADB), junto con la dirección de destino de la cabecera del paquete, que juntos identifican de forma única una asociación de seguridad para dicho paquete. Para un paquete entrante se realiza un procedimiento similar; en este caso IPsec utiliza las claves de verificación y descifrado de la base de datos de las asociaciones de seguridad.

En el caso de multicast, se proporciona una asociación de seguridad al grupo, y se duplica para todos los receptores autorizados del grupo. Puede haber más de una asociación de seguridad para un grupo, utilizando diferentes SPIs, y por ello permitiendo múltiples niveles y conjuntos de seguridad dentro de un grupo. De hecho, cada remitente puede tener múltiples asociaciones de seguridad, permitiendo autenticación, ya que un receptor sólo puede saber que alguien que conoce las claves ha enviado los datos. Hay que observar que el estándar pertinente no describe cómo se elige y duplica la asociación a través del grupo; se asume que un interesado responsable habrá hecho la elección.

2.4. Modos de funcionamiento

2.4.1. Modo transporte

En modo transporte, sólo la carga útil (los datos que se transfieren) del paquete IP es cifrada y/o autenticada. El enrutamiento permanece intacto, ya que no se modifica ni se cifra la cabecera IP; sin embargo, cuando se utiliza la cabecera de autenticación (AH), las direcciones IP no pueden ser traducidas, ya que eso invalidaría el hash. Las capas de transporte y aplicación están siempre aseguradas por un hash, de forma que no pueden ser modificadas de ninguna manera (por ejemplo traduciendo los números de puerto TCP y UDP). El modo transporte se utiliza para comunicaciones ordenador a ordenador.

Una forma de encapsular mensajes IPsec para atravesar NAT ha sido definido por RFCs que describen el mecanismo de NAT-T.

El propósito de este modo es establecer una comunicación segura punto a punto, entre dos hosts y sobre un canal inseguro. La imagen 1 ilustra esto:

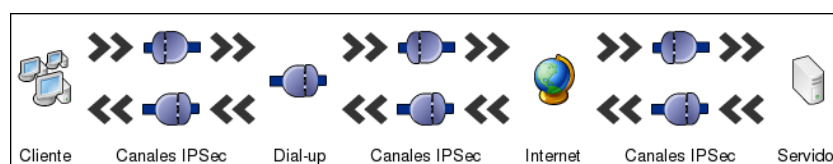


Figura 1: Esquemático de IPsec en modo transporte

2.4.2. Modo túnel

En el modo túnel, todo el paquete IP (datos más cabeceras del mensaje) es cifrado y/o autenticado. Debe ser entonces encapsulado en un nuevo paquete IP para que funcione el enrutamiento. El modo túnel se utiliza para comunicaciones red a red (túneles seguros entre routers, ejemplo, para VPNs) o comunicaciones ordenador a red u ordenador a ordenador sobre Internet. El propósito de este modo es establecer una comunicación segura entre dos redes remotas sobre un canal inseguro. La imagen 2 ilustra esto:

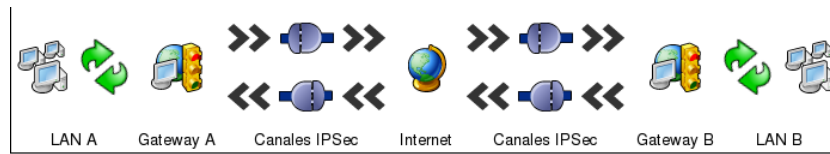


Figura 2: Esquemático de IPsec en modo túnel

2.5. Protocolo AH

AH está dirigido a garantizar integridad sin conexión y autenticación de los datos de origen de los datagramas IP. Para ello, calcula un Hash Message Authentication Code (HMAC) a través de algún algoritmo hash operando sobre una clave secreta, el contenido del paquete IP y las partes inmutables del datagrama. Este proceso restringe la posibilidad de emplear NAT, que puede ser implementada con NAT transversal. Por otro lado, AH puede proteger opcionalmente contra ataques de repetición utilizando la técnica de ventana deslizante y descartando paquetes viejos. AH protege la carga útil IP y todos los campos de la cabecera de un datagrama IP excepto los campos mutantes, es decir, aquellos que pueden ser alterados en el tránsito. En IPv4, los campos de la cabecera IP mutantes (y por lo tanto no autenticados) incluyen TOS, Flags, Offset de fragmentos, TTL y suma de verificación de la cabecera. AH opera directamente por encima de IP, utilizando el protocolo IP número 51. Un cabecera AH mide 32 bits, la figura 3 muestra un diagrama de cómo se organizan:

- next hdr: Identifica cuál es el siguiente protocolo, es decir, cual es el protocolo que será autenticado, cuál es el payload.
- AH len: El tamaño del paquete AH.
- RESERVED: Reservado para futuras aplicaciones. Debe ser igual a 0.
- Security parameters index (SPI): Indica los parametros de seguridad, que en combinación con los parámetros IP, identifican la asociación de seguridad del paquete.
- Sequence Number: Es un número creciente usado para prevenir ataques por repetición. El número está incluido en los datos encriptados, así que cualquier alteración será detectada.
- Authentication Data: Contiene el valor de identificación de integridad. Puede contener relleno. Se calcula sobre el paquete entero, incluidas la mayoría de las cabeceras. El que recibe calcula otra vez el hash, y si este no coincide, el paquete se descarta.

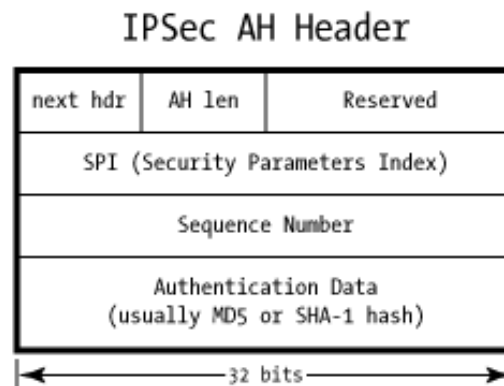


Figura 3: Cabecera IPSec AH

2.6. Protocolo ESP

Añadir encriptación hace que ESP sea un poco más complicado que AH: ESP incluye cabecera y campos para dar soporte a la encriptación y a una autenticación opcional. Además, provee los modos de transporte y túnel.

Los sistemas de encriptación más utilizados son DES, triple-DES, AES o Blowfish para asegurar la carga útil de “ojos indiscretos”. El algoritmo usado para una conexión en particular es definido por la Security Association (SA), y esta SA incluye no sólo la el algoritmo, también la llave usada.

A diferencia de AH, que da una pequeña cabecera antes de la carga útil, ESP rodea la carga útil con su protección. Los parámetros de seguridad Index y Sequence Number tienen el mismo propósito que en AH, pero agrega relleno al final del paquete antes del campo “siguiente campo” y el opcional “Authentication data”.

Es posible usar ESP sin ninguna encriptación (usar el algoritmo NULL), sin embargo estructura del paquete es de la misma forma y no brinda ninguna confidencialidad a los datos que estamos transmitiendo.

