TALLER DE SEGURIDAD INFORMÁTICA

CURSO 2020

Informe - Laboratorio 01
Grupo 03

Autores:

Agustín RIEPPI 4.754.760-1

Guillermo COELHO 5.043.831-6

6 de septiembre de 2020

Índice

1. Introducción **2**

2. Implementación **2**

2.1. Escenario 2

2.2. Configuración IPSec 3

2.3. Funciones criptográficas 4

2.3.1. Encriptado 5

2.3.2. Desencriptado 5

2.4. Verificación 5

3. Mejoras y trabajo a futuro **8**

4. Conclusión **8**

1. Introducción

Este laboratorio tiene como fin entender el funcionamiento y la implementación de algunas de las características de IPSec [1].
IPSec es un conjunto de protocolos que tienen como objetivo agregar seguridad (confidencialidad, integridad y autenticación) a los paquetes de la capa IP.
En este trabajo se trabaja con una subconjunto de algunas de las características de IPSec, con el objetivo de implementar un programa que simule el comportamiento de IPsec sobre mensajes de ICMP [2] entre 2 host.

2. Implementación

En el laboratorio se plantea una realidad simplificada en la cual se asume una serie de hipótesis de trabajo sobre el funcionamiento de IPSec:

- Solo se usa con el protocolo Encapsulating Security Payload (ESP)
- Se utiliza IPSec en modo transporte
- El algoritmo de cifrado es 3DES-CBC
- La configuración de claves y la creación de las Security Association (SA) se realiza de forma manual utilizando la herramienta setKey [3]

A partir de estas hipótesis y del programa proporcionado por el Grupo de Seguridad Informática de la UDELAR (ping-sec) que simula la herramienta Ping del protocolo ICMP, se implementa la funciones encargadas de encriptar y desencriptar el payload de los paquetes enviados y recibidos a través del programa.

2.1. Escenario

El escenario de elección fue trabajar sobre una maquina física con dos maquinas virtuales corriendo Linux con una arquitectura de 32 bits. Las distribuciones utilizadas para la verificación son Linux Mint Debian Edition 4 y Ubuntu 16.04, tomando en cuenta que soportan las funcionalidades necesarias.
Las maquinas virtuales son configuradas como 'Host 1' y se le asigna la IP: 192.168.1.100, desde donde se usa el programa sin tener ningún tipo de configuración IPsec. Por

otro lado, se configura el 'Host 2' como la segunda máquina virtual a la cual se le asigno la IP: 192.168.1.200, a esta se le aplica la configuración IPsec nativo con las correspondientes claves y algoritmos.
También como complemento se prueba desde una maquina física con Linux 64 bits (Elementary Os 5.7) usando el programa proporcionado, a una maquina virtual configurada con IPsec nativo.

2.2. Configuración IPSec

Se crea un archivo de configuración en el 'Host 2' para usar IPsec nativo, el mismo se conforma de 3 partes.
La primera se encarga de vaciar claves SA y las SPD.
La segunda asigna las SA correspondientes a una de entrada y otra de salida. Para esto se usa el comando 'add' el cual añade una SA, este requiere las IP de origen y destino, el SPI, el algoritmo (cada tipo de algoritmo tienen su correspondiente flag, -A para autenticación y -E cifrado) y por último se especifica la clave, esta puede ser en ASCII entre comillas dobles o en hexadecimal con el prefijo 0x.
La tercera y última, se configuran las SPD, se usa el comando 'spdadd', este comando define los paquetes que filtro se le aplica (Descartar, transfereir o proteger con IPSec), con qué protocolo y clave. Lo primero que se especifica son las IP de origen y destino, en nuestro caso 'Host 2' es el que lleva el archivo de configuración, por lo que la IP de destino sera la 192.168.1.200 y de origen la 192.168.1.100. Sabiendo esto lo siguiente es especificar los paquetes, protocolo y puerto que se va a cifrar, en este caso se usa 'any' para indicar que se le aplique IPSec a cualquier paquete. Al final se le indica qué política emplear (-P), aquí se indica la dirección de los paquetes (in/out), recordando el escenario en el que se esta trabajando, la acción que se va a aplicar (ipsec), el protocolo (esp), el modo (transport) y el nivel (require).

```
#!usr/sbin/setkey -f

#vaciar las SAD y SPD
flush;
spdflush;

#SAs para ESP
add 192.168.1.200 192.168.1.100 esp 0x00000200 -E 3des-cbc "ips3ctasi1key3descbc
in02";
add 192.168.1.100 192.168.1.200 esp 0x00000100 -E 3des-cbc "ips3ctasi1key3descbc
in01";

#Seguridad
spdadd 192.168.1.100 192.168.1.200 any -P in ipsec
        esp/transport//require;

spdadd 192.168.1.200 192.168.1.100 any -P out ipsec
        esp/transport//require;
```

Figura 1: Archivo de configuración de IPsec nativo

Con el archivo de configuración realizado (en el caso de que se use IPsec nativo, como lo fue en una de las pruebas realizadas antes de usar el programa, se deberá crear un archivo así en el otro host de forma análoga) se lo puede cargar con el comando 'setkey'. Para esto se necesita permisos root y la ruta en el cual se encuentra el archivo.
Para inicializar entonces se usa el comando **sudo setkey -f /etc/ipsec.conf** y para comprobar que se haya aplicado se puede usar **setkey -D** para verificar las SA y **setkey -DP** para las SPD.

