ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

INFORME DE MATERIA DE GRADUACIÓN

“Análisis y Estudio de herramientas para prevenir y solucionar amenazas de Seguridad en Sistemas de Voz Sobre IP”

Previa a la obtención del Título de:

- **INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**
- **INGENIERO EN TELEMÁTICA**

Presentada por:

Jessica Vanessa Gaibor Ortega

Pablo Raúl Caicedo Arellano

GUAYAQUIL - ECUADOR

AÑO: 2010

AGRADECIMIENTO

A nuestro director, el Ing. Gabriel Astudillo, así como a las demás personas que nos han brindado su apoyo para la culminación de esta tesis.

También debemos agradecer a todos los que han aportado en ayudarnos a cumplir con nuestra meta: amigos, profesores, y autoridades de este prestigioso establecimiento educativo.

DEDICATORIA

Dedico este trabajo a Dios, quien me ha dado fortaleza y su inmensa ayuda en cada paso que doy, quien me ha levantado de los momentos más difíciles.
A mi padre que creyó en mí y me dio la oportunidad de estudiar, con gran esfuerzo e inmenso amor y paciencia.

Vanessa Gaibor Ortega

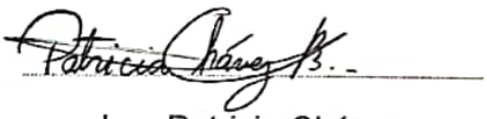
Dedico este trabajo a todos aquellos que durante este largo camino, han creído en mí, y me han ayudado a formar las bases de mi vida profesional.
A todas las personas que han cooperado para que pudiera cumplir con mis objetivos y tareas del día a día de una manera profesional.

Pablo Caicedo Arellano

TRIBUNAL DE SUSTENTACION

Ing. Gabriel Astudillo

PROFESOR DE LA MATERIA DE GRADUACION



Ing. Patricia Chávez

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

DECLARACIÓN EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestas en esta tesis nos corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL).

Jessica Vanessa Gaibor Ortega

Pablo Raúl Caicedo Arellano

RESUMEN

El presente proyecto realiza el análisis de algunas de las amenazas que afectan a las redes de VoIP, aprovechando cualquier debilidad de la red. Se mostrara la importancia de la seguridad en un sistema VoIP.

Se escaneara la red, con la finalidad de encontrar dispositivos que la conforman, así como las extensiones con la que dispone la PBX.

Se trabajara sobre el escenario en la que dos máquinas de la PBX se comunican entre sí, y una tercera maquina intrusa que ha logrado ingresar a la red captura la comunicación establecida entre los atacados, logrando así grabar las llamadas que se establecen entre las dos víctimas, utilizando la técnica de eavesdropping, la cual tiene como objetivo interceptar datos de una transmisión de manera no autorizada.

Se usaran ciertas herramientas contenidas en el sistema Operativo Backtrack, instalado en la maquina atacante a fin de recalcar la importancia de la seguridad en un ambiente VoIP.

Finalmente se estudiará el método de encriptación VPN como medida para prevenir estos ataques.

ÍNDICE DE CONTENIDO

ABREVIATURAS

VoIP	Voz Sobre IP
IP	Protocolo de Internet
Softphone	Software instalado en PCs que simulan teléfonos IP
IVR	Interactive Voice Response
TDM	Multiplexación por división de tiempo
FXS	Foreign Exchange Station
FXO	Foreign Exchange Office
STMP	Simple Mail Transfer Protocol
HTTP	Hypertext Transfer Protocol
RTP	Real-time Transport Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
IAX2	Inter-Asterisk exchange protocol
NAT	Network Address Translation
LAN	Local Area Network
MitM	Man-in-the-middle

MAC	Media Access Control
VLAN	Virtual LAN
VPN	Virtual Personal Network

ÍNDICE DE FIGURAS

INTRODUCCIÓN

La tecnología VoIP es vulnerable en muchos aspectos, debido a que para su funcionamiento efectivo depende de algunos factores, como son las capas y protocolos sobre las cuales trabaja, así como la red por la que se transmite, y dispositivos que intervienen para su funcionamiento.

Las llamadas establecidas en un sistema VoIP, generalmente se ven expuestas a riesgos de seguridad y privacidad debido a que se transmiten por internet o por redes potencialmente inseguras.

Por este motivo, debido a la gran cantidad de información importante que puede verse comprometida por un sistema inseguro de Voz Sobre IP, resulta preocupante que cualquier intruso pueda dañar el buen desenvolvimiento de nuestro sistema.

Por lo tanto, en este proyecto se estudiaran algunas herramientas que pueden ser usadas simplemente para obtener información o usarla para perjudicar a un ente que trabaje bajo esta tecnología, de la misma manera se expondrá una opción para evitar este tipo de ataques.

CAPITULO 1
ANTECEDENTES Y JUSTIFICACIÓN

1.1 ANTECEDENTES

Desde sus inicios, la tecnología VoIP es usada principalmente por su gratuidad, por las grandes ventajas que brinda a grandes y pequeñas empresas, como lo es la transportación de servicios por redes IP (internet), en vez de ser transportado por una red telefónica convencional. Lamentablemente como en toda tecnología existen debilidades, de las cuales personas mal intencionadas se aprovechan. Existen numerosas herramientas de ataques contra sistemas VoIP, comunidades de Hackers con objetivos destructivos hacia entidades que usan esta tecnología en su diario vivir.

1.2 JUSTIFICACIÓN

El presente proyecto ha sido desarrollado con el fin de mostrar algunos de los tantos ataques que se pueden realizar en un sistema VoIP, y exponer una de las medidas de contingencia que hoy en día se puede implementar para su correcto funcionamiento.

1.3 DESCRIPCIÓN DEL PROYECTO

El estudio de seguridad realizado a lo largo de este trabajo tiene como propósito, alcanzar los siguientes objetivos:

1.3.1 Objetivos Generales

Mediante el análisis de algunas herramientas hacker usadas para encontrar información o dañar el buen desempeño de un sistema VoIP, mostrar la importancia de tomar medidas de seguridad para evitar este tipo de ataques.

1.3.2 Objetivos específicos

Realizar los siguientes ataques:
Identificación de los dispositivos VoIP disponibles en la red.
Reconocimiento de extensiones SIP e IAX de la PBX.
Captura de llamadas e implementación de VPN dentro del sistema VoIP para evitar ataques al sistema.

1.4 METODOLOGÍA

Para cumplir con los objetivos descritos en este proyecto se instalara el Sistema Operativo Backtrack en la maquina intrusa que forma parte del sistema VoIP. Backtrack contiene las herramientas usadas a lo largo del proyecto para realizar los diferentes ataques. Finalmente se realizara una secuencia de instrucciones tanto en el servidor del Sistema VoIP como en la máquina de cada usuario donde se encuentra cada softphone para levantar VPN dentro del sistema como contramedida ante los ataques analizados.

1.5 PERFIL DE LA TESIS

El objetivo principal de esta tesis es el análisis de herramientas usadas en ataques a sistemas VoIP y el estudio de medidas para prevenir y solucionar estas amenazas de Seguridad.

En el capítulo 2, se expone la teoría fundamental de lo que es una red Voz Sobre IP. Concepto, ventajas y protocolos del software libre Asterisk.

En el capítulo 3 se detalla el uso cada una de las herramientas analizadas en este proyecto para ejecución de los ataques al sistema VoIP.

Finalmente en el capítulo 4 se muestra el procedimiento a realizar para la implementación de VPN dentro de la red VoIP como medida de prevención ante los ataques analizados en el capítulo 3.

CAPITULO 2
VOZ SOBRE IP y ASTERISK

2.1 Red VoIP

La Voz sobre IP es una tecnología de telefonía que permite la transmisión de voz en forma de paquetes de datos a través de redes IP.

La telefonía IP es una aplicación inmediata de esta tecnología, de tal manera que permite la realización de llamadas telefónicas ordinarias sobre redes IP u otras redes de paquetes, utilizando PCs, gateways y teléfonos estándares.

La ventaja real de esta tecnología es la transmisión de voz en forma gratuita, ya que viaja como datos.

2.1.1 Infraestructura básica de una red VoIP

Los sistemas VoIP cuentan con una amplia variedad de equipos y características, siendo las más importantes los equipos procesadores o administradores de llamadas IP, los cuales se encargan de establecer el control, acceso y estado de los diferentes integrantes de la red telefónica, además de gateways de voz que sirven de puente entre servicio de VoIP y el servicio local telefónico.

También intervienen, equipos intermedios, como lo son los routers, encargados de llevar las reglas de tráfico y transporte entre redes IP, firewalls o cortafuegos encomendados para proteger a la red de ataques externos a

ésta, y protocolos de señalización telefónica IP basados en un conjunto de estándares que determinan cuáles operaciones (detección de estado, establecimiento de llamada, gestión y mantenimiento de la voz) se efectúan en la red telefónica.

Dentro de la estructura básica de una red VoIP hay que diferenciar tres elementos fundamentales:

- **Terminales:** Son los dispositivos que usan los usuarios para comunicarse. Implementados tanto en hardware como en software, realizan las funciones de los teléfonos tradicionales.

- **Gateways:** De forma transparente se encargan de conectar las redes VoIP con las redes de telefonía tradicional.

- **Gatekeepers:** Se encargan de realizar tareas de autenticación de usuarios, control de admisión y de ancho de banda, encaminamiento, servicios de facturación, entre otros.

La figura 2.1 muestra la manera en que se interconectan los equipos que componen la infraestructura de una central telefónica para que esta funcione correctamente.

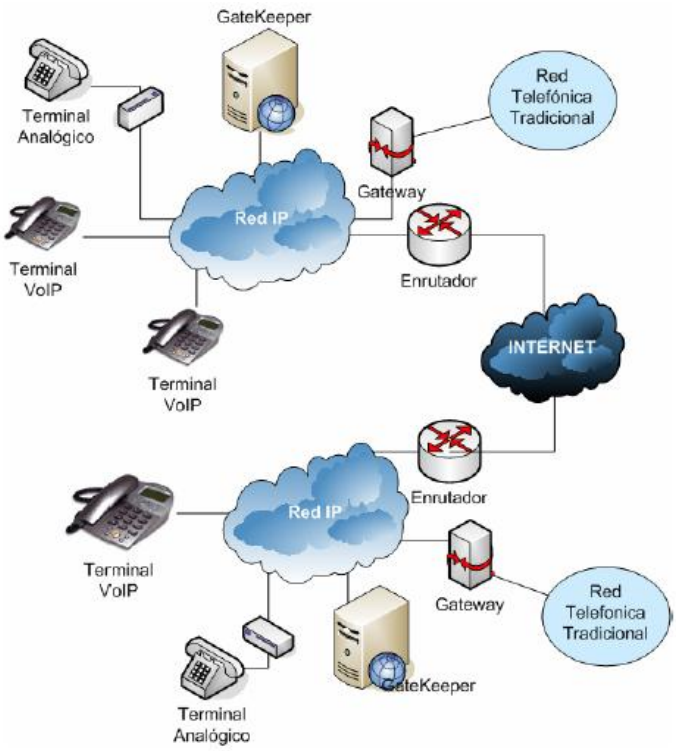


Figura2.1 Estructura Básica de dos delegaciones de una misma empresa conectadas telefónicamente a través de Internet

2.2 ASTERISK

Asterisk es un software de código abierto, es decir totalmente gratuito y de alcance libre. Este software contiene algunas librerías que se modifican con el fin de configurarlas según la estructura, cantidad de equipos terminales y funcionamiento que se desea tener en la central telefónica a implementar.

