Universidad de Buenos Aires

Facultades de Ciencias Económicas, Ciencias Exactas y Naturales e Ingeniería

Carrera de Especialización en Seguridad Informática

Trabajo Final

Ataque de man in the middle para protocolo sip mediante análisis de caja negra

Autor: Ing. Ignacio Benedini

Tutor de Trabajo Final: Ing.Diego, Alonso

Año de presentación: 2013

Cohorte: 2011

Declaración Jurada de origen de los contenidos

Por medio de la presente, el autor manifiesta conocer y aceptar el Reglamento de Trabajos Finales vigente y se hace responsable que la totalidad de los contenidos del presente documento son originales y de su creación exclusiva, o bien pertenecen a terceros u otras fuentes, que han sido adecuadamente referenciados y cuya inclusión no infringe la legislación Nacional e Internacional de Propiedad Intelectual.

FIRMADO
Ignacio Benedini
DNI 28.863.191

Resumen

El protocolo SIP permite el establecimiento de sesiones entre dos o más equipos, el cual puede ser usado para la trasmisión de video, audio o cualquier otro formato multimedia.

En la actualidad, el protocolo ha cobrado importancia en los sistemas de telefonía IP, dado su versatilidad y capacidad en el manejo de sesiones. Este manejo permite el uso de diferentes funciones tales como transferencia de llamadas, conferencia, llamada en espera, etc.

Durante el transcurso del trabajo se analizaran los componentes básicos del protocolo, las funciones principales del mismo, y como se inicia y finaliza una sesión. Además se analizaran los tipos de ataques más conocidos en lo que respecta a seguridad informática.

Basándonos en la premisa que el ataque más peligros es aquel que pasa inadvertido a los ojos de la víctima, se encuentra que el ataque de mayor criticidad es aquel conocido como hombre en el medio. Se analizara en detalle dicho ataque y que es lo que debe realizar un atacante para cumplir dicho fin. Ya analizado los distintos métodos de ataque, se plantearán contramedida contra los mismos; buscando asegurar una conexión SIP segura.

El protocolo SIP es considerado un protocolo joven, el cual aun necesita ser revisado y modificado. En la actualidad existen varios documentos RFCs que proponen diversas mejoras en el mismo. Es por esto que con el fin de asegurar una conexión segura, es necesaria el uso de otros protocolos o sistema como TLS e IPsec que sirvan de apoyo al mismo.

INDICE

NÓMINA DE ABREVIATURAS (ACRÓNIMO EN INGLES) IV
INTRODUCCIÓN 1
CAPITULO 1 - ENTENDIENDO EL PROTOCOLO SIP 2
1.1 INTRODUCCIÓN 2
1.2 HISTORIA DEL PROTOCOLO 2
1.3 FUNCIÓN DEL PROTOCOLO 3
CAPITULO 2 - ESTRUCTURA DEL PROTOCOLO SIP 5
2.1 CARACTERÍSTICAS 5
2.2 IDENTIFICADOR UNIVERSAL DE RECURSOS - URI 5
2.3 COMPONENTES DE UNA RED TÍPICA 7
CAPITULO 3 - COMUNICACIÓN ENTRE DISPOSITIVOS 9
CAPITULO 4 – SEGURIDAD EN SIP 11
4.1 SESIÓN SEGURA 11
4.2 BASIC AUTHENTICATION 11
4.3 AUTHENTICATION DIGEST 11
4.4 S/MIME 14
4.5 TLS 15
4.6 IPSEC 16
CAPITULO 5 – VULNERABILIDADES Y ATAQUES 18
5.1 VECTOR DE ATAQUE 18
5.2 ATAQUES TÍPICOS 19
5.3 TÉCNICAS DE ATAQUE 22
CAPÍTULO 6 - CONTRAMEDIDAS PARA MITIGAR LAS AMENAZAS 29
CONCLUSIONES 32
BIBLIOGRAFÍA ESPECÍFICA 34
BIBLIOGRAFÍA GENERAL 37
INDICE ESPECÍFICO 39

Nómina de abreviaturas (*Acrónimo en ingles)*

3GPP	3rd Generation Partnership Project
ARP	Address Resolution Protocol
CBC	Cipher-Block Chaining
HTTP	Hypertext Transport Protocol
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IPS	Intrusion Predictive System
IPsec	Internet Protocol security
MD5	Message-Digest Algorithm 5
NAT	Network Address Translation
OSI	Open System Interconnection
PKI	Public Key Identificator
s/MIME	Secure/Multipurpose Internet Mail Extensions
SCIP	Simple Conference Invitation Protocol
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VLAN	Virtual LAN
VPN	Virtual Private Network

Introducción

En este documento se analizará la vulnerabilidad que presenta frente al protocolo *SIP* el ataque conocido como hombre en el medio.

El uso del protocolo *SIP* crece exponencialmente, en especial en el rubro de las telecomunicaciones para centrales telefónicas. Esto se debe a que el este esta enfocado en el manejo de sesiones, donde cada sesión implica una llamada telefónica, realizada en la central. Además el protocolo ofrece una amplia gama de opciones para el manejo de las mismas, lo que permite el desarrollo de funciones tales como conferencias, transferencias de llamadas, llamada en espera, etc.

Para entender las implicancias del ataque en análisis primero hay que entender que es el protocolo SIP y como funciona. Esto se analizará en el documento dentro de los capítulos uno, dos y tres

Entendiendo como funciona y los procesos que se realizan por ejemplo para el inicio y cierre de una sesión. Mientras que en el cuarto capítulo se analizaran los diferentes métodos existentes que permiten asegurar la confidencialidad, integridad y autenticidad de cada mensaje. Este proceso tiene como fin el de mitigar el riesgo de ser atacado.

En el quinto capítulo analizaremos que amenazas existen. También revisaremos los diferentes métodos conocidos que pueden lograr con éxito que el atacante pueda interceptar la llamada entre dos o más extensiones específicas.

Por último abordaremos las diferentes contramedidas que uno puede realizar para defenderse ante estos ataques.

Capitulo 1 - Entendiendo el Protocolo SIP

1.1 Introducción

El protocolo *SIP* creado por el *IETF*, fue creado con el fin de estandarizar la inicialización, modificación y finalización de sesiones de usuarios en donde intervienen elementos multimedia, tales como video, voz, mensajería instantánea, etc.

La sintaxis del protocolo *SIP* es similar a la de los protocolos *HTTP* y *SMTP*, usado para servicios de páginas Web y distribución de emails respectivamente. Todo sistema usado para las telecomunicaciones se encuentra referenciado dentro del modelo *OSI*.

El modelo posee 7 capas o categorías. La capa 1, capa física, hace referencia al medio físico usado para establecer las conexiones. La capa 2, capa de enlace de datos, refiere a la topología de red implementada, características de las tramas implementadas y el método empleado para el control de errores durante la transmisión.

Por otro lado, la capa 6, nivel de presentación, analiza la forma en que los datos recibidos serán analizados, por ultimo la capa 7, capa de aplicación, define los protocolos a ser usados por los diferentes servicios disponibles. El protocolo *SIP* se encuentra dentro de dicha capa.

1.2 Historia del protocolo

La estructura del protocolo *SIP* usado actualmente es el resultado de la mezcla de dos proyectos presentados a la *IETF* para el manejo de sesiones. En 1996 Mark Handley y Eve Schooler presentaron el proyecto conocido en la actualidad como *SIPv.1*. El mismo año el Dr. Henning Schulzrinne presentó el proyecto conocido como *SCIP* basado en el protocolo *HTTP*. Éste proponía el uso de una misma dirección para recibir tanto correos electrónicos como invitaciones a conferencias multimedia. No utilizaba al *SDP* para la descripción de los contenidos sino que creaba un mecanismo propio.

El *IETF* decidió combinar ambos en un único protocolo llamado Session Initiation Protocol o *SIP*v2. En 1999 se publico la propuesta de protocolo RFC 2543[1], y se creó el grupo de trabajo *SIP*. Esta misma propuesta fue revisada, y mejorada, creando así en 2002 la propuesta final para el protocolo *SIP*, el RFC 3261[2].

1.3 Función del protocolo

Las funciones principales del protocolo son:

- Manejo de la sesión
- Disponibilidad
- Movilidad
- Capacidad
- Establecimiento de la sesión

La función principal del protocolo *SIP* es la de dar inicio y cierre de cada sesión. Durante el inicio de la sesión, el usuario envía un paquete de notificación, por ejemplo el *INVITE*, a los demás participantes de la comunicación. Cada uno de estos integrantes tiene la facultad de aceptar o rechazar dicho pedido de inicio de la sesión.

La disponibilidad es una función que permite al usuario establecer y reportar su estado. Este estado comúnmente varía entre disponible, ocupado, no molestar, no disponible o desconectado. Esto tiene como fin facilitar la comunicación entre usuarios, por ejemplo un usuario puede saber si el otro se encuentra usando el dispositivo *SIP* o no. En ciertos productos SIP, el usuario puede modificar dicho estado de disponibilidad manualmente.