El relleno sirve para poder usar algoritmos de encriptación orientados a bloques, dado que se tiene que crear una carga a encriptar que tenga un tamaño múltiplo de su tamaño de bloque. El tamaño del relleno viene dado por el campo pad len. El campo next hdr brinda el tipo (IP, TCP, UDP, etc) de la carga útil, aunque esto sea usado como un punto para volver hacia atrás en el paquete para ver que hay en el AH.

Además de la encriptación, ESP puede proveer autenticación con la misma HMAC de AH. A diferencia de AH, esta autentifica sólo la cabecera ESP y la carga útil encriptada, no todo el paquete IP. Esto no hace que la seguridad de la autenticación más débil, pero nos da algunos beneficios importantes.

Cuando un forastero examina un paquete IP que contiene datos ESP, es prácticamente imposible adivinar que es lo que tiene dentro, excepto por los datos encontrados en la cabecera IP (siendo interesantes las direcciones IP de origen y destino). El atacante va a saber casi seguro que son datos ESP (está en la cabecera que son datos ESP), pero no va a saber de que tipo es la carga útil. El gráfico 4 muestra las cabeceras del protocolo.

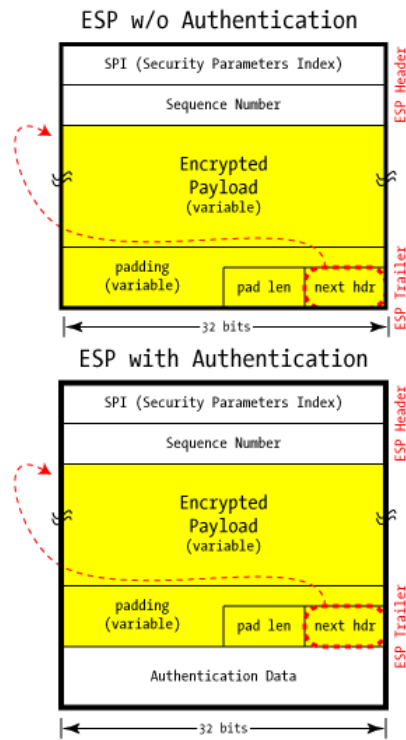


Figura 4: Cabeceras IPsec ESP

3. Desarrollo

3.1. Configuración de las máquinas virtuales

Para realizar la práctica se levantarán cuatro máquinas virtuales con el sistema VirtualBox. Las Vm fueron conectadas de la forma como muestra la figura 5 y, como se mencionó previamente cada una cumple la funcionalidad de cada componente de la red.

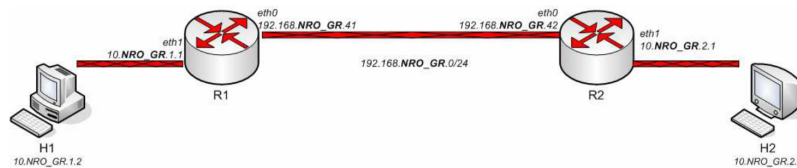


Figura 5: red de VMs

3.2. Configuración de adaptadores

A continuación se configurarán los adaptadores de red de cada VM para poder armar la red propuesta.

