PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

MAESTRÍA EN REDES DE COMUNICACIÓN

INFORME FINAL CASO DE ESTUDIO PARA UNIDAD DE TITULACIÓN ESPECIAL

TEMA:

"Esquema de seguridad para una central VoIP, en software libre en su implementación Elastix"

Priscila Maldonado Mendieta

Quito – 2016

AUTORÍA

Yo, ***Priscila Galina Maldonado Mendieta***, portador de la cédula de ciudadanía No. ***1104601636***, declaro bajo juramento que la presente investigación es de total responsabilidad del autor, y que se ha respetado las diferentes fuentes de información realizando las citas correspondientes. Esta investigación no contiene plagio alguno y es resultado de un trabajo serio desarrollado en su totalidad por mi persona.

_____

Priscila Galina Maldonado Mendieta.

Contenido

Figuras

Tablas.

1. Introducción

La tecnología VoIP posee gran flexibilidad y muchas características de valor agregado con relación a la telefonía convencional y su infraestructura, es por eso que ha tomado gran importancia en la actualidad y con ello ha aumentado la utilización de esta tecnología.

Debido a la gran utilización de VoIP, en estos tiempos se han hecho más evidentes las vulnerabilidades que tienen estos servicios, los problemas que presentan las redes VoIP, son en gran medida similares a los problemas de seguridad de las redes de datos. Estas vulnerabilidades implican desde un escaneo de dispositivos SIP, denegación de servicio, intercepción de llamadas ilegítimamente hasta generar altos costos a la víctima, debido al uso fraudulento de llamadas no autorizadas.

Este trabajo presenta un estudio de los protocolos de VoIP (SIP, RTP, SDP) para de esta forma comprender las falencias que poseen y de las cuales nos convierte en víctimas frente a un ataque, además de un estudio de los protocolos de seguridad utilizados en este tipo de redes como: SRTP, TLS, lo que permitirá determinar cuáles son sus principales ventajas y así evaluar cuál sería el más óptimo al momento de implementar una red VoIP.

En segunda instancia se pretende desarrollar un método para implementar redes seguras de VoIP, partiendo de un análisis de vulnerabilidades e implementación de contramedidas basadas en un SBC (Controlador de Sesión de Borde) de código abierto que permita mitigar el impacto que producen los ataques a este tipo de redes.

Es por ello, que para el presente caso de estudio se realiza una implementación práctica de laboratorio utilizando la central Elastix y el sistema BLOX SBC como controlador de sesión de borde, una vez establecida la aplicación práctica y mediante las diferentes configuraciones se realizará distintas pruebas de seguridad y se presentarán los resultados.

2. Justificación

A lo largo del tiempo las comunicaciones han ido evolucionando y sus requerimientos de seguridad, en primera instancia se toma como medida la utilidad de un firewall el cual es un elemento que brinda seguridad a una red cuando se interconecta con otras, permitiendo el tráfico de salida y bloqueando el de entrada no autorizado.

El firewall es un elemento importante en redes de datos, pero no es suficiente para la seguridad en redes de VoIP. Las conexiones de voz sobre IP son incompatibles con NAT de manera que algunas amenazas pueden atravesar el firewall de forma transparente debido a que se basan en NAT. Una posibilidad es abrir excepciones en el NAT en el firewall para la voz sobre IP, pero no es una solución porque compromete la seguridad y no protege contra ataques de denegación de servicio e intrusión.

El control de intrusión no sólo se maneja a nivel de red en la cual un firewall puede realizar esta función, sino principalmente a nivel de aplicación, con el fin de controlar que las llamadas sean autorizadas, para evitar ataques e intrusiones. El control se complica ya que las sesiones de voz sobre IP se crean de forma aleatoria a medida que se establecen llamadas.

Para enfrentar estos riesgos se requiere de un nuevo elemento, que forme parte activa en las sesiones de voz sobre IP para que se establezcan de forma legítima, segura y fiable. Por todo lo antes mencionado en el presente caso de estudio se pretende realizar una implementación con SBC (Controlador de sesión de borde) de código abierto ya que no es un sistema muy difundido.

3. Antecedentes

La primera central telefónica se estableció en 1877 en la ciudad de Boston [13], de ahí hasta la actualidad se están desarrollando programas que realizan las funciones de una central telefónica, ya sea basado en software libre o sistemas pagos. Sin embargo como en toda red se encuentra expuesta a ataques informáticos los cuales pueden producir que el atacante mediante escaneo tenga conocimiento de la red interna de una empresa, exista interrupción en el servicio y hasta genere problemas en facturación por el uso fraudulento de la red VoIP.

Según estudio de ataques a sistemas VoIP planteado por Angelos Keromytis presenta vulnerabilidades basadas en los efectos, obteniendo como resultado el 48% debido a ataques de denegación de servicio y la segunda más grave con el 20% es cuando el atacante permite controlar un dispositivo remotamente, en tercer lugar presenta el estudio con un 13% de ataques por acceso a los servicios, luego con un 12% debido a ataques al usuario donde incluye XSS la cual es una de la técnica de hacking a nivel de aplicación que aprovecha las vulnerabilidades en el código de una aplicación web permitiendo a un atacante obtener datos del usuario y el eavesdropping de acuerdo al estudio realizado por Angelos está inmerso en esta categoría ya que es un ataque de usuario, al escuchar las conversaciones que realiza el mismo, y por último con un 8% cuando el atacante accede a los datos. [3]

Por otro lado Angelos plantea en base a las características comunes que presentan los ataques ocupando el primer lugar con un 57% por denegación de servicio, en segundo lugar con 21% Eavesdropping, 19% amenazas sociales y 4% abuso de servicio.

Es por ello que se requiere dar seguridad a nivel de central en redes VoIP, y de esta formas mitigar en gran medida las vulnerabilidades existentes.

4. Objetivos

Objetivo General:

Implementar un esquema de seguridad en un ambiente de laboratorio, que permita la mayor mitigación de los ataques informáticos a los que se encuentran expuestas las centrales VoIP.

Objetivos Específicos:

1. Estudiar los componentes de VoIP que se van a utilizar en el implementación.
2. Describir algunas herramientas que permitan la explotación de vulnerabilidades sobre la central de VoIP implementada.
3. Presentar las vulnerabilidades existentes en la implementación a través de las herramientas empleadas.
4. Describir las funciones que desempeña un SBC (Session Border Controller) en las redes VoIP
5. Implementar controles de seguridad a través de BLOX como SBC (Session Border Controller) en open source para garantizar la mayor mitigación a los ataques informáticos a los que se encuentran expuestas la central VoIP.

5. Desarrollo Caso de Estudio.

5.1 Introducción

5.1.1 Protocolo de inicio de sesión (SIP)

SIP fue creado por el **IETF MMUSIC Working Group,** para crear una arquitectura multimedia más completa. Es un protocolo de control y señalización utilizado para telefonía IP. SIP es un protocolo, abierto, escalable y ampliamente soportado, que no depende de ningún fabricante y se integra fácilmente con otros protocolos y aplicaciones que lo han convertido en un estándar de la telefonía IP que funciona tanto sobre UDP como TCP. [27]

SIP es un protocolo de señalización porque maneja establecimiento, control y terminación de las sesiones de comunicación. Las sesiones incluyen: llamadas telefónicas, transferencias de datos multimedia, y conferencias en tiempo real. [27]

A continuación se presenta los protocolos que interactúan con la comunicación SIP a detalle:

- **RTP -** Real Time Transport Protocol, cuando SIP establece la llamada se produce un intercambio de paquetes RTP que transportan realmente el contenido de voz y video.
- **SDP –** Session Description Protocol, los mensajes del protocolo SDP son transportados por el protocolo SIP y son utilizados para la negociación de las capacidades de los participantes (códecs, versión del protocolo, etc)

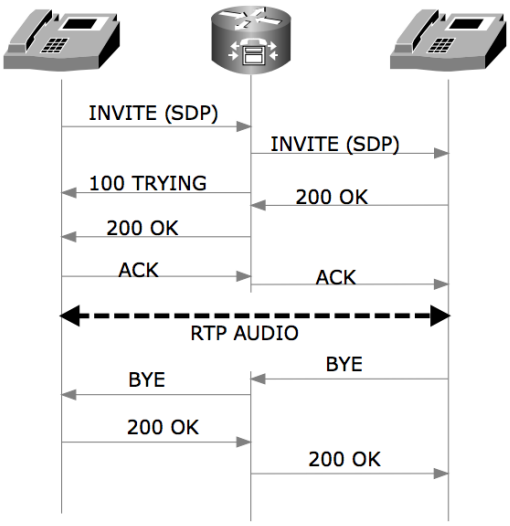


Figura 5. 1 Llamada SIP [7]

Se puede observar en la Figura 5.1 el intercambio de mensajes para establecer una llamada mediante terminales SIP.

Dentro de una red SIP se encuentran los siguientes componentes: Agentes de Usuario (UA) y servidores. Entre los agentes de usuario, se encuentran **agentes de usuario clientes (UAC)** que son los que inician las peticiones de llamada y los **agentes de usuario servidor (UAS)** que reciben las peticiones del UAC. [27]

La estructura de SIP se encuentra basada en protocolos SMTP y HTTP por su similitud. Por ello en la infraestructura SIP los clientes son identificados por direcciones definidas muy similares a las direcciones de correo: **user@host** ó **user@dominio.** Por otro lado los SIP responses son en códigos de 3 dígitos tomando similitud a HTTP: **200 ok, 404 not found.** [27]

Los mensajes SIP y su función se resumen en la Tabla 5.1.

Mensaje	Explicación
INVITE	Permite invitar un usuario para participar en una sesión. Se hace referencia al **RFC3261.** [16]
ACK	Utilizado para responder a un mensaje de estado SIP, confirma el establecimiento de una sesión. Se hace referencia al **RFC3261.**
OPTION	Solicita información sobre las capacidades de un servidor. Este mensaje se envía antes de establecer una llamada. Se hace referencia al **RFC3261.**
BYE	Indica la terminación de una sesión. Se hace referencia al **RFC3261.**
CANCEL	Una conexión puede interrumpirse antes de establecerse la llamada. Se hace referencia al **RFC3261.**
REGISTER	Registra al agente de usuario. El registro del terminal se realiza con parámetros que lo identifican y permiten la comunicación. Se hace referencia al **RFC3261.**
INFO	Son utilizados para intercambiar información entre los terminales.
PRACK	Realiza la misma tarea que un mensaje ACK, pero es una respuesta provisional.
SUBSCRIBE	Es utilizado por un terminal para establecer una sesión de intercambio de datos estadísticos y de actualización de estados. Se hace referencia al **RFC3265** [3]
NOTIFY	Permite el intercambio de información del estado de un terminal que ya es parte de una sesión de intercambio de datos estadísticos y de actualización

	de datos.

Tabla 5. 1 Mensajes SIP [6]

En la Tabla 5.2 se presenta un listado de códigos de respuesta a las peticiones SIP los cuales hacen referencia en el **RFC2543.** [16]

Código	Significado	Ejemplo
1xx	Mensajes temporales	100 TRYING (intentando) 180 RINGING (timbrado)
2xx	Respuestas de éxito.	200 OK (llamada establecida exitosamente)
3xx	Respuestas de redirección.	301 MOVED PERMANENTLY (indica que el terminal cambio de dirección IP)
4xx	Respuesta de fallo de método.	401 UNAUTHORIZED (indica un fallo de autenticación)
5xx	Respuestas de fallos de servidor.	500 INTERNAL ERROR (indica error interno del servidor)
6xx	Respuestas de fallos globales.	600 BUSY EVERYWHERE (indica que el sistema está completamente ocupado.

Tabla 5. 2 Códigos de respuesta a las peticiones SIP [16]

En la Figura 5.2 mediante WIRESHARK se muestra como se realiza el intercambio de mensajes SIP como: **INVITE, ACK, BYE** y códigos de respuesta **401 UNAUTHORIZED, 100 TRYING, 180 RINGING, 200 OK** mediante una llamada.

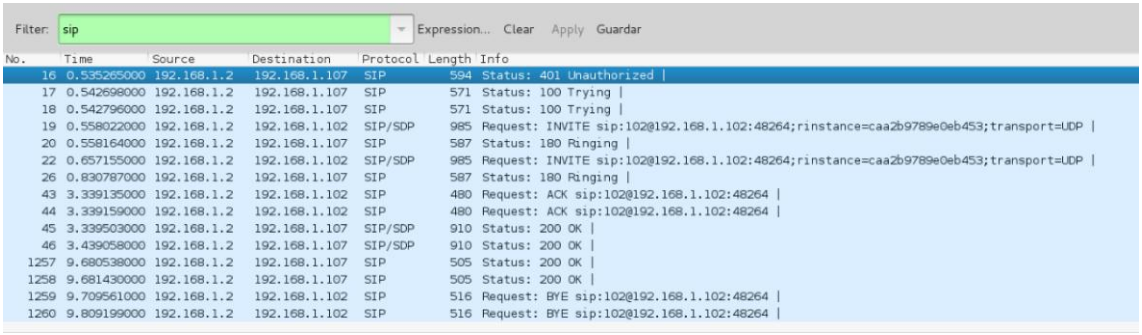


Figura 5. 2 Mensajes SIP.

5.1.2 Protocolo SDP

El protocolo de descripción de sesión (SDP o Session Description Protocol) es encapsulado por los mensajes SIP y sirve para describir sesiones en tiempo real. Se encuentra definido en el **RFC 2327**. [21]
SDP fue diseñado para anunciar información necesaria para los participantes. Actualmente su uso está en la negociación de las capacidades de una sesión multimedia para telefonía IP. [6]

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

Cuando los mensajes SDP son interceptados, permiten que el atacante conozca muchas características de los terminales como: códecs, número de teléfono, puertos y protocolo utilizado para transportar la voz e información de conexión. Una vez que se obtenga la dirección y el número de puerto donde se enviarán los datos multimedia, se puede realizar ataques directos a los datos de voz.

En la Figura 5.3 muestra la interceptación de los mensajes SDP encapsulados en SIP, mediante la terminal 102 a través de la petición INVITE.

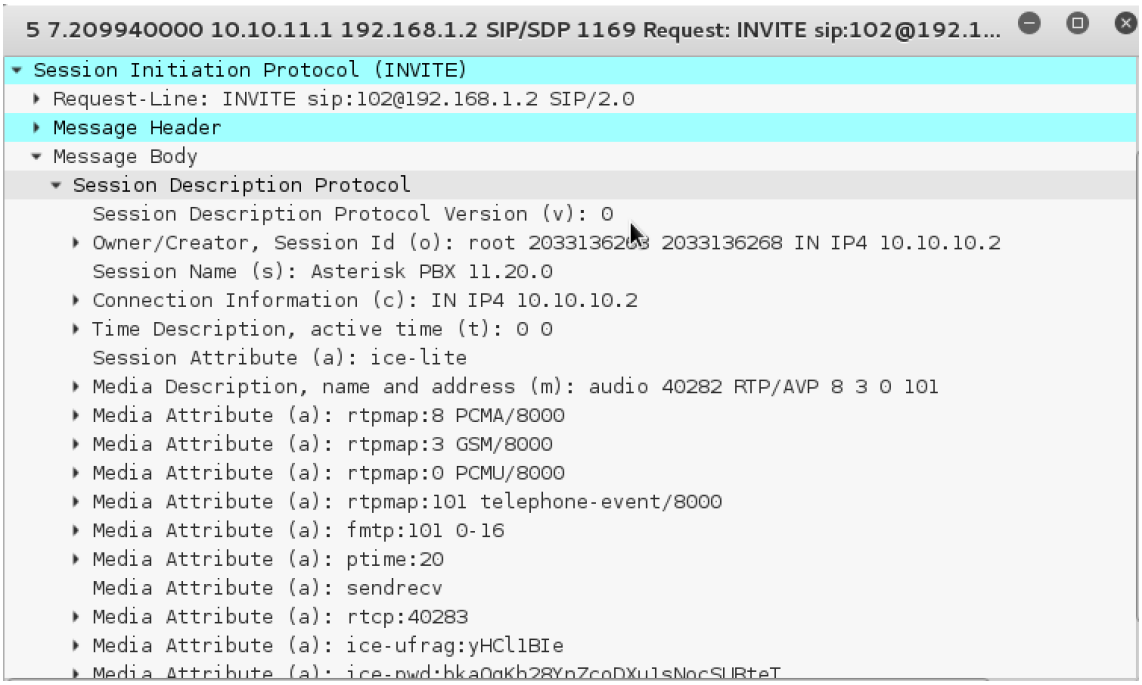


Figura 5. 3 Mensaje INVITE con SDP encapsulado.

5.1.3 Protocolo RTP

El protocolo de transporte de tiempo real, RTP por sus siglas en inglés Real-time Transport Protocol se encarga de transportar los datos de servicios de tiempo real como aplicaciones de audio y video, asegurando la calidad de servicio (QoS) de los mismos.

RTP se encuentra definido en el **RFC 3550** [15]. Entre las funciones se encuentran: la identificación del tipo de datos, la numeración secuencial de los paquetes, la medición de tiempo y el reporte de la calidad de comunicación.

RTP trabaja en la capa de transporte sobre UDP. Sin embargo RTP cuenta con algunas características que UDP no tiene, como un sistema de checksum para detección de errores y secuenciación de paquetes, permitiendo que la aplicación pueda reordenar los paquetes que no se han recibido en orden.