Otra función que ayuda a facilitar la comunicación entre usuarios, es la de la movilidad. Esta función fue creada en base al hecho de que la cantidad de dispositivos de telecomunicaciones por usuario se ha incrementado drásticamente. Hoy en día un profesional tiene por lo menos tres teléfonos: casa, oficina y celular; sin olvidar dispositivos de chats, sistemas de videoconferencia, etc.

La función de movilidad permite a un usuario usar un único número proveniente de su agente, el cual se registrara al dispositivo. De este modo, al momento que el usuario recibe un pedido de inicio de sesión, este pedido es dirigido al dispositivo registrado en ese momento.

Debido a la versatilidad del protocolo ante la disponibilidad de crear sesiones para diferentes medios (telefonía, video, chat, etc.), y a la función de movilidad; el protocolo posee la capacidad de establecer los recursos disponibles entre los dispositivos involucrados, y cuales van a ser usados durante el establecimiento de la sesión. Por ejemplo: dado que un usuario puede registrar su agente que puede o no soportar videoconferencia, es necesario tener un control de los recursos que posee el dispositivo en uso.

Por último, el protocolo *SIP* permite efectuar modificaciones sobre una sesión ya establecida. Esto posibilita el uso de incontables funciones tales como transferencia, conferencia o llamada en espera. Además de modificar o ampliar la cantidad de sistemas multimedia en uso durante el transcurso de la misma. Por ejemplo durante una comunicación telefónica, los usuarios pueden habilitar una sesión del chat entre ellos, para complementar dicha comunicación telefónica.

Capitulo 2 - Estructura del Protocolo *SIP*

2.1 Características

Un mensaje *SIP* puede ser tanto un pedido como una confirmación. El propósito de éste se encuentra definido por el contenido de su *URI* y cabezal. El mismo esta formado de texto plano, lo que facilita su lectura e interpretación tanto por un usuario como por el atacante. [Ver Figura.01]

Cada mensaje de mensaje *SIP* esta compuesto por tres partes:

1. *URI*, en principio representa la identidad del mensaje (función y tipo): pedido o confirmación. El RFC 3261[2] establece que todo dispositivo *SIP* debe ser capaz de interpretar al menos los siguientes mensajes de pedido *REGISTER*, *INVITE*, *ACK*, CANCEL, *BYE* y *OPTIONS*. En cuanto a los mensajes de respuesta estos se encuentran clasificados por números: 1XX, 2XX, 3XX, 4XX, 5XX, 6XX.

2. El cabezal, provee información de los miembros participantes de la comunicación, además del *caller-id*, la cantidad de veces que este mensaje fue retransmitido, largo del paquete, marcas usadas para la identificación de los usuarios, etc.

3. El cuerpo, describe las características a ser usadas en la sesión: protocolos, velocidad de transmisión, tipo de codificación, direcciones *IP* involucradas, etc.

2.2 Identificador universal de recursos - URI

El *URI* esta basado en el concepto utilizado por el protocolo *HTTP*. El *URI* es semejante al *URL* el cual provee una forma de encontrar una dirección *HTTP*. Este campo identifica al destinatario original de la solicitud. Pero debido a los diferentes procesos y enrutamientos posibles que se presentan en una comunicación, el destinatario podría aceptar o no solicitudes cuando el campo encabezado no es reconocido por el servidor.

Figura.01- Partes de un mensaje *SIP*

El RFC que establece las características del protocolo, recomienda que esta solicitud sea aceptada, aunque no se reconozca el *URI*. Dado que el URI se pudo ver modificado al provenir de un dominio distinto. En el caso que se decida no aceptar dicha solicitud, el destinatario debe enviar un mensaje con código 403 al emisor del mensaje.

La nomenclatura del *URI* divide al mismo en cinco partes: mensaje (hace referencia al tipo de mensaje que se envía), tipo de protocolo, la extensión involucrada, el dominio al que pertenece al extensión y el puerto en uso, y el método de autenticación en uso, [Ver Figura.02]

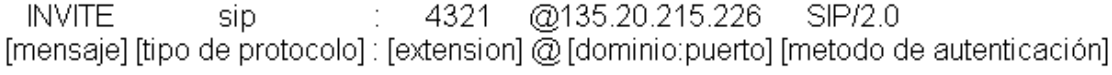

Figura.02- *URI*

Dentro del *URI* el campo que indica el tipo de protocolo a usar durante la comunicación misma, define si este es *TCP* o *TLS*. Por ejemplo, al usar *TCP* el *URI* a usar se expresa *INVITE SIP*: 147@10.1.100, mientras usando *TLS* se expresa *INVITE SIP*S: 147@10.1.100.

Cabe destacar que el protocolo a usar se define entre cada dispositivo que forma parte de una comunicación. Por ejemplo un cliente se conecta al servidor solicitando llamar a otro cliente, la conexión entre el primer cliente y el servidor puede ser *TLS* y la conexión entre el servidor y el segundo cliente puede ser *TCP* (ya que el segundo cliente no soporta el método de encripción usado por *TLS*)

2.3 Componentes de una red típica

Dentro de una red *SIP* se encuentran 5 componentes fundamentales. Estos son:

- El servidor o *UAS*
- El cliente o *UAC*
- El servidor para logueo
- El servidor para re-direccionamiento
- El servidor *Proxy*

SIP esta basado en una comunicación cliente-servidor. Un dispositivo en una comunicación puede asumir tanto el rol del cliente como el de un servidor, dependiendo de quien envía la solicitud de inicio de sesión.

El Servidor se define como aquel que recibe los pedidos provenientes de los clientes *SIP* (*UAC*) y responde a estos según el caso.

El *UAC* esta definido como todo equipo que inicia el pedido de sesión a un servidor (*UAS*), los mensajes típicos son: *INVITE*, *ACK*, *OPTIONS*, *BYE*, CANCEL y *REGISTER*.

Durante el pedido de inicio de sesión entre un *UAC* y un *UAS*, el *UAC* enviara toda la información requerida para establecer la sesión (protocolo a ser usado, dirección *IP*, puerto, etc.).

Esta información es dinámica, en ciertos casos esto puede dificultar la configuración de los *Firewalls*. El motivo principal de dicha variación es que

el usuario puede tener asignado y disponible más de un dispositivo a la vez, lo cual implica tipo de recursos y capacidades diferentes entre si.

Por su parte el servidor de logueo es aquel que permite la relación entre dirección IP y extensión del usuario. Esto se logra en el momento que el cliente manda un mensaje *REGISTER* al servidor, modificando así el registro relacionado a la extensión, adicionando un nuevo dispositivo.

El servidor para re-direccionamiento es aquel que permite que una de las principales funciones del protocolo *SIP* pueda ser realizada, la movilidad. El servidor permite al usuario cambiar su dirección IP o el tipo de dispositivo que está en uso, todo esto sin perder la conectividad con la red, manteniendo siempre la identidad *SIP* del usuario. Por ejemplo, si el usuario Bob se encuentra registrado a la red usando el teléfono de su casa; él puede loguearse a su celular, haciendo que todas las conexiones sean direccionadas a su celular, sin perder conexión o las aplicaciones asignadas a dicha identidad *SIP*.

El Servidor *Proxy* se lo clasifica como un mediador entre dos o más sub-redes, dado que es el encargado de re-direccionar los paquetes provenientes, ya sea de un *UAS* o *UAC* a su respectivo destino, dentro de otra sub-red. Dado un servicio de organización y seguridad entre organizaciones o subredes. Cabe aclarar que los servidores *Proxy* no son *Firewalls*.

Las sub-redes pueden ser tanto parte de una única organización, o de diferentes organizaciones. En ambos casos, el servidor de *Proxy* ofrece el re-direccionamiento de paquetes, y el bloqueo de paquetes provenientes de redes o clientes no autorizados.

El servidor *Proxy* también permite el mapeo de la red, por ejemplo, permite relacionar una identidad *SIP* de una red externa a una identidad de la red interna.