3.2.1. Configuración del adaptador de red de H1

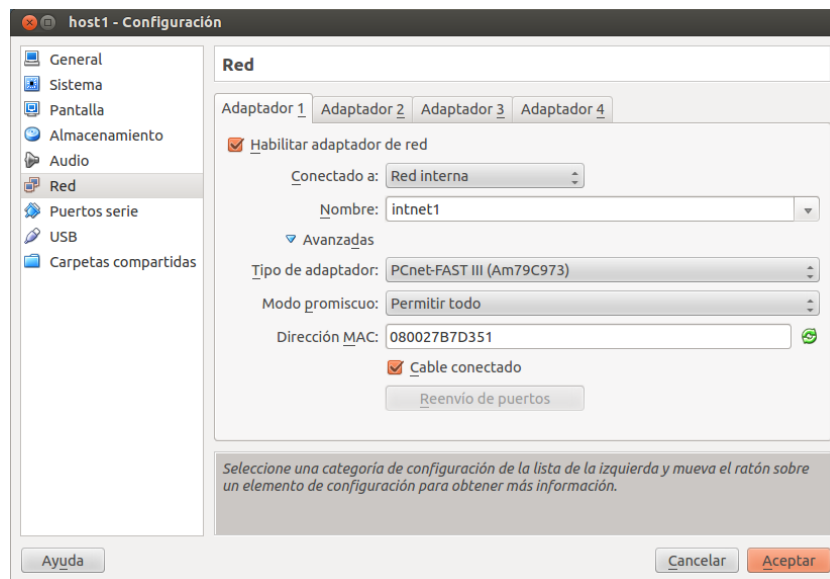


Figura 6: Configuración del adaptador de red de H1

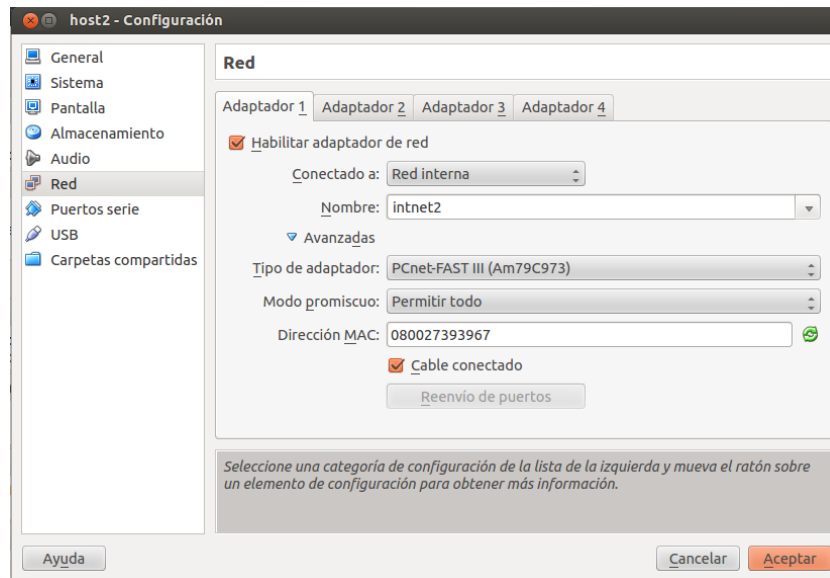


Figura 7: Configuración del adaptador de red de H2

3.2.2. Configuración del adaptador de red de H2

3.2.3. Configuración del adaptador de red de R1

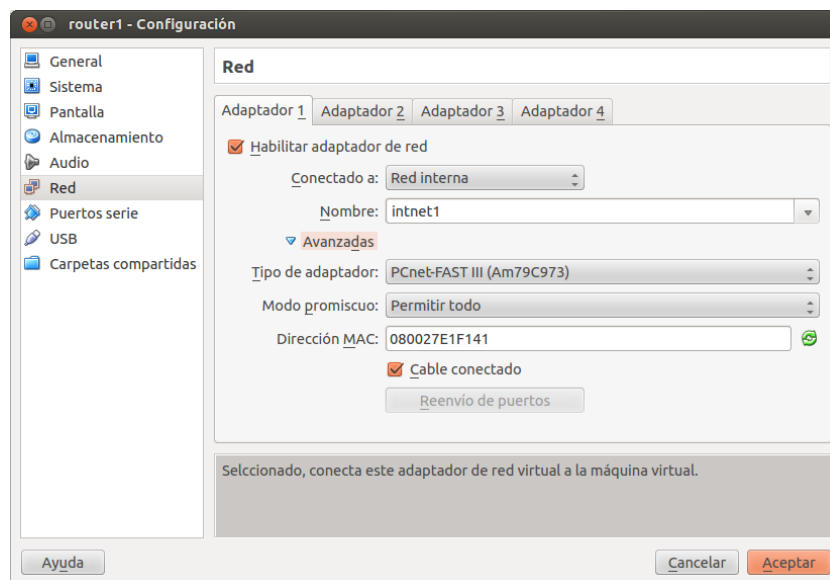


Figura 8: Configuración del adaptador 1 de red de R1

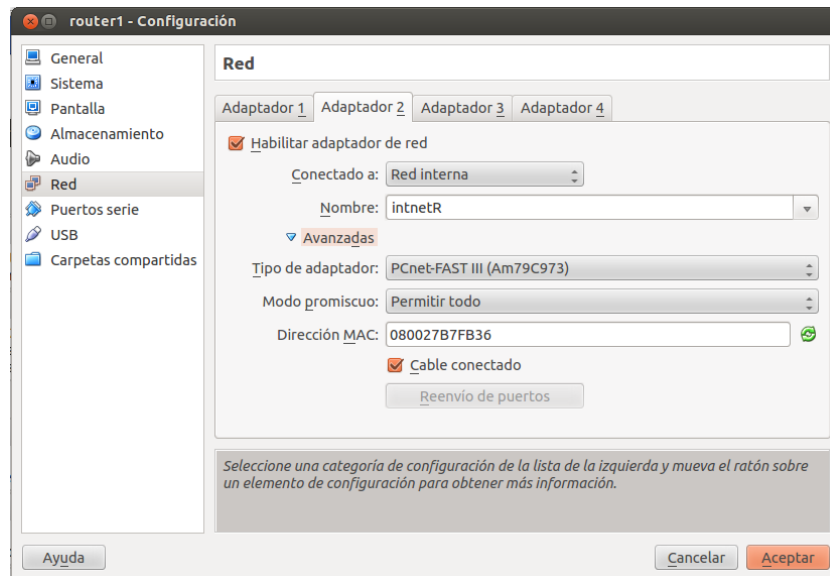


Figura 9: Configuración del adaptador 2 de red de R1

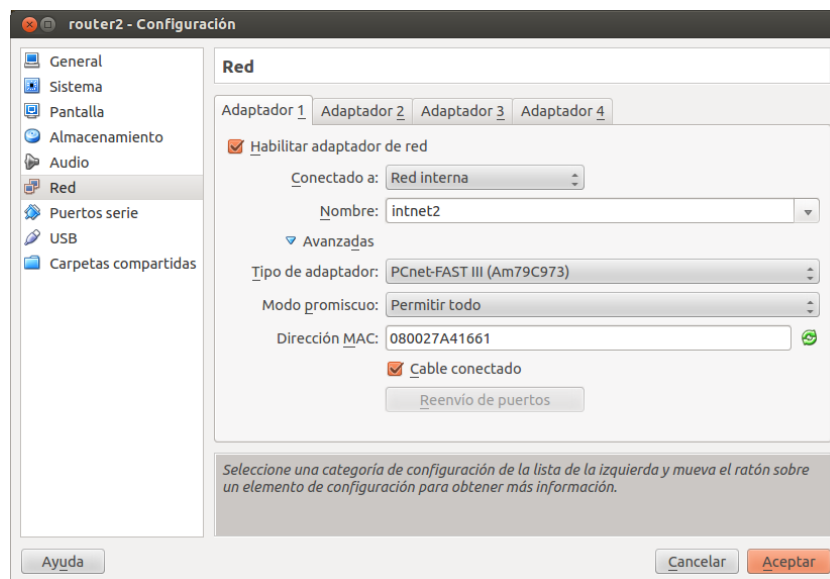


Figura 10: Configuración del adaptador 1 de red de R2

3.2.4. Configuración del adaptador de red de R2

3.3. Configuración de redes y VMs

A continuación se configurarán las redes así como los routers y hosts. Primero se debe elegir en que redes se va a trabajar, para ello hay que determinar en el archivo `/crypto/conf/config.sh` el número de grupo. En la presente prueba se utilizó el número 1. A su vez hay que asegurarse que el tipo de túnel sea

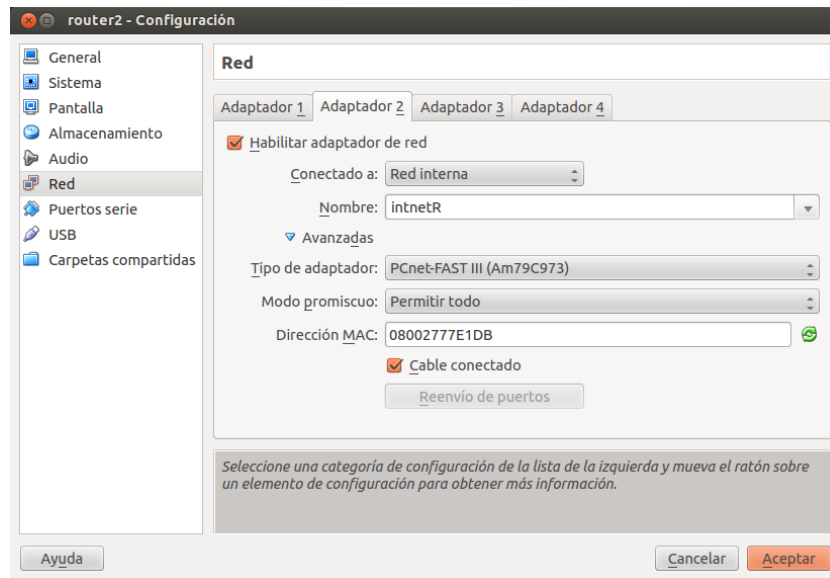


Figura 11: Configuración del adaptador 2 de red de R2

ESP-RSA. Estos pasos deben realizarse para cada VM. Se mostrará una sola imagen a modo ilustrativo(ver figuras números 12 y 13).