Una característica importante de RTP es que, gracias a un protocolo conocido como RTP-HC (Real-Time Protocol - Header Compression), permite la compresión del encabezado del paquete disminuyendo su tamaño, permitiendo reducir los 40 bytes de encabezado en RTP/UDP/IP de 2 a 5 bytes, eliminando los encabezados que se repiten en todos los paquetes, mejorando considerablemente el desempeño de la red. Además RTP utiliza los protocolos RTCP y SDP. RTCP es el protocolo de control de RTP y utiliza el encabezado

del RTP, además ocupa el campo de carga útil para enviar estadísticas. Por otro lado RTP, utiliza SDP para intercambiar datos de descripción de la llamada. [6]

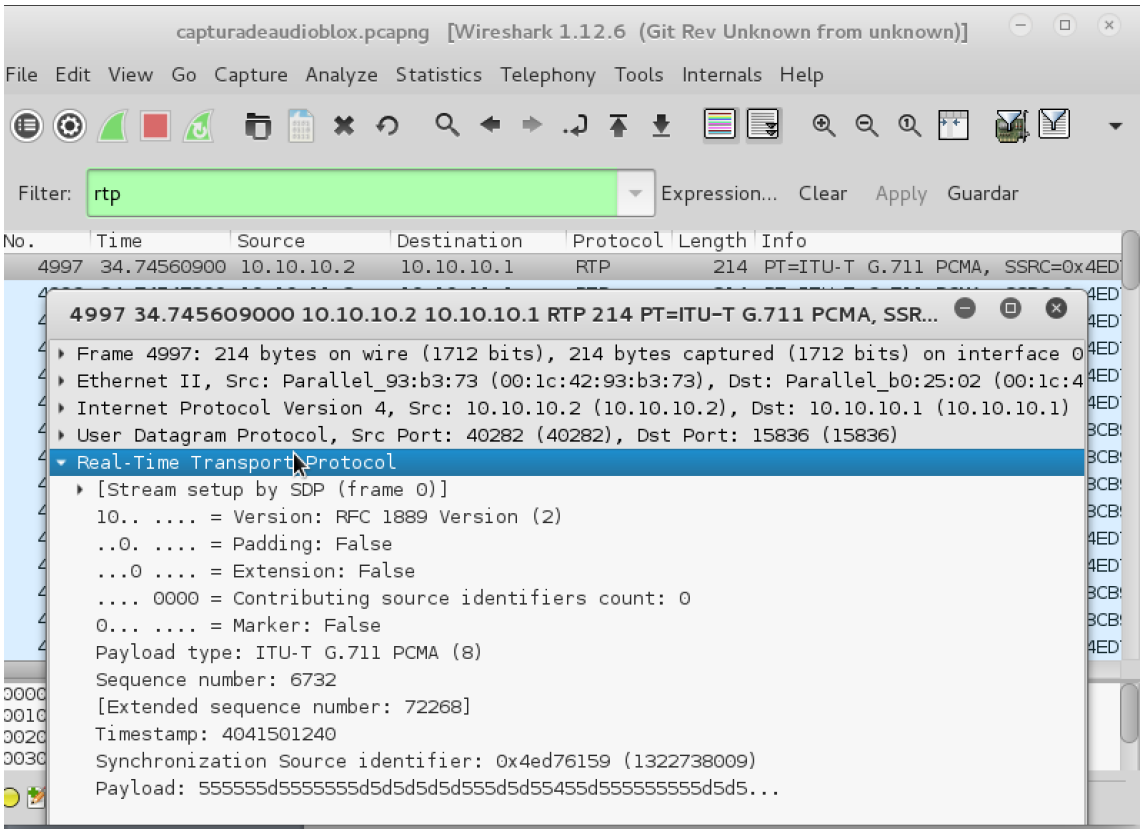


Figura 5. 4 Mensaje RTP

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

El protocolo de transporte de tiempo real seguro (Secure Real-time Transport Protocol o SRTP) provee seguridad a los protocolos RTP y RTCP. Este protocolo está definido en el RFC 3711 [20]. SRTP se caracteriza por lograr buenos resultados con poco incremento del tamaño del paquete transmitido, debido a, que la encriptación no produce aumento en la carga útil del mensaje. SRTP es flexible y no depende de ninguna administración específica de llaves.

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

- **NULL:** el modo NULL es utilizado cuando sólo se necesita autenticación, por lo tanto la carga útil no va cifrada.

- **AES** Segmented Integer Counter Mode: conocido como AES-CM, no incrementa de manera considerable el tamaño para la carga útil encriptada, el mismo que es 220 bytes.
- **AES-f8:** es un modo utilizado en UMTS (Universal Mobile Telecommunications System, redes 3G). A diferencia de AES-CM, cambia en la retroalimentación de la salida y la función inicial de encriptación. [6]

En la Figura 5.5 se muestra el establecimiento del protocolo SRTP.

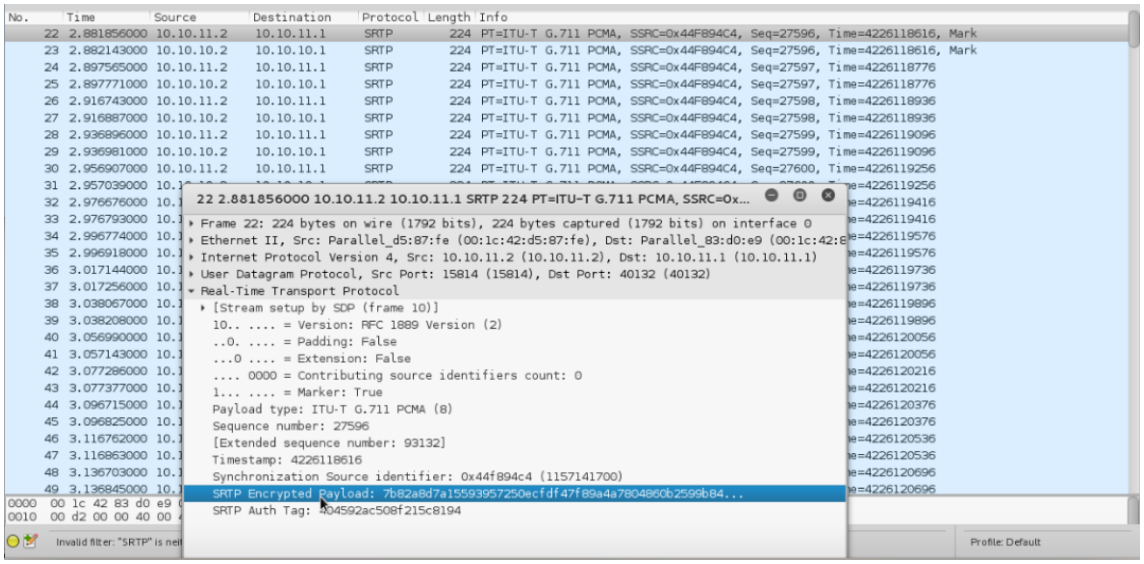


Figura 5. 5 Mensaje SRTP

En la Figura 5.6 se observa mediante Wireshark y su módulo VoIP Calls el aseguramiento de una llamada a través de SRTP.

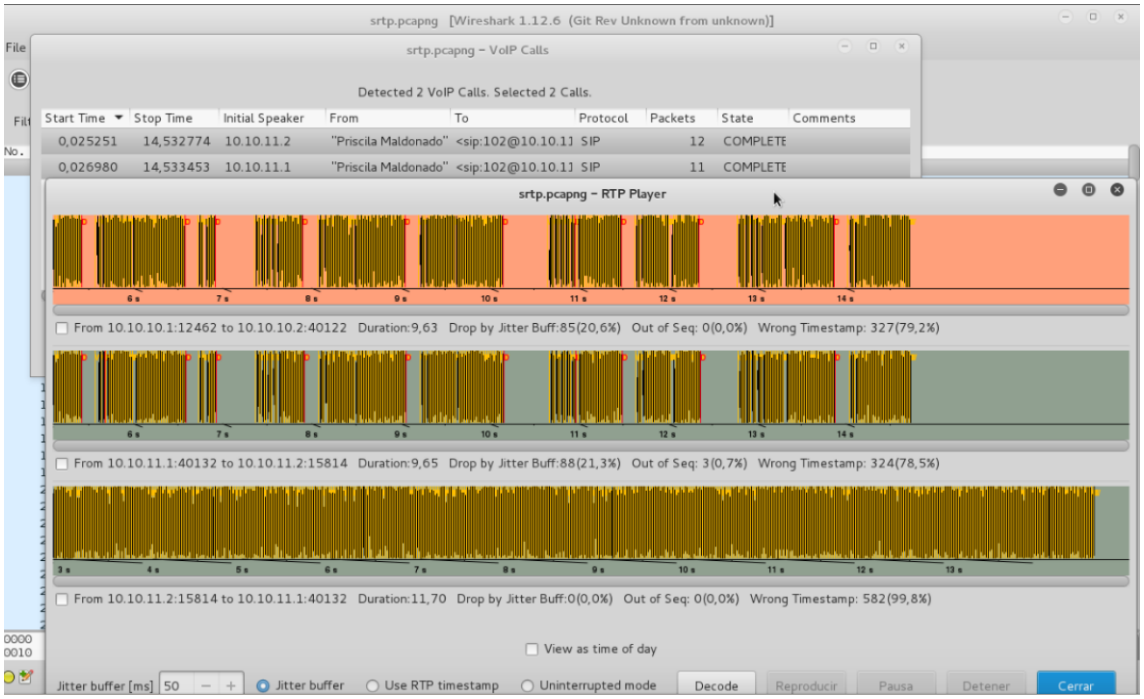


Figura 5. 6 Llamada VoIP con seguridad SRTP

5.1.5 Seguridad de la capa de transporte (TLS).

Transport Layer Security (TLS) se encuentra basado en Secure Sockets Layer SSL en español capa de puertos seguros.

TLS establece comunicaciones seguras por encima de la capa de transporte, porque funciona sobre TCP. TLS ofrece seguridad punto a punto, es decir si la comunicación pasa por varios dispositivos, cada uno de los dispositivos deben utilizar TLS, de no ser así la información será transmitida sin encriptación.

TLS utiliza una infraestructura pública de llaves (con sus siglas en inglés PKI, Public Key Infraestructure). PKI es el conjunto de dispositivos que permiten a un usuario firmar digitalmente mensajes utilizando clave privada, de manera que otro usuario pueda verificar dicha firma utilizando la clave pública del usuario, la cual se encuentra contenida en el certificado que ha sido emitido por una autoridad certificadora (CA) de la PKI. [6]

El establecimiento de la comunicación consta de 3 etapas:

- **ETAPA 1:** Al empezar la comunicación los extremos negocian el algoritmo de cifrado que van a ejecutar.

- **ETAPA 2:** Se lleva a cabo el intercambio de llaves y acuerda los algoritmos de firma.
- **ETAPA 3:** Al establecerse la comunicación, se emplea tanto el algoritmo de clave simétrica para cifrar como el algoritmo de firma. [6]

SIP utiliza TLS para proteger sus encabezados, sin embargo, al utilizar TLS este a su vez utiliza TCP, por tanto TLS no brinda seguridad al tráfico RTP debido a que RTP funciona sobre UDP.

5.1.6 Central Telefónica IP Privada (Private Branch Exchange, IP –PBX)

5.1.6.1 Elastix

Elastix fue creado y es permanentemente desarrollado por Palo Santo Solutions, iniciándose así como una interfaz de reporte para llamadas de Asterisk, fue liberado en Marzo del 2006, y ha evolucionado hasta convertirse en una distribución basada en Asterisk. [25]

Elastix contiene un conjunto de características y funciones relacionadas con los servicios que ofrece entre los principales se mencionan los siguientes: Telefonía IP, Conferencias, Servidor de Correo y Servidor de Mensajería Instantánea. [25]

En la Tabla 5.3 se menciona a detalle algunas características y funciones:

GENERAL	PBX
• Monitor de Recursos del Sistema. • Configurador de parámetros • Control de apagado/re-encendido de la central vía web. • Control de Acceso a la Interfaz, basado en ACLs • Administración Centralizada de Actualizaciones • Soporte para backup/restore a través de Web	• Grabación de llamadas • Correo de Voz • Codecs soportados: ADPCM, G.711 (A-Law & µ-Law), G.722, G.723.1 (pass through), G.726, G.728, G.729, GSM, iLBC (opcional) entre otros. • IVR Configurable y Flexible • Servidor DHCP para asignación dinámica de IPS. • Panel de Operador basado en Web • Reporte de detalle de llamadas

● Soporte para temas o skins ● Soporte para configuración de fechas en el servidor, hora y zonas horarias	(CDR) ● Reportes de uso de canales ● Asterisk en tiempo real ● Centro de Conferencias con Salas Virtuales. ● Soporte para protocolos SIP e IAX, entre otros ● Correo de voz-a-email ● Identificación de llamadas (Caller ID) ● Troncalización ● Rutas entrantes y salientes. ● Administración centralizada vía Web ● Cliente de Email basado en Web ● Administración de Lista de Email ● Soporte para cuotas ● Soporte Antispam ● Basado en Postfix para un alto volumen de correos

Tabla 5. 3 Características de la central Elastix. [25]

Elastix es una herramienta empresarial de código abierto distribuida bajo licencia GLPv2, es decir no tiene costo relacionado con el licenciamiento o sus funciones. El uso de Elastix está sujeto a las condiciones descritas en la licencia ya sea para uso comercial o personal.

Las versiones de Elastix son versiones completas sin limitaciones de uso o características, ni la adición de módulos; ni la adición de usuarios.

Por otro lado se presenta los requerimientos mínimos de hardware como muestra la Tabla 5.4

Propósito	Número de canales	Mínimo Recomendado
Sistema como hobby	No más de 5	400 MHz x86, 256 MB RAM
SOHO (Oficina pequeña / oficina en casa)	5 a 10	1GHz x86, 512 MB RAM
Empresa Pequeña Empresa	10 a 25	3GHz X86, 1GB RAM
Empresa mediana/grande	Más de 25	Dual CPUs o múltiples servidores

Tabla 5. 4 Requerimientos mínimos de hardware. [12]

5.1.7 Códecs

Un códec es la abreviatura de codificador / decodificador. Básicamente es un algoritmo capaz de transformar una señal o flujo de datos (stream). Los códecs pueden codificar el flujo de datos y recuperarlo del mismo modo. Su uso está muy extendido para la codificación de señales de audio y video dentro de un formato [11].

Los códecs dependen de 3 factores fundamentales:

- Recursos de CPU
- Consumo de ancho de banda
- Calidad de la comunicación

A continuación en la Tabla 5.5 se muestra los diferentes códecs de voz utilizados en VoIP.

Como se observa el códec G.711 requiere mayor ancho de banda en comparación a los otros códecs, obteniendo en Ethernet un ancho de banda de 87.2 kbps, sin embargo es gratuito, y ha permitido la implementación para este caso de estudio.

Información de códec				Cálculos de ancho de banda					
Velocidad de bits y códec (kbps)	**Ejemplo de tamaño del códec (bytes)**	**Ejemplo de intervalo del códec (ms)**	**Mean Opinion Score (MOS)**	**Tamaño de la carga útil de voz (bytes)**	**Tamaño de la carga útil de voz (ms)**	**Paquetes por segundo (PPS)**	**Ancho de banda MP o FRF.12 (kbps)**	**Ancho de banda c/cRTP MP o FRF.12 (kbps)**	**Ancho de banda Ethernet (kbps)**
G.711 (64 kbps)	80 bytes	10 ms	4,1	160 bytes	20 ms	50	82,8 kbps	67,6 kbps	87,2 kbps
G.729 (8 kbps)	10 bytes	10 ms	3,92	20 bytes	20 ms	50	26,8 kbps	11,6 kbps	31,2 kbps
G.723.1 (6.3 kbps)	24 bytes	30 ms	3,9	24 bytes	30 ms	34	18,9 kbps	8,8 kbps	21,9 kbps
G.723.1 (5.3 kbps)	20 bytes	30 ms	3,8	20 bytes	30 ms	34	17,9 kbps	7,7 kbps	20,8 kbps
G.726 (32 kbps)	20 bytes	5 ms	3,85	80 bytes	20 ms	50	50,8 kbps	35,6 kbps	55,2 kbps
G.726 (24 kbps)	15 bytes	5 ms		60 bytes	20 ms	50	42,8 kbps	27,6 kbps	47,2 kbps
G.728 (16 kbps)	10 bytes	5 ms	3,61	60 bytes	30 ms	34	28,5 kbps	18,4 kbps	31,5 kbps

Tabla 5. 5 Códecs de Voz. [7]

5.1.8 Terminales

Los endpoints o terminales también conocidos como cliente o agente, puede ser un teléfono IP o softphones. Los softphones son una aplicación que se ejecuta en un computador o smartphone.

Un endpoint debe soportar al menos un estándar de señalización entre (SIP, H323, IAX2) que establece y finaliza las llamadas, cuenta con un software de interacción con el usuario el cual provee de funcionalidades además de la telefonía como: buzón de voz, autenticación, conferencias.

Entre las funciones que poseen los endpoints se encuentra la codificación y decodificación tanto de los paquetes de voz transmitidos y recibidos, el soporte de códecs permitirá determinar la calidad de la voz en la red VoIP.

Por otro lado los teléfonos IP y softphones se pueden convertir en herramientas utilizadas por un atacante para tener acceso a una red VoIP, sin embargo el obtener acceso no involucra que la red VoIP deje de funcionar, a pesar de que son dispositivos menos controlables, debido a, la movilidad del usuario. Los atacantes pueden acceder a la configuración de la red VoIP ya que cuentan con información en sus configuraciones, entre ellas la dirección de la IP –PBX y datos del usuario.

En términos de seguridad el terminal debe contar con configuraciones de certificados de seguridad o mínimo con autenticación de contraseñas lo suficientemente robustas, si es el caso en esta última acotación, al encontrarse un terminal comprometido por el atacante a éste le pueda resultar más difícil extraer las contraseñas.

Los softphones que son soportados en computador en relación a sus vulnerabilidades presentan las mismas de éste, ya que se encuentran en la misma red de datos, por lo tanto este tipo de softphones tienen acceso tanto a la red de datos y a la red de voz. Estos terminales no permiten separar la red de voz con la de datos, de esta manera cualquier ataque realizado a la red de datos se puede ver afectado directamente a la red de voz. Sin embargo en los softphones localizados en smartphones como aplicación no se presenta este problema y se puede configurar el certificado SSL a través del cifrado para brindar seguridad en la implementación y el desarrollo de pruebas de VoIP. [6]

5.1.9 Conceptos de seguridad

Como principales conceptos asociados a la seguridad de la información se presenta confidencialidad, integridad y disponibilidad.

5.1.9.1 Confidencialidad.

La confidencialidad se define como: "el acceso a la información por parte únicamente de quienes estén autorizados", según la norma ISO 27001 [4]. La pérdida de la confidencialidad de la información en VoIP se presenta al divulgar información confidencial a través del teléfono. [10]

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

5.1.9.2 Integridad.

La integridad es definida como: "el mantenimiento de la exactitud y completitud de la información y sus métodos de proceso", según la norma ISO 27001 [4]. La integridad permite que no se realicen modificaciones no autorizadas de la información [10].

Para evitar vulneraciones de integridad debe notificarse los cambios que se realicen, ya que cualquier cambio provocado o de manera involuntaria viola el principio de integridad.

5.1.9.3 Disponibilidad.

El principio de disponibilidad según la norma ISO 27001 la define como: "acceso a la información y los sistemas de tratamiento de la misma por parte de los usuarios autorizados cuando lo requieran" [4]

Por tanto, los recursos deben estar disponibles cuando un usuario los necesite. Es así, que existe indisponibilidad del sistema no permite realizar una llamada ya sea porque el sistema se encuentra ocupado por otro usuario lo cual ocasiona que la red colapse. Sin

embargo, cuando existe falla involuntaria ya sea errores debido al hardware o software también se puede violar el principio de disponibilidad.

Para tratar de evitar indisponibilidad en un sistema se puede contar con seguridad de manera que no deje de funcionar si es víctima de ataques que pueda provocar un mal funcionamiento. Otro mecanismo importante es contar con redundancia en los equipos que intervienen en la comunicación. [10]

Los conceptos anteriormente descritos se resumen en la Tabla 5.6.

Concepto	**Definición**	Mecanismo de seguridad
Confidencialidad	Acceso a la información por parte únicamente de quienes estén autorizados.	Contraseñas y encriptación.
Integridad	El mantenimiento de la exactitud y completitud de la información y sus procesos.	Paridad
Disponibilidad	Acceso a la información y los sistemas de tratamiento de la misma, por parte de los usuarios autorizados cuando lo requieran.	Redundancia.