Capitulo 3 - Comunicación entre dispositivos

Como se menciono con anterioridad, el protocolo *SIP* tiene como propósito establecer o cerrar una sesión entre dos o más servidores/dispositivos. El RFC 3665[3] establece un proceso típico de establecimiento y cierre de sesión entre dos dispositivos [Ver Figura-03]

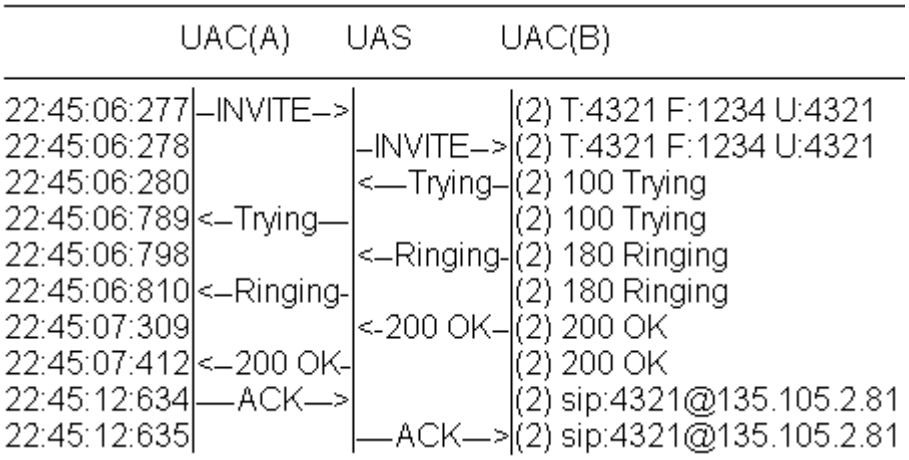

Figura.03 – Inicio de Sesión

El cliente A envía un mensaje de *INVITE* solicitando establecer una sesión con el cliente B. Si el servidor identifica al cliente emisor como un cliente registrado, le envía un mensaje *100-Trying* demostrando que el servidor recibió la solicitud, la cual esta siendo analizada.

Como siguiente paso, el servidor re-direcciona la solicitud previamente recibida hacia el cliente B, el cual acepta la misma, y envía los mensajes *180-Ringing*, y *200-OK* al servidor, indicando así que esta listo para iniciar la sesión. Estos mensajes serán re-direccionadas al cliente A, quien responderá con un mensaje *ACK*, iniciando así la sesión multimedia.

Como se puede observar el *UAS* es aquel que autentica al *UAC* que solicita el inicio de una sesión.

En cuanto al cierre de sesión, éste se realiza de forma simple. El cliente que desee cerrar una sesión tan solo envía un mensaje *BYE* al otro cliente. El destinatario del mensaje tan solo responderá con un mensaje *200-*

OK, indicando que recibió el mensaje con éxito y procederá a cerrar la sesión [Ver Figura.04]

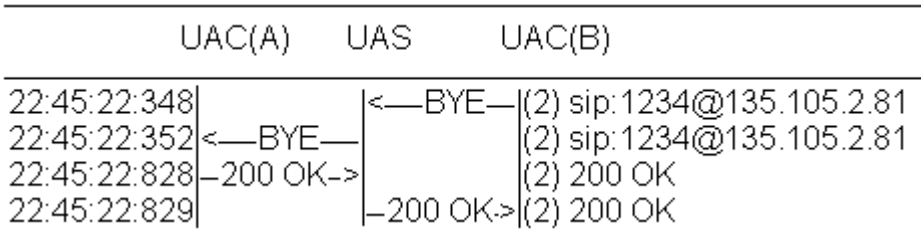

Figura.04- Cierre de Sesión

En cuanto al proceso de logueo y deslogueo de los clientes al servidor, el RFC 3261[2] sugiere el siguiente proceso: una interacción entre el servidor y el cliente en donde el servidor le solicita al cliente que se autentifique. Para ello el servidor envía un desafío al cliente. El cliente debe ser capaz de enviar la respuesta del mismo [Ver Figura.05]. La autentificación de este protocolo es unidireccional; esto implica que se solicita al cliente que se autentique pero no al servidor.

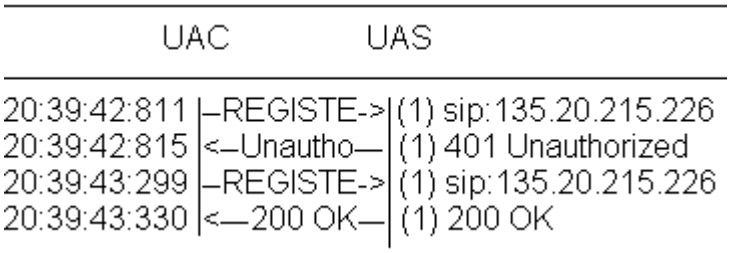

Figura-05- Logueo de un cliente

Con el fin de finalizar la sesión del servidor se envía nuevamente el mensaje *REGISTER*, cuyo valor de *Contact* es (*), conteniendo la respuesta al desafío previamente recibido.

En el capitulo siguiente se vera el proceso en detalle.

Capitulo 4 – Seguridad en *SIP*

4.1 Sesión Segura

Con el fin de establecer una sesión segura, se puede recurrir tanto a las propiedades y funcionalidades del protocolo *SIP* como a las herramientas de seguridad de la red misma.

En lo que respecta a la seguridad otorgada gracias a las funcionalidades y propiedades del protocolo (capa de aplicación del modelo *OSI*), las principales son *basic authentication*, *digest authentication* y *s/MIME*. Por otra parte las herramientas comúnmente usadas sobre conexiones *SIP* son *TLS* y *IPsec* (capa de transporte y red respectivamente)

4.2 *Basic Authentication*

Como se menciono en el punto 1.2, el RFC 2543[1] fue el primer documento donde se describió el protocolo SIP. En él se promovía el uso de *Basic Authentication*, el cual consiste en un sistema de pregunta-respuesta. El receptor del pedido de sesión solicita al emisor que se identifique, para ello envía una pregunta en texto plano. El emisor, debe responder dicha pregunta. Para lograr dicha autenticación ambas partes conocen la respuesta a la pregunta enviada.

Este sistema de autenticación es inseguro dado que tanto la pregunta como la respuesta pasan por la red en formato de texto plano, y sin sello de tiempo.

4.3 *Authentication digest*

Como ya se discutió en puntos anteriores, las funciones y propiedades del protocolo fueron revaluadas, creando así un nuevo RFC. Dicho RFC plantea una mejora en el sistema de autenticación de los mensajes. Este nuevo sistema es conocido como *Authentication digest*, también conocido como *SIP 2.0*.

El sistema esta basado en el sistema utilizado para paginas webs, el *HTTP 1.1*, el cual esta definido en el RFC 2617[4]. Ambos sistemas están enfocados en la autenticación de los participantes, pero no aseguran ni la integridad y confidencialidad de dichos mensaje recibido. *Además ambos usan un sistema de pregunta-respuesta en donde la pregunta enviada se encuentra oculta mediante un hash en MD5.*

Con el fin de mejorar el sistema de autenticación sugerido en el HTTP 1.1, se agregaron ciertas condiciones creando así el sistema SIP 2.0. Entre las condiciones agregadas esta:

- El formato del *URI* en uso dentro la pregunta enviada por el UAS es URI = SIP-URI o SIPS-URI; dependiendo si se usa el protocolo TLS o no.
- En el sistema HTTP 1.1 se solicita que se verifique que el URI del mensaje y el URI usado dentro de la pregunta sean iguales. En el RFC 3621[5], esta condición queda descartada, dado que el sistema permite el direccionamiento de mensajes entre diferentes dominios. Esto lleva a que el URI del mensaje vario, pero sin alterar el URI ubicando dentro de la pregunta, el cual pertenece al UAC de origen.
- En el caso de que el cuerpo del mensaje este vacio (valor nulo) se asume que el valor de hash MD5 es igual a "d41d8cd98f00b204e9800998ecf8427e"

El proceso da inicio en el momento que un UA, solicita identificación al emisor de un mensaje entrante. El UA receptor es conocido como UAS y el UA emisor es el UAC.

Al momento que un usuario envía un mensaje al *UAS*, éste responde a dicho mensaje, enviando un mensaje 4XX. En éste último se encuentra el campo *Proxy-Authenticate* o *WWW-Authenticate*. Esto varía según el mensaje 4XX específicamente enviado, y por ende según el mensaje de pedido inicialmente enviado por el *UAC*.

Por ejemplo si el *UAC* envía un mensaje *REGISTER* el *UAS* enviara un *401-Unauthorized*. En el caso de que el *UAC* envía un *INVITE* el *UAS* enviara un *407 Proxy Authentication Required*. El campo Proxy-Authenticate se encuentra en el mensaje 407 y el campo *Proxy-Authenticate* en el mensaje 401. La información presentada en ambos campos, es la misma. [Ver Figura-06]

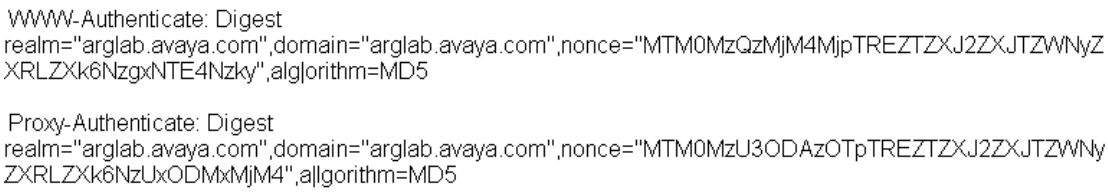

Figura.06-Autenticación

Una vez que el *UAC* recibe el mensaje 4XX, responde enviando el mismo mensaje que mando con anterioridad, solo que agrega el campo *Authorization* o *Proxy-Authorization*, ya sea que el mensaje recibido sea 401 o 407 respectivamente. [Ver Figura.07].