Asterisk ofrece funcionalidades de: buzones de voz, IVR, etc., ya disponibles en centrales telefónicas tradicionales, adicionalmente ofrece servicios de: grabación de llamadas, extensiones remotas. Trabaja sobre una plataforma de servidor Linux.

Entre algunas de las ventajas que presenta sobre la telefonía convencional, se destacan:

2.2.1 Escalabilidad

Asterisk es una PBX con capacidad de servicio de hasta múltiples usuarios distribuidos en diferentes sucursales de una gran Empresa.

2.2.2 Competitividad en costos

Por ser un sistema gratuito de código abierto (Open Source), por su efectividad, arquitectura, entre otros, se ha convertido en una herramienta básica en grandes y pequeñas empresas.

2.2.3 Flexibilidad

Asterisk permite la conexión a redes públicas de telefonía tradicional e integrarse fácilmente a otras centrales Asterisk.

2.3 PROTOCOLOS

Hoy en día, existen dos protocolos para transmitir voz sobre IP, ambos definen la manera en que los dispositivos de este tipo deben establecer la comunicación entre sí.

2.3.1 Protocolo SIP

SIP es un protocolo simple de señalización y control, generalmente usado para telefonía y videoconferencias sobre las redes IP. Su estructura está basada en otros protocolos como STMP y HTTP con los que guarda cierta similitud.

SIP es un protocolo abierto y ampliamente soportado que no depende de ningún fabricante. Su simplicidad, escalabilidad y facilidad para integrarse con otros protocolos y aplicaciones lo han convertido en un estándar de la telefonía IP.

SIP es un protocolo de señalización por lo que solo maneja el establecimiento, control y terminación de las sesiones de comunicación.

Dentro de una red SIP vamos a encontrar los siguientes componentes que intervienen en el proceso de autenticación, el cual se da cuando se va establecer una llamada: Agentes de Usuario (UA) y servidores.

Entre los agentes de usuario, podemos encontrar los UAC y los UAS:

- Los agentes de usuario clientes (UAC) que son los encargados de iniciar las peticiones de llamada y,
- Los agentes de usuario servidor (UAS) que reciben las peticiones del UAC.

Normalmente una vez que se ha establecido una llamada en un Sistema VoIP, se produce el intercambio de paquetes RTP que transportan realmente el contenido de la voz. SIP es un protocolo de aplicación y funcionará tanto sobre UDP como TCP.

SIP utiliza el puerto UDP 5060 para el establecimiento, negociación, y fin de la comunicación.

2.3.2 Protocolo IAX2

IAX: Es uno de los protocolos utilizado por Asterisk, para manejar conexiones VoIP entre servidores Asterisk, y entre servidores y clientes que también utilizan protocolo IAX.

El protocolo IAX ahora se refiere generalmente al IAX2, la segunda versión del protocolo IAX. El protocolo original ha quedado obsoleto en favor de IAX2.

IAX2 es robusto, lleno de novedades y muy simple en comparación con otros protocolos. Puede ser usado para transportar virtualmente cualquier tipo de datos, esta capacidad lo hace muy útil para realizar videoconferencias o realizar presentaciones remotas.

IAX2 utiliza un único puerto UDP, generalmente el 4569, para comunicaciones entre puntos finales (terminales VoIP), para señalización y datos. IAX2 es un protocolo casi

transparente a los cortafuegos y realmente eficaz para trabajar dentro de redes internas.

IAX es extremadamente flexible y puede ser utilizado con cualquier tipo de dato incluido vídeo, provee soporte para ser transparente a NAT.

CAPITULO 3
ATAQUES A UN SISTEMA VoIP

3.1 Introducción

En este capítulo se estudia algunas herramientas que son de utilidad para realizar diferentes ataques a un Sistema VoIP. Se hará un análisis de cada una de las herramientas, en el orden en que estas proporcionen información del Sistema VoIP que será de ayuda para la siguiente herramienta a tratar.

En primera instancia se realizara el escaneo de la red, a fin de identificar los softphone y el Servidor Asterisk del Sistema VoIP, luego, se encontraran las extensiones SIP e IAX de la PBX. Posteriormente se hará la captura de llamadas internas en el Sistema VoIP, y finalmente se atacara al servidor mediante el método de denegación de Servicio.

3.2 Identificación de Dispositivos de Red

En este punto se va a escanear la red a fin de identificar a los dispositivos del sistema VoIP, para esto se usara la herramienta smap.

3.2.1 Smap

Smap es una excelente herramienta de hackeo, que trabaja sobre Linux. Muy útil para la detección de dispositivos finales como lo son los Softphone y servidor Asterisk.

Esta herramienta permite la detección de las direcciones IP activas en nuestra PBX, es decir las que corresponden a

los Softphone y al servidor Asterisk, con tal solo digitar la línea de comando “smap dirección_IP_de_Red/CIDR, tal como se muestra a continuación:

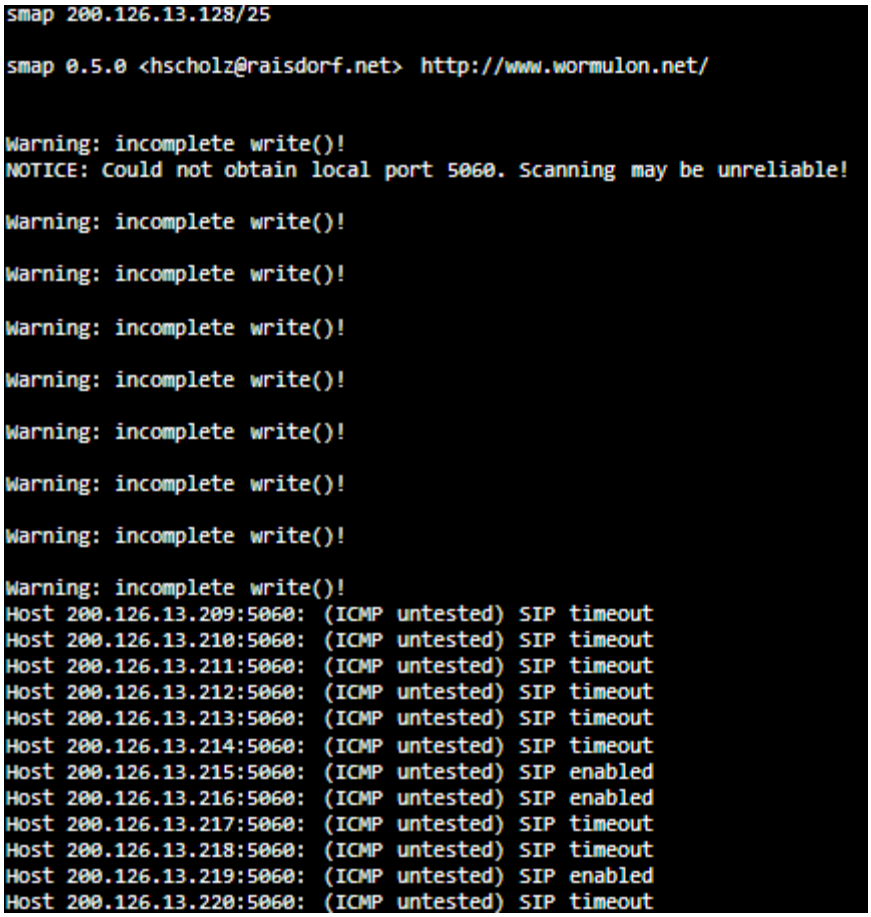


Figura 3.1 Escaneo del Sistema VoIP

Tal como se observa en la figura 3.1, en el escaneo realizado de la red 200.126.13.128/25, se encuentra que las direcciones: 200.126.13.215, 200.126.13.216 y 200.126.13.219, son calificadas por smap, como SIP enabled, es decir son direcciones IP activas de la red.

Para identificar a que dispositivos pertenecen dichas direcciones, se debe aplicar el comando: “smap –l dirección_IP” a cada una de las direcciones encontradas.

La figura 3.2 muestra que, aplicando el comando smap –l a la dirección IP 200.126.13.215, se encuentra que esta corresponde a un softphone de la PBX.

```
smap -l 200.126.13.215

smap 0.5.0 <hscholz@raisdorf.net> http://www.wormulon.net/

NOTICE: test_headers: cmpstr: "Via:From:To:Call-ID:CSeq:Content-
Length:Server:"
NOTICE: test_headers: Please add new cmpstr
Host 200.126.13.215:5060: (ICMP untested) SIP enabled
best guess (75% sure) fingerprint:
  Draytek UA-1.1.x or 1.2.x series

FINGERPRINT information:
newmethod=NR
accept_class=ignore
allow_class=ignore
supported_class=ignore
hoe_class=ignore
options=501
brokenfromto=NR
prack=NR
ping=NR
invite=100
headers found:
  Server: SJphone/1.73.375a (SJ Labs)


1 host scanned, 1 SIP enabled
```

Figura 3.2 Identificación del Softphone1 de la PBX

En la figura 3.3 se observa que la dirección IP 200.126.13.219 corresponde a otro softphone de la PBX.

```
smap -l 200.126.13.219

smap 0.5.0 <hscholz@raisdorf.net> http://www.wormulon.net/

NOTICE: test_headers: cmpstr: "Via:From:To:Call-ID:CSeq:Content-
Length:Server:"
NOTICE: test_headers: Please add new cmpstr
Host 200.126.13.219:5060: (ICMP untested) SIP enabled
best guess (75% sure) fingerprint:
  Draytek UA-1.1.x or 1.2.x series

FINGERPRINT information:
newmethod=NR
accept_class=ignore
allow_class=ignore
supported_class=ignore
hoe_class=ignore
options=501
brokenfromto=NR
prack=NR
ping=NR
invite=100
headers found:
  Server: SJphone/1.63.328a (SJ Labs)


1 host scanned, 1 SIP enabled
```

Figura 3.3 Identificación del Softphone2 de la PBX

Finalmente aplicando el comando smap –l a la dirección 200.126.13.216, en la Figura 3.4 se observa que esta dirección IP corresponde al Servidor Asterisk de la PBX.

```
smap -l 200.126.13.216


smap 0.5.0 <hscholz@raisdorf.net> http://www.wormulon.net/

NOTICE: test_accept: "Accept: application/sdp"
NOTICE: test_allow: "Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
SUBSCRIBE, NOTIFY"
NOTICE: test_supported: "Supported: replaces"
NOTICE: test_headers: cmpstr: "Via:From:To:Call-ID:CSeq:User-
Agent:Allow:Supported:Contact:Accept:Content-Length:"
NOTICE: test_headers: Please add new cmpstr
Host 200.126.13.216:5060: (ICMP untested) SIP enabled
best guess (77% sure) fingerprint:
  Asterisk PBX (unknown version)

FINGERPRINT information:
newmethod=501
accept_class=2
allow_class=201
supported_class=8
hoe_class=ignore
options=200
brokenfromto=404
prack=481
ping=501
invite=100
headers found:
  User-Agent: Asterisk PBX
```

Figura 3.4 Identificación del Servidor Asterisk de la PBX

3.3 Obtención de Extensiones

Ya hemos encontrado la dirección IP del Servidor Asterisk de la PBX analizada, por lo que podemos obtener las extensiones de este Sistema VoIP mediante el uso de dos herramientas: Sipscan y EnumIAX.