Tabla 5. 6 Conceptos de seguridad. [4]

5.2 Descripción de herramientas utilizadas para ataques en centrales VoIP.

5.2.1 Kali Linux

Kali Linux es una reconstrucción de Backtrack Linux basado en Debian GNU/Linux, es gratuita como su predecesor, fue fundada por Offensive Security Ltd. Mati Ahoaroni y Devon Kearns el cual fue lanzado el 13 de marzo del 2013. Kali es desarrollado en un entorno seguro, permite interactuar con repositorios oficiales, cada uno de los paquetes de Kali están firmados por su desarrollador. [19]

Kali linux es una distribución de Linux que tiene herramientas para realizar pruebas de penetración de seguridad y auditoría de seguridad. Cuenta con las siguientes características:

- Más de 600 herramientas de pruebas de penetración.
- Kali Linux como Backtrack es completamente gratis.
- Modelo de desarrollo de código abierto (Git tree), es decir que cualquier código de Kali se encuentra disponible para cualquier persona de manera que ésta pueda modificar y reconstruir para satisfacer sus necesidades específicas.
- Kali adhiere al FHS (Filesystem Hierarchy Standard) estándar de jerarquía del sistema de archivos, permitiendo a los usuarios LINUX localizar fácilmente archivos, archivos binarios, librerías etc.
- Kali evalúa constantemente sus parches de inyecciones para la robustez de su kernel.
- Kali Linux se desarrolla bajo un estricto control de sus repositorios.
- Cada uno de los paquetes de Kali son firmados por cada desarrollador individualmente.
- Kali tiene soporte multi-lenguaje, que permite incrementar el número de usuarios para que puedan trabajar en su idioma nativo y encontrar las herramientas específicas que cubran sus necesidades.
- Cualquier usuario puede personalizar Kali Linux.

Entre las herramientas que nos presenta Kali Linux para redes VoIP, se muestran en la Tabla 5.7.

Análisis VoIP	WIRESHARK
HERRAMIENTAS PARA	INVITEFLOOD
	RTPINSERTSOUND
	SVCRAK
	SVMAP
	SVCRASH
	SVREPORT
	SVWAR
	RTPBREAK

VoIP	RTPFLOOD
	RTPMIXSOUND

Tabla 5. 7 Herramientas de Kali Linux para VoIP

5.2.2 SIPVicious

La suite SIPVicious se compone de un conjunto de herramientas basadas en el lenguaje Python, que se pueden utilizar para probar sí la configuración SIP de nuestro servidor Elastix es segura. [14]

A continuación se detalla las herramientas que componen la suite de SIPVicious:

- **SVMAP:** Permite escanear una dirección IP o un rango de direcciones IP para determinar si existen dispositivos SIP.
- **SVCRASH:** Presenta los intentos de detener las búsquedas no autorizadas SVWAR y SVCRACK.
- **SVCRACK**: De acuerdo al número de intentos pretende obtener la contraseña de una extensión SIP.
- **SVWAR:** Permite escanear el número de extensiones y determinar si están protegidas con contraseña.
- **SVREPORT:** Genera reportes.

5.2.3 Wireshark.

Wireshark es analizador de protocolos de red, es la continuación de un proyecto que comenzó en 1998, basado en software libre, y se ejecuta sobre la mayoría de sistemas operativos: Linux, Solaris, FreeBSD, Android, Mac OS X, Microsoft Windows, por citar algunos. Para capturar paquetes se debe seleccionar la interfaz de red, y ejecutar con permisos de superusuario (root) considerando que posee una gran cantidad de analizadores de protocolo. [29]

Entre las características principales de Wireshark se detallan a continuación.

- Inspección profunda de cientos de protocolos que se van incrementando todo el tiempo, los cuales se capturan permitiendo así hacer un análisis offline (realizando una captura previa).
- Utiliza licencia GLP.
- Robusto, tanto en modo promiscuo como en modo no promiscuo.
- Basado en la librería pcap.
- Mecanismo de filtrado para obtener la información deseada de manera específica.
- Admite el formato estándar de archivos tcpdump.
- Reconstrucción de sesiones TCP.

Específicamente en VoIP, wireshark permite analizar protocolos como SIP, H323, ISUP y MGCP con el correspondiente RTP, así no solamente se puede capturar los paquetes sino también decodificarlos y escuchar las llamadas en caso que no tenga seguridad la red VoIP mediante su módulo de **VoIP Calls.**

VoIP Calls permite mostrar información de todas las llamadas que cruzan por la red, permite también realizar un filtrado de una llamada en particular o más de una, de esta manera realizar el correcto análisis de los mensajes enviados durante la conversación.

5.2.4 INVITEFLOOD

INVITEFLOOD es una herramienta para realizar inundaciones SIP/SDP INVITE a través de UDP/IP. Esta herramienta se encuentra ya instalada en la distribución Kali Linux. [23]

INVITEFLOOD permite enviar mensajes INVITE en grandes cantidades de manera que el receptor SIP colapse, ya que el servidor intentará atender todas las llamadas generadas consumiendo gran cantidad de recursos, por esta razón es utilizado para realizar denegación de servicio.

5.2.5 RTPFLOOD

RTPFLOOD es una herramienta que es utilizada para inundar dispositivos que procesan RTP. La inundación se realiza con paquetes UDP que contienen data RTP. Con el fin de obtener un ataque exitoso usando RTPFLOOD, se debe conocer el puerto que escucha RTP en el dispositivo al cual se realizará el ataque. El puerto por defecto RTP de Zoiper es el 8000.

RTPFLOOD es una herramienta basada en Python que es parte de la distribución de Kali. [1]

5.3 Identificación de vulnerabilidades a través de pruebas de penetración a sistemas VoIP.

La telefonía VoIP, hoy en día ha reemplazado de manera masiva a la telefonía tradicional (PSTN), ya sea por la conectividad y bajo costo que representa, sin embargo al implementar las centrales VoIP no se da la importancia debida a la seguridad, sino a parámetros como el envío y recibo de los datos, voz, la calidad de servicio.

5.3.1 Escenario.

Para desplegar los vectores de ataque sobre la infraestructura de comunicaciones VoIP, se inserta un atacante sobre la red de datos en el cual se plasmará las herramientas que permitan recopilar información para elg análisis de vulnerabilidades. Por lo cual se incorpora la distribución Kali Linux en su versión 2.0, que cuenta con las herramientas para el análisis de vulnerabilidades a diferentes servicios entre ellos VoIP.

El atacante puede ser parte de la red interna o puede encontrarse entre la red interna y el internet, de aquí se pueden derivar lo que se conoce como ataques internos y externos de red teniendo así acceso a la información que despliega la central telefónica y todos los dispositivos que se encuentren en dicho segmento de red, como se muestra en la Figura 5.7.

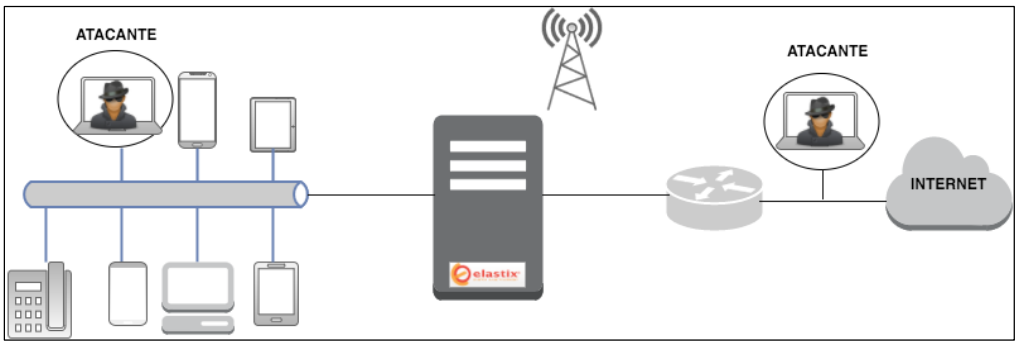


Figura 5. 7 Escenario para implementación de vectores de ataque.

5.3.2 Recopilación de información: Footprinting.

El atacante como primera instancia realizará **footprinting** en busca de obtener la mayor información, en este caso corresponde a las características de la infraestructura de comunicaciones de VoIP, y de esta manera plantear las herramientas que permitan realizar un análisis de las vulnerabilidades de la central.

Como herramienta principal se presenta Kali, la cual contiene la suite SIPVicious que es un conjunto de herramientas que se pueden utilizar para sistemas VoIP basados en SIP. [14]

En la suite SIPVicious se encuentra la herramienta SVMAP, la cual realiza un scanner a nivel SIP, que nos servirá para listar los dispositivos SIP que se encuentran en un rango IP, como se muestra en la Figura 5.8.

Figura 5. 8 Escaneo de dispositivos SIP.

De manera que a través de esta aplicación se envía un requerimiento SIP ”INVITE“ que permite realizar este intercambio de mensajes, y de esta manera se obtiene el software que tiene instalada la central como se muestra en la Figura 5.8 la central Elastix tiene instalada FPBX-2.11.0.

5.3.3 Reconocimiento de puertos y servicios: Escaneo.

Con la información obtenida previamente acerca de la red, se pretende adquirir información acerca de los puertos y servicios que se encuentran habilitados en la central telefónica, para lo cual se utiliza el aplicativo **NMAP**, que advierte los puertos habilitados con el servicio que están prestando como se muestra en la Figura 5.9.

NMAP tiene una base de datos exclusiva para identificar dispositivos dedicados a VoIP, utilizando el siguiente comando, **nmap** –O –P0 192.168.1.0-254

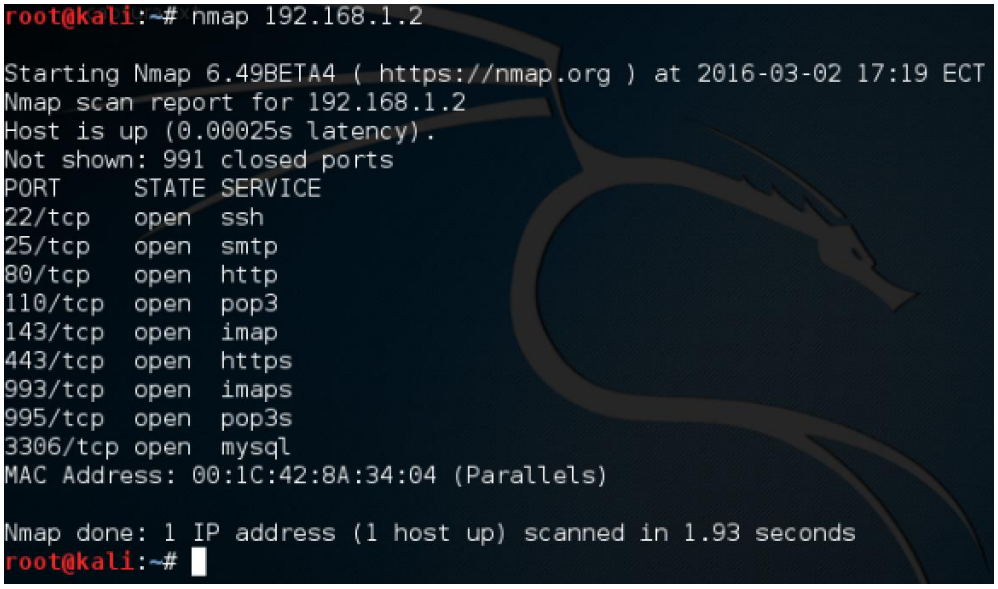


Figura 5. 9 Escaneo de puertos.

5.3.4 Enumeración: Fingerprinting

Después de obtener un listado de servicios y direcciones IP consistente, se trata de buscar agujeros de seguridad, es así que mediante el método SIP Register el cual es generado por el usuario para el gestor de llamadas VoIP, que para este ataque es un servidor SIP aquí el atacante envía solicitudes de registro a varias extensiones y usuarios de la red VoIP, para así poder listar las extensiones no utilizadas o los usuarios no registrados. Para ello se emplea la herramienta SVWAR de la suite SIPVICIOUS, que cumple la función de mostrar la numeración de extensiones de una PBX. Como se muestra en la Figura 5.10. [14]

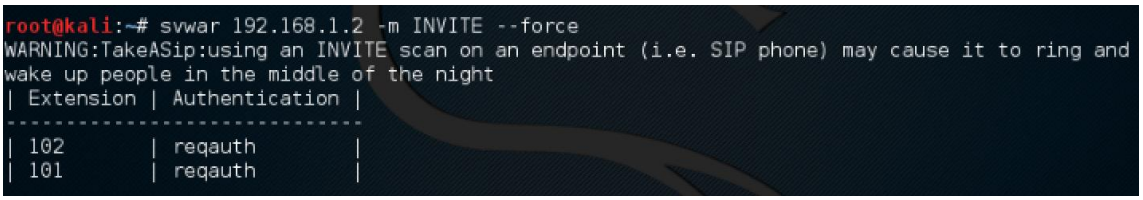


Figura 5. 10 Obtención de extensiones de una PBX.

5.3.5 Ataques por fuerza bruta

Al realizar este ataque se pretende que a través de múltiples combinaciones de caracteres se pueda obtener autenticación SIP mediante las extensiones registradas a la central VoIP. [14]

Para esto se cuenta con un diccionario, y la utilización de la herramienta SVCRACK la misma que se encuentra en la suite de SIPVICIOUS. Como se observa en la Figura 5.11 la extracción de contraseñas tanto de la extensión 101 como 102.

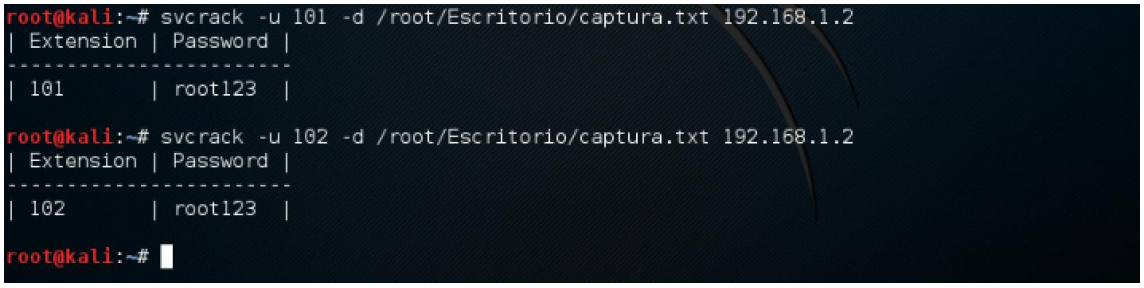


Figura 5. 11 Ataque por fuerza bruta

5.3.6 Ataque de denegación de servicio.

Este vector de ataque tiene como finalidad causar la indisponibilidad de servicio al degradar el rendimiento de la red, en este caso el de VoIP en las centrales telefónicas es decir es inutilizable para los usuarios.

Para este propósito se basa en el envío de paquetes construidos para explotar alguna vulnerabilidad, colapso en el flujo de datos y de la red o sobrecarga de procesos en los dispositivos de la víctima. Los ataques DDoS o ataques de denegación distribuidos, son ataques simples pero realizados desde múltiples terminales de manera coordinada.

Como en todo sistema, la telefonía IP con respecto a sus aplicaciones o elementos de comunicación que interviene en la comunicación, trabajan sobre puertos específicos, colapsar estos puertos con tráfico innecesario puede ocasionar una denegación de servicio y que usuarios autorizados no puedan hacer uso del sistema. La suplantación de identidad (del destinatario de la llamada o de algún otro dispositivo VoIP) generalmente deriva en una denegación de servicio. En redes VoIP basadas en el protocolo SIP, es posible enviar mensajes como INVITE CANCEL, o ICMP Port Unreacheable, UDP para afectar RTP, con el objetivo indisponer el servicio e impedir su correcto funcionamiento.

Los sistemas VoIP son vulnerables por diferentes razones:

- Dependen de calidad de servicio.
- Los ataques de DoS se basan en atacar los dispositivos de red y/o inundar la red de tráfico inútil para degradar su funcionamiento, y de esta manera causar q los paquetes se pierdan o se retrasen.

Para este propósito se utiliza la herramienta INVITEFLOOD, la cual envía mensajes INVITE de manera masiva. La herramienta va aumentando el número de secuencia de los mensajes INVITE.

En la Figura 5.12 se observa a la extensión **102** la cual pertenece a la PBX 192.168.1.2 y se encuentra activa.

Figura 5. 12 Extensión 102

En la Figura 5.13 se puede observar la ejecución del comando INVITEFLOOD [23]:

root@kali# inviteflood eth3 102 192.168.1.101 192.168.1.2 1000000 –a atacante –D 5060 -v

El argumento eth3 es la interfaz, el segundo argumento 102 es la extensión hacia la cual va dirigido el ataque. Luego el tercer y cuarto argumento es la dirección IP origen y la dirección IP destino respectivamente, el quinto argumento es la cantidad de paquetes los cuales servirán para realizar la denegación de servicio, el sexto argumento es un alias de usuario, el séptimo argumento es el puerto destino y el último argumento es la verbosidad la cual puede ser opcional.

```
root@kali:~# inviteflood eth3 102 192.168.1.101 192.168.1.2 100000000000 -a atacante -D 5060

inviteflood - Version 2.0
              June 09, 2006

source IPv4 addr:port    = 192.168.1.101:9
dest   IPv4 addr:port    = 192.168.1.2:5060
targeted UA              = 102@192.168.1.101

Flood User Alias: atacante

Flooding destination with 1215752192 packets
sent: 5671402C 5671356
exiting...
```

Figura 5. 13 Ataque de DoS a través de INVITEFLOOD.

En la Figura 5.14 se puede observar la denegación de servicio a la extensión **102**.

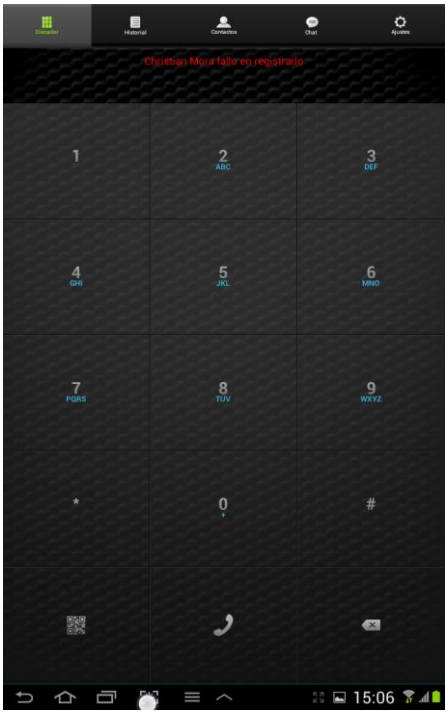


Figura 5. 14 DoS dela extensión 102

Por otro lado se comprueba mediante **WIRESHARK** la inundación de paquetes **INVITE** mediante la Figura 5.15**.**

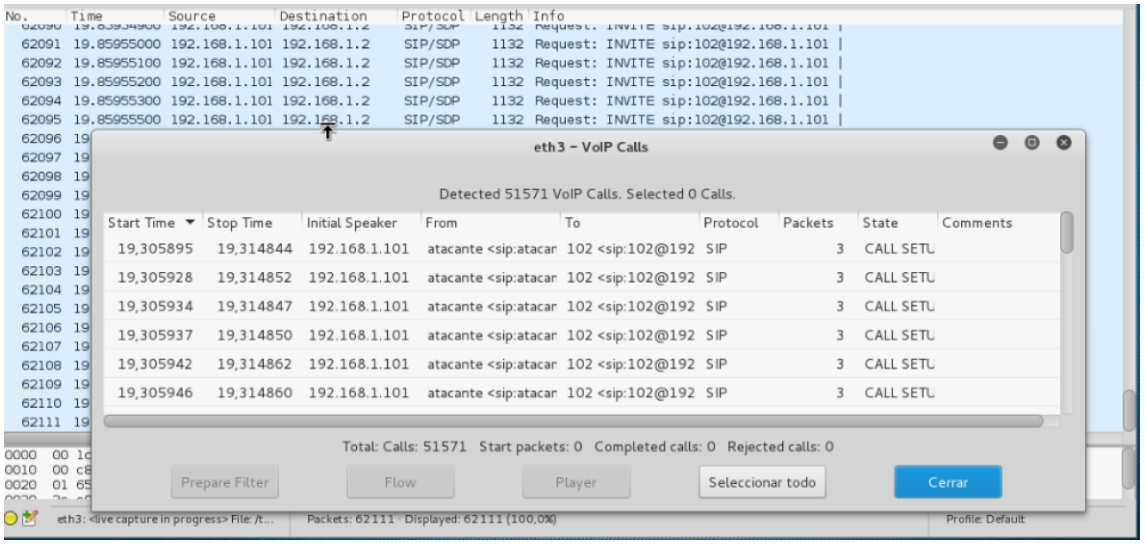


Figura 5. 15 Inundación de paquetes INVITE a través de INVITEFLOOD.