La información contenida en el campo de *Authentication digest* esta formada por:

- *realm*: Es un campo obligatorio y tiene como propósito identificar las credenciales de un mensaje. Comúnmente es el dominio al cual el servidor *Proxy* de origen pertenece.
- *domain*: especifica el domino del *URI*
- *nonce*: es una cadena de datos generados por cada desafío nuevo. Esta cadena es comúnmente el hash *MD5* de un paquete de datos que contiene por ejemplo un sello de tiempo (asegurando así que cada cadena tenga un tiempo de vida limitado) y la contraseña del dispositivo. Es en base al contenido de este cadena que el UAC obtiene la respuesta a la pregunta enviada.
- *Algorithm* Por el momento el protocolo *SIP* solo soporta MD5

Para una correcta respuesta, el *UAC* debe enviar no solo los valores previamente recibidos más la respuesta obtenida del campo *nonce*, sino también el *URI* y el nombre de la extensión.

En el caso de que la respuesta a la pregunta no sea la correcta, el *UAS* enviara un mensaje *401 Unauthorized,* reportando así que el *UAC* no esta autorizado a iniciar dicha sesión.

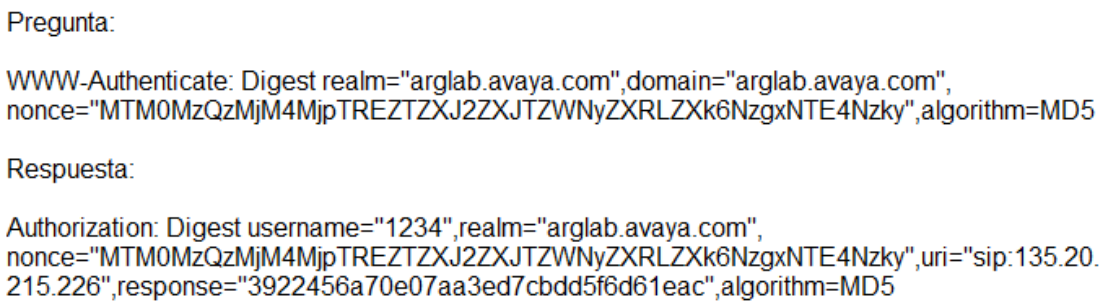

Figura 07- Respuesta al *Digest Authetication*

4.4 *S/MIME*

s/MIME esta definido por el RFC 2633[6], el cual especifica las condiciones de uso de una clave publica de cifrado o *PKI* entre ambos usuarios finales. Para esto el usuario utiliza un certificado, el cual forma parte del *URI* en el mensaje enviado. Usando dicho sistema el protocolo *SIP* envía mensajes donde solo el contenido del cuerpo se encuentra protegido por *s/MIME*.

El sistema permite no solo verificar la autenticidad, sino también la integridad y confidencialidad del mensaje, pero no en su totalidad dado que el cabezal del mismo queda sin encriptar, quedando así expuestas a ataques. Otra desventaja es la necesidad de un certificado por cada usuario y que dicho certificado pueda ser validado. En la actualidad, tal como lo indica el RFC 3261[2] en el punto 23.1, no existe un ente regulatorio a nivel mundial que expida estos certificados.

4.5 *TLS*

Con el fin de brindar protección a la conexión *SIP*, en la capa de transporte, se usan los paquetes *TLS* como medio de transmisión. Este mecanismo usa un certificado X.509. El RFC 5246[7] detalla el sistema de transmisión de *TLS*, mientras que el RFC 5630[8] contempla el uso del mismo en *SIP*, el cual permite proteger la autenticidad, integridad y confidencialidad de los mensaje enviados.

Esta condición se especifica en el *URI* del mensaje donde se modifica el mismo de *SIP*:bob@biloxy.com a *SIP*s:bob@biloxy.com. Esta protección se configura en cada punto o dispositivo que interactúa durante la transmisión. Cada uno de estos debe ser capaz de soportar dicho protocolo. En el caso de que alguno de los dispositivos involucrados no de soporte a dicho protocolo, se observara una degradación de la seguridad en ese mismo tramo de la red.

Al momento que un usuario inicia un pedido de sesión segura a un servidor, este inicia una conexión *TLS* por donde se enviarán los mensajes *SIP*. El servidor responderá a dicho pedido enviando su certificado público, el cual es validado por el usuario.

En el paso siguiente, se intercambian claves de sesión para la encripción y desencripción segura de los paquetes *TLS* a ser transmitidos. Una vez establecida la conexión, el servidor realiza el mismo proceso, con el siguiente dispositivo que va a interactuar en la comunicación.

El RFC 5630 en el punto 3.1.2 hace hincapié en las limitaciones que posee este sistema al momento de establecer una mutua autenticación. Dichas limitaciones se presentan en ambientes que involucran *NATs*, *Firewalls* o dispositivos similares. También la conexión se ve limitada dado el hecho de que se requiere certificados a ambos puntos. Esto se ha comprobado que es impráctico para ciertos ambientes, en particular para ambientes de alta concentración de tráfico.

4.6 *IPsec*

Es un conjunto de protocolos que interactúan, según el modelo *OSI* en la capa de red, con el fin de asegurar la autenticación, integridad y confidencialidad del flujo de paquetes transmitidos. Para lograr dicha protección el *IPsec* cifra cada uno de los paquetes. Dado que el sistema funciona en una capa inferior a la capa de transporte, el *IPsec* presenta mayor flexibilidad al momento de proteger la información.

El sistema fue originalmente creado en 1995 al publicarse los RFC 1825[9] y 1829[10]. Luego estos RFC fueron remplazados por los RFC 2401[11] y 2412[12]. En 2005 se publico una nueva generación de *IPsec*, RFC 4301[13] y 4309[14].

En referencia al uso de *IPsec* sobre una comunicación *SIP* se creo el RFC 3329[15]. En él se detalla un marco de trabajo entre *SIP* y *IPsec*. Al igual que el sistema TLS, esta conexión necesita ser habilitada en cada dispositivo que interactúa en la comunicación.

El RFC establece las condiciones básicas para la creación de una conexión *IPsec* entre dos dispositivos. El documento también sugiere el uso de otros métodos de seguridad implementados en conjunto con el sistema, por ejemplo *TLS* y *Authentication Digest*.

IPsec establece un canal de comunicación segura. Para ello se efectúa un intercambio de claves mediante el sistema *Diffie-Hellman*, creando así una clave de secreto compartido. Dicha clave se usara para cifrar la comunicación *IKE*. Una vez establecida dicha conexión, se negocia entre ambos dispositivos pertenecientes al canal, estableciendo una Asociación de Seguridad (SA). La negociación consiste en un mínimo de dos SAs unidireccionales.

El sistema de transmisión a usar son paquetes *UDP*. En la actualidad, el sistema *IPsec* posee dos versiones. La versión *IKEv.2* ofrece varias mejoras respecto a la versión anterior, entre ellas esta, requerir menor cantidad de paquetes a transmitir para realizar la negociación de la

Asociación de Seguridad, da soporte a la telefonía IP y brindar protección contra ataques de negación de servicio.

En el RFC se presentan tres nuevos campos a ser usados en el cabezal de un mensaje *SIP*: *security-client, security-server* y *security-verify*. Estos campos son usados a fin de poder establecer una conexión I*Psec* durante al inicio de una negociación para habilitar una nueva sesión.

A modo de explicar el uso de estos nuevos campos, se plantea el caso en el que un *UAC* solicite iniciar una nueva sesión. Para ello envía un mensaje de I*NVITE* al *UAS*, estando en el medio entre el *UAS* y el *UAC*, un *proxy*.

El mensaje de *INVITE* es enviado por el *UAC* al *proxy*. El servidor antes de direccionar el mensaje al *UAS*, envia un mensaje *402-Extension required* al *UAC*. Dentro del mensaje se encuentra la solicitud del proxy de iniciar una conexión *IPsec*.

En respuesta al mensaje recibido, el *UAC* envía un mensaje *ACK* e inicia una conexión *IPsec* con el *proxy*, para luego enviar nuevamente el mensaje *INVITE* original (incluyendo los parámetros de seguridad establecidos en la conexión *IPsec*).

Capitulo 5 – Vulnerabilidades y Ataques

5.1 Vector de Ataque

Sobre la base de que todo sistema, en especial aquel que involucra la interacción entre varios dispositivos, no esta exento de tener vulnerabilidades de seguridad, sin importar lo avanzado que este sea, se analizaran los diferentes tipos de ataques posibles sobre el protocolo *SIP*.