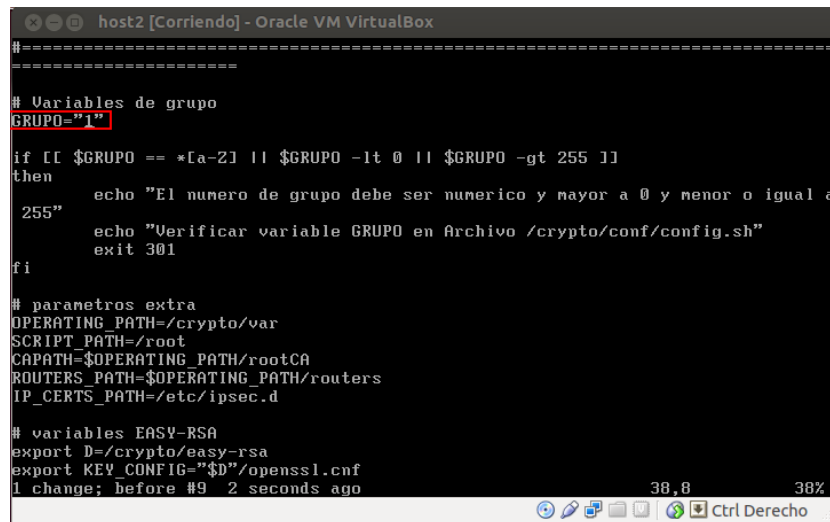
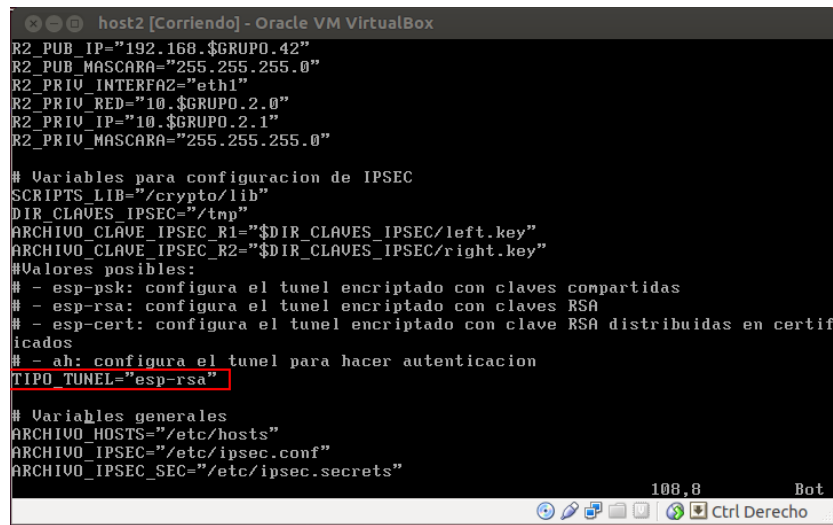


Figura 12: chequeo el número del grupo



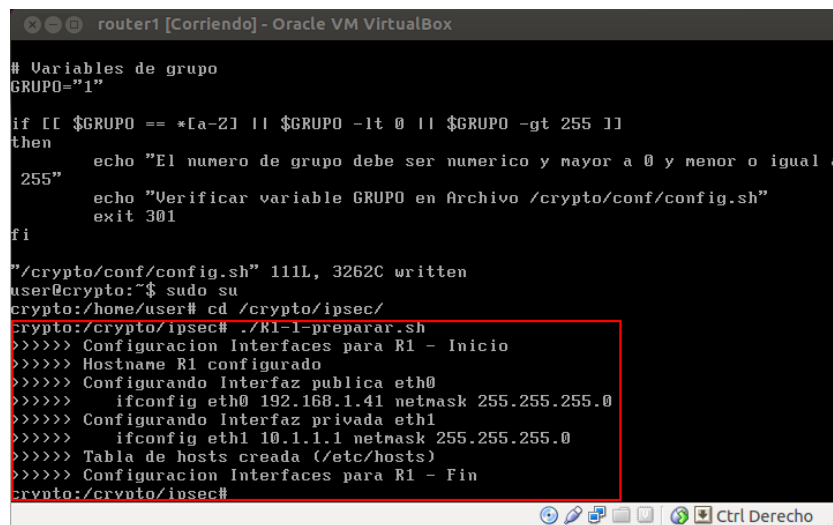
```
host2 [Corriendo] - Oracle VM VirtualBox
R2_PUB_IP="192.168.$GRUPO.42"
R2_PUB_MASCARA="255.255.255.0"
R2_PRIV_INTERFAZ="eth1"
R2_PRIV_RED="10.$GRUPO.2.0"
R2_PRIV_IP="10.$GRUPO.2.1"
R2_PRIV_MASCARA="255.255.255.0"

# Variables para configuracion de IPSEC
SCRIPTS_LIB="/crypto/lib"
DIR_CLAVES_IPSEC="/tmp"
ARCHIVO_CLAVE_IPSEC_R1="$DIR_CLAVES_IPSEC/left.key"
ARCHIVO_CLAVE_IPSEC_R2="$DIR_CLAVES_IPSEC/right.key"
#Valores posibles:
# - esp-psk: configura el tunel encriptado con claves compartidas
# - esp-rsa: configura el tunel encriptado con claves RSA
# - esp-cert: configura el tunel encriptado con clave RSA distribuidas en certificados
# - ah: configura el tunel para hacer autentificacion
TIPO_TUNEL="esp-rsa"

# Variables generales
ARCHIVO_HOSTS="/etc/hosts"
ARCHIVO_IPSEC="/etc/ipsec.conf"
ARCHIVO_IPSEC_SEC="/etc/ipsec.secrets"
```

Figura 13: Chequeo el tipo de túnel

3.3.1. Configuración de R1



```
router1 [Corriendo] - Oracle VM VirtualBox

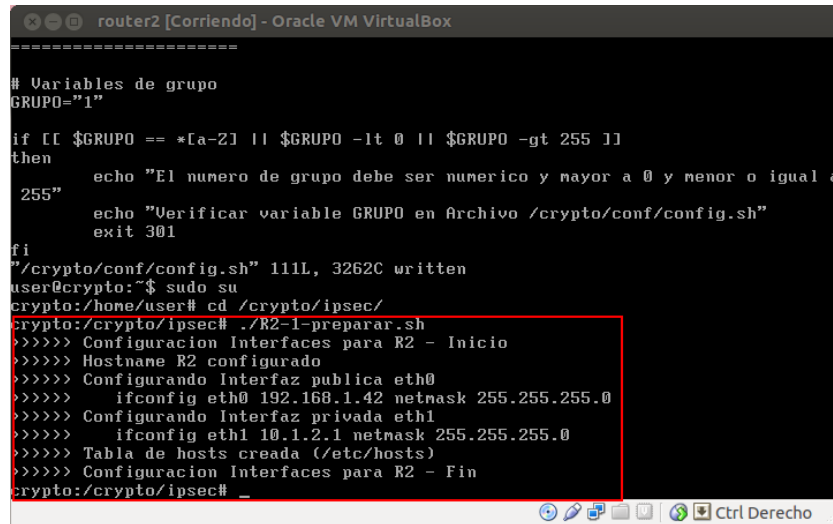
# Variables de grupo
GRUPO="1"

if [[ $GRUPO == *[a-z] || $GRUPO -lt 0 || $GRUPO -gt 255 ]]
then
    echo "El numero de grupo debe ser numerico y mayor a 0 y menor o igual a 255"
    echo "Verificar variable GRUPO en Archivo /crypto/conf/config.sh"
    exit 301
fi

"/crypto/conf/config.sh" 111L, 3262C written
user@crypto:~$ sudo su
crypto:/hone/user# cd /crypto/ipsec/
crypto:/crypto/ipsec# ./R1-1-preparar.sh
>>>>> Configuracion Interfaces para R1 - Inicio
>>>>> Hostname R1 configurado
>>>>> Configurando Interfaz publica eth0
>>>>> ifconfig eth0 192.168.1.41 netmask 255.255.255.0
>>>>> Configurando Interfaz privada eth1
>>>>> ifconfig eth1 10.1.1.1 netmask 255.255.255.0
>>>>> Tabla de hosts creada (/etc/hosts)
>>>>> Configuracion Interfaces para R1 - Fin
crypto:/crypto/ipsec#
```