5.3.6.1 RTPFLOOD

RTPFLOOD es una herramienta que viene en la suite de aplicaciones de Kali Linux y requiere parámetros de los mensajes **RTP** como el **timestramp,** para provocar una inundación de paquetes. [1]

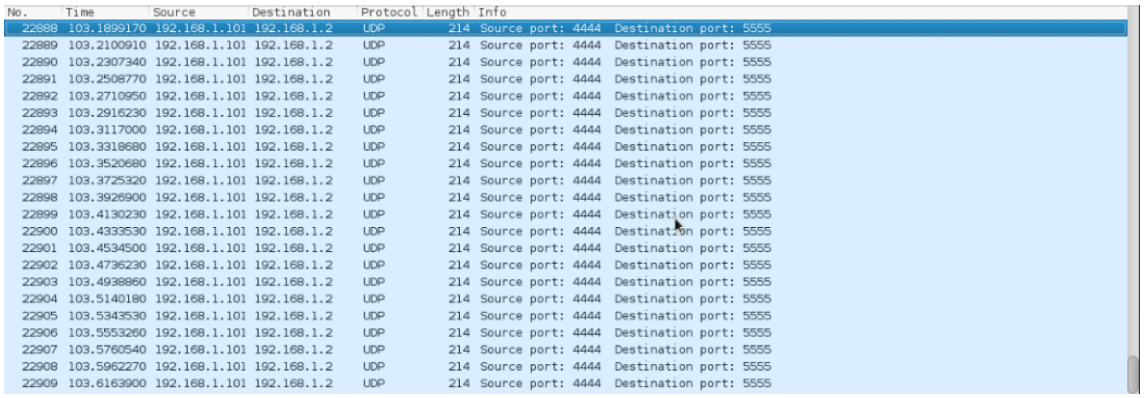
El primer argumento es la dirección IP de origen desde donde se realiza el ataque. El segundo argumento es la dirección IP de destino en este caso la central telefónica. El tercer y cuarto argumento son los puertos de origen y destino respectivamente. El quinto argumento es el número de paquetes que serán enviados. El sexto es el número de secuencia. El séptimo es el timestamp y por último el SSID como se muestra en la Figura. 5.16.

root@kali# rtpflood 192.168.1.101 192.168.1.2 4444 5555 10000000000 2 123456 atacante

Una solución para mitigar este tipo de ataques es **SRTP.**

Figura 5. 16 Utilización de la herramienta RTPFLOOD.

Por otro lado se puede comprobar mediante **WIRESHARK** la inundación de paquetes **UDP** mediante la Figura 5.17



No.	Time	Source	Destination	Protocol	Length	Info
22888	103.1899170	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22889	103.2100910	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22890	103.2307340	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22891	103.2508770	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22892	103.2710950	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22893	103.2916230	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22894	103.3117000	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22895	103.3318680	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22896	103.3520680	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22897	103.3725320	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22898	103.3926900	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22899	103.4130230	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22900	103.4333530	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22901	103.4534500	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22902	103.4736230	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22903	103.4938860	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22904	103.5140180	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22905	103.5343530	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22906	103.5553260	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22907	103.5760540	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22908	103.5962270	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555
22909	103.6163900	192.168.1.101	192.168.1.2	UDP	214	Source port: 4444 Destination port: 5555

Figura 5. 17 RTPFLOOD en Wireshark.

5.3.7 Eavesdropping.

La intercepción o eavesdropping (“escuchar secretamente"), es el término con el cual se conoce a la captura de información (cifrada o no), por parte del atacante al que no va dirigida esta información. En VoIP la intercepción de información es interceptar conversaciones VoIP que se transmiten por la red y que no es enviada para los intrusos.

El **eavesdropping** en VoIP es diferente a la intercepción de datos presentes en redes tradicionales, en VoIP se encuentran presente dos partes dentro de lo que respecta a la comunicación: **la señalización y el flujo de datos** en los cuales intervienen protocolos diferentes. En la señalización interviene el protocolo SIP mientras que en el flujo de datos presenta el protocolo RTP sobre UDP. El impacto puede ser pasivo al momento de escuchar la conversación pero se puede generar un impacto de forma activo en la

comunicación al instante de insertar nuevos datos, en este caso audio que provocaría impedir que los datos lleguen al destino.

Para conseguir interceptar la comunicación se presentan herramientas como **ethereal/wireshark** que permitirá capturar todo el tráfico del segmento de red de interés, utilizando la técnica **Main in the Midle.**

Mediante **Wireshark,** para acceder al análisis de llamadas VoIP, se debe realizar los pasos como se muestra en la Figura 5.18

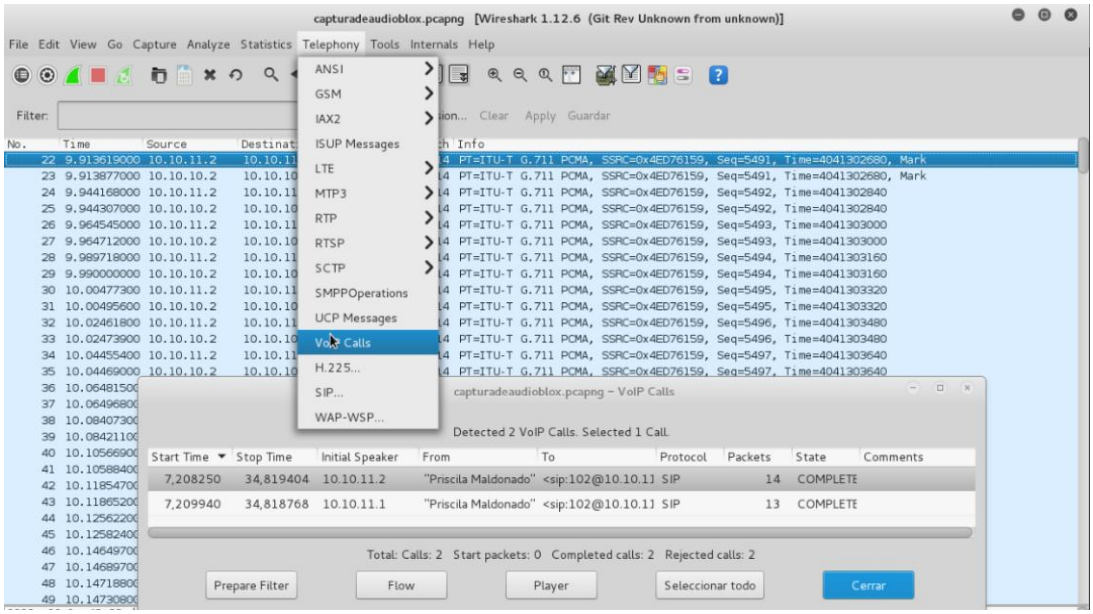


Figura 5. 18 Intercepción de llamadas

Se puede ver a detalle en la Figura 5.19 que se realiza la llamada desde Priscila Maldonado la cual es la extensión 202 con la sesión **tag** = as53716f43 hacia la extensión 102.

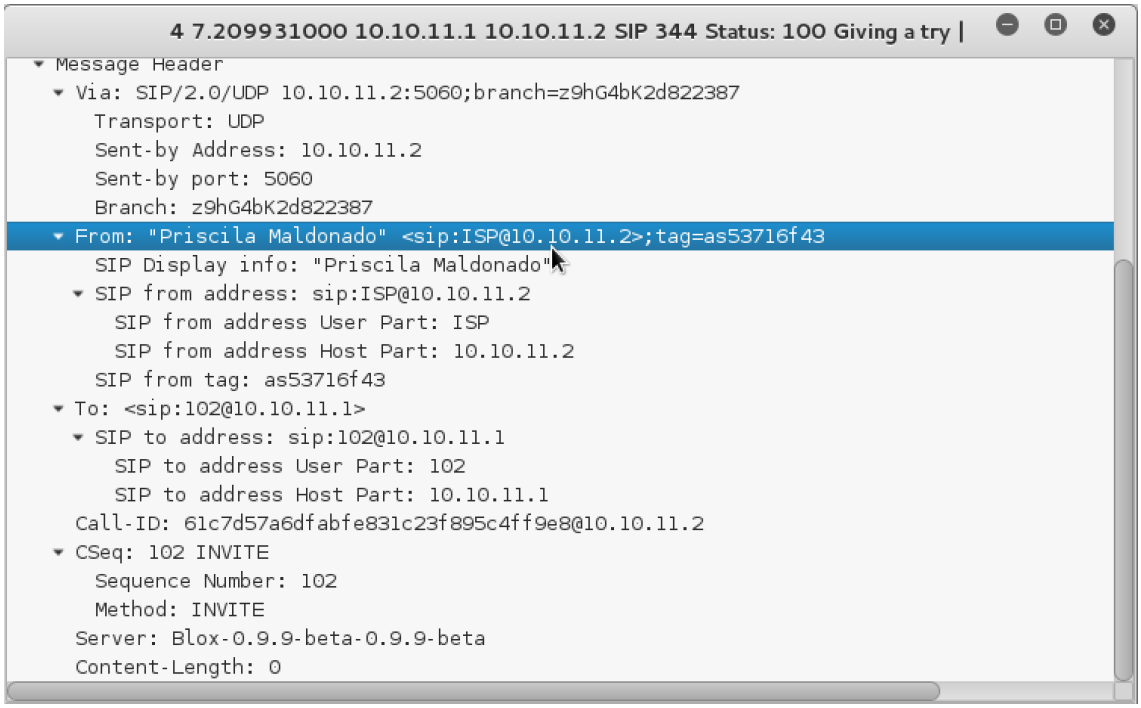


Figura 5. 19 Detalle del eavesdropping

La lista de llamadas VoIP muestra la siguiente información por cada llamada como se detalla y se observa en la Figura 5.20. Según el indicativo de la página oficial de Wireshark. [22]:

- **Start Time:** Tiempo de inicio de la llamada.
- **Stop Time:** Tiempo de finalización de la llamada.
- **Initial Speaker:** La IP origen del paquete que inició la llamada.
- **From:** Para llamadas SIP analizar el campo "From" del mensaje INVITE.
- **To:** Para llamadas SIP analizar el campo "To" del mensaje INVITE
- **Protocol:** SIP.
- **Packets:** Número de paquetes de la llamada.
- **State:** El estado actual de la llamada. Entre los cuales pueden ser:
    - **CALL SETUP:** Llamada en estado setup (Setup, Proceeding, Progress o Alerting)
    - **RINGING:** Timbrando
    - **IN CALL:** Llamada establecida
    - **CANCELLED:** llamada fue liberada antes de conectar desde quien inició la llamada.

- **COMPLETED:** llamada fue completada y luego liberada
- **REJECTED:** llamada fue liberada antes de conectar por el destinatario
- **UNKNOWN:** no se conoce el estado de la llamada

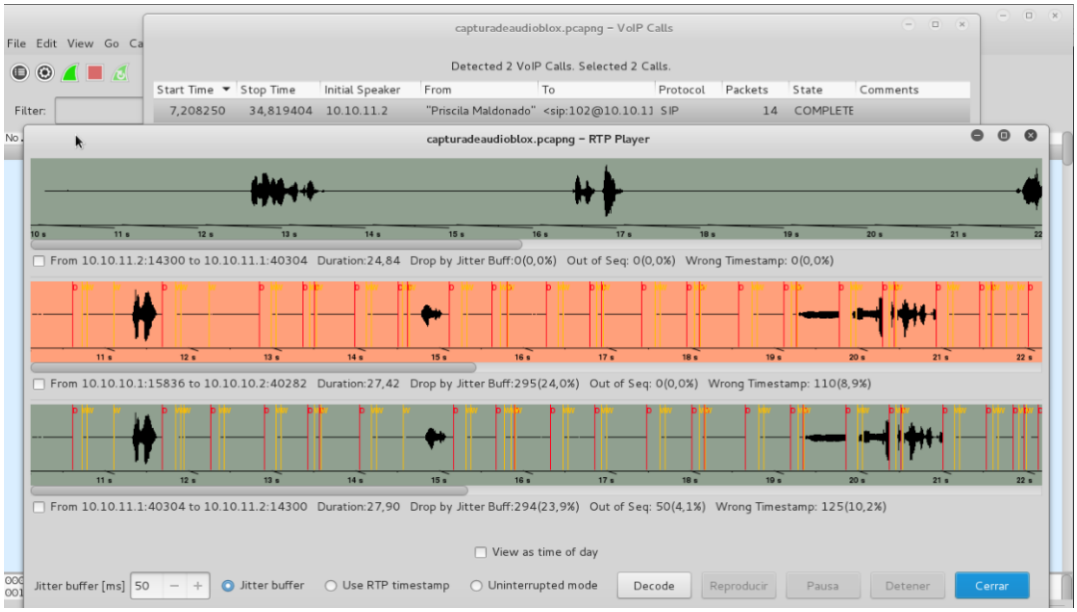


Figura 5. 20 Intercepción de llamadas mediante VoIP Calls

Para filtrar una llamada específica, se selecciona la o las llamada requeridas y se debe presionar en el botón "Prepare Filter". Wireshark presentara el filtro de los paquetes que intervienen en la llamada. Como se muestra Figura 5.21.

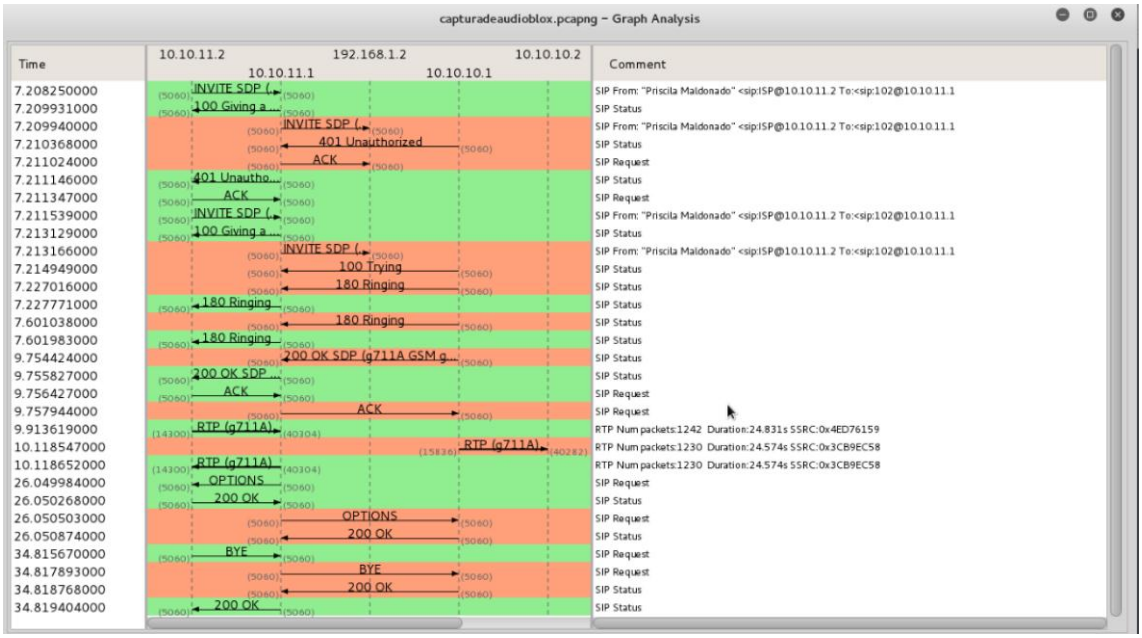


Figura 5. 21 Análisis del filtro de llamada en Wireshark.

5.4 Session Border Controller (SBC)

5.4.1 Funcionalidades de un SBC

El término SBC relativamente no es específico, ya que no ha sido estandarizado. Usualmente el dispositivo SBC se encuentra entre la red interna y el proveedor de servicios, ya que es un dispositivo de borde, básicamente, el SBC gestiona tanto la media como la señalización de las llamadas VoIP. Como se observa en la Figura 5.22. [17]

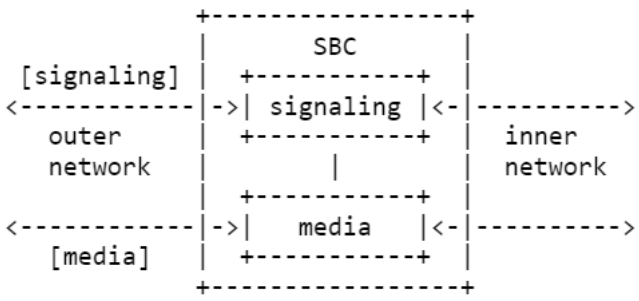


Figura 5. 22 Arquitectura SBC. [17]

Se puede instalar como un appliance o máquina virtual y dispone de funcionalidades que harán que la red de telefonía IP sea mucho más segura y se integre mejor con el equipamiento SIP de diferentes fabricantes y proveedores de servicios. [17]

A continuación se enumeran las funciones que se utilizan en las implementaciones de las redes de comunicaciones:

- SEGURIDAD: Es una de las funciones básicas que brinda el SBC a la red y otros dispositivos SIP. El SBC oculta al exterior la topología de red interna, de manera que no sea divulgada a terceros, evitando que los equipos sean expuestos a ataques de denegación de servicio o aún más efectivo un ataque de DDoS.

- ENCRIPTACIÓN: Por otro lado requieren del uso de certificados digitales para cifrar todo el tráfico a través del intercambio de llaves, de manera que, al ser víctima de Easvesdropping en la comunicación, el atacante no pueda escuchar la conversación ni pueda ver los parámetros necesarios para establecer la comunicación. Estas amenazas se pueden evitar utilizando los protocolos TLS

(Transport Layer Security) y SRTP (Secure Real-Time Transport Protocol) para proteger la señalización y los canales de voz respectivamente.

- POLÍTICAS DE ACCESO: La gestión de políticas permite al SBC controlar el uso no autorizado de los servicios de VoIP. Control de acceso es una función que permite limitar el número de sesiones establecidas para no sobrepasar el límite soportado por la WAN.