2.3. Funciones criptográficas

La tarea central de este laboratorio es implementar las funciones de encriptado y desencriptado de forma tal, que el resultado de estas operaciones sea compatible con IPSec bajo las hipótesis de trabajo definidas.
Para las operaciones criptográficas se utiliza la librería Gcrypt [4]. El algoritmo que se utiliza es triple-DES [5], este tiene como característica que trabaja con bloques de 8 bytes y utiliza una clave de 168 bits, pero el método del la libreria Gcrypt requiere que se use una clave de 192 bits porque los bits mas significativos son ignorados.
Como el algoritmo es utilizado en modo Cipher Block Chaining (CBC) [6] necesita un initialization vector (IV), y como el algoritmo cifra en bloques de 8 bytes este vector

tiene que ser del mismo tamaño.

2.3.1. Encriptado

La función de encriptado recibe el payload del mensaje, el largo del mensaje, la clave de encriptado, y el algoritmo de cifrado.
Lo primero que se realiza es generar un nonce random del largo necesario para ser usado como IV. Luego se instancia un manejador encargado del cifrado proporcionado por Gcrypt, al cual se le configura la clave recibida y el IV generado.
Una vez realizada la configuración se procede a encriptar el payload a través del manejador instanciado.
Como es indicado en el protocolo ESP el IV utilizado tiene que se agregado al frente del payload encriptado, por lo cual luego del encriptado es necesario desplazar la información 8 bytes hacia la derecha y en el espacio liberado insertar el IV.
Por ultimo se cierra el manejador instanciado para que pueda remplazar con ceros toda la información sensible asociada al manejador.

2.3.2. Desencriptado

Para el desencriptado se recibe las claves y definición de algoritmo para el desencriptado, además se recibe el payload encriptado y el largo de este.
Como se sabe que la información que se recibe cumple con el estándar del protocolo ESP en IPSec,y que el largo del IV utilizado por el método de cifrado es de 8 bytes, se procede a extraer el IV del header de la data recibida (Los 8 primeros bytes), luego se desplaza la información 8 bytes hacia la izquierda sobrescribiendo el IV extraído.
La instanciación del manejador se realiza de manera análoga a la de encriptado.
Luego se utiliza se desencripta la data y se procede a cerrar el manejador.

2.4. Verificación

A la hora de la verificación se encontró algunos problemas con la implementación del programa ping-sec, uno de los mas relevantes es el manejo incorrecto del numero de secuencia del SPI de los mensajes, otro problema relevante es que para interceptar la terminación de programa a través del teclado, se utiliza un numero identificador definido como una constante en vez de utilizar la información propor-

cionada por el SO para identificarlo, el problema que genera es que el numero identificador puede variar según el SO. Solventados y/o obviados estos inconvenientes se realiza una serie de verificaciones para probar el correcto funcionamiento del programa ping-sec.
Para la verificación se hace uso del programa WireShark [7], el cual ayuda a analizar el trafico de red. Primero se hace un análisis del trafico utilizando el programa ping ofrecido por el SO teniendo los host configurados con IPSec, esto se realiza para ver el correcto funcionamiento de IPSec y para entender la estructura de los mensajes encriptados siguiendo los protocoles de IPSec. Para poder ver el contenido dentro de los paquetes se hace uso de la funcionalidad ofrecida por Wireshark para desencriptar los mensajes cifrados con ESP.