Figura 14: Configuración de las interfaces de R1

3.3.2. Configuración de R2



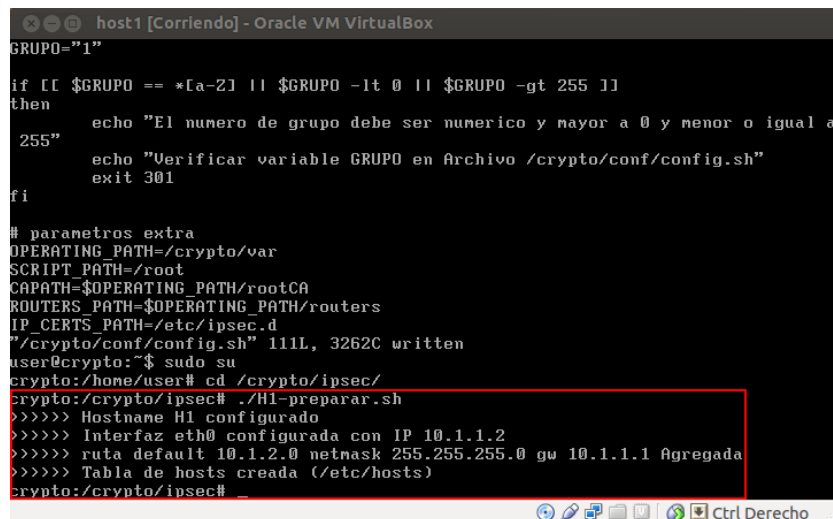
```
=====
# Variables de grupo
GRUPO="1"

if [[ $GRUPO == *[a-z] || $GRUPO -lt 0 || $GRUPO -gt 255 ]]
then
    echo "El numero de grupo debe ser numerico y mayor a 0 y menor o igual a
    255"
    echo "Verificar variable GRUPO en Archivo /crypto/conf/config.sh"
    exit 301
fi

"/crypto/conf/config.sh" 111L, 3262C written
user@crypto:~$ sudo su
crypto:/home/user# cd /crypto/ipsec/
crypto:/crypto/ipsec# ./R2-1-preparar.sh
>>>>> Configuracion Interfaces para R2 - Inicio
>>>>> Hostname R2 configurado
>>>>> Configurando Interfaz publica eth0
>>>>> ifconfig eth0 192.168.1.42 netmask 255.255.255.0
>>>>> Configurando Interfaz privada eth1
>>>>> ifconfig eth1 10.1.2.1 netmask 255.255.255.0
>>>>> Tabla de hosts creada (/etc/hosts)
>>>>> Configuracion Interfaces para R2 - Fin
crypto:/crypto/ipsec# _
```

Figura 15: Configuración de las interfaces de R2

3.3.3. Configuración de H1



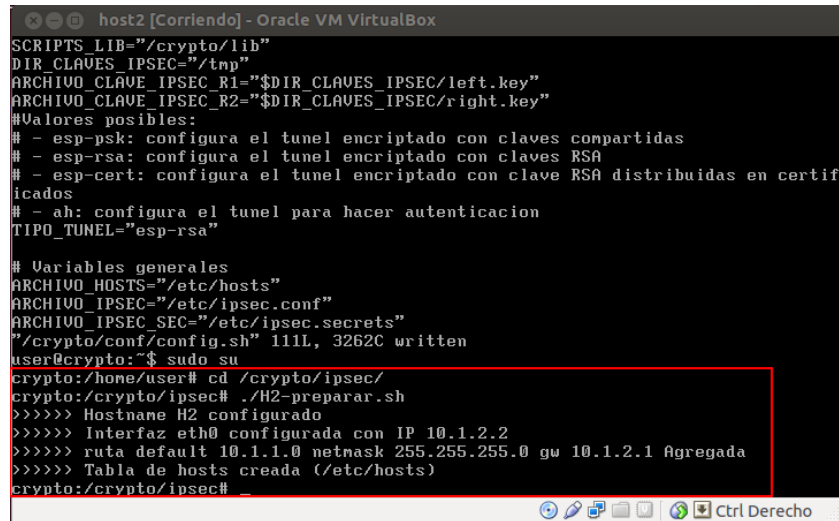
```
GRUPO="1"

if [[ $GRUPO == *[a-z] || $GRUPO -lt 0 || $GRUPO -gt 255 ]]
then
    echo "El numero de grupo debe ser numerico y mayor a 0 y menor o igual a
    255"
    echo "Verificar variable GRUPO en Archivo /crypto/conf/config.sh"
    exit 301
fi

# parametros extra
OPERATING_PATH=/crypto/var
SCRIPT_PATH=/root
CAPATH=$OPERATING_PATH/rootCA
ROUTERS_PATH=$OPERATING_PATH/routers
IP_CERTS_PATH=/etc/ipsec.d
"/crypto/conf/config.sh" 111L, 3262C written
user@crypto:~$ sudo su
crypto:/home/user# cd /crypto/ipsec/
crypto:/crypto/ipsec# ./H1-preparar.sh
>>>>> Hostname H1 configurado
>>>>> Interfaz eth0 configurada con IP 10.1.1.2
>>>>> ruta default 10.1.2.0 netmask 255.255.255.0 gw 10.1.1.1 Agregada
>>>>> Tabla de hosts creada (/etc/hosts)
crypto:/crypto/ipsec# _
```

Figura 16: Configuración de las interfaces de H1

3.3.4. Configuración de H2



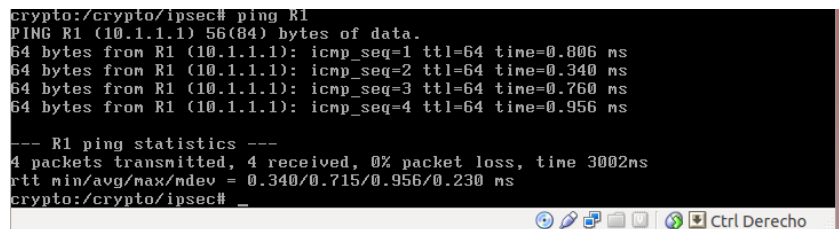
```
host2 [Corriendo] - Oracle VM VirtualBox
SCRIPTS_LIB="/crypto/lib"
DIR_CLAVES_IPSEC="/tmp"
ARCHIVO_CLAVE_IPSEC_R1="$DIR_CLAVES_IPSEC/left.key"
ARCHIVO_CLAVE_IPSEC_R2="$DIR_CLAVES_IPSEC/right.key"
#Valores posibles:
# - esp-psk: configura el tunel encriptado con claves compartidas
# - esp-rsa: configura el tunel encriptado con claves RSA
# - esp-cert: configura el tunel encriptado con clave RSA distribuidas en certificados
# - ah: configura el tunel para hacer autentificacion
TIPO_TUNEL="esp-rsa"

# Variables generales
ARCHIVO_HOSTS="/etc/hosts"
ARCHIVO_IPSEC="/etc/ipsec.conf"
ARCHIVO_IPSEC_SEC="/etc/ipsec.secrets"
"/crypto/conf/config.sh" 111L, 3262C written
user@crypto:~$ sudo su
crypto:/home/user# cd /crypto/ipsec/
crypto:/crypto/ipsec# ./H2-preparar.sh
>>>>> Hostname H2 configurado
>>>>> Interfaz eth0 configurada con IP 10.1.2.2
>>>>> ruta default 10.1.1.0 netmask 255.255.255.0 gw 10.1.2.1 Agregada
>>>>> Tabla de hosts creada (/etc/hosts)
crypto:/crypto/ipsec#
```