- ENRUTAMIENTO DE LLAMADAS: El SBC permite configurar reglas de enrutamiento de llamadas que permitirán funcionalidades como el LCR (Least Cost Routing) o el balanceo de carga entre diferentes SIP Trunk.

- La FIJACIÓN DE DESAJUSTE DE CAPACIDAD permite la comunicación entre usuarios agentes con diferentes capacidades o extensiones, un ejemplo es la conexión con dos usuarios agentes que soporten diferentes códecs, que se encuentren en diferentes dominios de direcciones o diferentes versiones de IP, dándose la comunicación a través de esta función de manera transparente

- INTEROPERABILIDAD: Aunque SIP se considera un estándar, es extremadamente flexible.

- RELACIÓN SIP NAT: Se refiere a las modificaciones de mensajes específicos requeridos para asistir a un usuario agente en el mantenimiento SIP y la conexión media, cuando existe un dispositivo NAT localizado ente el UA y PROXY/REGISTRADOR o posiblemente otro UA. La principal propuesta de la función del NAT transversal es mantener el control de la conexión hacia el UA detrás de NAT, esto puede ser alcanzado generando tráfico de red periódico que mantiene establecido el NAT. En los SBC el NAT transversal es requerido cuando el NAT está fuera del SBC no en casos donde el SBC actúa como NAT.

- TRANSCODIFICACIÓN DE MEDIA: La encriptación media es necesaria para permitir que diferentes tipos de media crucen a través de equipos diversos y también para permitir un uso óptimo del ancho de banda disponible.

5.4.2 SBC BLOX.

Blox es un SBC Session Border Controller basado en software de código abierto, que se utiliza para controlar la señalización de VoIP y stream media. Ofrece escalabilidad con funciones avanzadas. Blox ofrece transcodificación media, seguridad, enrutamiento basado en políticas.

Es un controlador de sesión de borde (SBC) responsable del establecimiento, ejecución y finalización de las llamadas. SBC permite controlar las llamadas que atraviesan por la red y superar algunos problemas que son causados por Firewall y NAT para llamadas VoIP.

Es un dispositivo de borde que se encuentra localizado entre la red de área local (LAN), y el ISP, siendo para las pruebas correspondiente en este caso de estudio el ISP una central Elastix . SBC vigila el tráfico SIP en tiempo real entre las fronteras de las redes basadas en SIP, lo que garantiza la seguridad de la red privada, como se observa en la Figura 5.23.

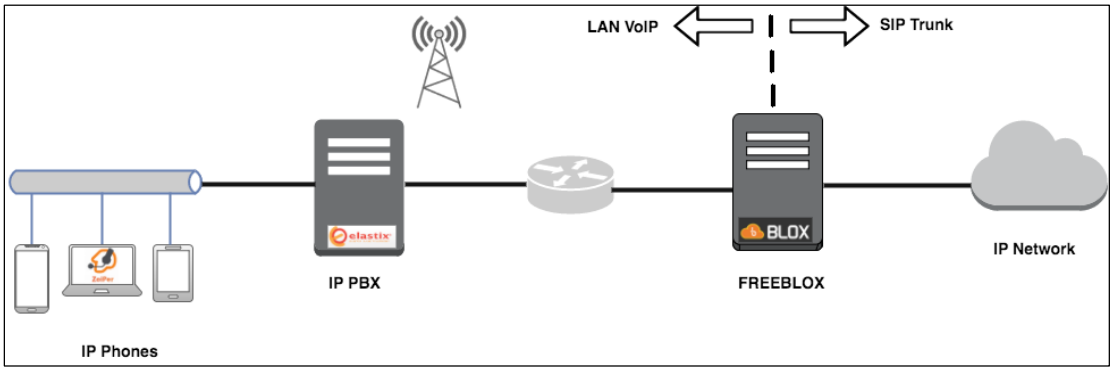


Figura 5. 23 Implementación Blox

A continuación, en Tabla 5.8. Se muestra la información correspondiente a la configuración de red.

Nombre del Servidor	Dirección IP	Dirección Mac
Elastix-PBX	192.168.1.2	00:1C:42:D7:86:46
BLOX	**LAN** 10.10.10.2 **WAN** 10.10.11.1	00:1C:42:93:B3:73 00:1C:42:83:D0:E9
Elastix- ISP	192.168.3.2	00:1C:42:CF:43:41

Tabla 5. 8 Configuración de red

Por otro lado FreeBlox es la interfaz de usuario diseñada para BLOX SBC, de manera que el usuario pueda configurar las características y la administración.

SBC Blox funciona con DPI (inspección de paquetes sobre el tráfico VoIP), el cual soporta firmas para claves de malwares, vulnerabilidades en implementaciones SIP como extensiones, enumeraciones, ataques de DoS, cracking de contraseñas entre otras.

Una ventaja de Blox es que es soportado en PBX open source como Asterisk, FreeSwitch, TrixBox. Adicional a lo antes mencionado se encarga de problemas comunes observados en implementaciones VoIP como el SIP NAT. [28]

A continuación se presenta la Tabla 5.9 las características de SBC BLOX.

Capacidades técnicas de firewall	SI
DPI sobre tráfico SIP.	SI
Número de llamadas simultáneas	Hasta 250 *
No. of VoIP firmas	30 aprox
Codecs de Audio	G.722,G.729,G.711 A-Law, G.711 U-Law, G.723, G.726
Logging	Local log viewer, Syslog

Tabla 5. 9 Especificaciones técnicas de Blox [28]

5.4.2.1 Funcionalidades de SBC Blox.

Entre las funcionalidades principales que posee Blox se presentan las siguientes [5]:

- Normalizar la señalización SIP para que el cliente sea compatible con el proveedor de servicio.
- Resolver problemas de NAT transversal para permitir la adopción de SIP, SIP trunking y comunicaciones unificadas de seguridad que permitan la señalización SIP y la comunicación media afines, al atravesar el firewall. Sin esta función, la mayoría de las empresas tendrán un solo sentido, audio.
- Security through deep packet inspection (DPI): Es una poderosa manera de no solamente proteger SIP sino también a la red. Es una forma de filtrado de paquetes

de red que examina los datos en la cabecera UDP /TCP, a medida que pasa por BLOX. DPI puede ser eficaz contra los ataques de desbordamiento de buffer, denegación de servicio (DoS), y un pequeño porcentaje de gusanos que se ajustan dentro de un solo paquete.

- Control a través de la autenticación: Muchos proveedores de servicios requieren la autenticación del usuario con su red. IP-PBX no admiten esta función. Al implementar Blox el requisito del proveedor puede cumplirse con independencia de que se utilice IP-PBX.
- Cifrado: características de cifrado son inherentes en el protocolo SIP y cuando se utiliza entre dos sitios es capaz de minimizar cualquier oportunidad a las partes relacionadas para interceptar la llamada. Esto ofrece la máxima privacidad, incluso a través de Internet pública.
- Detección de Intrusión / Prevención: El sistema de detección de intrusos (IDS) y de prevención de intrusos (IPS) en el módulo de software de seguridad mejorada, Blox permite detectar ataques de denegación de servicio basado en SIP y bloquear los paquetes de señalización SIP maliciosos diseñados para atacar terminales SIP, servidores u otros dispositivos de la empresa LAN. Esto asegura la red de la empresa, Blox maneja los ataques mientras que los servidores y otros dispositivos de SIP en la red puedan seguir disponibles.

5.5 Implementación de SBC BLOX como solución de seguridad a centrales VoIP.

En la mayoría de las empresas se tiene la idea de que al presentar un dispositivo Firewall, configurado para dar seguridad a su red no es necesario ningún otro dispositivo, pero en análisis el Firewall como tal permite bloquear puertos para VoIP.

En VoIP las sesiones son dinámicas lo que causa una limitante en Firewall debido a que en lo que respecta a habilitar puertos por tiempos muy prolongados pueden ser víctima de ataques en estos puertos.

El SBC resuelve la limitante de los Firewall mediante B2BUA (Back to back user agent) cuya funcionalidad es dividir el canal de comunicación en dos llamadas para las cuales hace de mediador de toda la señalización SIP desde el establecimiento hasta la finalización de la llamada.

Aunque en el sitio oficial de Blox no se menciona la utilización de herramientas que hacen posible su funcionamiento, Blox integra algunas herramientas de software libre las mismas que en la configuración por CLI se pueden encontrar como: OpenSIPS y Snort.

- OpenSIPS

OpenSIPS es una herramienta que se encuentra inmersa en BLOX, OpenSIPS es un proxy/servidor de código abierto SIP para voz, video, mensajería instantánea, presencia y cualquier otra extensión SIP. [24]

OpenSIPS es un servidor de múltiples funciones de señalización de usos múltiples SIP, que puede actuar como SIP Router/Switch, registrador SIP, servidor de aplicaciones, servidor de redireccionamiento, balanceador de carga/Dispatcher con conmutación por error, Back-to-Back agente de usuario, servidor de IM, controlador de borde de sesión, servidor NAT transversal para SIP y tráfico RTP, IP gateway (SMS, XMPP), para la capa de transporte soporta TCP, TLS, SCTP, UDP IPv4 e IPv6.

OpenSIPS posee arquitectura modular, es decir módulo de interfaz plug and play para ampliar la funcionalidad del servidor.

OpenSIPS se recomienda para cualquier tipo de escenario SIP / servicio por:

- ✓ Alto rendimiento - millones de llamadas simultáneas
- ✓ Flexibilidad de enrutamiento e integración - escritura de enrutamiento para la implementación de la lógica de enrutamiento personalizado, varias API de interfaz.
- ✓ Más de 120 módulos para proporcionar características, para la manipulación de SIP y operaciones de back-end.

- Snort

Blox cuenta con Snort como IDS, el cual es redes VoIP, se lo relaciona directamente con el DPI, que es un sniffer de paquetes y un detector de intrusos. Se encuentra disponible bajo licencia GLP, contiene una gran cantidad de filtros o patrones ya predefinidos. Esta funcionalidad lo convierte en un sistema flexible de firmas de ataques. [26]

Snort tiene una base de datos de ataques que se actualiza constantemente, los usuarios pueden crear firmas basadas en los ataques nuevos lo que beneficia a todos los usuarios de snort.

En lo que se refiere a Blox, se puede ir actualizando por medio de la interfaz gráfica y soporta hasta 30 firmas aproximadamente.

5.5.1 Requisitos mínimos de hardware para SBC BLOX.

Hasta 90 llamadas simultáneas, cumpliendo estas características de hardware, de acuerdo al fabricante:

- Procesador Dual Core con arquitectura de 64 bits
- 2 GB de RAM
- 2 interfaces de red (10/100/1000Mbps)
- Espacio de disco duro de 80 GB
- Descargar e Instalar Blox en un dispositivo de 64 bits.
- Configurar la red para Blox. (WAN/LAN), con la siguientes instrucciones http://blox.org/DownloadInstallationGuide
- Una tarjeta de transcodificación (Opcional) en caso de requerir transcodificación
- Configurar mediante la interfaz gráfica FreeBlox

5.5.2 Instalación de BLOX

A continuación se detalla los pasos para la descarga e instalación necesaria para obtener Blox y acceder a la configuración de FreeBlox.

PASO 1.-Descargar **.iso** de Blox (**Blox-0.9.9-22-beta-x86_64.iso**) desde http://blox.org/downloads , la imagen ISO de BLOX SBC se compone de un sistema basado en LINUX específicamente CentOS. Como se muestra en la Figura 5.24.

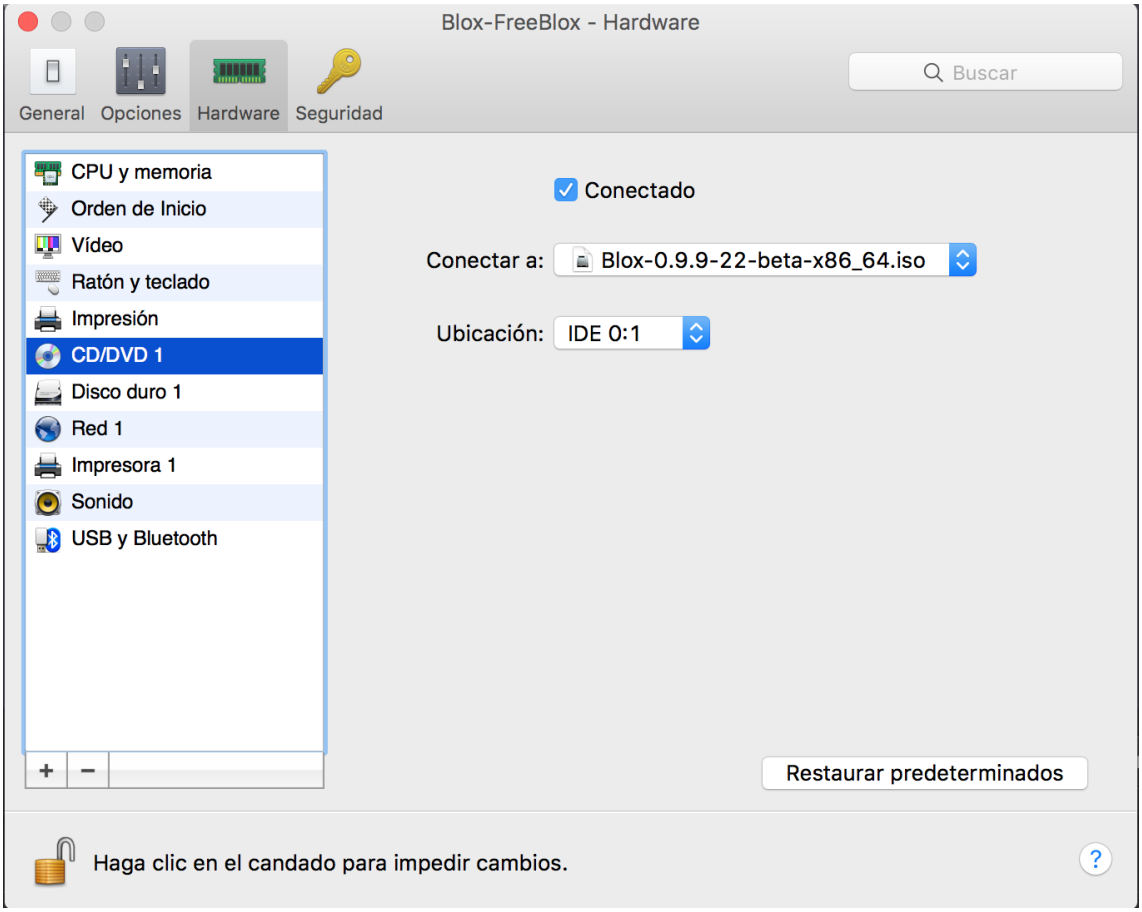


Figura 5. 24 Implementación de Blox sobre máquina virtual.

Paso 2.- Se debe **BOOTEAR** el servidor con la imagen **Blox-0.9.9-22-beta-x86_64.iso** Como se muestra en la Figura 5.25.

Figura 5. 25 Instalación de BLOX

Paso 3.- Iniciar el proceso de instalación, escogiendo la opción **Install,** seleccionar la hora, idioma, idioma del teclado, configurar la contraseña root y comprobar la misma. Configure una contraseña robusta ya que Blox, no permite contraseñas débiles. Como se muestra en la Figura 5.26.

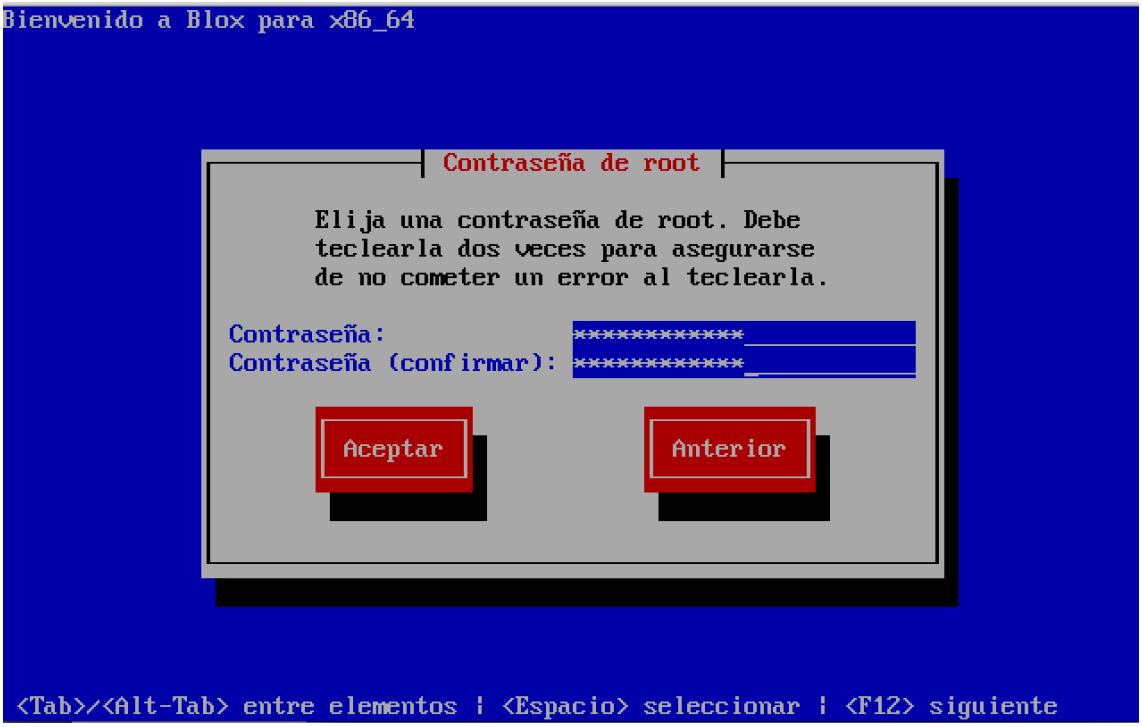


Figura 5. 26 Configuración de contraseña root.

Paso 4.- Seleccionar el disco de instalación, para iniciar el proceso de instalación. Como se muestra en la Figura 5.27.

Figura 5. 27 Wizard de particiones.

Paso 5.- Observe en la Figura 5.28 la instalación del paquete terminada.