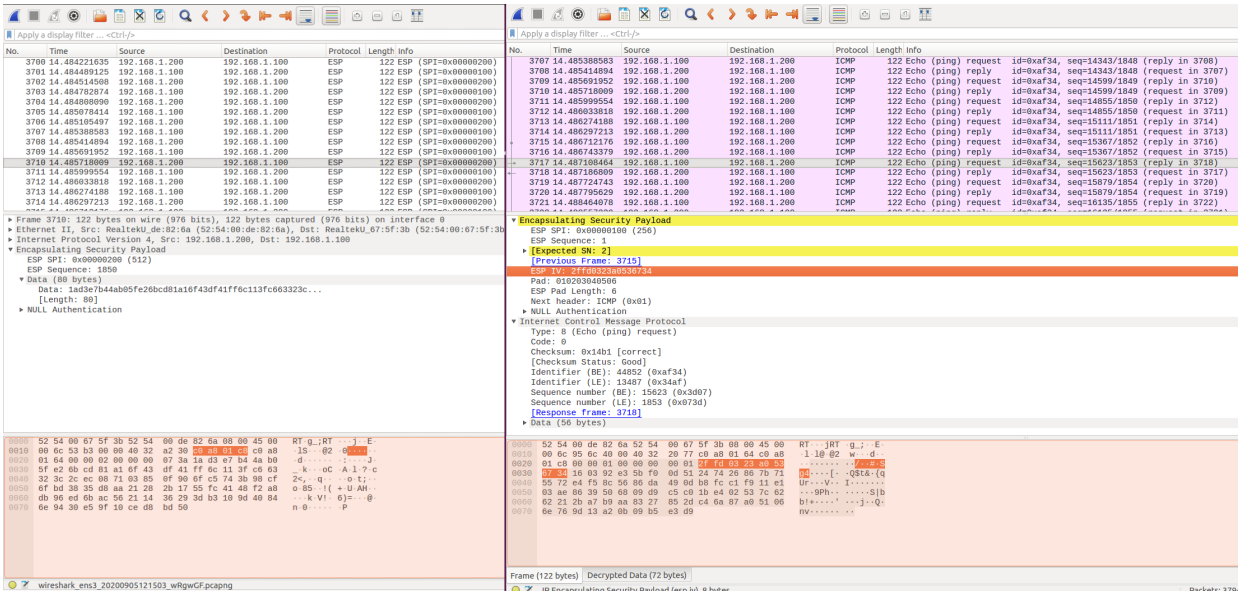

Figura 2: Izquierda: Sin desencriptar, Derecha: Desencriptado

Luego de confirmar el correcto funcionamiento y estudiar la estructura de los mensajes se procede a verificar el correcto funcionamiento del programa ping-sec. Para esto se hace uso de WireShark en ambos host para ver que esta funcionando bien en ambas puntas de la comunicación.