3.3.1 SipScan

Sipscan es una herramienta que trabaja sobre Windows, cuya utilidad es obtener extensiones SIP de una PBX, tan solo conociendo la dirección IP del Servidor.

Se debe ingresar en los campos “Target SIP Server” y “Target SIP Domain”, la dirección IP del Servidor. Se debe escoger en “Transport”, el protocolo UDP y en port ingresar el puerto 5060, que es sobre el cual trabaja SIP.

Figura 3.5 Obtención de las extensiones SIP de la PBX

La figura 3.5 muestra que para la PBX analizada, cuya IP del Servidor es: 200.126.13.216, las extensiones SIP encontradas con ayuda de la herramienta SipScan son: 601 y 602.

3.3.2 EnumIAX

EnumIAX es una herramienta de hackeo que trabaja sobre Linux y que sirve para la obtención de extensiones IAX de una PBX. Para esto se debe ingresar el comando enumiax seguido de la direccion IP del Servidor. Aplicando este comando a la direccion IP del servidor ya encontrado, se observa en la figura 3.6 que las extensiones IAX de la PBX son: 603 y 604:

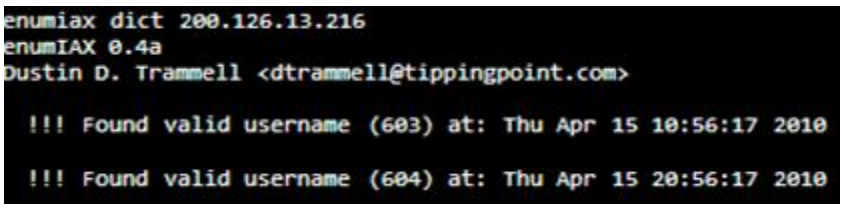


Figura 3.6 Obtención de las extensiones IAX de la PBX

3.4 Eavesdropping

Eavesdropping es una técnica que tiene como objetivo capturar información cifrada o no por parte de un intruso al cual no iba dirigida dicha información.

Refiriéndonos a la telefonía IP, es la intercepción de las conversaciones VoIP por parte de individuos que no participan en la conversación.

Eavesdropping en VoIP requiere interceptar la señalización y los streams de audio o flujos de datos de una conversación. La

señalización trabaja con el protocolo SIP, y el flujo de datos usa el protocolo RTP.
Eavesdropping se realiza con Sniffers, los cuales son programas que se encargan de monitorear los paquetes que circulan en la red, estos pueden ser ubicados en una estación de trabajo conectada a la red y pueden ser ejecutados por un intruso u otra persona de acceso legítimo.

Un sniffer trabaja poniendo la tarjeta de red de la computadora que hace el Eavesdropping en modo promiscuo, el cual desactiva el filtro de verificación de direcciones de destino provocando así que los paquetes enviados a la red de la PBX analizada, lleguen a esta tarjeta de red, realizando así la captura de información que no le corresponde a esta máquina.

Existen excelentes sniffer como ethereal/wireshark que permiten capturar todo el tráfico del segmento de red donde se encuentra.

Generalmente las Empresas cuentan con redes conmutadas, lo que podría suponer que se eliminaría el Eavesdropping por el uso de switches, sin embargo esto deja de ser cierto usando otra técnica denominada: “Man in the Middle”, usando envenenamiento ARP.

3.4.1 Man in the Middle

Man in the middle es una técnica en la cual el atacante envía a los usuarios de la red avisos con la MAC de su

máquina falseada o “spoofeada”, logrando así que los paquetes IP lleguen a su host.

Esto lo logra por medio de un ARP spoofing, con lo que el atacante podrá capturar, analizar y escuchar comunicaciones VoIP.
El término spoofing se explicara en el siguiente contexto:

3.4.2 Arp Spoofing

ARP Spoofing es una técnica usada en redes Ethernet conmutadas, es decir que cuenta con switches como dispositivos intermedios. Esta técnica tiene como finalidad, suplantar la identidad de un usuario de la red, mediante la falsificación de su tabla ARP. La figura 3.7 muestra de forma gráfica como el atacante que realiza esta técnica recibe información que no le corresponde.

El protocolo ARP, es el encargado de traducir direcciones IP a direcciones MAC, lo cual es de utilidad para esta técnica ya que el protocolo Ethernet trabaja mediante direcciones MAC.

Esta técnica se basa en el proceso que se genera cuando una comunicación se va establecer, la cual se da de la siguiente manera:

Cuando un host quiere comunicarse con otro host de una determinada IP, este emite una trama ARP-Request a la

dirección de Broadcast pidiendo la MAC del host poseedor de la IP con la que desea comunicarse. El ordenador con la IP solicitada responde con un ARP-Reply indicando su MAC.

La técnica ARP Spoofing falsifica la tabla ARP (relación IP-MAC) que guardan los host y switches de la red, enviando tramas ARP-REPLY a la red indicando su MAC como destino válido para una IP específica, capturando la información que le iba dirigida.

Hasta el momento con las técnica mencionada se logra enviar paquetes a la víctima a fin que esta cambie su tabla ARP, de tal manera que el intruso logra direccionar a su tarjeta de red, la información que no le iba dirigida, información que se podrá capturar con el uso de un sniffer que capture para nuestro caso el proceso de autenticación que se da cuando se va establecer una llamada y así capturar dicha conversación.

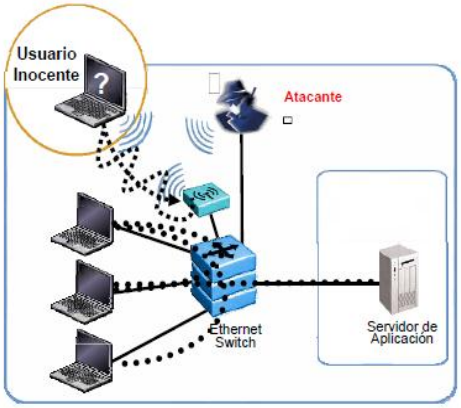


Figura 3.7 ARP Spoofing

3.4.3 Proceso de Autenticación

El proceso de autenticación consiste en un flujo de peticiones y Request que aseguran que la llamada se establece de forma correcta. Cada dispositivo debe autenticarse para lograr dicha comunicación. La figura 3.8, muestra el proceso de autenticación, el cual se detalla a continuación:

PETICION El usuario notifica al servidor que desea realizar una llamada.

DESAFIO El servidor responde con un mensaje que incluye un valor aleatorio (*nonce*) junto al dominio contra el que se va a autenticar (*realm*).

PETICION El cliente envía una respuesta cifrada al servidor en un mensaje de tipo *response*, indicando el *nonce*, el *realm* junto con el nombre de usuario, el uri y la contraseña.

AUTENTICACION: Una vez que el servidor recibe estos datos, compara el valor de la respuesta del cliente con el resultado del cifrado que realizo por su cuenta, si son iguales, se establece la comunicación.
La siguiente captura resume lo expuesto:

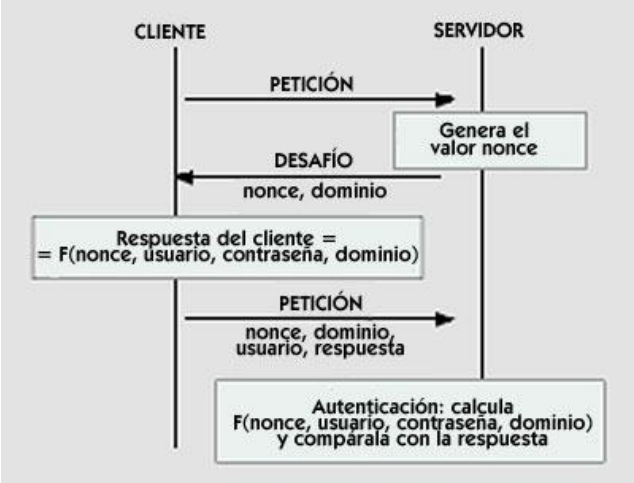


Figura 3.8 Proceso de Autenticación para el establecimiento de una llamada entre dos usuarios de la PBX

Para que un intruso pueda escuchar una llamada que no le corresponde, la que se da entre dos usuarios de una PBX, debe capturar los mensajes que se dan en el proceso de autenticación. Esto se logra generando peticiones falsas ARP con la ayuda de la herramienta ettercap, para direccionar el proceso de autenticación a la tarjeta de red intrusa y capturarlas con el sniffer wireshark.

3.4.4 Ejecución de Ettercap

Ettercap es una herramienta que realiza el envenenamiento ARP aplicando la técnica man the middle.

En este proyecto se usa la versión de Ettercap NG-0.7.3. El procedimiento a realizar es el siguiente:

Una vez abierto Ettercap, tal como muestra la figura 3.9, se debe hacer clic en la opción "Sniff" y escoger: "Unified sniffing..."

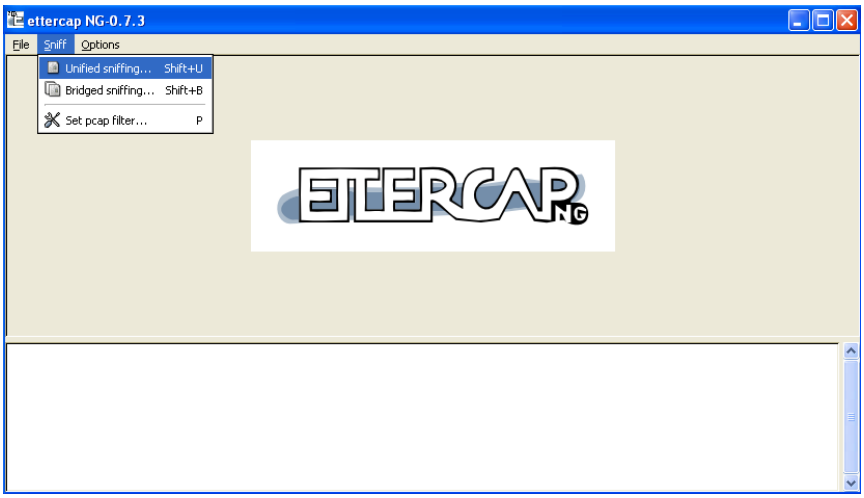


Figura 3.9 Paso1 Ettercap

Con lo que se muestra la opción “Network interface”, tal como se observa en la figura 3.10, en la cual se debe escoger la interfaz que conecta al atacante con la red VoIP que se quiere esnifar y cliquear en "OK".