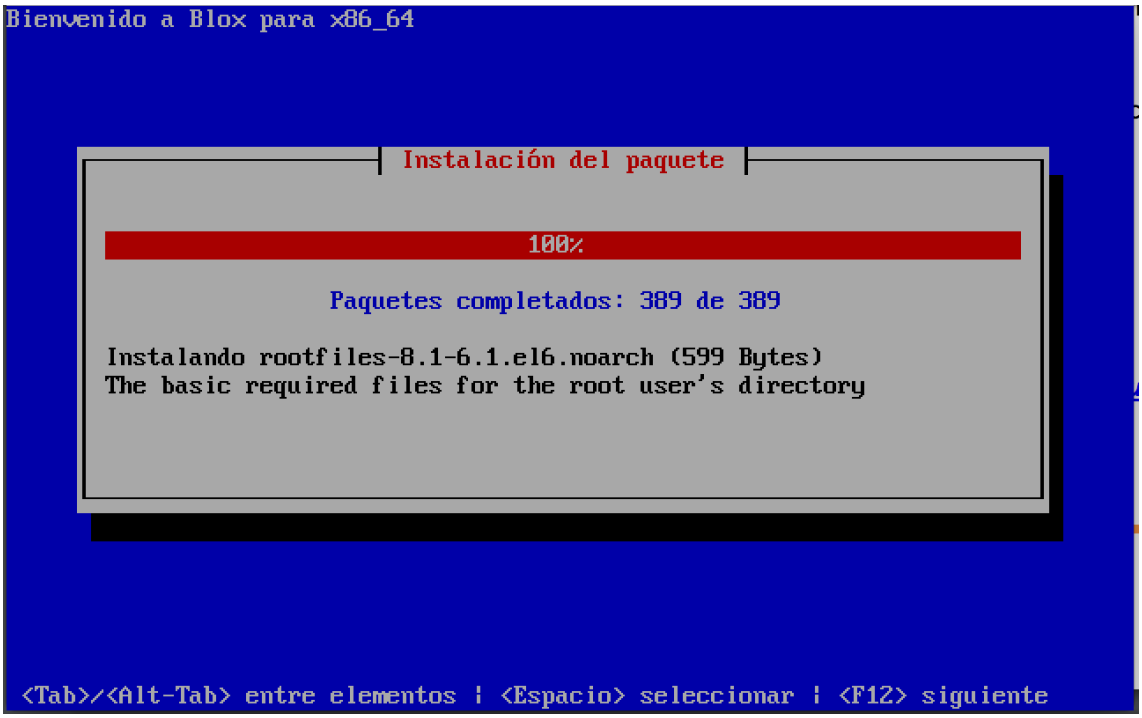


Figura 5. 28 Instalación de paquetes

Paso 6.- A continuación se muestra el login por consola de **BLOX** en la Figura 5.29.

Figura 5. 29 Login de Blox

Paso 7.- Con el comando **ifconfig** se muestran las interface de red que posee Blox. Como se observa en la Figura 5.30.

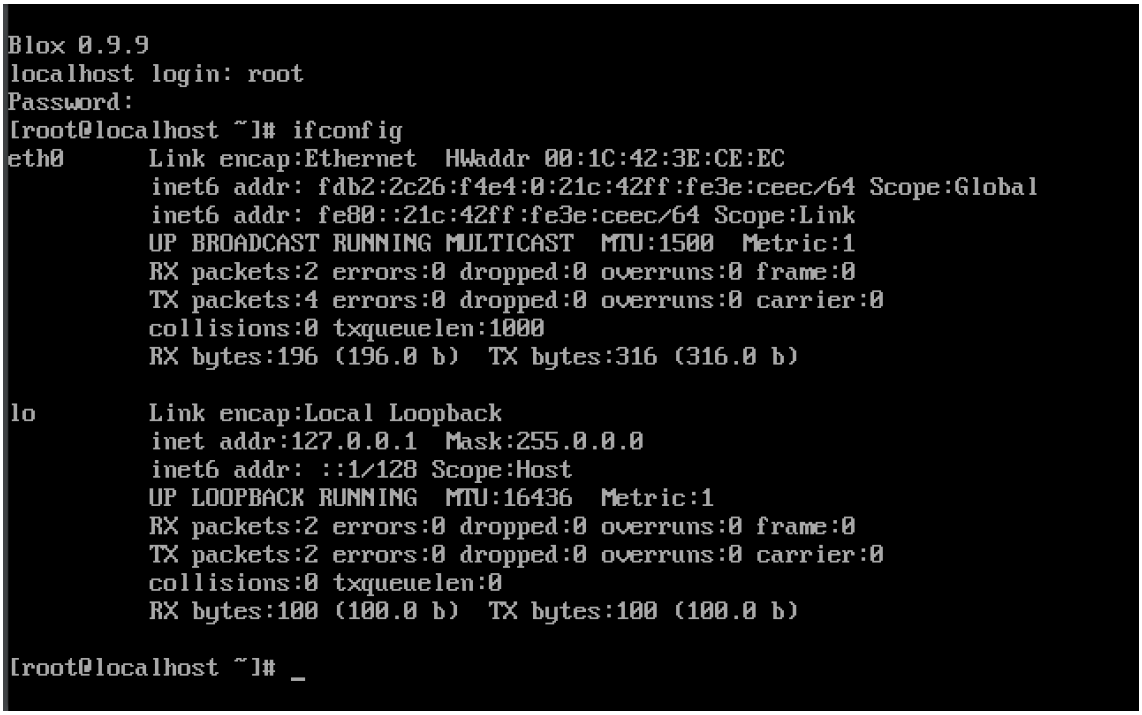


Figura 5. 30 Visualización de interfaces.

Paso 8.- Configuración de red.

Se necesita configurar 2 interfaces de red y una interfaz virtual para media.

Como primer paso se configura la interfaz **LAN eth0** de forma estática en el archivo de configuración **/etc/sysconfig/network-script/ifcf-eth0** Como se muestra en la Figura 5.31.

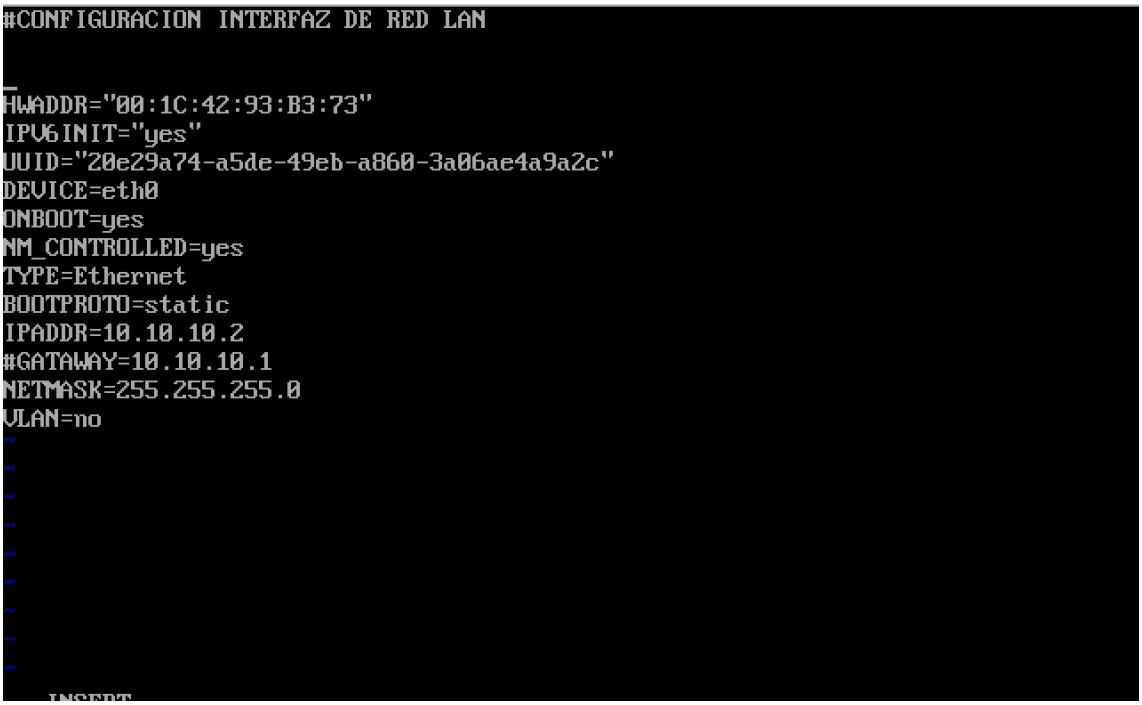


Figura 5. 31 Configuración de red LAN.

Se configura la interfaz **WAN eth1** de forma estática en el archivo de configuración **/etc/sysconfig/network-script/ifcf-eth1.** Como se muestra en la Figura 5.32.

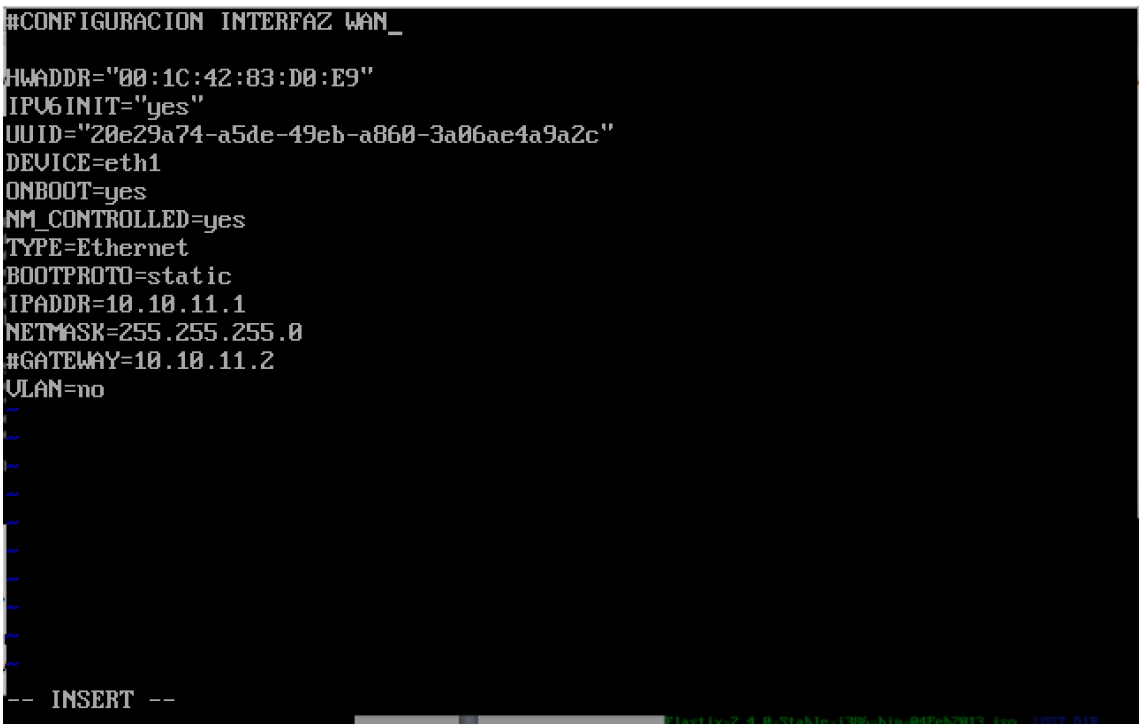


Figura 5. 32 Configuración de red WAN.

Por último se configura la sub interfaz **eth0:1** de forma estática en el archivo de configuración **/etc/sysconfig/network-script/ifcf-eth0:1** Como se muestra en la Figura 5.33.

Figura 5. 33 Configuración Interfaz virtual

Tras realizar estas configuración se procede a reiniciar el servicio de red mediante el comando **service network restart**

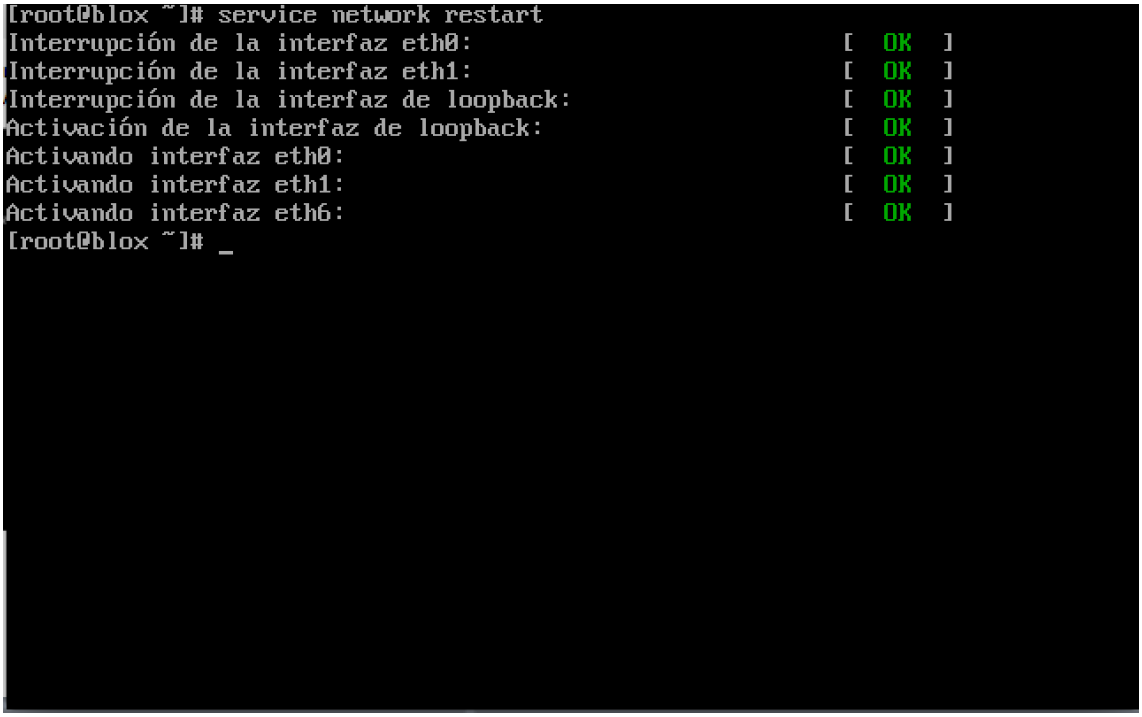


Figura 5. 34 Reinicio de servicio de Red

Se debe revisar las interfaces configuradas mediante el comando **ifconfig.** Como se muestra en la Figura 5.35.

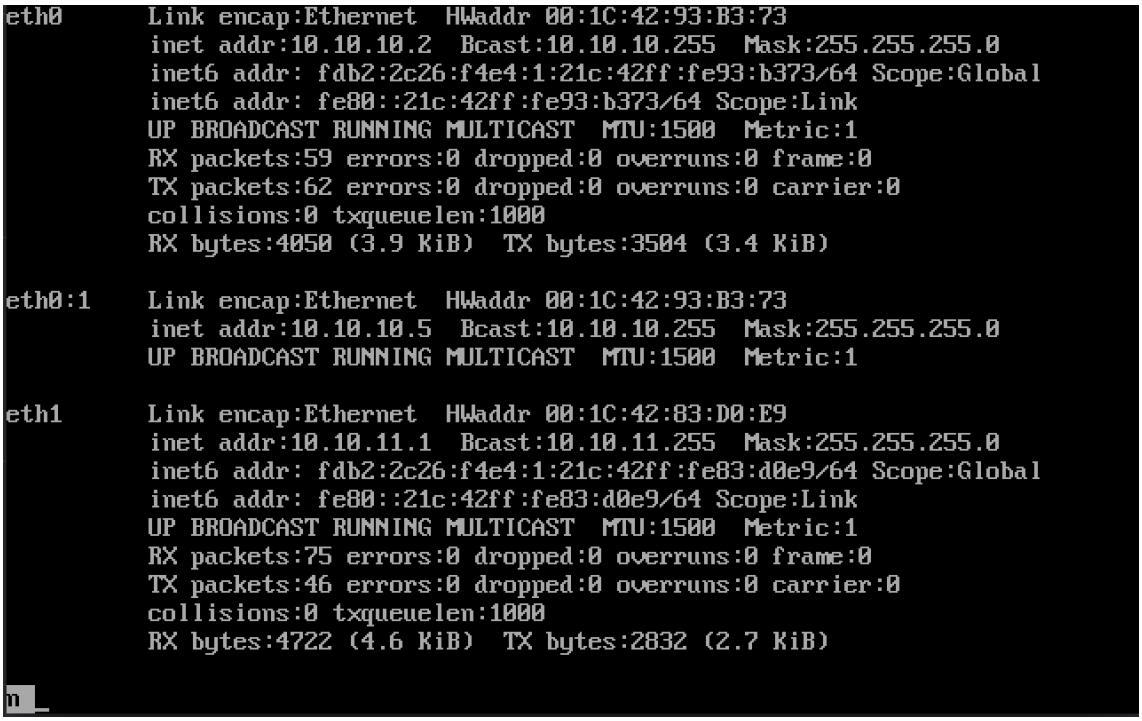


Figura 5. 35 Actualización de Interfaces de Red

5.5.3 Instalación de FreeBlox.

Paso 1.- Copiar **freeblox-0.9.0-22.x86_64.rpm** en BLOX server. Como se muestra en la Figura 5.36.

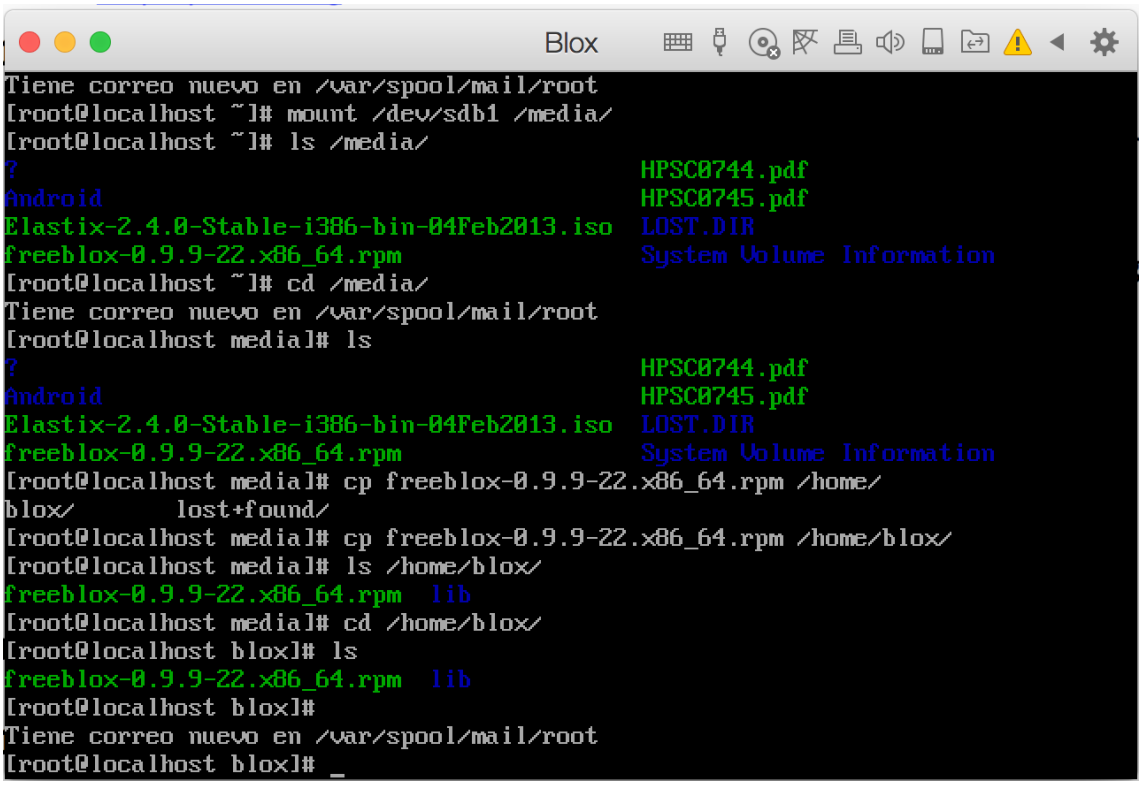


Figura 5. 36 Copia de Instalador Freeblox

Paso 2.- Para iniciar sesión con Blox Server, ir a la carpeta FreeBlox-rpm y ejecutar los comandos que se muestran en la Figura 5. 37.