Host 2 - IPSec configurado

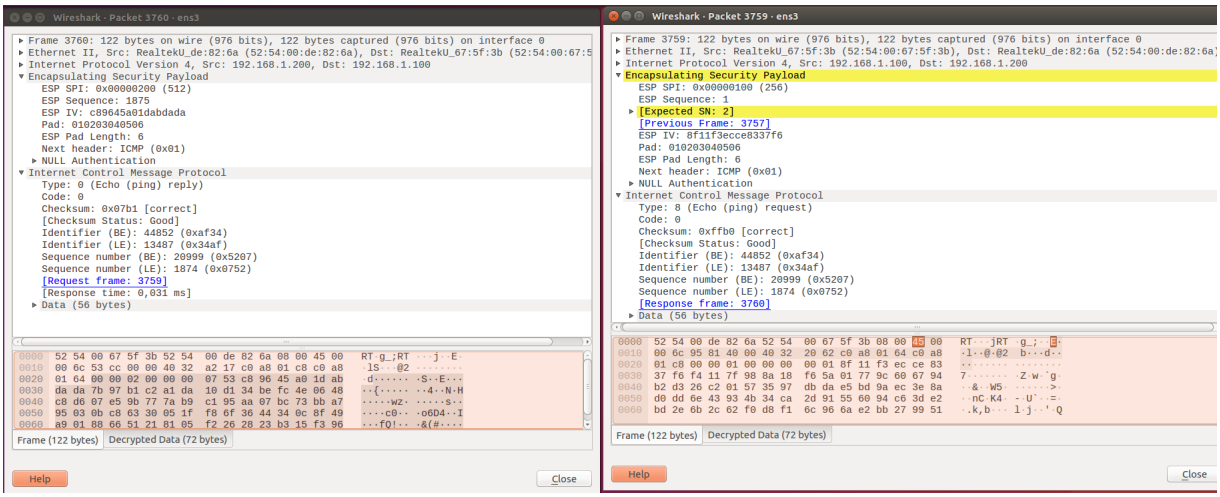

Figura 3: Izquierda: Echo Replay, Derecha: Echo Request

Host 1 - Utilizando programa Ping-Sec

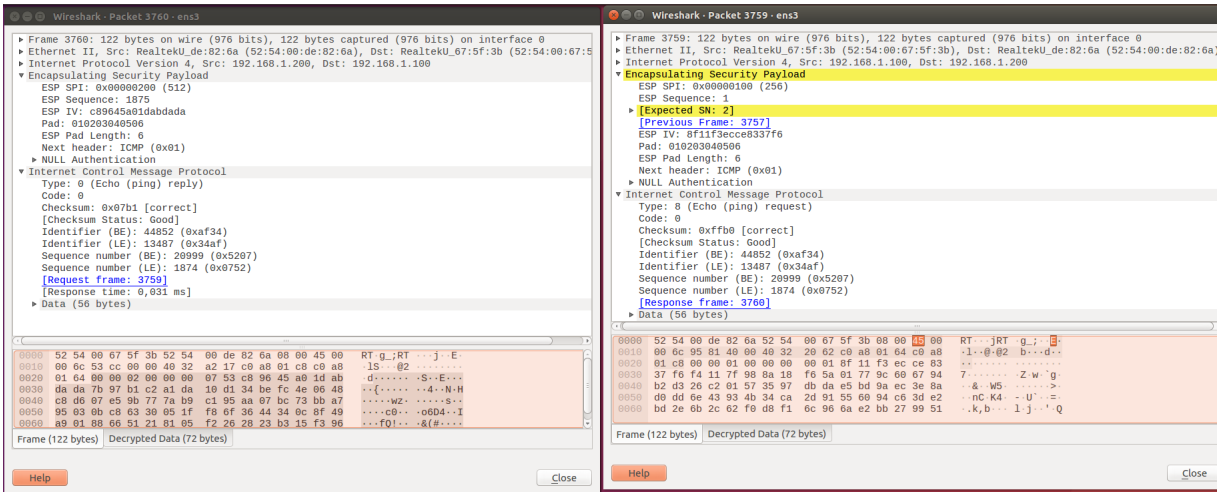

Figura 4: Izquierda: Echo Replay, Derecha: Echo Request

Como es posible observar, en ambos casos los mensajes vienen formados correctamente, dado que los mensajes son enviados con los cabezales de ESP necesarios y la data correspondiente a ICMP viene encriptada.
Por ultimo utilizando las estadísticas proporcionadas por el programa ping-sec cuando se cancela se aprecia que el ping realizado esta respondiendo correctamente y

todos los paquetes tienen respuesta.

3. Mejoras y trabajo a futuro

Las mejoras y trabajos a futuro son divididas en 2 secciones, la relacionadas directamente a la funciones de cifrado y las relacionadas al programa ping-sec en general. Empezado por el programa ping-sec, se encontraron una serie de detalles que podría ser mejorados, algunos de estos son:

- Utilizar el identificador SIGINT ofrecido por el SO para identificar que se quiere matar el proceso a través del teclado (Ctrl + c)
- Utilizar números de secuencia de SPI en los mensajes de forma incremental, tal como se recomienda en la documentación de IPSEC
- A la hora de mostrar información de los bytes recibidos de una IP determinada, en algunos casos la IP es mostrada de forma inversa, por lo cual habría que revisar el uso del funciones htonl y ntohl [8].

Alguna de las mejores que se podría realizar a la funciones criptográficas son:

- Generar un mapeo automático del algoritmo de cifrado recibido en las funciones a las utilizas por la librería Gcrypt
- Un mejor manejo de errores, tomar decisiones y distintos accionares en base a los tipos de errores.
- Mejorar la inserción o remoción del IV dentro del payload recibido, haciendo uso del manejo de punteros y no dezplazar toda la información.

4. Conclusión

En este laboratorio se vio a bajo nivel como podría ser una posible implementación de IPSec. En la cual es posible apreciar que si bien hay muchos protocolos y reglas definidas a seguir para el correcto funcionamiento, la complejidad añadida a la capa IP no es tan grande. Y trayendo como ventaja el hecho de agregar seguridad de forma invisible para la capas superiores.

Referencias

[1] [RFC4301] S. Kent, Network Working Group. Request for Comments. Security Architecture for the Internet Protocol. 2005.

[2] [RFC792] J. Postel, Network Working Group. Request for Comments. Internet Control Message Protocol. 1981.

[3] SetKey - Linux man page : `https://linux.die.net/man/8/setkey`. Visitada setiembre de 2020.

[4] Gnupg - The Libgcrypt Reference Manual: `https://gnupg.org/documentation/manuals/gcrypt.pdf`. Visitada setiembre de 2020.

[5] Triple DES (3DES): `http://www.crypto-it.net/eng/symmetric/3des.html`. Visitada setiembre de 2020.

[6] Modes of block chiphers: `http://www.crypto-it.net/eng/theory/modes-of-block-ciphers.html`. Visitada setiembre de 2020.

[7] WireShark: `http://www.wireshark.org`. Visitada setiembre de 2020.

[8] Ubuntu, Byte Order: `http://manpages.ubuntu.com/manpages/bionic/es/man3/byteorder.3.html`. Visitada setiembre de 2020.

[9] [RFC4303] S. Kent, Network Working Group. Request for Comments. IP Encapsulating Security Payload(ESP). 2005.