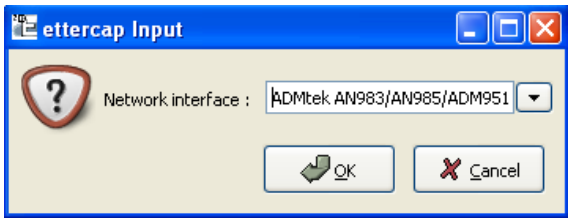


Figura 3.10 Selección de la tarjeta de red en ettercap

El siguiente paso es buscar los host de la red, para esto se hace clic en "Hosts" > "Scan for hosts". La figura 3.11 muestra el proceso de búsqueda que realiza la herramienta

ettercap, para encontrar la cantidad de host presentes en la red.

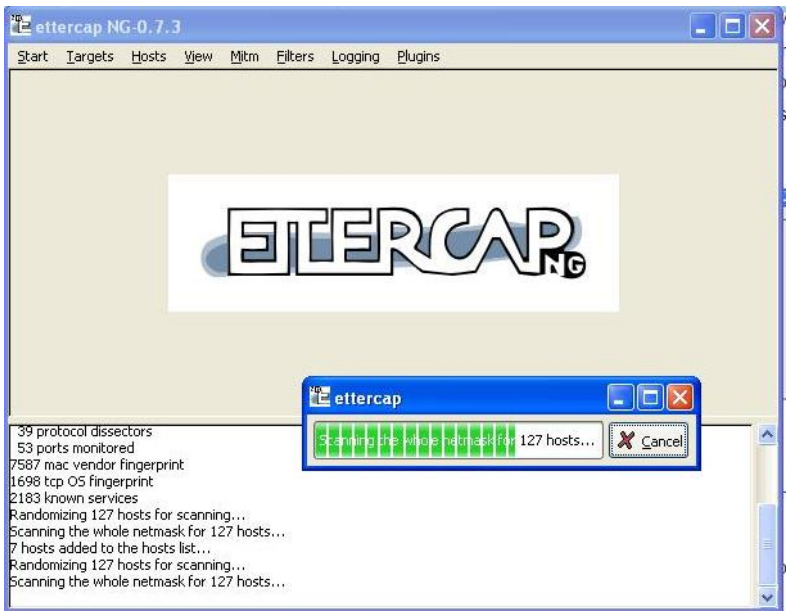


Figura 3.11 Detección de la Cantidad de Host en la red

Terminado el proceso de búsqueda, se debe hacer clic en "Hosts" > "Hosts list" para visualizar los host encontrados por ettercap.

Para poner en ejecución ettercap se deben agregar a las tarjetas 1 y 2, las direcciones 200.126.13.215 y 200.126.13.219 que corresponden a los softphone involucrados en la conversación a sniffar y la dirección 200.126.13.216 que corresponde al servidor Asterisk ya encontrado anteriormente. Esto se realiza seleccionando cada una de estas direcciones IP y seleccionando: "Add to Target 1" y "Add to Target 2".

La figura 3.12 muestra como las direcciones IP 200.126.13.215, 200.126.13.219 y 200.126.13.216 ya han sido agregadas a TARGET1 y TARGET2, con lo que para comenzar a esnifar se debe hacer clic en: "Start" > "Start Sniffing".

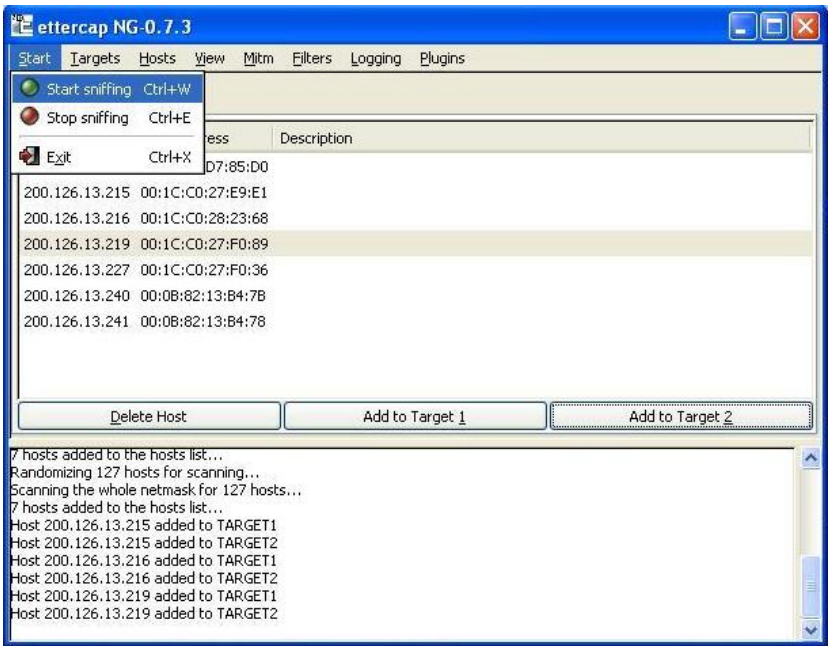


Figura 3.12 Inicio del Sniffeo de la conversación con Ettercap

Para realizar el envenenamiento ARP, se debe cliquear sobre la opción "MitM" y escoger "Arp Poisoning...", con lo que ettercap muestra un cuadro con dos opciones de las cuales se debe escoger: "Sniff remote connections" y "OK".

La figura 3.13 muestra a ettercap realizando la técnica Man in the Middle en la PBX:

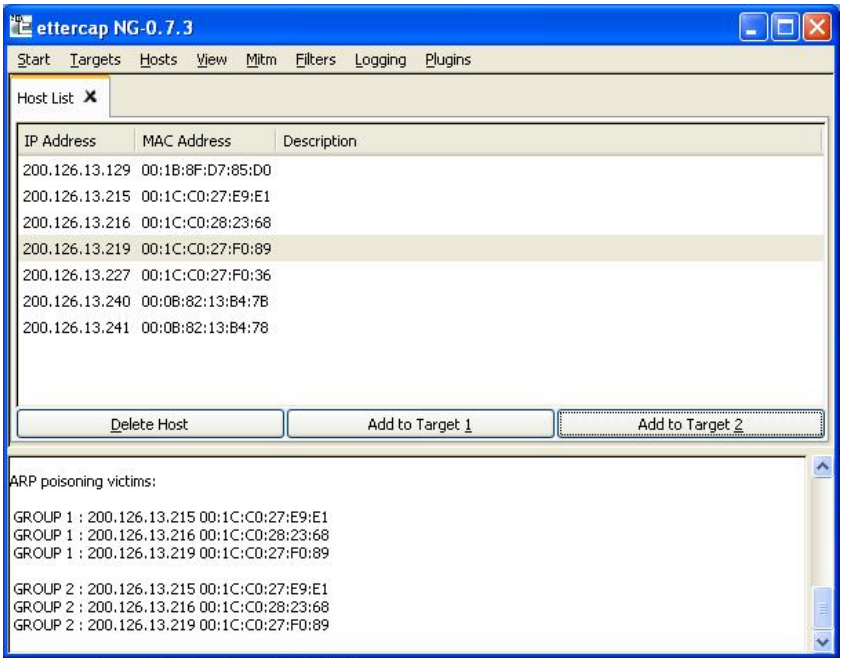


Figura 3.13 MitM con Ettercap

Por último para detener el sniffeo, se debe hacer clic en la opción "MitM" y escoger "Stop MitM attack(s)". Luego hacer clic en Start y escoger “Stop Sniffing”.

Mientras se ejecuta Ettercap, se debe correr la herramienta wireshark para realizar la captura de las llamadas internas de la PBX.

3.4.5 Captura de llamadas con Wireshark

Wireshark es un sniffer de paquetes que captura todo el tráfico de datos que ingresa por la tarjeta de red del equipo donde se encuentra instalado. El procedimiento realizado para la captura de una llamada se detalla a continuación:

Tal como especifica la figura 3.14, se debe cliquear sobre el icono “Show the Capture options”, con lo que se despliega una ventana en la cual hay que habilitar “Capture packets in promiscuous mode”, opción que habilita la tarjeta del intruso en modo promiscuo. En Interface escoger: Local y seleccionar la tarjeta de red que conecta al intruso con el sistema VoIP. Finalmente cliquear Start.

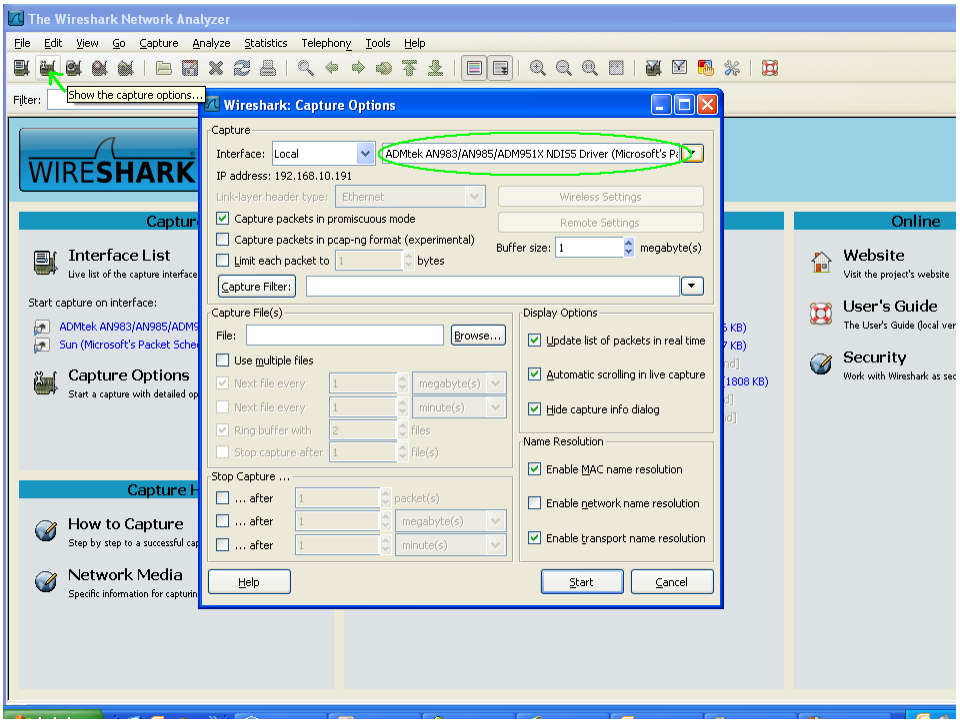


Figura 3.14 Inicio de Wireshark

Con los parámetros seleccionados, wireshark empieza a capturar paquetes con destino a las direcciones IP agregadas en Ettercap. Es decir con destino a los 2 softphone de la PBX, y al servidor Asterisk. El flujo de

paquetes capturados por wireshark, se puede visualizar en la figura 3.15, que se muestra a continuación:

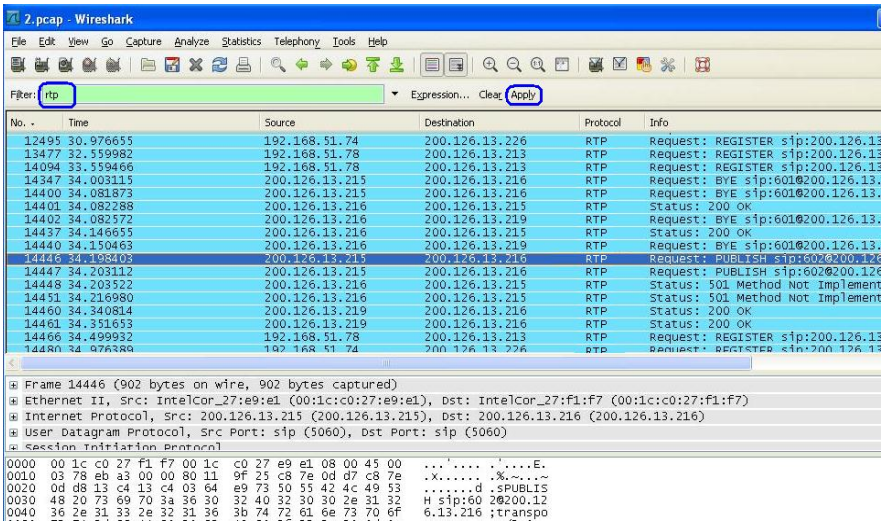


Figura 3.15 Captura paquetes de una llamada

Realizado la captura de paquetes que se consideren necesarios, se debe cliquear sobre la opción: “Stop the running live capture” y guardar los paquetes capturados haciendo clic en File Save As.