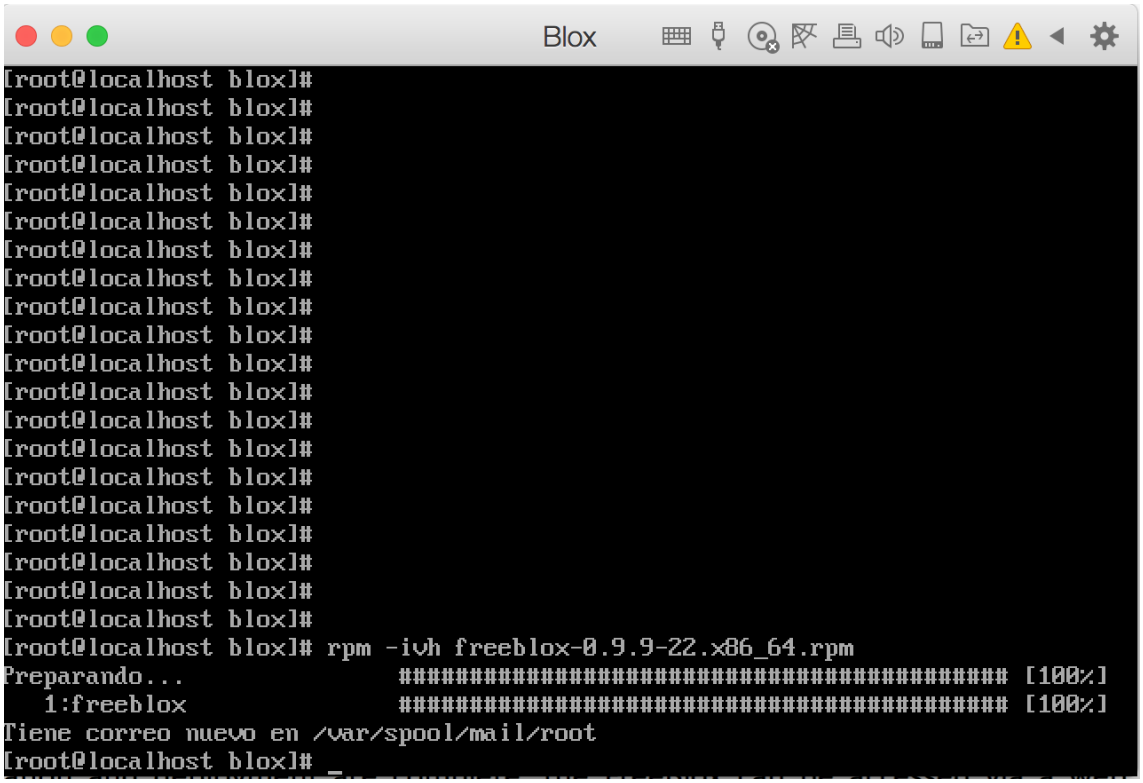


Figura 5. 37 Instalación de FreeBlox

Paso 3.- Reiniciar el sistema usando el comando **reboot**. Y empezar a utilizar la interfaz gráfica FreeBlox para la configuración de Blox SBC. Como se muestra en la Figura 5.38

Figura 5. 38 Login de Blox mediante FreeBlox

5.5.4 Configuración de BLOX

5.5.4.1 Dashboard

El Dashboard es el primer inicio de sesión, es la interfaz gráfica de usuario Web que proporciona información general del estado de la configuración de FreeBlox (Figura 5. 39) presenta:

- El status del sistema, el uso de memoria, uso de flash y el uso de la CPU.
- El panel superior muestra el firmware.
- Status de Red, IP, MAC, LAN, WAN y puerta de enlace del dispositivo.
- Panel de resumen de alertas de seguridad, alarmas.
- Status DPI.

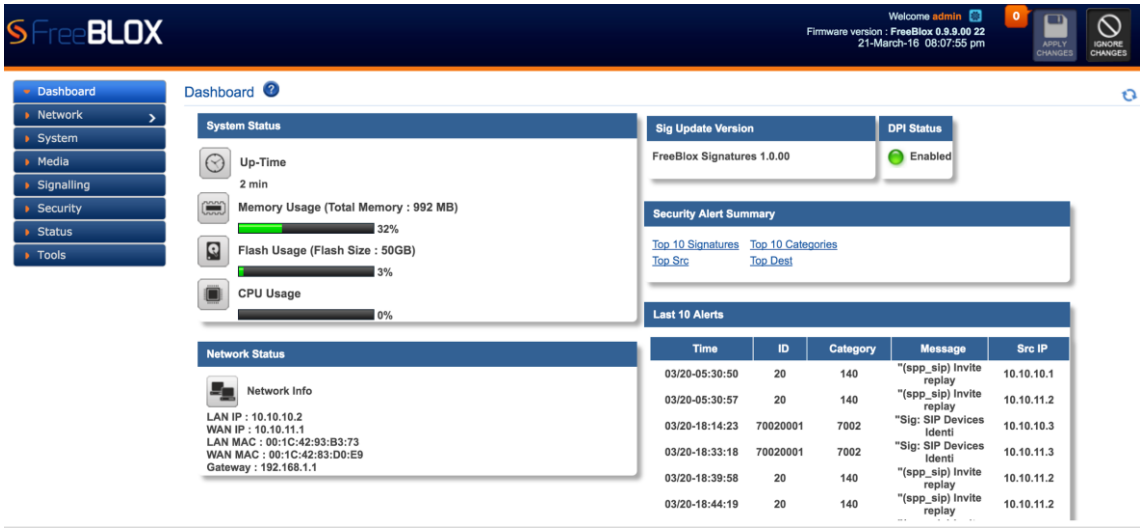


Figura 5. 39 Dashboard

5.5.4.2 Configuración de red.

Proporciona información detallada acerca de las interfaces, rutas y dispositivos de acceso a FreeBlox.

La interfaz LAN transporta la señalización SIP, que entra y sale de SBC Blox. Las interfaces de Blox son los adaptadores Ethernet, permitiendo al usuario configurar el nombre de host, configuración IP en modo estático, dirección IP/ máscara, puerta de enlace y DNS, permite además activar o desactivar el acceso SSH al dispositivo, permitir o denegar la solicitud ICMP ping, como se muestra en la siguiente Figura 5.40

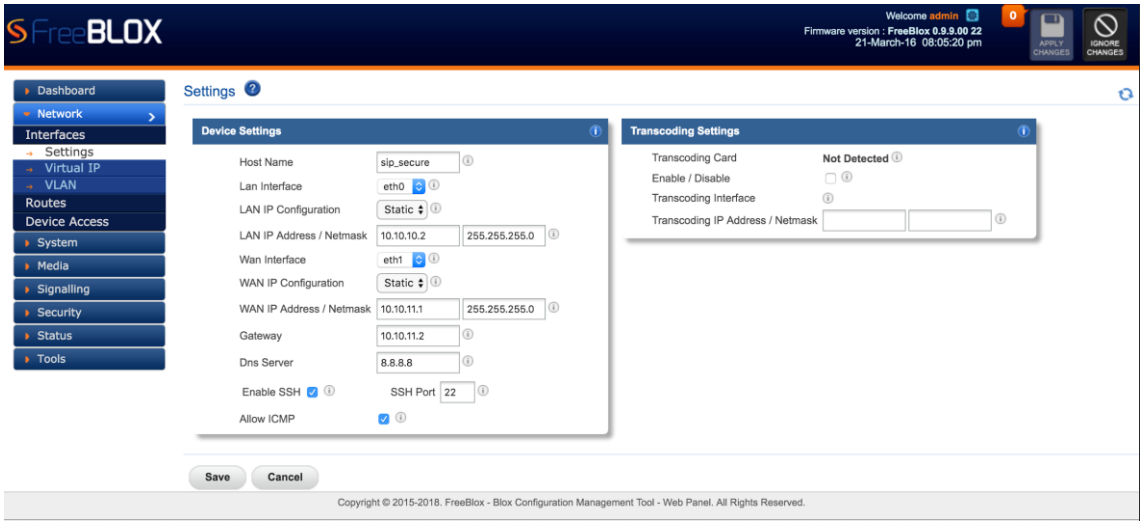


Figura 5. 40 Configuración de Red

La configuración de red, para el segmento LAN en la interfaz **eth0** se configura la IP 10.10.10.2 y para el segmento WAN la interfaz **eth1** se configura la IP 10.10.11.1.

En caso de contar con tarjeta de transcodificación, Blox la detecta, en este caso el usuario activa o desactiva el funcionamiento de la misma, si es activada se puede configurar una IP y máscara de red. (La tarjeta de transcodificación es mandatoria para SRTP y T38)

En Blox principalmente la configuración de red consta de 4 interfaces, interfaz interna, externa, media o también denominada virtual de medios de comunicación y transcodificación.

El usuario puede configurar la IP virtual y VLAN para acceder a FreeBlox.

5.5.4.3 IP Virtual

Es una dirección IP asignada para múltiples aplicaciones que se alojan en un servidor o múltiples nombres de dominio, en lugar de asignarlo a una tarjeta de red específica (NIC).

Como se observa en la Figura 5.41 se ha creado una interfaz en este caso para la administración siguiendo los parámetros de la Tabla 5.10.

Name	Especificar el nombre de la dirección IP de referencia para el usuario.
Interfaces	Seleccionar la interfaz de la lista en la cual se desea crear la IP virtual. Puede ser en la interfaz WAN/LAN.
IP Address	Configurar la dirección IP virtual.
Netmask	Configurar la máscara de red de IP Virtual.
Description	Proporcionar la descripción para la IP Virtual (Opcional)

Tabla 5. 10 Parametrización de IP Virtual.

La IP que fue configurada para IP Virtual es la **10.10.10.5,** en la interfaz **eth0** que pertenece al segmento **LAN**

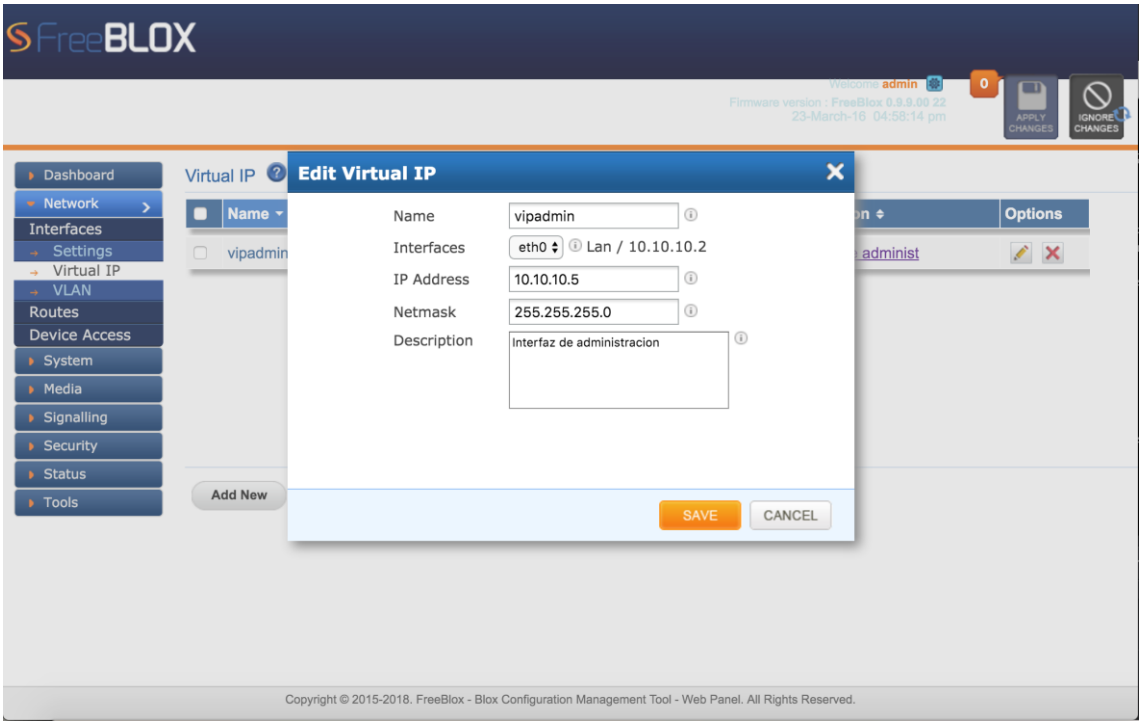


Figura 5. 41 Configuración Virtual IP

5.5.4.4 Configuración Date/Time.

FreeBlox permite la configuración manual y NTP, para registro de hora y fecha. De esta manera se puede seleccionar la zona horaria o la configuración NTP, añadir la IP o dominio del servidor NTP.

Se ha optado por configurar un servidor NTP, para la sincronización de FreeBlox con la central como se muestra en la Figura 5.42. El cliente FreeBlox apunta a la dirección **192.168.1.2** el cual es la central Elastix y para la configuración es el servidor NTP.

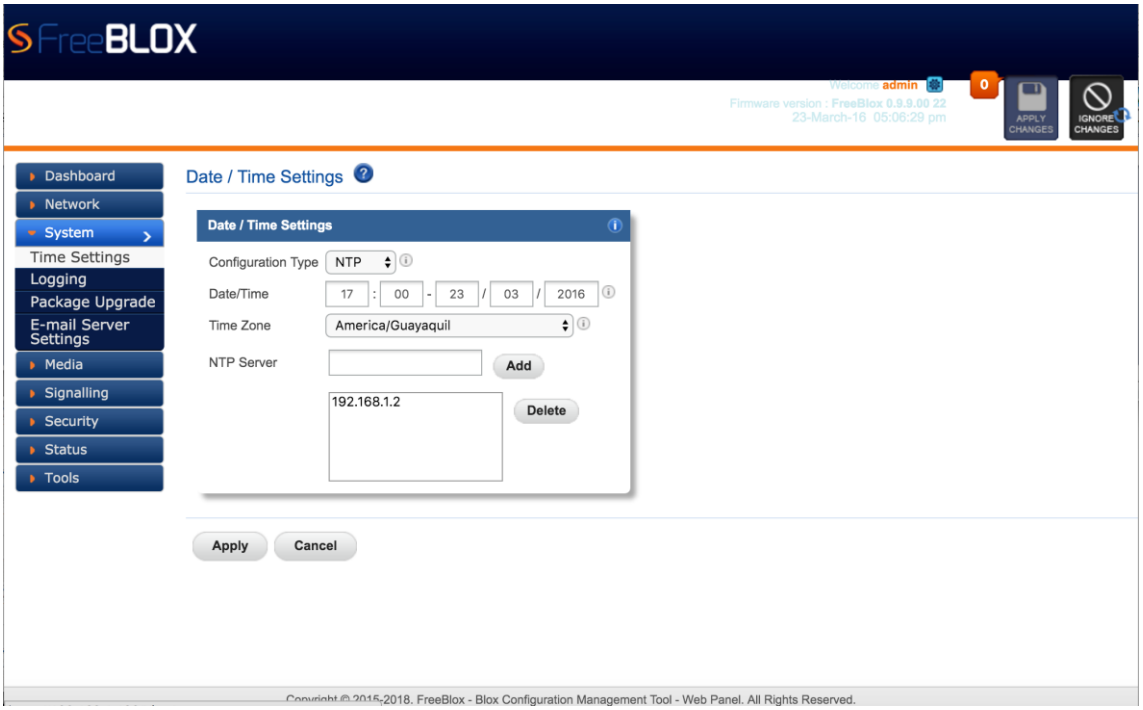


Figura 5. 42 Configuración NTP cliente

A continuación se indica la configuración del servidor NTP **192.168.1.2** configurado en Elastix, en el archivo **ntp.conf.** Como se observa en la Figura 5.43

```
# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery

# Permit all access over the loopback interface.  This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
restrict -6 ::1

# Hosts on local network are less restricted.
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 0.centos.pool.ntp.org
server 1.centos.pool.ntp.org
server 2.centos.pool.ntp.org
server 3.centos.pool.ntp.org
server 192.168.1.2

#broadcast 192.168.1.255 key 42         # broadcast server
#broadcastclient                        # broadcast client
```

Figura 5. 43 Configuración NTP server Elastix.

5.5.4.5 Perfil Media

El perfil de Media se encarga de la canalización de media a la entrada y salida de Blox, permitiendo al usuario configurar el intervalo de puertos de comunicación, transcodificación o media en general. Este perfil permite realizar la transcodificación de un códec a otro, además de las funciones de media de RTP /SRTP streaming RTP.

Para la configuración del perfil de media se realiza la parametrización que se observa en la Tabla 5.11.

Name	Introduzca el nombre para el perfil.
Description	Proporcionar una breve descripción para el perfil de media. (Opcional)
External Interface	Seleccionar la dirección IP WAN para enviar fuera del SBC.
Internal Interface	Seleccionar la dirección IP LAN para recibir por el SBC
Media Interface	El usuario selecciona la Interfaz de Media, que puede ser una IP virtual, o una IP de transcodificación.
RTP Port Start	Puerto de inicio para RTP (4000)
RTP Port End	Puerto fin para RTP (4500)
TOS	Cualquier valor entre (0-63) Por defecto 10.

Tabla 5. 11 Parametrización de Perfil Media

En este caso se configuró FreeBlox para comunicación de media en general es decir RTP. La cual es una subinterfaz de interfaz **eth0:1** correspondiente a la interfaz LAN en la IP **10.10.10.6.** Como se observa en la Figura 5.44

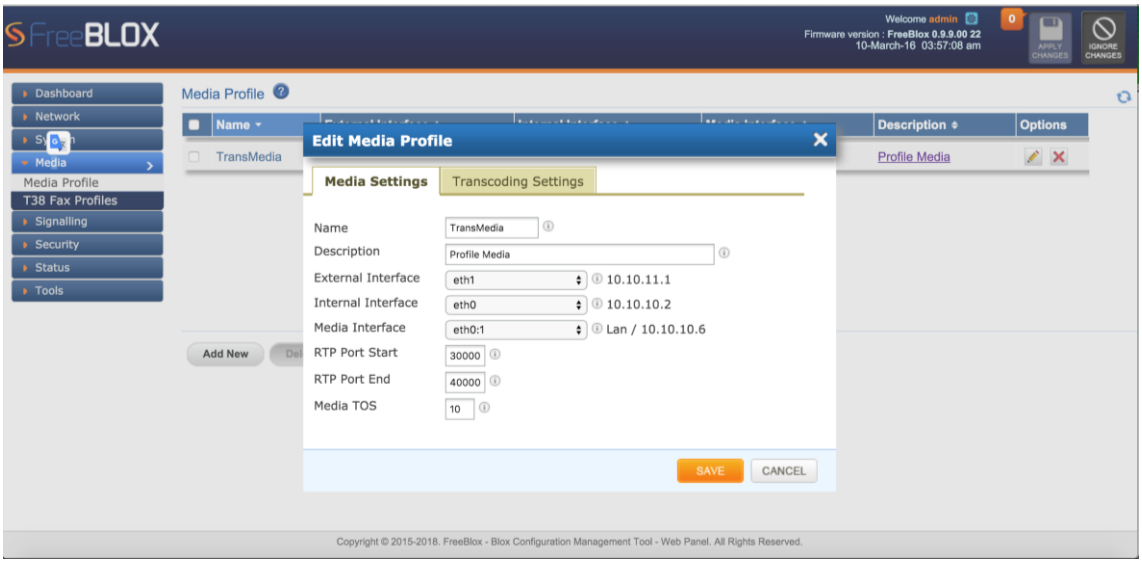


Figura 5. 44 Configuración Perfil Media

5.5.4.6 Señalización

En la sección de señalización permite configurar perfiles SIP, configuración Trunk, usuarios móviles, Least Cost Routing y TLS.