El atacante, sin importar cual es el motivo que lo lleva a tratar de atacar un sistema, ya sea por fama o plata, tiene tiempo, y recursos para analizar el funcionamiento del sistema a atacar, y así encontrar sus respectivas vulnerabilidades.

El concepto de vector de ataque hace referencia al método que utiliza el atacante, en su mayoría, estos se realizan de forma remota.

En referencia al protocolo *SIP*, este protocolo es vulnerable a paquetes *SIP* malformados/distorsionados; aunque tampoco esta exento de ser quebrado por ataque de fuerza bruta.

El procedimiento de un ataque típico a este protocolo costa de tres pasos básicos:

- Análisis de las capacidades y características de la victima
- Desarrollo de los paquetes *SIP* a ser usados para el ataque, en base a la información recopilada en el paso anterior.
- Testeo del método desarrollado atacando a la victima.

Con el fin de analizar las características de la victima es posible usar los mensajes *INVITE* o *OPTIONS*. La respuesta enviada por el *UAC* a estos mensajes ayuda a descifrar las características de la victima.

La principal diferencia en el uso de ambos mensajes es en la forma que el *UAC* responde a dicho pedido. El mensaje de *OPTIONS* es usado para verificar las características técnicas del dispositivo. Mientras que el mensaje de *INVITE* es usado para dar inicio a una sesión. En ambos casos,

el *UAC* responde a dicho mensaje con información tal como: los métodos que el dispositivo de destino soporta, códec, extensión, etc. Pero solo con el mensaje de *INVITE* el dispositivo de destino emite una alerta de inicio de sesión (por ejemplo, que el teléfono emita un sonido)

Usando la información provista por el *UAC* como respuesta al mensaje I*NVITE* o *OPTIONS*, más un análisis de paquetes realizado sobre la red, uno puede conseguir toda la información necesaria para poder crear los diferentes paquetes *SIP* a ser usados durante el ataque. La construcción de dichos mensajes debe contemplar si el atacante y la victima se encuentran en el mismo dominio o red.

En este documento nos basaremos en la idea de que tanto el atacante como la victima están en un único dominio. Este, a su vez, es un escenario crítico ya que implica que el atacante es un usuario registrado en la red, además que las medidas de seguridad y prevención son mínimas en comparación a las implementadas para evitar que atacantes entren y se registren al dominio en uso.

5.2 Ataques Típicos

Los ataques más conocidos se pueden según su propósito en:

Negación de servicio (*Denial of Service* o *DoS*): Como su nombre lo indica, este ataque tiene como propósito negarle a los servidores atacados la posibilidad de brindar servicio por un tiempo indefinido. Este ataque consiste en saturar el sistema de computado o la red incrementado drásticamente el flujo de información que el dispositivo recibe, logrando así saturarlo. Un ejemplo conocido de este ataque es el que reporto la empresa Nodo50, el día 2 de enero a las 12:00 PM UTC, donde ellos emitieron una comunicación anunciando que estaban siendo victimas de un ataque de negación de servicio. Reportaron que el tráfico entrante había alcanzado los 760 Mbps y más de 1 millón de paquetes UDP por segundo. [16]

Escuchas telefónicas o Escuchas secretamente (*Eavesdropping*): Se presenta comúnmente en las redes informáticas. Consiste en interceptar y

grabar mensajes o comunicaciones que cruzan dicha red. Este proceso no esta enfocado en un usuario en particular.

Dado que los usuarios comúnmente divulgan información crítica durante sus comunicaciones (ejemplo, número de tarjetas de crédito), el atacante busca poder recolectar dicha información.

Hombre en el Medio (*Man in the Middle*): Es un caso particular del ataque de “escucha telefónica”. Este ataque esta enfocado en interceptar la comunicación realizadas por un usuario específico. Esto implica que el atacante tiene acceso a toda la información entrante o saliente de la victima.

Existen varios ejemplos donde se muestra este ataque y la gravedad del mismo. Un ejemplo de esto es el famoso caso de Rupert Murdoch dueño del diario *News of the World*. [17]

Se acusa al diario de *News of the World y a Rupert Murdoch*, como dueño del diario, de intervenir el celular de una niña desaparecida, con el solo fin de recopilar información. En este caso, el atacante no solo recopilo información, sino altero información de la victima. El atacante borro los mensajes de voz de la victima, ya que el correo de mensajes de voz estaba lleno. Esto lo hizo con el simple hecho de que pudieran entrar nuevos mensajes en el correo de voz de la niña desaparecida.

La policía Británica, quien también tenía el teléfono intervenido, asumió que la niña había borrado dichos mensajes, modificando así el curso de su investigación. Fue tal la gravedad de las acciones realizadas por el diario y sus periodistas, que el diario *News of the World* tuvo que cerrar.

Robo de identidad (*Spoofing attack*): implica la usurpación de la identidad de un usuario de la red, por tiempo indeterminado. De esta forma el atacante, adquiere todos los permisos y privilegios asociados al usuario original.

Esto se logra tanto engañando al usuario original para que revele información clave o irrumpiendo en el dispositivo en uso o de forma personal (robo de cartera o billetera, al revisar los registros, facturas, etc.)

Llamadas fraudulentas (*Call Fraud*): Consiste en poder utilizar los recursos del dispositivo de comunicación sin intención de abonar por el uso de los mismos.

La técnica más popular de esta es conocida como *By Pass*. Esta consiste en el ingreso de tráfico internacional a un país de forma ilegal, evitando abonar las tasas entre operadores. Esto no solo implica una evasión sino también un ataque. El atacante debió adultera el origen genuino de la llamada, lo cual tiene implicaciones legales.

Ingeniería Social (*Social Engineer*): Es la técnica usada para obtener información confidencial de un usuario, a través de la manipulación de la victima, tal como el engaño y las mentiras.

En la práctica existen incontables ejemplos de este ataque. Por ejemplo uno es que el atacante se hace pasar por el administrador de un equipo que el usuario utiliza, y le solicita su contraseña para realizar ciertos cambios en su cuenta. Otro ejemplo es el envió de mensajes, comúnmente vía emails, a un gran grupo de personas informando que ha recibido un premio monetario, pero para recibirlo debe cargar los datos de su tarjeta de crédito en una página web. Este último ejemplo se lo conoce como *phising*.

Como se puede observar, los diferentes ataques presentados se encuentran interrelacionados. Por ejemplo, uno puede usar las técnicas de ingeniería social con el fin de realizar un fraude telefónico.

Durante el transcurso de este punto hemos repasado los ataques más conocidos, y el fin de cada uno de ellos. Basándonos en la premisa que el ataque más peligros es aquel que pasa inadvertido a los ojos de la victima, al momento del hecho, los ataque de escuchas telefónicas, llamadas fraudulentas, hombre en el medio e ingeniería social, son las más peligrosas. Efectuando una comparación entre estos cuatro tipos de ataque, se

encuentra una diferencia entre ellos en lo que respecta a la efectividad de los mismos.

El ataque de escucha telefónica posee un nivel bajo de efectividad, dado que el atacante debe poder localizar aquella comunicación en curso, en la cual se estén revelando datos útiles.

En el caso de las llamadas fraudulentas, el proveedor puede filtrar los números usados comúnmente por el atacante. Además dicho ataque queda expuesto al momento de comparar las llamadas realizadas por el sistema y las llamadas facturadas.

La ingeniera social tiene un alto grado de efectividad, especialmente si se aplica sobre personas que no están familiarizadas con dicha amenaza. Por tal motivo, es posible reducir drásticamente su efectividad al capacitar a las personas y a fomentar la toma de conciencia ante dicha amenaza.

Por último el ataque de hombre en el medio. Este posee una alta tasa de efectividad, dado que es difícil de rastrear o monitorear, ya que figura en el sistema como una llamada más. Comúnmente son difíciles de localizar a menos que el atacante cometa un error que termine por exponerse, por ejemplo revelar el acceso a información confidencial. Además, dado que esta enfocado a usuarios específicos, el atacante empieza a recopilar información útil desde el comienzo del ataque.

5.3 Técnicas de Ataque

- **Uso de paquete *REGISTER***

El RFC 3327[18] esta enfocado en describir las características y funcionalidades del mensaje *SIP* conocido como *REGISTER.* Este mensaje tiene como función la de asociar una dirección de contacto temporal con una Dirección de registro.

Se observa del RFC varias particularidades del mensaje que permite una interacción entre cliente-servidor (UAC-UAS). Dichas particularidades

son utilizadas por los atacantes para poder asumir la identidad de la victima en el servidor.

El ataque consiste en borrar todo relación entre dirección de contacto temporal y la dirección de registro. Luego que el atacante borra dichas asociaciones entre el servidor y la victima, crea un nuevo en el cual asocia su propia dirección de contacto con la dirección de registro de la victima. Asumiendo así la identidad de dicha victima.