En la captura realizada se puede observar los procesos de autenticación de las llamadas entre el softphone 200.126.13.215 y el softphone 200.126.13.219.

Para una mejor visualización de la o las llamadas capturadas, se debe escoger la opción VoIP Calls del menú Telephony, con lo que se despliega la ventana que se muestra en la figura 3.16:

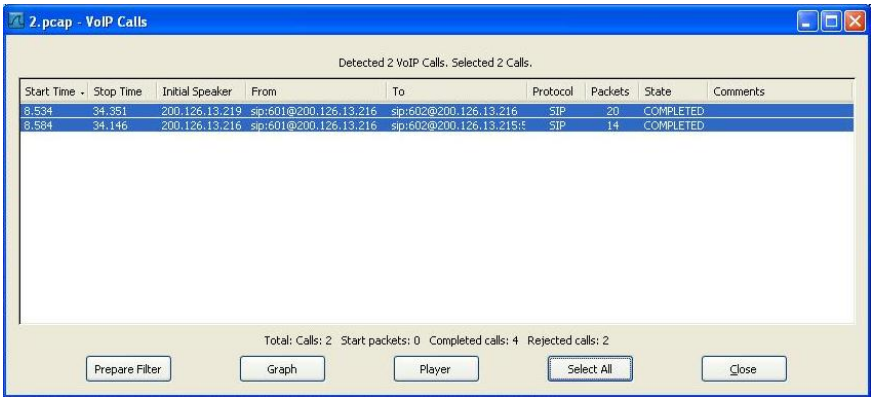


Figura 3.16 VoIP Calls de Wireshark

En la figura 3.16 se observa que la llamada se realizó desde la extensión 601 del softphone 200.126.13.219 a la extensión 602 del softphone 200.126.13.215.

Seleccionando la opción Graph, se muestra la ventana de la figura 3.17, la cual detalla en forma gráfica el proceso de autenticación que se realizó ante el servidor de la PBX 200.126.13.216 para el establecimiento de la llamada capturada entre el softphone 200.126.13.219 y el softphone 200.126.13.215.

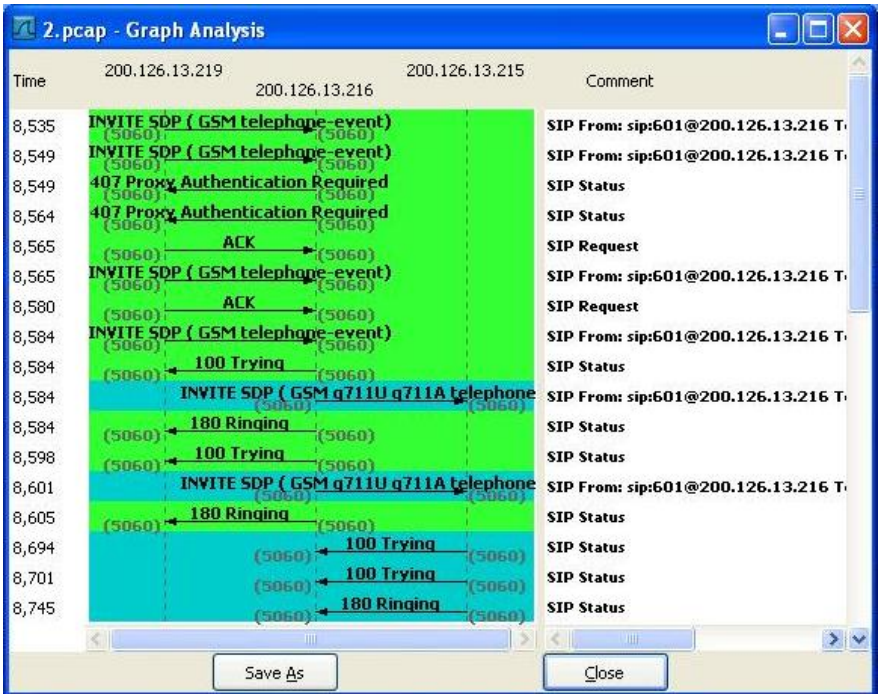


Figura 3.17 Proceso de Autenticación de la llamada capturada con Wireshark

Para finalmente escuchar la llamada, se filtró los paquetes RTP de la conversación capturada, los cuales ya fueron obtenidos en la figura 3.15, para esto se escribió RTP en el campo Filter y se hizo clic en Apply, luego se escogió la opción RTP del menú Telephony y haciendo clic en la opción Streams Analysis, se obtuvo una ventana del mismo nombre, en la que para guardar la conversación se hizo clic en Save Payload As, obteniendo así la siguiente ventana:

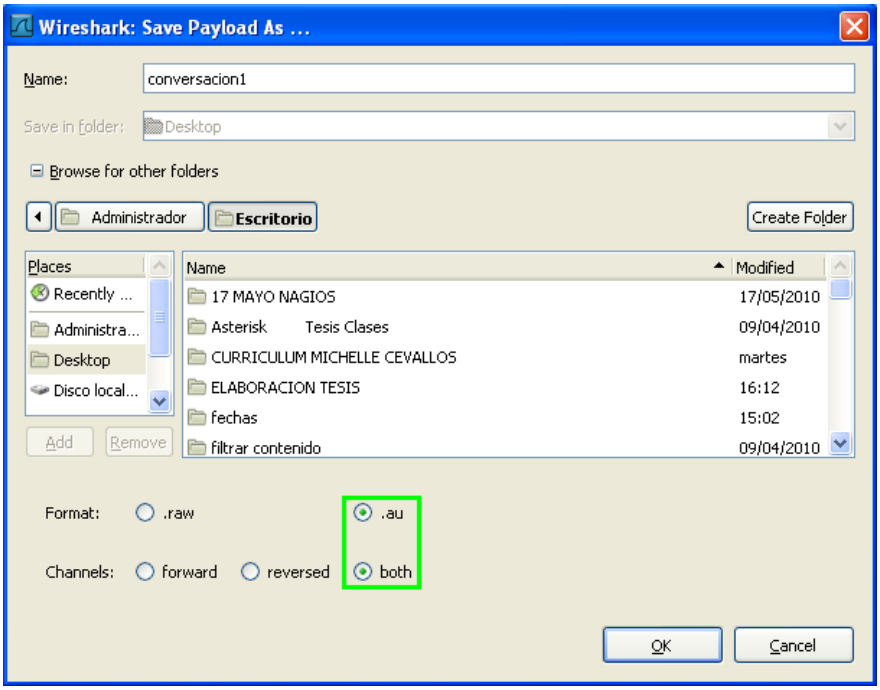


Figura 3.18 Creación del archivo .au de la llamada capturada.

En la figura 3.18, en el campo Name se escribió el nombre del archivo con en el que se guardó la conversación capturada, se escogió la ubicación del archivo en places, el formato .au, y en channels se escogió both, ya que el flujo de la llamada es en ambos sentidos.

Tal como se muestra en figura 3.19, el archivo “conversacion1” fue guardado en el escritorio, el cual puede ser escuchado desde Windows Media.

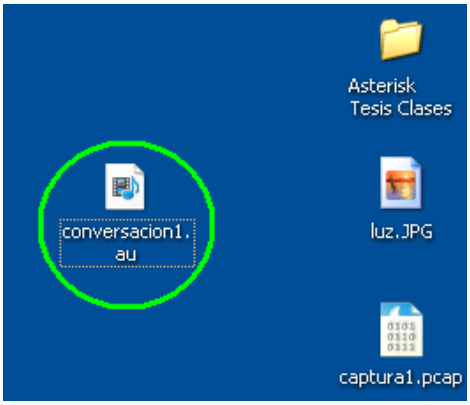


Figura 3.19 Archivo reproducible que contiene la llamada sniffeada.

3.5 Denegación de Servicio

Un ataque por denegación de servicio tiene como objetivo impedir el normal comportamiento de un sistema, que para nuestro estudio, es de VoIP. Este ataque causa que un servicio o recurso sea inaccesible a usuarios legítimos.

El ataque a analizar consiste en enviar múltiples peticiones al servidor PBX, lo que provoca la saturación del mismo, de tal manera que no puede procesar otras llamadas. Para este ataque se usa la herramienta Sipsak:

3.5.1 Sipsak

Sipsak es una herramienta que trabaja sobre Linux, su función es enviar múltiples peticiones al servidor de un sistema VoIP con la finalidad de saturarlo, de tal manera que

no pueda procesar más peticiones, con lo que si un dispositivo terminal de la PBX desea comunicarse con otro dispositivo, al enviar una petición al servidor, este hará caso omiso, provocando un mal funcionamiento de la PBX.

Para este proyecto, la línea de comando usada de esta herramienta es: sipsak –F –s sip:601@200.126.13.216. Donde F activa el modo Flood o inundación, opción que hace que las peticiones al servidor vayan en aumento, 601 es el usuario de destino de los Request enviados por el servidor 200.126.13.216, en respuesta de las múltiples peticiones que le están siendo enviadas con la herramienta sipsak.

3.6 BackTrack

BackTrack es un completo sistema operativo que trabaja sobre Linux, el cual contiene una serie de herramientas enfocadas a la seguridad y al hack.

Por la gran utilidad que proporciona BackTrack, al contener todas las herramientas requeridas para hacer los diversos ataques analizados anteriormente, fue instalado en la maquina intrusa para la elaboración de este proyecto.

Para esto se instaló en la maquina intrusa, una máquina virtual: Virtual Box, posteriormente se descargó de la web la imagen ISO de BackTrack y se instaló sobre la máquina virtual.