Figura 17: Configuración de las interfaces de H2

3.4. Verificación de la comunicación

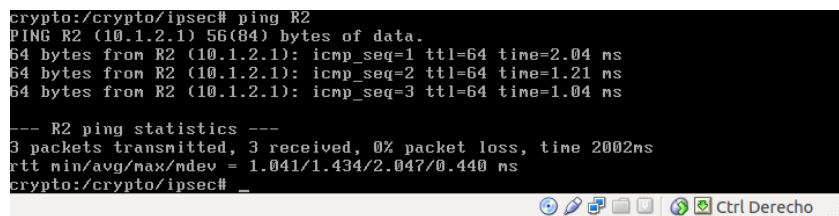
Para verificar la comunicación se realizarán pings entre los siguientes pares H1 a R1, R1 a R2 y H2 a R2. Ver figuras 18, 19 y 20



```
crypto:/crypto/ipsec# ping R1
PING R1 (10.1.1.1) 56(84) bytes of data.
64 bytes from R1 (10.1.1.1): icmp_seq=1 ttl=64 time=0.806 ms
64 bytes from R1 (10.1.1.1): icmp_seq=2 ttl=64 time=0.340 ms
64 bytes from R1 (10.1.1.1): icmp_seq=3 ttl=64 time=0.760 ms
64 bytes from R1 (10.1.1.1): icmp_seq=4 ttl=64 time=0.956 ms

--- R1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.340/0.715/0.956/0.230 ms
crypto:/crypto/ipsec#
```

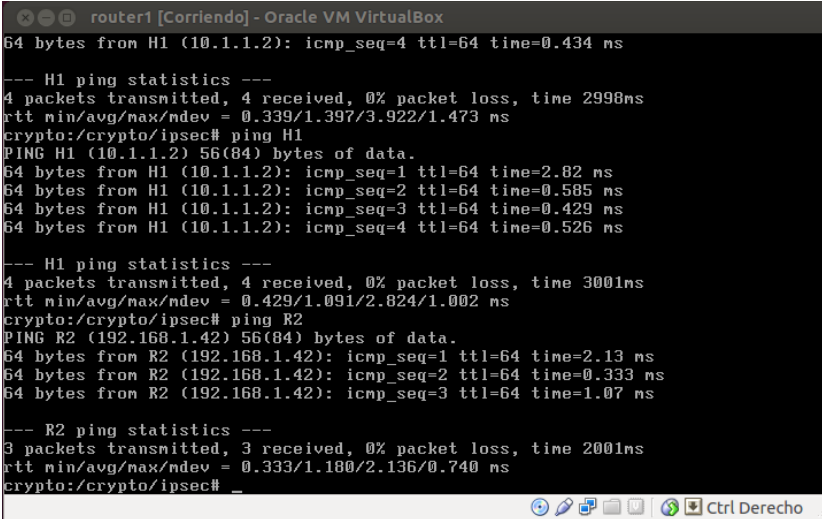
Figura 18: Pings de H1 a R1



```
crypto:/crypto/ipsec# ping R2
PING R2 (10.1.2.1) 56(84) bytes of data.
64 bytes from R2 (10.1.2.1): icmp_seq=1 ttl=64 time=2.04 ms
64 bytes from R2 (10.1.2.1): icmp_seq=2 ttl=64 time=1.21 ms
64 bytes from R2 (10.1.2.1): icmp_seq=3 ttl=64 time=1.04 ms

--- R2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.041/1.434/2.047/0.440 ms
crypto:/crypto/ipsec#
```

Figura 19: Pings de H2 a R2



```
router1 [Corriendo] - Oracle VM VirtualBox
64 bytes from H1 (10.1.1.2): icmp_seq=4 ttl=64 time=0.434 ms

--- H1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.339/1.397/3.922/1.473 ms
crypto:/crypto/ipsec# ping H1
PING H1 (10.1.1.2) 56(84) bytes of data.
64 bytes from H1 (10.1.1.2): icmp_seq=1 ttl=64 time=2.82 ms
64 bytes from H1 (10.1.1.2): icmp_seq=2 ttl=64 time=0.585 ms
64 bytes from H1 (10.1.1.2): icmp_seq=3 ttl=64 time=0.429 ms
64 bytes from H1 (10.1.1.2): icmp_seq=4 ttl=64 time=0.526 ms

--- H1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.429/1.091/2.824/1.002 ms
crypto:/crypto/ipsec# ping R2
PING R2 (192.168.1.42) 56(84) bytes of data.
64 bytes from R2 (192.168.1.42): icmp_seq=1 ttl=64 time=2.13 ms
64 bytes from R2 (192.168.1.42): icmp_seq=2 ttl=64 time=0.333 ms
64 bytes from R2 (192.168.1.42): icmp_seq=3 ttl=64 time=1.07 ms

--- R2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.333/1.180/2.136/0.740 ms
crypto:/crypto/ipsec# _
```


Figura 20: Pings de H1 a R2

Se puede observar que no se perdieron los pings, por lo tanto la red está bien configurada.

3.5. Configuración túnel IPSEC

A continuación se configurará el túnel IPsec entre R1 y R2.

3.5.1. Generación de claves IPSEC



```
crypto:/crypto/ipsec# ./R1-2-generarclaves.sh
>>>>> Generacion de claves en R1 - Inicio
>>>>> usando archivo: /etc/ipsec.secrets
>>>>> Generacion de claves en R1 - Fin
crypto:/crypto/ipsec# _
```

Figura 21: Generar Claves de IPsec en R1



```
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 22: Generar Claves de IPsec en R2

3.5.2. Obtención de la claves IPSEC

```
crypto:/crypto/ipsec# ./R1-3-obtenerclaveremota.sh
>>>>> Copiado de clave IPSEC de R2 - Inicio
root@r2's password:
right.key                               100% 436    0.4KB/s   00:00
>>>>> Clave IPSEC de R2 copiada
>>>>> Copiado de clave IPSEC de R2 - Fin
crypto:/crypto/ipsec# _
```

Figura 23: Obtener Claves de IPSec en R1

```
crypto:/crypto/ipsec# ./R2-3-obtenerclaveremota.sh
>>>>> Copiado de clave IPSEC de R1 - Inicio
root@r1's password:
Permission denied, please try again.
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 24: Obtener Claves de IPSec en R2