Para lograr esto el atacante primero envía un mensaje *REGISTER* al servidor, en el cual su valor de campo *EXPIRE* esta en cero y no hay valor para el campo *CONTACT* [ver Figura.08]. De esta forma el atacante solicita al servidor el listado de las asociaciones existentes con respecto a la dirección de registro de la victima. Para desvincular dichas asociaciones el atacante envia otro mensaje de *REGISTER*. Dicho mensaje contiene el campo *EXPIRE* en cero y al campo *CONTACT* se le asigna el valor (*). [Ver Figura.08 y Figura.09]

En este punto no existe asociación alguna entre alguna dirección de contacto temporal con la dirección de registro de la victima. Con el fin de asumir la identidad de la victima, el atacante envía un tercer mensaje *REGISTER* asumiendo la identidad de la victima

```
REGISTER sip:135.20.215.226 SIP/2.0
Via: SIP/2.0/UDP 135.105.9.63:5060;branch=z9hG4bK-d8754z-d12b64fd060ce970-1---d8754z-;rport
Max-Forwards: 70
Contact:     ;rinstance=9064ae93d627d197
To: <sip:1234.20.215.226>
From: <sip:1234.20.215.226>;tag=d02f5ac6
Call-ID: NjM2ZmFkYjE1MzY0M2IyOGM2YWFhMzE3ZmJhOTAwNDg
CSeq: 1 REGISTER
Expires: 0
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
Content-Length: 0
```

Figura.08- Listado de enlaces

```
REGISTER sip:135.20.215.226 SIP/2.0
Via: SIP/2.0/UDP 135.105.9.63:5060;branch=z9hG4bK-d8754z-d12b64fd060ce970-1---d8754z-;rport
Max-Forwards: 70
Contact:*;rinstance=9064ae93d627d197
To: <sip:1234.20.215.226>
From: <sip:1234.20.215.226>;tag=d02f5ac6
Call-ID: NjM2ZmFkYjE1MzY0M2IyOGM2YWFhMzE3ZmJhOTAwNDg
Cseq: 1 REGISTER
Expires: 0
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
Content-Length: 0
```

Figura.09- Eliminación de enlaces.

.Al momento que otro usuario del sistema quiera comunicarse con el usuario recientemente atacado, el servidor direccionara los mensajes a la dirección de contacto que le pertenece al atacante; y es el quien direccionara nuevamente dichos mensajes hacia la victima. La victima responderá a dichos mensajes, enviándolos a su emisor, el atacante.

En conclusión, todo mensaje enviado desde el servidor a la victima es recibido por el atacante. A su vez, la victima responderá a todo paquete entrante, enviándolo a su emisor, el atacante. El atacante se encuentra en el medio de la comunicación entre la victima y el servidor.

- **Uso de paquetes 3XX**

Los mensajes 3XX son usados para el re-direccionamiento de una comunicación. Este ataque se basa en el uso de los paquetes 301 y 302. El paquete 301 indica al destinatario del mensaje que el cliente se movió permanentemente a otra locación o dirección *IP*; mientras que el mensaje 302 indica que el cambio de locación es temporal [Ver Figura.10]

```
SIP/2.0 302 Moved Temporarily
From: <sip:1234@135.20.215.226>;tag=d02f5ac6
To: <sip:1234@135.20.215.226>;tag=0AFAF68CB82C921E5EE97C8173DFC9631343
43238230706
Call-ID: NjM2ZmFkYjE1MzY0M2IyOGM2YWFhMzE3ZmJhOTAwNDg.
CSeq: 1 INVITE
Via: SIP/2.0/UDP
135.105.9.80:5060;received=135.105.9.80;branch=z9hG4bK-d8754z-2ab61423b4925579-1---d8754z-;rport=5060

Contact: <sip:1234@135.105.9.80:5060;rinstance=8e998aef1829def5>
Content-Length: 0
Server: Avaya SIP Enablement Services
Organization: arglab.avaya.com
```

Figura.10- Mensaje 302 *Moved Temporaly*

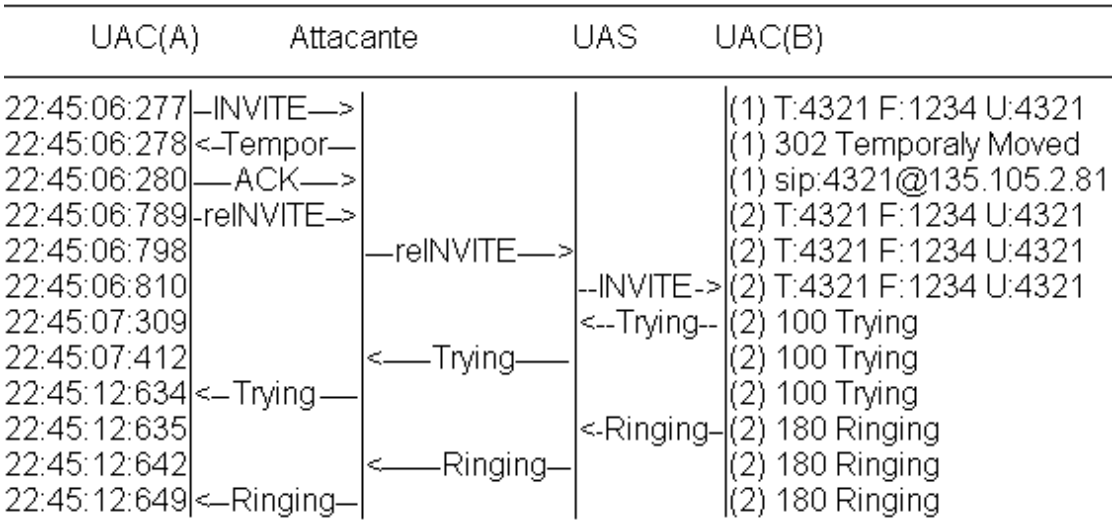

Figura-11-Uso del Mensaje 302

La victima manda un mensaje de pedido, por ejemplo *INVITE* el cual es interceptado por el atacante. Este envía un mensaje 301, donde el atacante asume la identidad de destino y le informa a la victima que la extensión de destino se movió temporalmente a una nueva dirección *IP*, siendo esta la dirección IP del atacante. De esta forma el atacante asume por la identidad de la victima, ya sea esta un cliente o servidor. Forzando así que toda comunicación pase por el. [Ver Figura.11]

Dicho ataque se puede realizar tanto usando mensajes 301 0 302.La diferencia principal en emitir mensaje 301 o 302 como medio de ataque, es el tiempo que uno asume la identidad de la victima (temporal o permanente)

- **Uso de paquete REFER**

El RFC 3515[19] esta enfocado en determinar las características y propiedades del mensaje *REFER*. Este mensaje es usado para establecer una suscripción temporal entre los *UA* presentes. Para ello el mensaje usa los parámetros *To* (que indica el origen) y *Refer-To* (el cual indica el destino de la acción) Con este mensaje se amplian los servicios posibles, agregando funciones como transferencia de llamadas, conferencias, etc.

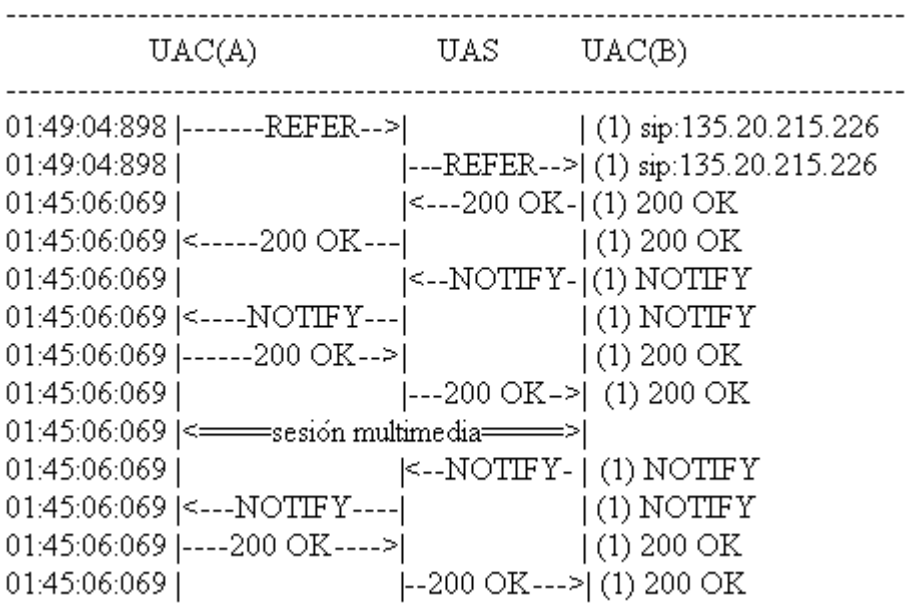

Figura.12- Uso típico del mensaje *REFER*

Este mensaje de suscripción obliga una respuesta por parte del destino enviando un mensaje *NOTIFY*. [Ver Figura-12] La suscripción agrega a una sesión ya establecida, a otro integrante más. Los participantes de la sesión no reciben notificación de esto.