CAPITULO 4

IMPLEMENTACION DE VPNs PARA EVITAR ATAQUES A UN SISTEMA VOIP

4.1 Introducción

Las siglas VPN significan: virtual personal network. VPN es una técnica que permite enlazar 2 o más redes, de tal manera que simulan ser una sola red privada, permitiendo así la conexión entre ellas como si los usuarios de ambas redes estuviesen dentro de una misma red. VPN generalmente es usado por usuarios remotos para acceder a redes LAN.

Este método se usara como contramedida de protección ante los ataques analizados a lo largo de este proyecto, ya que el uso de VPNs permitirá al sistema VoIP usar aplicaciones y enviar datos a través de este de forma segura.

En este proyecto se levantara una interfaz túnel en el Servidor Asterisk y una interfaz túnel en cada una de las máquinas que contienen los softphone pertenecientes a la PBX atacada, creando así túneles entre el servidor Asterisk y cada uno de los usuarios de la PBX, de tal manera que una maquina intrusa al tratar de capturar paquetes que no le corresponden lo hará por la interfaz que la une a la red VoIP, no logrando interceptar paquetes de la red, ya que estos pasaran a través de los túneles VPN. Todos los datos que se transmiten entre clientes y servidor estarán totalmente encriptados a través de una clave RSA. Las VPNs se levantaran de la forma en que se muestra en la figura 4.1.

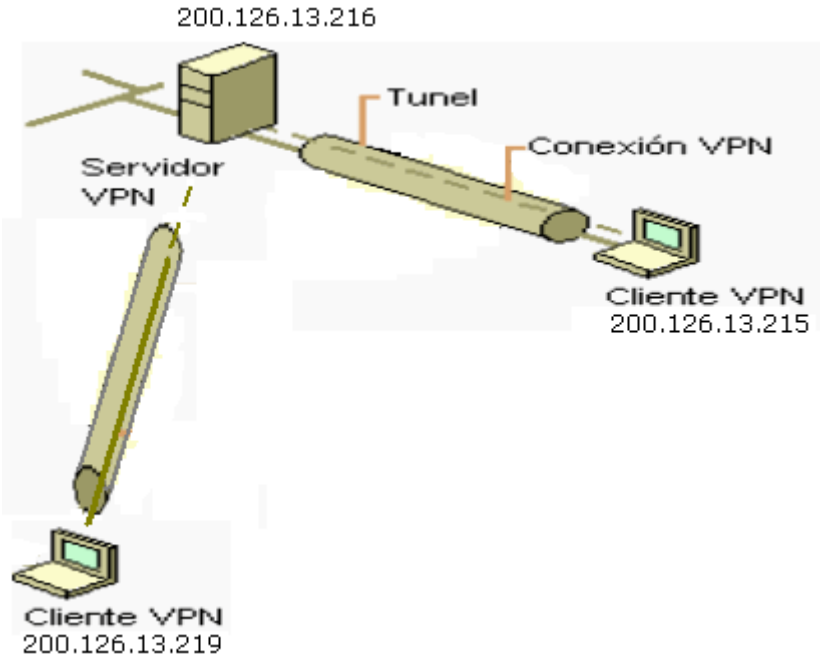


Figura 4**.1** Esquema de las VPNs a levantar en la PBX

Para levantar una VPN entre cada máquina Windows (donde está configurado un softphone) y la maquina Linux (Servidor Asterisk), se debe realizar un procedimiento tanto en el servidor como en la maquina Windows, de la misma manera se debe configurar un firewall para los permisos de acceso a la red. El procedimiento realizado en este proyecto para la implementación de cada VPN, se detalla a continuación:

4.1.1 Configuración del Servidor OpenVPN

1. Desde el root, se creó el fichero **/etc/yum.repos.d/AL-Server.repo**: #vi /etc/yum.repos.d/AL-Server.repo, y se ingresó el siguiente contenido:

```
[AL-Server]
name=AL Server para Enterprise Linux $releasever
mirrorlist=http://www.alcancelibre.org/al/el$releasever
/al-server
gpgcheck=1
gpgkey=http://www.alcancelibre.org/al/AL-RPM-KEY
```

2. Se importo la firma digital de **Alcance Libre** ejecutando lo siguiente desde el root:

```
#rpm --import http://www.alcancelibre.org/al/AL-RPM-KEY
```

3. Se instalo desde los depósitos yum, los paquetes RPM de OpenVPN, Shorewall y vim-enhanced.

```
#yum -y install openvpn shorewall vim-enhanced
```

Los siguientes procedimientos se realizaron sin salir del directorio: /etc/openvpn/

4. Dentro del directorio /etc/openvpn/ se copio los ficheros openssl.cnf, whichopensslcnf, pkitool y vars, localizados en /etc/openvpn/easy-rsa/2.0/:

```
cp /usr/share/openvpn/easy-rsa/2.0/openssl.cnf ./
cp /usr/share/openvpn/easy-rsa/2.0/whichopensslcnf ./
cp /usr/share/openvpn/easy-rsa/2.0/pkitool ./
cp /usr/share/openvpn/easy-rsa/2.0/vars ./
```

5. Se editó las últimas líneas del fichero **/etc/openvpn/vars**, que corresponden a lo siguiente:

```
export KEY_COUNTRY="US"
export KEY_PROVINCE="CA"
export KEY_CITY="SanFrancisco"
export KEY_ORG="Fort-Funston"
export KEY_EMAIL=me@myhost.mydomain
```

Reemplazandolo por:

```
export KEY_COUNTRY="ECU"
export KEY_PROVINCE="Guayas"
export KEY_CITY="Guayaquil"
export KEY_ORG="hotmail"
export KEY_EMAIL=jeyvane@hotmail.com
```

Datos que corresponden a la ubicación del servidor VPN y el dominio o departamento de la organización o Empresa.

6. Para que se carguen las variables de entorno configuradas, se ejecutó la siguiente línea de comando:

```
source /etc/openvpn/./vars
```

7. Se ejecuto el fichero **/usr/share/openvpn/easy-rsa/2.0/clean-all** con sh.

```
sh /usr/share/openvpn/easy-rsa/2.0/clean-all
```

Esta línea de comando realiza una eliminación recursiva sobre el directorio: **/etc/openvpn/keys**, lo que significa que elimina todos los certificados y firmas digitales que hubieran existido con anterioridad.

8. Se creó el certificado del servidor:

```
sh /usr/share/openvpn/easy-rsa/2.0/build-ca
```

9. Se creó el fichero dh1024.pem, el cual contiene los parámetros del protocolo **D**iffie-**H**ellman, de 1024 bits:

```
sh /usr/share/openvpn/easy-rsa/2.0/build-dh
```

El protocolo **D**iffie-**H**ellman permite el intercambio secreto de claves entre dos partes, que en nuestro estudio corresponden a cada uno de los usuarios de la PBX y el servidor Asterisk. Este protocolo es usado para el cifrado de una sesión.

10. Se genero la firma digital con la siguiente línea de comando:

```
sh /usr/share/openvpn/easy-rsa/2.0/build-key-server
server
```

11. Se creó los certificados para ambos usuarios de la PBX (para los usuarios 200.126.13.215 y 200.126.23.219). Con las líneas de comando:

```
sh /usr/share/openvpn/easy-rsa/2.0/build-key cliente1
sh /usr/share/openvpn/easy-rsa/2.0/build-key cliente2
```

12. Se hizo uso de los certificados creados y las configuraciones realizadas, en el fichero: **vi /etc/openvpn/servidorvpn-udp-1194.conf**, editándolo con lo siguiente:

```
port 1194
proto udp
```

```
dev tun
#---- Seccion de llaves -----
ca keys/ca.crt
cert keys/server.crt
key keys/server.key
dh keys/dh1024.pem
#----------------------------
server 10.10.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
keepalive 10 120
comp-lzo
persist-key
persist-tun
status openvpn-status-servidorvpn-udp-1194.log
verb 3
```

Se ingreso la IP 10.10.0.0, porque es recomendable usar una red privada para evitar conflictos entre los host del Sistema VoIP cuando el túnel se encuentre activo. Con mascara 255.255.255.0 porque permitirá a 253 clientes conectarse a la VPN.

13. Se usó el mandato restorecon sobre el directorio /etc/openvpn a fin de asignar los contextos adecuados.

```
restorecon -R /etc/openvpn/
```

14. Se crearon los ficheros ipp.txt y openvpn-status-servidorvpn-udp-1194.log:

```
cd /etc/openvpn/
touch ipp.txt
touch openvpn-status-servidorvpn-udp-1194.log
```

15. Se aplicó contextos de lectura y escritura (openvpn_etc_rw_t) a los ficheros que contiene el directorio /etc/openvpn:

```
cd /etc/openvpn/
chcon -u system_u -r object_r -t openvpn_etc_rw_t
ipp.txt
chcon -u system_u -r object_r -t openvpn_etc_rw_t
openvpn-status-servidorvpn-udp-1194.log
```

Esto cambia los contextos a usuario de sistema (system_u), rol de objeto (object_r) y tipo configuración de OpenVPN de lectura y escritura (openvpn_etc_rw_t).

16. Se inició el servicio openvpn:

```
service openvpn Start
```

17. Para que el servicio de OpenVPN esté activo en el siguiente inicio del sistema, se utilizó el mandato **chkconfig** de la siguiente forma:

```
chkconfig openvpn on
```

4.1.2 Configuración del muro cortafuegos con Shorewall.

En este punto se realizó la configuración del firewall shorewall el cual ya fue instalado en el paso 3 del procedimiento de la configuración del Servidor OpenVPN.

Shorewall (Shoreline Firewall) es una robusta herramienta de alto nivel usada para la configuración de muros cortafuego, esta solo necesita configurar algunos datos en algunos ficheros para crear las reglas a través de iptables.

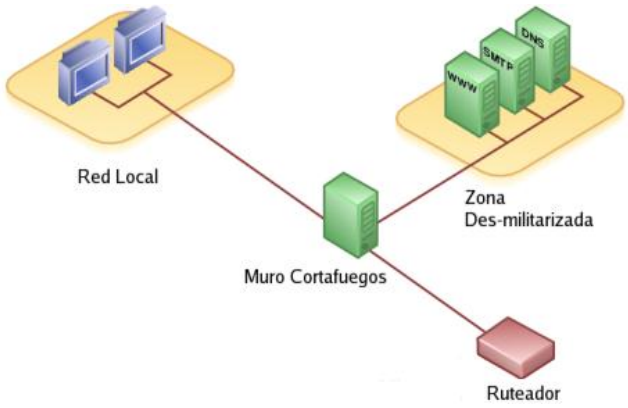


Figura 4.2 Grafica de las 3 interfaces del Cortafuego Shorewall

En la figura 4.2 se observan las 3 interfaces presentes en el firewall shorewall, las cuales se detallan a continuación:

- La interfaz de acceso a internet.
- La interfaz para acceso hacia una **DMZ**, tras la cual se podrán colocar servidores.
- La Interfaz para acceso hacia la **LAN**, en nuestro caso la PBX analizada.