3.5.3. Configuración de IPSEC

```
crypto:/crypto/ipsec# ./R1-4-configurar.sh
>>>>> Creacion de archivo de configuracion IPSEC en R1 - Inicio
>>>>> Archivo de configuracion IPSEC: /etc/ipsec.conf
>>>>> Leyendo clave de R1...
>>>>> Leyendo clave de R2...
>>>>> Creacion de archivo de configuracion IPSEC en R1 - Fin
>>>>> Iniciando servicio IPSEC
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
>>>>> Servicio IPSEC Iniciado
crypto:/crypto/ipsec# _
```

Figura 25: Configuración de IPSec en R1

```
crypto:/crypto/ipsec# ./R2-4-configurar.sh
>>>>> Creacion de archivo de configuracion IPSEC en R1 - Inicio
>>>>> Archivo de configuracion IPSEC: /etc/ipsec.conf
>>>>> Leyendo clave de R1...
>>>>> Leyendo clave de R2...
>>>>> Creacion de archivo de configuracion IPSEC en R1 - Fin
>>>>> Iniciando servicio IPSEC
Initializing IPsec netlink socket
NET: Unregistered protocol family 15
ipsec_setup: Stopping Openswan IPsec...
ipsec_setup: stop ordered, but IPsec does not appear to be running!
ipsec_setup: doing cleanup anyway...
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/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
>>>>> Servicio IPSEC Iniciado
crypto:/crypto/ipsec# _
```

Figura 26: Configuración de IPsec en R2

3.5.4. Inicialización del enlace IPSEC

```
/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=>0xc915ba31
<0x231c6fc6 xfrm=AES_0-HMAC_SHA1 NATD=none DPD=none}
>>>>> Establecimiento de conexion IPSEC - Fin
crypto:/crypto/ipsec# _
```

Figura 27: Inicialización del enlace IPsec

3.6. Verificación del túnel

Para la verificación del túnel primero se observa que estén las tablas de routeo correspondientes en ambos routers y se realizarán pings de H1 a H2 y viceversa para determinar que se pueden comunicar entre sí. Ver gráficos 28, 29, 30 y 31

```
crypto:/crypto/ipsec# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.1.0 * 255.255.255.0 U 0 0 0 eth0
10.1.1.0 * 255.255.255.0 U 0 0 0 eth1
10.1.2.0 * 255.255.255.0 U 0 0 0 eth0
crypto:/crypto/ipsec# _
```

Figura 28: Tabla del router R1

```
crypto:/crypto/ipsec# route
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0      *              255.255.255.0   U     0      0      0 eth0
10.1.1.0         *              255.255.255.0   U     0      0      0 eth0
10.1.2.0         *              255.255.255.0   U     0      0      0 eth1
crypto:/crypto/ipsec#
```

Figura 29: Tabla del router R2

```
crypto:/crypto/ipsec# ping H2
PING H2 (10.1.2.2) 56(84) bytes of data.
64 bytes from H2 (10.1.2.2): icmp_seq=1 ttl=62 time=8.02 ms
64 bytes from H2 (10.1.2.2): icmp_seq=2 ttl=62 time=6.15 ms
64 bytes from H2 (10.1.2.2): icmp_seq=3 ttl=62 time=15.9 ms

--- H2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 6.156/10.043/15.951/4.247 ms
crypto:/crypto/ipsec#
```

Figura 30: pings de H1 a H2 con IPSec

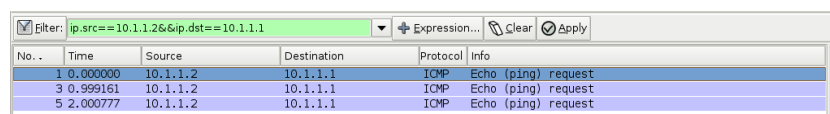
```
crypto:/crypto/ipsec# ping H1
PING H1 (10.1.1.2) 56(84) bytes of data.
64 bytes from H1 (10.1.1.2): icmp_seq=1 ttl=62 time=4.44 ms
64 bytes from H1 (10.1.1.2): icmp_seq=2 ttl=62 time=2.02 ms
64 bytes from H1 (10.1.1.2): icmp_seq=3 ttl=62 time=3.90 ms

--- H1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 2.020/3.456/4.448/1.039 ms
crypto:/crypto/ipsec#
```

Figura 31: pings de H2 a H1 con IPSec

3.7. Análisis del tráfico

A continuación se correrá el wireshark sobre la interfaz eth0, y se observará que tipo de tráfico hay.



No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.1.1.2	10.1.1.1	ICMP	Echo (ping) request
3	0.999161	10.1.1.2	10.1.1.1	ICMP	Echo (ping) request
5	2.000777	10.1.1.2	10.1.1.1	ICMP	Echo (ping) request

Figura 32: tráfico en la red entre H1 y R1

Se puede observar solo los paquetes ICMP enviados por el host 1. No se observan paquetes ESP ya que esa comunicación no está todavía en el túnel IPSec.

Ahora se observará el tráfico entre R1 y R2 al hacer un ping tanto de H2-H1, ver figura 33

1	0.000000	192.168.1.42	192.168.1.41	ESP	ESP (SPI=0xda38cfd)
2	0.000109	10.1.2.2	10.1.1.2	ICMP	Echo (ping) request
3	0.000116	192.168.1.41	192.168.1.42	ESP	ESP (SPI=0xc7f4d553)
4	1.003609	192.168.1.42	192.168.1.41	ESP	ESP (SPI=0xda38cfd)
5	1.003609	10.1.2.2	10.1.1.2	ICMP	Echo (ping) request
6	1.005230	192.168.1.41	192.168.1.42	ESP	ESP (SPI=0xc7f4d553)
7	2.003239	192.168.1.42	192.168.1.41	ESP	ESP (SPI=0xda38cfd)
8	2.003239	10.1.2.2	10.1.1.2	ICMP	Echo (ping) request
9	2.004275	192.168.1.41	192.168.1.42	ESP	ESP (SPI=0xc7f4d553)
10	3.007525	192.168.1.42	192.168.1.41	ESP	ESP (SPI=0xda38cfd)
11	3.007525	10.1.2.2	10.1.1.2	ICMP	Echo (ping) request
12	3.008632	192.168.1.41	192.168.1.42	ESP	ESP (SPI=0xc7f4d553)
13	4.007411	192.168.1.42	192.168.1.41	ESP	ESP (SPI=0xda38cfd)
14	4.007411	10.1.2.2	10.1.1.2	ICMP	Echo (ping) request

Figura 33: tráfico dentro del túnel

Como se puede observar, los paquetes entre los routers (192.168.1.41 a 192.168.1.42 y viceversa) son paquetes ESP propios del túnel. También se ve como el túnel encripta con los paquetes ESP los request de los paquetes ICMP.

Analizando algún paquete ESP podemos verificar por ejemplo, como el datagrama IP tiene el valor 50 (0x32 en hexadecimal) en el campo del protocolo (ver figura 34). A su vez, se observa que está en modo túnel el enlace ya que los destinos y orígenes están en encapsulados dentro del datagrama y en estas direcciones. En el exterior del datagrama IP figuran las IPs de la red que conecta los routers (ver figura 35).