De esta forma el atacante puede usar este tipo de mensaje para auto-invitarse o referir la llamada a un *UA* controlado por el atacante.

- **Envenenamiento de los paquetes *ARP***

Todo dispositivo de red, incluyendo aquellos que usan el protocolo *SIP*, contiene una tabla temporal de *ARP*. Esta contiene la relación entre

dirección *MAC* y dirección *IP* de los dispositivos que este estuvo interactuando recientemente. Con esto se reduce significativamente la necesidad de realizar una búsqueda usando paquetes *ARP Reques.* Este paquete es pedido que manda el dispositivo para buscar que dirección IP tiene asignada una dirección *MAC* específica.

.Ejemplo típico de una comunicación *ARP*: una persona llega a la oficina, prende su computadora, y necesita imprimir un documento. La computadora para enviar la orden de impresión a la impresora, primero debe ubicar dicha impresora. Para ello realiza un búsqueda con *ARP request* buscando que dirección *IP* que tiene el dispositivo con la dirección *MAC* buscada. Solo la impresora responderá, enviando un *ARP Reply* a la computadora.

El atacante, quien se encuentra analizando el trafico en la red, es capaz de enviar numerosos mensajes *ARP replay* al momento que observa que un dispositivo envió un *ARP request*. Este *ARP replay* forzara a la viticma a asociar la dirección MAC buscada por la computadora con la dirección *IP* del atacante. [Ver Figura.13]

De esta forma el atacante puede hacerse pasar por el servidor *SIP* (*UAS*) ante un cliente *SIP* (*UAC*). Todo paquete que el atacante reciba, lo re-enviara al servidor *SIP* real, previamente modificado.

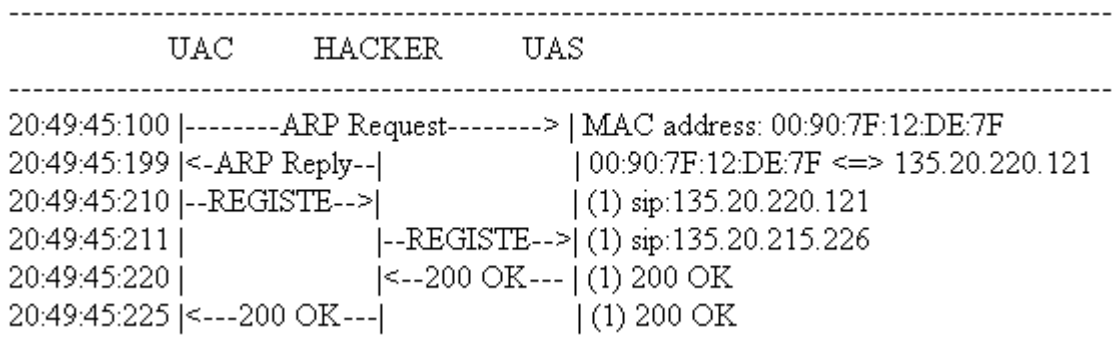

Figura.13- Envenenamiento de los paquetes ARP

Esta técnica permite al atacante interceptar todas las comunicaciones entrantes y salientes a la victima. De esta forma el atacante puede recolectar

toda la información que necesita. Por ejemplo el atacante puede extraer las peticiones de registro de la victima, y de allí uno puede realizar un ataque de diccionario o de fuerza bruta contra los hashes obtenidos de forma offline. Logrando obtener así la clave real del *UA*.

- **Ataque al sistema de autenticación mediante mensaje BYE**

El atacante envía al *UAC* un mensaje de *INVITE*. El *UAC* reaccionara a este iniciando el proceso para establecer una nueva sesión, y emitirá una alerta, por ejemplo una alerta sonara o visual, para el usuario de dicho *UAC* este informado. [Ver Figura.14]

El usuario atenderá dicha comunicación entrante. Como el atacante no envia ningún paquete multimedia (audio, video, etc). La victima procederá a cancelar la comunicación.

Al momento de la cancelación el *UAC* enviara un mensaje *BYE* al atacante. El atacante responderá al *UAC* enviando un mensaje *407 Proxy authentication* con un valor de *Authentication digest* conocido por el. Como resultado, la victima resolverá el *Authentication digest* exponiendo el hash de su clave, la cual se puede encontrar con un ataque por fuerza bruta.

ATACANTE	UAC	
22:45:20:095	–INVITE–>	(2) T:4321 F:1234 U:4321
22:45:20:096	<–Trying—	(2) 100 Trying
22:45:20:114	<-Ringing-	(2) 180 Ringing
22:45:29:330	<–200 OK-	(2) 200 OK
22:45:20:505	—ACK—>	(2) sip:4321@135.20.215.226
22:45:20:506	<:ses.mult>	
22:45:21:804	—BYE—>	(2) sip:1234@135.105.2.81
22:45:21:805	<-Proxy A-	(2) 407 Proxy Authentication Required
22:45:22:348	—BYE—>	(2) sip:1234@135.105.2.81

Figura.14- Ataque por mensaje *BYE*

Capítulo 6 - Contramedidas para mitigar las amenazas

En el capítulo anterior se analizaron los diferentes ataques posibles que un atacante puede realizar para conseguir con éxito interceptar la información que se transmite entre dos usuarios específicos. Con el fin de contrarrestar o mitigar dichas amenazas existen ciertas técnicas, dispositivos y consejos que se pueden aplicar.

Con el fin de contrarrestar los ataques en la capa de aplicación se recomienda que todos mensajes *REGISTER* y *OPTIONS* soliciten autenticación. Con esto minimizamos la posibilidad de que el atacante obtenga información del sistema.

También se recomienda configurar el servidor para que solicite autenticación por cada mensaje de *CANCEL* o *BYE*, recibido. Esto disminuirá significativamente la amenaza de un ataque de negación de servicio.

Además existen procesos como el que presenta el RFC 4488[20], que sugiere agregar un campo más al mensaje *REFER*, para que el receptor de dicho mensaje no tenga que mandar un mensaje *NOTIFY* como respuesta. Esto reduce la posibilidad de que el mensaje *REFER* sea usado para un ataque de negación de servicio, pero a su vez facilita el uso del mismo mensaje para un ataque de hombre en el medio. Por ello, dado que el mensaje REFER se usa solo para funciones particulares, se recomienda negar dicho mensaje si las funciones en uso no la requieren.

En la capa de transporte no es recomendable el uso de paquetes *UDP* debido a que es altamente vulnerable. En cambio se recomienda usar paquetes *TCP*. Para mejorar aun más la seguridad, es recomendado usar el protocolo *TLS*. En particular entre las conexiones entre los servidores, buscando asegurar la autenticidad, integridad y confidencialidad de los mensajes SIP transmitidos.

El sistema de transporte *TLS*, basado en el *SSL*, posee en la actualidad tres versiones: v1.0, v1.1 y v1.2 (RFC 2246[21], 4346[22] y 5246[7]

respectivamente). En la versión v1.0 se encontró un problema crítico de vulnerabilidad en las políticas del sistema implementadas sobre el *CBC*. Con el fin de evitar dicha vulnerabilidad se recomienda usar solamente RC4 para establecer la conexión o actualizar el sistema. La versión v1.1 posee protección contra dicho ataque.

El sistema *TLS*v1.2 presenta mejoras con respecto a la versión anterior. Principalmente se remplazo *MD5* por *SHA-256*, debido a las vulnerabilidades encontradas en el.

En la capa de red se recomienda bloquear los puertos no en uso por los dispositivos. Además, de ser posible se recomienda la implementación de conexión *IPsec* o *VPNs* entre los dispositivos. Al menos sobre conexiones críticas, por ejemplo la conexión entre dos servidores.

Se recomienda el uso de *VLANs* para segmentar la red, limitando el daño causado ante un ataque.

Por ultimo se recomienda el uso de *IDS* o *IPS* con el fin de analizar, prevenir y controlar la comunicación *SIP* entrante y saliente del sistema. Este sistema no solo deberá analizar que cada paquete cumpla con las especificaciones estipuladas en cada RFC, sino también analizar que la sucesión de paquetes enviados no formen parte del accionar de un atacante.

Ciertas medidas a considerar para la configuración del dispositivo:

- Este sistema deberá emitir alertas ante la primera falla de autenticación.
- Almacenar todos los mensajes *REGISTER* emitidos diariamente que no cumplan con un patrón prefijado.
- Analizar que cada mensaje sea gramaticalmente correcto: por ejemplo, según lo estipula el RFC 3261[2], un call-id valido es “a84b4c7”, mientras que un call-id invalido puede ser a84b4c7@dominio.com o “a84b4c7, a84b4c7, a84b4c7”.