Para la configuración del Cortafuegos Shorewall, se editaron los siguientes ficheros:

a) Fichero de configuración /etc/shorewall/zones

En este fichero, se registra las zonas de Internet (net), de Red Local (loc) y el túnel virtual **tun** para el VPN (rem). La zona **fw,** no se registró debido a que ya está presente en el fichero **/etc/shorewall.conf** como configuración predefinida. Se añade la zona **rem** para el túnel con el tipo **ipv4**, antes de la última línea.

```
#ZONE   DISPLAY         OPTIONS
fw      firewall
net     ipv4
loc     ipv4
# OpenVPN ----
rem     ipv4
#LAST LINE -- ADD YOUR ENTRIES BEFORE THIS ONE -- DO NOT
REMOVE
```

b) Fichero de configuración /etc/shorewall/interfaces

En éste fichero se establece las interfaces para las tres diferentes zonas, expuestas en la figura 3.2. Es decir se establecen las interfaces correspondientes a internet (net), al área local (loc) y al túnel (rem), y en todas se solicita se calcule automáticamente la dirección de transmisión (Broadcast):

```
#ZONE   INTERFACE       BROADCAST       OPTIONS
GATEWAY
net     ppp0            detect
loc     eth0            detect
# OpenVPN ----
rem     tun0            detect
#LAST LINE -- ADD YOUR ENTRIES BEFORE THIS ONE -- DO NOT
REMOVE
```

c) Fichero de configuración /etc/shorewall/policy

En este fichero se establece como se accederá desde una zona hacia otra y hacia la zona de Internet.

```
              #SOURCE         DEST    POLICY  LOG
LIMIT:BURST
loc             net     ACCEPT
fw              net     ACCEPT

# OpenVpn ----
fw              rem             ACCEPT
net             rem             ACCEPT
rem             fw              ACCEPT
rem             net             ACCEPT
rem             loc             ACCEPT
# ------------
net             all     ACCEPT    info
all             all     ACCEPT  info
#LAST LINE -- ADD YOUR ENTRIES BEFORE THIS ONE -- DO NOT REMOVE
```

Se añade la política para permitir el acceso de los miembros de la **VPN** hacia las zonas del cortafuego, la red pública y la red local. Todo definido antes de la última línea del fichero.

d) Fichero de configuración /etc/shorewall/rules

Debido a que todos los puertos están cerrados de modo predefinido, se habilita el puerto configurado para el túnel vpn.

Se permite las conexiones desde Internet hacia el firewall y viceversa por el puerto UDP [1194].

```
ACCEPT     net      fw       udp      1194
ACCEPT      fw       net      udp      1194
```

```
#LAST LINE -- ADD YOUR ENTRIES BEFORE THIS ONE -- DO NOT
REMOVE
```

e) Fichero de configuración /etc/shorewall/tunnels

En este fichero se establece el tipo de conexión. Ya que esto es configurado en un servidor VPN, se indica al cortafuego el puerto, la zona, y la red en que trabaja el Servidor.

```
#TYPE                    ZONE     GATEWAY          GATEWAY
#                                                  ZONE
openvpnserver:1194       rem      192.168.10.0/0
#LAST LINE -- ADD YOUR ENTRIES BEFORE THIS ONE -- DO NOT
REMOVE
```

Para aplicar los cambios, se reinicia el firewall **shorewall** con el mandato **service**: `service shorewall restart`

Con esto la interfaz del túnel se encuentra levantada en el servidor. Con el comando ifconfig, tal como muestra la figura 4.3, se verá adicional a las interfaces existentes, la interfaz túnel que la denominamos tun0:

```
tun0  Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-
00-00
      inet addr:10.10.0.1  P-t-P:10.10.0.2  Mask:255.255.255.255
      UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
      RX packets:73 errors:0 dropped:0 overruns:0 frame:0
      TX packets:96 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      RX bytes:24242 (23.6 KiB)  TX bytes:41235 (40.2 KiB)
```

Figura 4.3 Captura de la interfaz túnel levantada en el servidor Asterisk

4.1.3 Configuración del Cliente OpenVPN

Se levanta el otro lado del túnel (en el cliente), para esto se copia los archivos: ca.crt, cliente1.crt y cliente1.key, generados en el directorio /etc/openvpn/ keys del servidor, al directorio C:/Archivos de Programa/OpenVPN/config, del cliente Windows. Adicional se crea el archivo de configuración **cliente1-udp-1194.ovpn** dentro del mismo directorio, cuyo contenido es el siguiente:

```
client
dev tun
proto udp
remote 192.168.10.204 1194
float
resolv-retry infinite
nobind
persist-key
persist-tun
#-------------------SECCION DE LLAVES---------------------
ca C:\\archivos de programa\\OpenVPN\\config\\ca.crt
cert C\\archivos de programa\\OpenVPN\\config\\ca
server.crt
key C: \\archivos de programa\\OpenVPN\\config\\ca
server.key
ns-cert-type server
#-----------------------------------------------------------
comp-lzo
verb 3
ns-cert-type server
```

Con esto, en la maquina Windows aparece el icono para la conexión al servidor Asterisk a través del túnel, como se muestra en la figura 4.4. En el cual, al hacer clic derecho cuenta con la opción Connect.

Figura 4.4 Menú del cliente OpenVPN para efectuar la conexión al servidor VPN

Al hacer clic en la opción Connect, aparece una ventana de notificación que muestra los procesos de verificación e intento de conexión al servidor VPN, con la última línea que indica “Initialization sequence completed”.

Con esto, tal como se observa en la figura 4.5, el icono de notificación del cliente OpenVPN se muestra de color verde y muestra un valor aleatorio IP correspondiente a la red VPN.

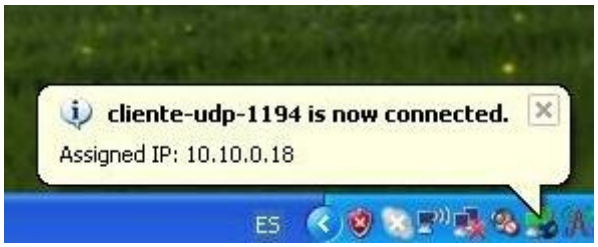


Figura 4.5 Mensaje de notificación al conectarse al servidor VPN

Esta IP puede ser vista también en el Command device de Windows con el comando ipconfig.

Este procedimiento se realizó en la maquina 200.126.13.215, en la que quedo establecida la ip de túnel 10.10.0.18. Se realizó la misma configuración de Cliente OpenVPN en la maquina 200.126.13.219 y se estableció una ip de túnel 10.10.06. La figura 4.6 muestra las ip de túneles asignados:

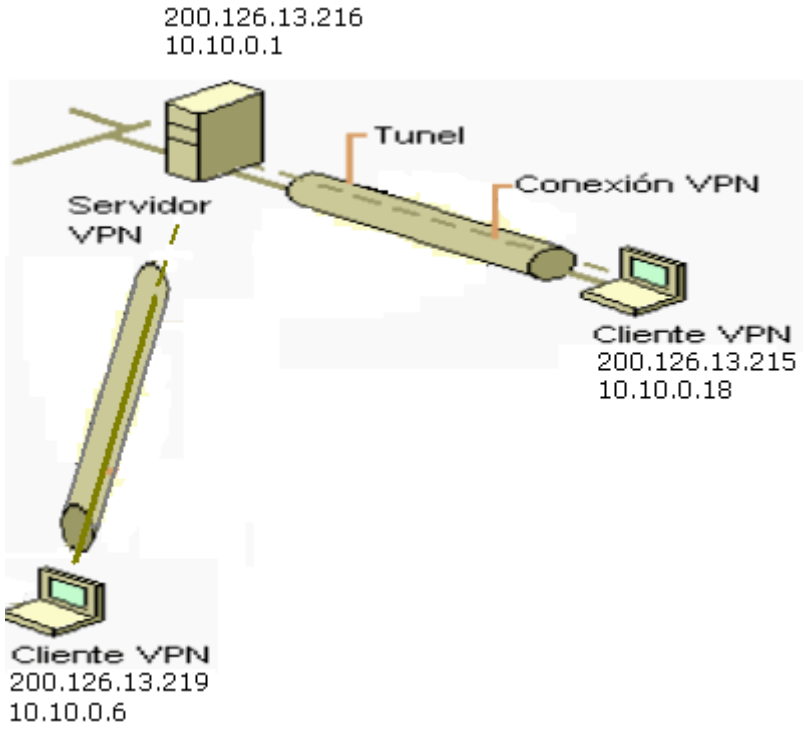


Figura 4.6 IP de túnel asignadas en la PBX

4.2 Captura de Llamada en un Sistema VoIP con VPN Levantada

Se realizó el proceso de captura de llamada entre el softphone de la maquina 200.126.13.219 y el softphone de la maquina 200.126.13.215 y

se comprueba que no es posible, tal como muestra la captura realizada en la figura 4.7:

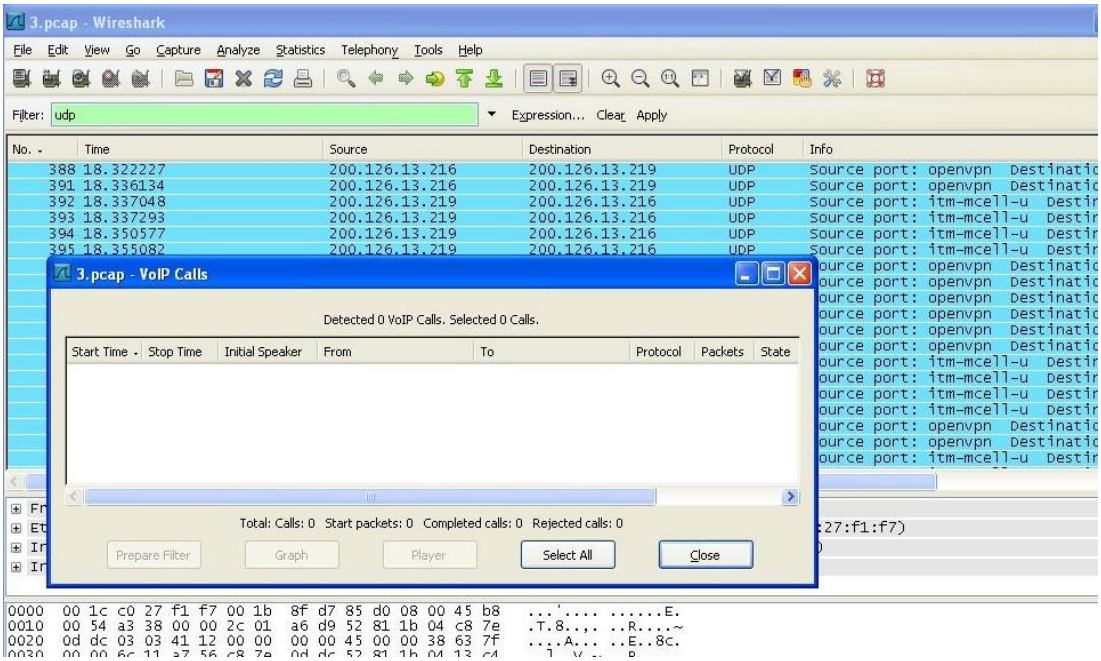


Figura 4.7 Captura realizada con Open VPN activo

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. Tal como se analizó durante todo este proyecto, es imprescindible tomar medidas de seguridad y prevención, durante la implementación de un sistema VoIP.