Header length: 20 bytes	
▶ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)	
Total Length: 152	
Identification: 0x0800 (2048)	
▶ Flags: 0x04 (Don't Fragment)	
Fragment offset: 0	
Time to live: 64	
Protocol: ESP (0x32)	
▶ Header checksum: 0xae90 [correct]	

Figura 34: protocolo utilizado por el túnel IPsec

Ethernet II, Src: CadmusCo_77:e1:db (08:00:27:77:e1:db), Dst: CadmusCo_e1:f1:41 (08:00:27:e1:f1:41)	
▼ Destination: CadmusCo_e1:f1:41 (08:00:27:e1:f1:41)	
Address: CadmusCo_e1:f1:41 (08:00:27:e1:f1:41)	
.....0..... = IG bit: Individual address (unicast)	
.....0..... = LG bit: Globally unique address (factory default)	
▼ Source: CadmusCo_77:e1:db (08:00:27:77:e1:db)	
Address: CadmusCo_77:e1:db (08:00:27:77:e1:db)	
.....0..... = IG bit: Individual address (unicast)	
.....0..... = LG bit: Globally unique address (factory default)	

Figura 35: Destinos y orígenes encriptados

Para establecer el enlace se ejecutan dos fases, las cuales son el Main Mode (Identity Protection) y el Quick Mode.

El Main Mode es el encargado de generar un canal seguro en el cual se puedan intercambiar los datos necesarios de la segunda fase. El intercambio de las claves se realiza utilizando Diffie-Hellman y luego se autentica a cada una de las partes. En la fase Quick Mode es en donde se produce la negociación de las IPsec SA's para poder generar el túnel IPsec. Se actualiza la información utilizada para encriptar, así como también se actualizan de forma periodica las SA's. El gráfico número 36 muestra la comunicación.

85	52.145637	192.168.1.41	192.168.1.42	ISAKMP Identity Protection (Main Mode)
86	52.148897	192.168.1.42	192.168.1.41	ISAKMP Identity Protection (Main Mode)
87	52.170428	192.168.1.41	192.168.1.42	ISAKMP Identity Protection (Main Mode)
88	52.176068	192.168.1.42	192.168.1.41	ISAKMP Identity Protection (Main Mode)
89	52.210653	192.168.1.41	192.168.1.42	ISAKMP Identity Protection (Main Mode)
90	52.224003	192.168.1.42	192.168.1.41	ISAKMP Identity Protection (Main Mode)
91	52.252050	192.168.1.41	192.168.1.42	ISAKMP Quick Mode
92	52.260741	192.168.1.42	192.168.1.41	ISAKMP Quick Mode
93	52.380934	192.168.1.41	192.168.1.42	ISAKMP Quick Mode

Figura 36: comunicación entre routers al establecer conexión

3.8. Desencriptación del tráfico

Para desencriptar el tráfico se ejecuta el comando `setkey -D`.

Se agregan los datos obtenidos (mostrados en la figura 37) a la configuración del WireShark tal como está indicado en el enunciado del trabajo práctico.

La figura 38 muestra como el paquete de ICMP esta desencriptado. Se pueden apreciar las direcciones origen y destino dentro del paquete tanto del enlace R1-R2 como de los hosts H1-H2.

Al bajar el tunel y volver a levantarlo las claves se pierden y no es posible desencriptar el tráfico sin actualizar las mismas en el programa. Ver gráfico 39

```
R1:/crypto/ipsec# setkey -D
192.168.1.42 192.168.1.41
    esp mode=tunnel spi=841496426(0x3228376a) reqid=16385(0x00004001)
    E: aes-cbc 3bcf8acd c2d132e7 648f63e6 88bf87b4
    A: hmac-sha1 d687253b d518fdce 7afd603f 7a396c28 3cc206e8
    seq=0x00000000 replay=32 flags=0x00000000 state=mature
    created: Aug 22 02:12:56 2013 current: Aug 22 02:17:05 2013
    diff: 249(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=6719 refcnt=0
192.168.1.41 192.168.1.42
    esp mode=tunnel spi=2689933670(0xa0552166) reqid=16385(0x00004001)
    E: aes-cbc 4555fd21 5246d70e 42aaa252 980c4222
    A: hmac-sha1 8bb7b0fc 5b828f5d e1e418ed 1b7667f1 18aa2eee
    seq=0x00000000 replay=32 flags=0x00000000 state=mature
    created: Aug 22 02:12:56 2013 current: Aug 22 02:17:05 2013
    diff: 249(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=6719 refcnt=0
R1:/crypto/ipsec#
```

Figura 37: Desencriptando el tráfico

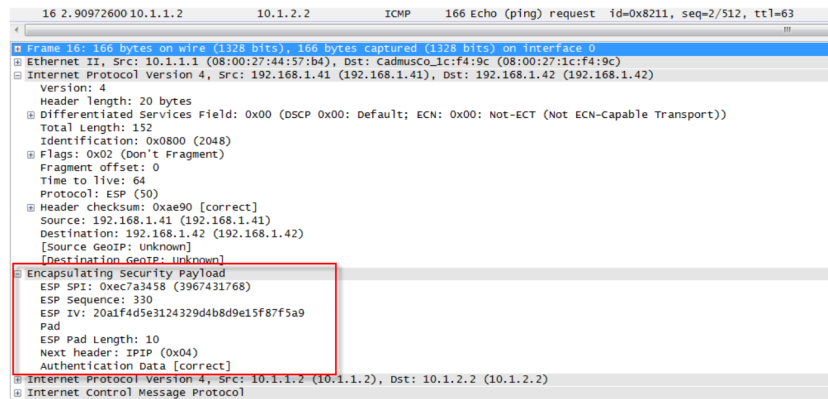


Figura 38: Tráfico descriptado

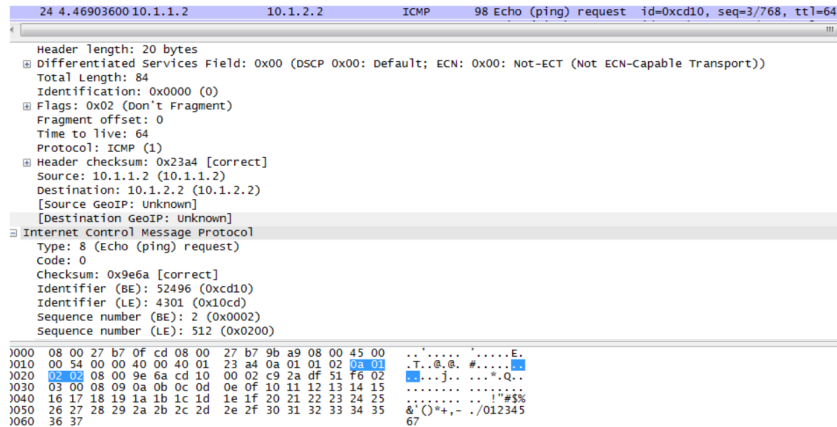


Figura 39: Tráfico encriptado nuevamente por tener claves erróneas

4. Conclusiones

Se comprobó que si un túnel IPSec en modo ESP es configurado correctamente entre dos redes privadas, estableciendo una VPN, es imposible sin conocer las claves de las SA's determinar el tráfico que se intercambia a través de ella. Por otro lado al usar el método de protección de identidad, además de impedir que personas ajenas a la comunicación puedan ver los paquetes, también se evita que tengan información acerca de quienes están intercambiando tráfico.