- Analizar que cada mensaje tenga el largo esperado. Por ejemplo el valor de un *URI* valido es *INVITE* Bob@dominio.com *SIP*/2.0, y de invalido puede ser *INVITE* Bob@dominio.com *SIP*/2222222222224.0. En este punto el RFC 3261[2] no es exacto en su declaración
- Semántica invalida
- Falta de información en el cabezal. Todo cabezal de mensaje *SIP* debe contener la información *From*, *To*, *Via*, *Call-id* y *Cseq*. Si estos no se encuentran presentes, el mensaje mismo debe ser descartado.
- Que el intercambio de mensajes entre dos dispositivos previamente autenticados, sea el esperado. Por ejemplo, una *UAS* que recibe un *INVITE*, inmediatamente después no debería recibir un mensaje *REGISTER* del mismo origen.

Conclusiones

Durante el transcurso del informe se analizó el protocolo *SIP*, su evolución, sus propiedades y funcionamiento, además de sus vulnerabilidades. En especial, se analizaron los riesgos que presenta ante ciertos tipos de ataque conocidos como hombre en el medio.

El protocolo *SIP* es considerado como un protocolo joven. El mismo pasó por una revisión completa con la creación del RFC 3621, publicado en el año 2002. Debido a esto su composición y las propiedades del mismo aún no contemplan todas las medidas de seguridad requeridas para asegurar la confidencialidad, integridad y autenticidad de cada uno de los mensajes SIP transmitidos. Estas tres características son los puntos fundamentales que cualquier sistema o protocolo que se considere seguro deben poder proteger.

En la actualidad el protocolo de sesión, el cual funciona sobre la capa de aplicación del modelo *OSI*, ofrece algunos sistemas de seguridad. Entre ellos esta el *s/MIME* y el *Authentication Digest*.

El *s/MIME* ofrece un sistema para verificar la autenticidad del mensaje, como la confidencialidad e integridad del mismo. Pero su efectividad es limitada dado que actúa solo sobre el cuerpo del mensaje. A lo largo del informe, se ha comprobado que el objetivo principal de los ataques es el cabezal del mensaje, y no el cuerpo del mismo.

El *Authentication Digest* ofrece un sistema de autenticación. Este sistema esta basado en un sistema de hash *MD5*, el cual ya se le conocen debilidades.

Con el fin de mejorar el mismo se sugiere cambiar dicha función de hash por otra, por ejemplo *SHA-256*. La función *SHA-256* es considerada en la actualidad una de las funciones hash más seguras.

Con el fin de asegurar la integridad de los mensajes, sería posible agregar la cantidad de bytes que ocupa el mensaje *SIP* en si. Dicho valor

deberá comprender el contenido del *URI*, el cabezal y el cuerpo del mensaje. Dicho valor formaría parte del valor de nonce que se encuentra dentro del campo usado por el Authentication Digest.

En conclusión, el protocolo SIP presenta un alto nivel de riesgo, quedando vulnerable ante diferentes tipos de ataques. Es por esto que se han creado RFCs que presentan opciones para mejorar dicha situación. El RFC 3621 plantea una forma segura de usar el protocolo TLS como medio de transmisión. El RFC 3329 plantea la interacción segura entre SIP y el sistema IPsec.

Por tal motivo se concluye que el protocolo requiere seguir siendo analizado y mejorado. Además de ser actualizado ante nuevas amenazas.

Bibliografía Específica

[16] Comunicado emitido por Nodo50

http://info.nodo50.org/Se-repite-el-ataque-de-denegacion.html

(Consultada el 13/8/2012)

[17] Diario La Nación: Caso Murdoch

http://www.lanacion.com.ar/1387619-murdoch

(Consultada el 14/8/2012)

[9] RFC 1825

http://www.ietf.org/rfc/rfc1825.txt (Consultada el 13/06/2012)

[10] RFC 1829

http://www.ietf.org/rfc/rfc1829.txt (Consultada el 13/06/2012)

[21] RFC 2246

http://tools.ietf.org/html/rfc2246.txt (Consultada el 23/8/2012)

[11] RFC 2401

http://www.ietf.org/rfc/rfc2401.txt (Consultada el 13/06/2012)

[12] RFC 2412

http://www.ietf.org/rfc/rfc2412.txt (Consultada el 13/06/2012)

[1] RFC 2543

http:/www.ietf.org/rfc/rfc2543.txt (Consultada el 12/06/2012)

[4] RFC 2617

http://www.ietf.org/rfc/rfc2617.txt (Consultada el 13/06/2012)

[6] RFC 2633

http://www.ietf.org/rfc/rfc2633.txt (Consultada el 12/06/2012)

[15] RFC 3327

http://www.ietf.org/rfc/rfc3327.txt (Consultada el 13/06/2012)

[18] RFC 3329

http://tools.ietf.org/html/rfc3329.txt ((Consultada el 12/8/2012)

[2] RFC 3261

http:/www.ietf.org/rfc/rfc3261.txt (Consultada el 12/06/2012)

[19] RFC 3515

http://tools.ietf.org/html/rfc3515.txt (Consultada el 12/8/2012)

[5] RFC 3621

http://www.ietf.org/rfc/rfc3621.txt (Consultada el 10/06/2012)

[3] RFC 3665

http://www.ietf.org/rfc/rfc3665.txt (Consultada el 10/06/2012)

[13] RFC 4301

http://www.ietf.org/rfc/rfc4301.txt (Consultada el 13/06/2012)

[14] RFC 4309

http://www.ietf.org/rfc/rfc4309.txt (Consultada el 13/06/2012)

[22] RFC 4346

http://tools.ietf.org/html/rfc4346.txt (Consultada el 23/8/2012)

[20] RFC 4488

http://tools.ietf.org/html/rfc4488.txt (Consultada el 15/8/2012)

[7] RFC 5246

http://www.ietf.org/rfc/rfc5246.txt (Consultada el 10/06/2012)

[8] RFC 5630

http://www.ietf.org/rfc/rfc5630.txt (Consultada el 13/06/2012)

Bibliografía General

- Dimitris Geneiatakis, Georgios Kambourakis, Tasos Dagiuklas, Costas Lambrinoudakis y Stefanos Gritzalis, SIP Security Mechanisms: A state-of-the-art review, Departamenteo de informatica y telecomunicaciones. Universidad de Aegean, Karlovassi, Samos, Grecia.
- Humberto Abdelnur, Tigran Avanesov, Michael Rusinowitch y Radu State,Abusing SIP authentication, INRIA, Nancy - Grand Est Campus Scientifique - BP 239 – 54506 Vandoeuvre-l`es-Nancy Cedex, France.
- Radvision, Understanding SIP Servers, RADVISION.Ltd, version 1.0, (2002)
- Marius Herculea, Tudor Mihai Blaga y Virgil Dobrota, Evaluation of Security and Countermeasures for a SIP-based VoIP Architecture, Universidad Técnica de Cluj-Napoca, Facultad de Electrónica, Telecomunicaciones e informática, 400027, Cluj-Napoca, Rumania
- Peter Burkholder, SSL Man-in-the-Middle Attacks, SANS Institute, (2002)

Páginas web:

- By Pass ilegal en redes

http://portal.oas.org/LinkClick.aspx?fileticket=LnSL7emTZ7c%3D&tabid=1851 (Consultada el 20 de Noviembre del 2012)

- Digest authentication

http://nioveav.blogspot.com.ar/2008/02/autenticacin-en-sip-para-linux.html

(Consultada el 13 de Agosto del 2012)

- IPsec-3GPP

http://www.arib.or.jp/IMT-2000/V310Sep02/T63/Rel5/33/A33203-520.pdf

(Consultada el 13 de Junio del 2012)

- VOIPSA

http://www.voipsa.org/Resources/tools.php

(Consultada el 29 de Agosto del 2012)

- Wikipedia: IETF

http://es.wikipedia.org/wiki/IETF

(Consultada el 29 de Agosto del 2012)

- Wikipedia: SIP

http://es.wikipedia.org/wiki/Session_Initiation_Protocol

(Consultada el 29 de Agosto del 2012)

- Wikipedia: URI

http://es.wikipedia.org/wiki/Uniform_Resource_Identifier

(Consultada el 29 de Agosto del 2012)

Indice Específico

Figura.01……Partes de un mensaje SIP……...6
Figura.02……URI……..6
Figura.03……Inicio de Sesión ……..…….…9
Figura.04……Cierre de Sesión…….10
Figura-05……Logueo de un cliente……..……10
Figura.06……Autenticación……...13
Figura 07……Respuesta al *Digest Authetication*……...14
Figura.08……Listado de enlaces……..23
Figura.09……Eliminación de enlaces……....24
Figura.10……Mensaje 302 *Moved Temporaly*……..25
Figura-11……Uso del Mensaje 302…….25
Figura.12……Uso típico del mensaje *REFER*……26
Figura.13……Envenenamiento de los paquetes *ARP*……..27
Figura.14……Ataque por mensaje *BYE*……..28