2. Existen muchas herramientas de alcance libre que pueden perjudicar al Sistema por lo que no es suficiente el correcto funcionamiento del mismo, sino que además considerar las medidas a tomar para no ser víctimas de un intruso mal intencionado.

3. Para implementar un exitoso Sistema VoIP, libre de amenazas de fraude, ataques DDoS, etc, deben ser considerados varios factores: firewalls, VPN.

4. Las herramientas usadas a lo largo de este proyecto, para realizar los diferentes ataques al sistema VoIP, son eficaces, es decir cumplen con las funciones por las cuales fueron implementadas.

5. El inconveniente que se dio en la implementación de cada una de las herramientas usadas, fue que, debido a que en primera instancia los paquetes de estas herramientas fueron descargados por separado, al compilarlos para su instalación, en todas las versiones encontradas de la web, se generaban errores, es decir no estaban correctamente implementados los

archivos .c y .h, contenidas en los paquetes. Esto fue solucionado con la implementación del sistema operativo backtrack en la maquina intrusa, ya que backtrack cuenta con estas herramientas listas para usar, y sin errores.

Recomendaciones

1. No hay mejor defensa para un Sistema VoIP, que evitar los ataques. Entre las medidas de prevención, a considerar se encuentran:

2. Abrir solo los puertos del servidor que sean necesarios, sobre los que funcionan las aplicaciones que procesa.

3. Los dispositivos de la red VoIP, como switch y router deben estar actualizados en términos de parches y actualizaciones de seguridad, tanto para los dispositivos existentes en la red, y los que se vayan agregando a medida que se extienda la red.

4. Hay que considerar que una pérdida de potencia puede provocar que la red se caiga, y en estos casos cuando un ataque de DoS se da, son difíciles de contrarrestar.

5. A pesar de la complejidad de emplear un firewall en VoIP por los múltiples requerimientos que necesita, es una medida excelente para proteger la red de voz.

6. Se recomienda prestar una especial atención a los accesos que se dan en la red.

7. Se recomienda usar las VPN como medida de seguridad a futuro en un sistema VoIP, ya que tal como se muestra en el presente proyecto son fáciles de implementar, sin necesidad de hacer cambios en la red.

ANEXOS

ANEXO A

Las siguientes descripciones contenidas en el anexo A y B de este proyecto han sido redactadas por el Sr. William López Jiménez en el manual: “VPN en servidor Linux y clientes Windows/Linux con OpenVPN + Shorewall” que se encuentra dividido en 2 partes en los enlaces: http://www.alcancelibre.org/staticpages/index.php/openvpn-clientes-win-linux-shorewall-P1 y http://www.alcancelibre.org/staticpages/index.php/openvpn-clientes-win-linux-shorewall-P2 respectivamente.

Descripción de Parámetros del archivo: /etc/openvpn/servidorvpn-udp-1194.conf

- **Port**: Especifica el puerto que será utilizado para que los clientes vpn puedan conectarse al servidor.

- **Proto**: tipo de protocolo que se empleará en a conexión a través de VPN.

- **dev**: Tipo de interfaz de conexión virtual que se utilizará el servidor openvpn.

- **ca**: Especifica la ubicación exacta del fichero de Autoridad Certificadora [.ca].

- **cert**: Especifica la ubicación del fichero [.crt] creado para el servidor.

- **key**: Especifica la ubicación de la llave [.key] creada para el servidor openvpn.

- **dh**: Ruta exacta del fichero [.pem] el cual contiene el formato de Diffie Hellman (requerido para **--tls-server** solamente).

- **server**: Se asigna el rango IP virtual que se utilizará en la red del túnel VPN.

- **Ifconfig-pool-persist**: Fichero en donde quedarán registrado las direcciones IP de los clientes que se encuentran conectados al servidor OpenVPN.

- **Keepalive 10 120**: Envía los paquetes que se manejan por la red una vez cada 10 segundos; y asuma que el acoplamiento es abajo si ninguna respuesta ocurre por 120 segundos.

- **comp-lzo**: Especifica los datos que recorren el túnel vpn será compactados durante la trasferencia de estos paquetes.

- **persist-key**: Esta opción soluciona el problema por llaves que persisten a través de los reajustes SIGUSR1, así que no necesitan ser re leídos.

- **Persist-tun**: Permite que no se cierre y re-abre los dispositivos TAP/TUN al correr los guiones up/down.

- **status**: fichero donde se almacenará los eventos y datos sobre la conexión del servidor [.log]

- **verb**: Nivel de información (default=1). Cada nivel demuestra todo el info de los niveles anteriores. Se recomienda el nivel 3 si usted desea un buen resumen de qué está sucediendo.

- **0 --**No muestra una salida excepto errores fatales. **1 to 4 –**Rango de uso normal. **5 --**Salida **R** y **W** caracteres en la consola par los paquetes de lectura y escritura, mayúsculas es usada por paquetes TCP/UDP minúsculas es usada para paquetes TUN/TAP.

ANEXO B

Descripción de Parámetros del archivo: cliente1-udp-1194.ovpn

- **Client:** Especifica el tipo de configuración, en este caso tipo cliente OpenVPN.

- **Port:** Especifica el puerto que será utilizado para que los clientes VPN puedan conectarse al servidor.

- **Proto:** tipo de protocolo que se empleará en a conexión a través de VPN

- **dev:** Tipo de interfaz de conexión virtual que se utilizará el servidor openvpn.

- **remote:** Host remoto o dirección IP en el cliente, el cual especifica al servidor OpenVPN. El cliente OpenVPN puede tratar de conectar al servidor con host: port en el orden especificado de las opciones de la opción --remote.

- **float:** Este le dice a OpenVPN aceptar los paquetes autenticados de cualquier dirección, no solamente la dirección cuál fue especificado en la opción --remote.

- **resolv-retry:** Si la resolución del nombre del anfitrión (hostname) falla para -- remote, la resolución antes de fallar hace una re-comprobación de n segundos.
- **nobind:** No agrega bind a la dirección local y al puerto.

- **ca:** Especifica la ubicación exacta del fichero de Autoridad Certificadora [.ca].

- **cert:** Especifica la ubicación del fichero [.crt] creado para el servidor.

- **key:** Especifica la ubicación de la llave [.key] creada para el servidor OpenVPN.

- **remote:** Especifica el dominio o IP del servidor así como el puerto que escuchara las peticiones para servicio VPN.

- **comp-lzo:** Especifica los datos que recorren el túnel VPN será compactados durante la trasferencia de estos paquetes.

- **persist-key:** Esta opción soluciona el problema por llaves que persisten a través de los reajustes SIGUSR1, así que no necesitan ser re leídos.

- **Persist-tun:** Permite que no se cierre y re-abre los dispositivos TAP/TUN al correr los guiones up/down.

- **verb:** Nivel de información (default=1). Cada nivel demuestra toda la Información de los niveles anteriores. Se recomienda el nivel 3 si usted desea un buen resumen de qué está sucediendo.

- **0 --** No muestra una salida excepto errores fatales. 1 to 4 –Rango de uso normal. **5 --**Salida R y W caracteres en la consola par los

paquetes de lectura y escritura, mayúsculas es usada por paquetes TCP/UDP minúsculas es usada para paquetes TUN/TAP.

BIBLIOGRAFÍA

[1] M4inFox del Bl4ck P0rtal, Sniffers, que son y cómo funcionan, http://www.bl4ck-p0rtal.org/foro/index.php?topic=7947.0, 2009.

[2] Grupo Solutec, Asterisk – Central Telefónica PBX http://www.solutecperu.com/spsac/asterisk-central-telefonica-pbx, 2009

[3] Suarez Eduardo y Villavicencio Nixon, WIRESHARK, http://www.scribd.com/doc/25350978/Manual-Rapido-de-WireShark, 2009.

[4] UNIVERSIDAD CENTRAL DE VENEZUELA RECTORADO DIRECCION DE TECNOLOGIA DE INFORMACION Y COMUNICACIONES, Manual de Usuario WireShark, http://ftp.ucv.ve/Documentos/Wireshark/Manual.doc, 2008.

[5] Ghislain Ndeuchi, Wireshark sniffing http://www.unappel.ch/public/100119-wireshark-xlite/, 2010.

[6] Gutiérrez Gil Roberto, Seguridad en VoIP: Ataques, Amenazas y Riesgos, http://www.scribd.com/doc/3160833/capitulo9-teoria-voladura-de-rocas, 2008.

[7] Endler & Collier, Hacking Exposed VoIP: Voice Over IP Security Secrets & Solutions - CHAPTER 3: ENUMERATING A VOIP NETWORK, http://www.hackingvoip.com/presentations/sample_chapter3_hacking_voip.pdf, 2006.

[8] usuario: rachelle090202 en monografías.com, Seguridad Para Sistemas Voz sobre IP, http://www.monografias.com/trabajos26/voz-sobre-ip/voz-sobre-ip2.shtml, 2006.

[9] Ibarra Corretge Saúl, Introducción a VoIP y Asterisk, http://www.slideshare.net/saghul/introduccin-a-asterisk-297883, 2007.

[10] Gorka Gorrotxategi – Iñaki Baz, Voz sobre IP y Asterisk, http://documentacion.irontec.com/cursoAsteriskVozIP-1-introduccion-SIP.pdf, 2006.

[11] Bytecoders, Husmeando en una red VoIP, http://bytecoders.homelinux.com/content/husmeando-en-una-red-voip.html, 2008.

[12] Luca Leone, Nicola Mondinelli, Pierpaolo Palazzoli, Matteo Valenza, VoIP Security Testing and Solutions, http://snortattack.org/docs/voip_en.pdf, 2007.

[13] Alvares Marañon Gonzalo, Denegación de servicio, http://www.iec.csic.es/CRIPTonOMICon/java/denegacion.html, 1997.

[14] Ibarra Corretge Saúl, (in)seguridad en VoIP, http://comunidad.asterisk-es.org/images/Inseguridad_en_voip.pdf, 2000

[15] López Jiménez William, VPN en servidor Linux y clientes Windows/Linux con OpenVPN + Shorewall [Parte 1], http://www.alcancelibre.org/staticpages/index.php/openvpn-clientes-win-linux-shorewall-P1, 2006.

[16] López Jiménez William, VPN en servidor Linux y clientes Windows/Linux con OpenVPN + Shorewall [Parte 2], http://www.alcancelibre.org/staticpages/index.php/openvpn-clientes-win-linux-shorewall-P2, 2007.

[17] Barrios Dueñas Joel, Cómo configurar un muro cortafuegos con Shorewall y tres interfaces de red, http://www.alcancelibre.org/staticpages/index.php/como-shorewall-3-interfaces-red, 1999.

[18] admin Libro VozToVoice, OpenVPN y Asterisk - Instalación y configuración, http://www.voztovoice.org/?q=node/103, 2008.

[19] Salgado Eliana, Redes Virtuales Privadas (VPN): Una elección de conectividad segura y eficiente, http://www.channelplanet.com/?idcategoria=10995, 2003.