- Perfil SIP.

Un perfil SIP contiene un conjunto de atributos SIP que se asocian a Blox. El perfil SIP utiliza una configuración de los puntos finales externos para conectarse con Blox, enlaza una dirección IP, el puerto y otros parámetros relacionados a SIP.

Contiene la configuración SIP UA. FreeBlox puede ser configurado para múltiples UA cada uno con una configuración diferente, por lo tanto un conjunto de diferentes IP y puerto para cada uno. La configuración se realiza en base a la Tabla 5.12.

Name	Introduzca el nombre para el perfil.
Description	Indicar una breve descripción para el nombre del perfil. (Opcional)
Interfaces	Seleccionar las interfaces respectivas en la lista desplegable para las redes interna (LAN) y externa (WAN). Ejemplo: LAN : eth0 10.10.10.2 WAN: eth1 10.10.11.1

SIP Protocol/Port	FreeBlox en el perfil SIP permite al usuario seleccionar múltiples protocolos (UDP, TCP y TLS) que pueden estar disponibles en la lista desplegable de protocolo. Y especifique el puerto SIP.
NAT Settings /IP Address	El usuario debe proporcionar algunas direcciones IP externas y se usa para comunicarse con (WAN). El usuario puede seleccionar la configuración de NAT como STUN o la dirección IP estática STATIC de la lista desplegable, en caso de contar con NAT.
Server Certificates	Si el protocolo TLS SIP está activado, el certificado del servidor debe activarse. El usuario puede seleccionar el certificado del servidor de la lista desplegable.
Domain	Es un nombre de dominio SIP.
Enable Keep alive	El usuario puede activar o desactivar el Keep Alive
Keep alive internal	Este campo espera un tiempo para que el mensaje de registro SIP se encuentre activo. Se especificará en el rango de 60 a 360.
SIP TOS	El usuario puede configurar el tipo de servicio de bytes (TOS) en los paquetes IP salientes para varios protocolos. El byte TOS es utilizado por la red para proporcionar un cierto nivel de calidad de servicio (QoS), incluso si la red está congestionada con el resto del tráfico.
Allow (IP:PORT)	El usuario puede seleccionar la IP interna (LAN) IP / externa (WAN), con su respectivo puerto.

Tabla 5. 12 Parametrización perfil SIP.

Para este propósito en FreeBlox se han configurado 2 perfiles SIP como se muestra en la Figura 5.45.

Figura 5. 45 Perfiles SIP

Se configura un perfil SIP para la interfaz **eth0.** Como se observa en la Figura 5.46.

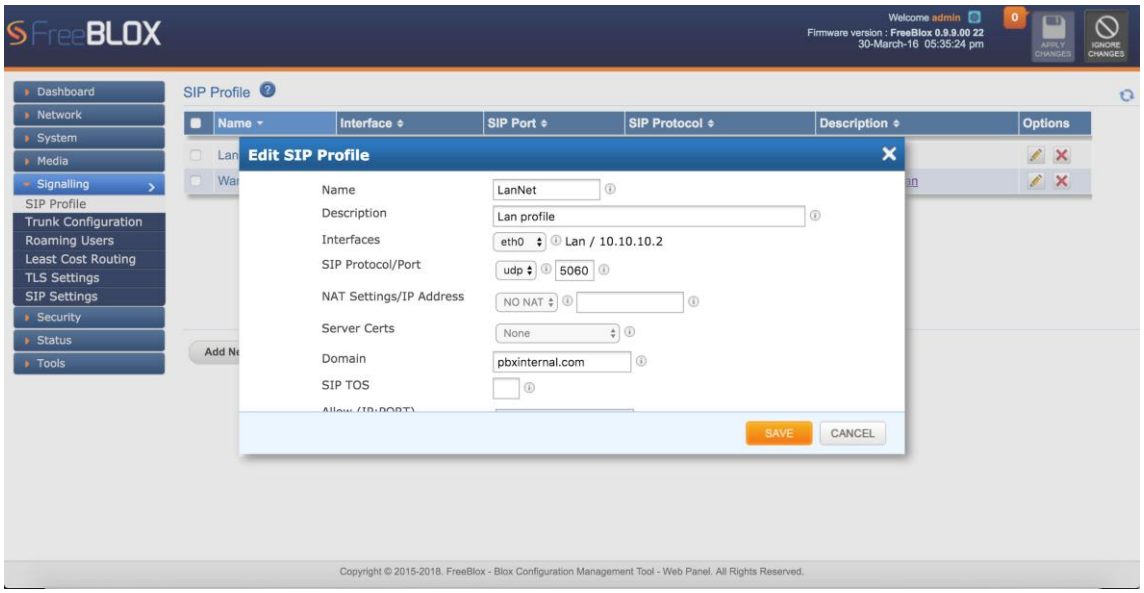


Figura 5. 46 Perfil SIP LAN

Se configura un perfil SIP WAN para la interfaz **eth1.** Como se observa en la Figura 5.47

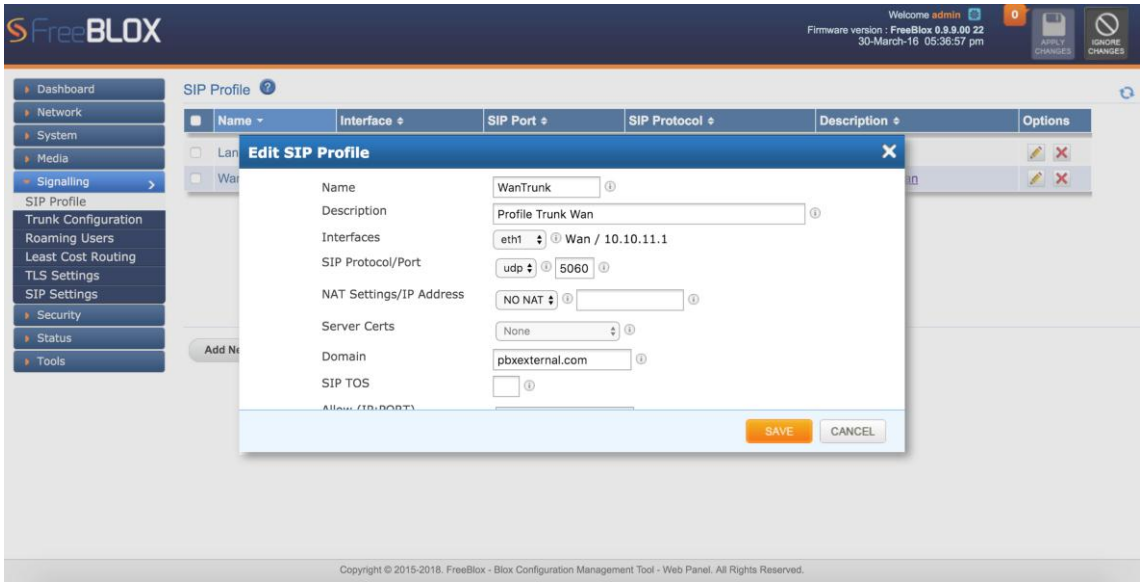


Figura 5. 47 Configuración SIP WAN

- Configuración Trunk.

SIP Trunks son utilizadas para conectar FreeBlox a un remoto SIP provider / User Agents. SIP Trunk puede ser utilizado para comunicación entre SIP carriers o con IP- PBX.

Las troncales SIP están obligadas a crear perfiles SIP, además un perfil SIP se puede conectar a varios Troncales SIP. En la Figura 5.48 se observa la troncal configurada.

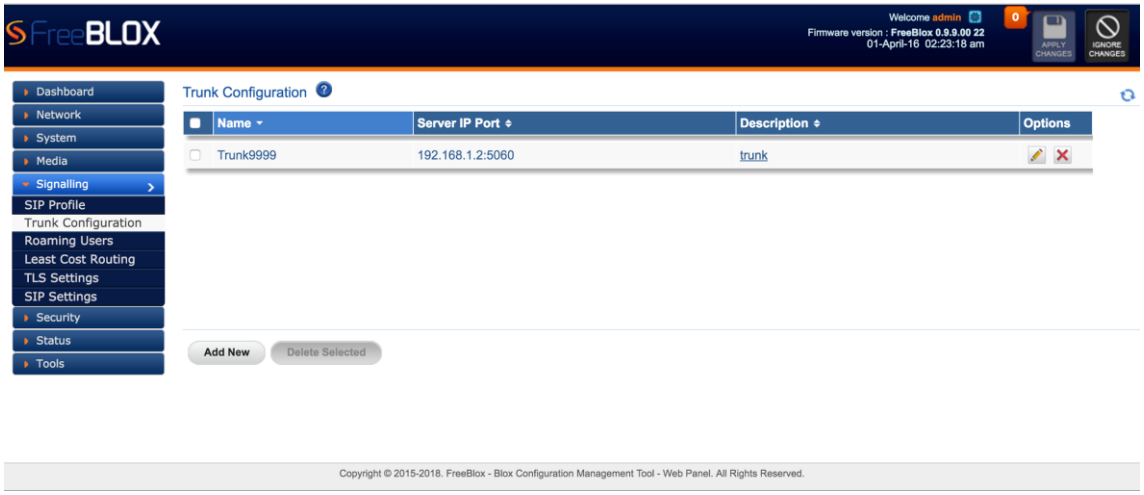


Figura 5. 48 Troncalización

A continuación en la Figura 5.49 se muestra la configuración de la troncal, siguiendo los parámetros de la Tabla 5.13.

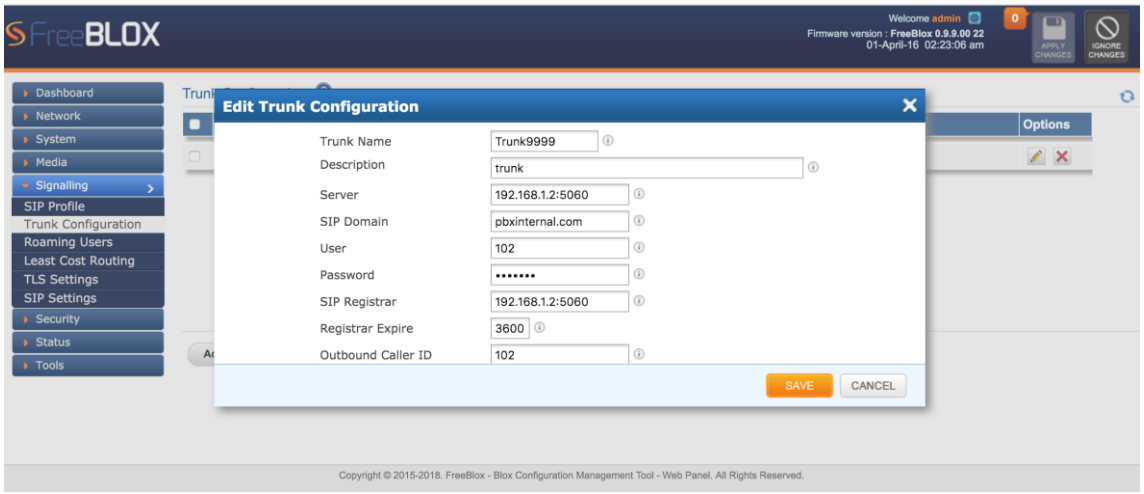


Figura 5. 49 Configuración Troncal.

Trunk Name	Introduzca el nombre para identificar la troncal.
Description	Descripción de la troncal.
Server,SIP(Domain/ IP:PORT)	Dirección IP y puerto del servidor para registrarse
User	El nombre de usuario ya sea el proporcionado por el proveedor o cualquier extensión para autenticar con la configuración de la troncal.
Password	Introducir la contraseña para la autenticación.
SIP Register (IP:Port)	Se especifica la dirección IP con el puerto de la PBX.
Register Expire	Tiempo de espera de caducidad para el registro de la troncal SIP. En un rango 360 -3600
Outbound Caller ID	Configurar el número de identificación de llamadas que se aplicaría para las llamadas salientes a través de la troncal.
Outbound Proxy URI	Opcional, configurar dirección IP y puerto proxy, garantizando que todos los paquetes SIP que se envían sean a través del proxy especificado.
User Agent	Especifique el usuario SIP Agente utilizado por defecto Blox-<versión>.
Internal SIP Profile	Perfil LAN, el usuario puede seleccionar el perfil SIP interno de la lista desplegable para PBX
External SIP Profile	Perfil WAN, el usuario puede seleccionar el perfil SIP externo de la lista desplegable para el proveedor SIP.

Media Profile	En este campo el usuario puede seleccionar el Perfil media puede ser un general u otro en caso de tener un perfil de transcodificación.
Media Encryption (LAN)	La función de cifrado de medios usando RTP seguro (SRTP) proporciona la capacidad de cifrar paquetes de media LAN. SRTP es un perfil de seguridad para RTP que añade confidencialidad, autenticación de mensajes, y protección de este protocolo.
Media Encryption (WAN)	La función de cifrado de medios usando RTP seguro (SRTP) proporciona la capacidad de cifrar paquetes de media WAN. SRTP es un perfil de seguridad para RTP que añade confidencialidad, autenticación de mensajes, y protección de este protocolo.
T38 Profile	Opcional, seleccionar el perfil T38 del menú desplegable en caso de disponer.(FAX)
Add Prefix	Es un campo opcional, en el que el usuario puede añadir un número como prefijo para la troncal en particular. Especificar añadir prefijo antes del número marcado.
Strip Digits	0 (Permite al usuario especificar el número de dígitos que se extrae del número marcado)
Allow Inbound	Marcar para permitir llamadas entrantes.
Inbound URI	Proporcionar una dirección IP de entrada con su respectivo puerto / PBX para la troncal
Max Inbound	El usuario puede limitar el número de llamadas entrantes, que pueden estar llegando a través de ese tronco en particular
Allow Outbound	Este campo permite al usuario activar o desactivar las llamadas de salida.
Max Outbound	El usuario puede limitar el número de llamadas salientes que pasan por la troncal en particular.

Tabla 5. 13 Parametrización Troncal.

Media Encryption LAN / WAN y el perfil T38 sólo se pueden utilizar si la tarjeta de transcodificación está conectado con el Blox. Por lo que estos campos están desactivados para perfil de soporte.

A continuación se muestra la configuración en la PBX e ISP para realizar el SIP TRUNK como se muestran en las Figuras 5.50 y 5.51 respectivamente.

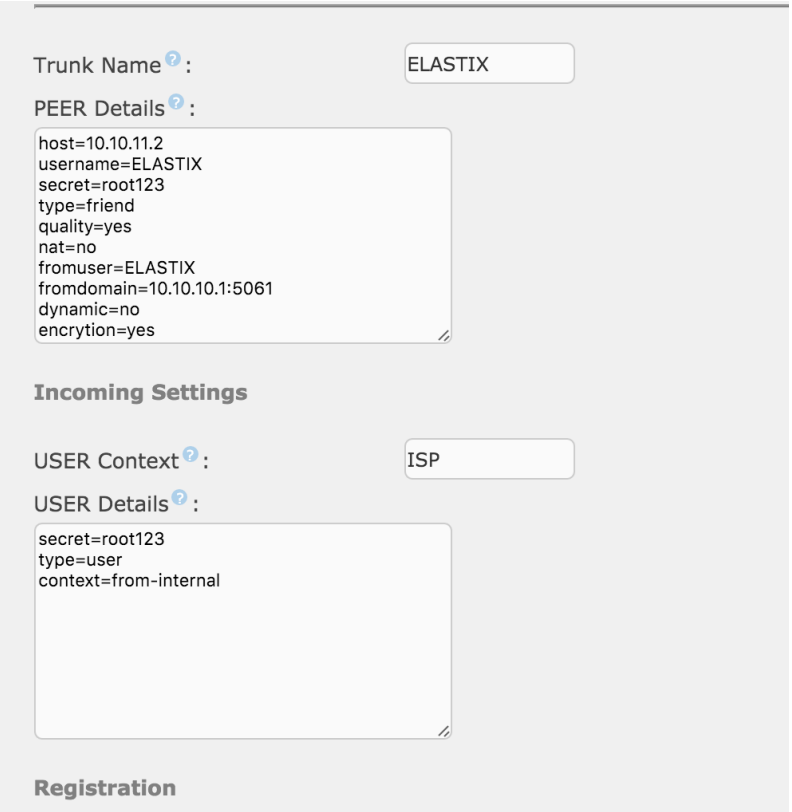


Figura 5. 50 Configuración Trunk para PBX

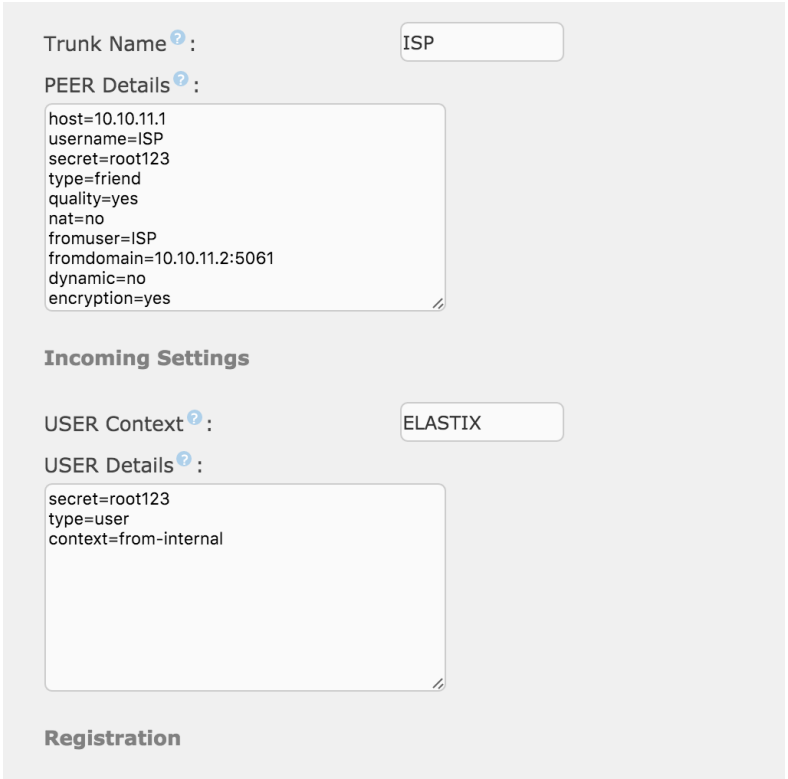


Figura 5. 51 Configuración SIP Trunk para ISP

5.5.4.7 Configuración TLS

- Device Root CA

En esta sección, previo a la emisión de certificados se debe realizar los pasos que se detallan a continuación:

- Configuración de host en los clientes y servidor
- Configuración de servidor NTP.

Previamente este documento muestra la configuración del servidor NTP y el cliente NTP.

A continuación se muestra como realizar la configuración del servidor como se observa en la Figura 5.52 mediante la edición del archivo **/etc/hosts**

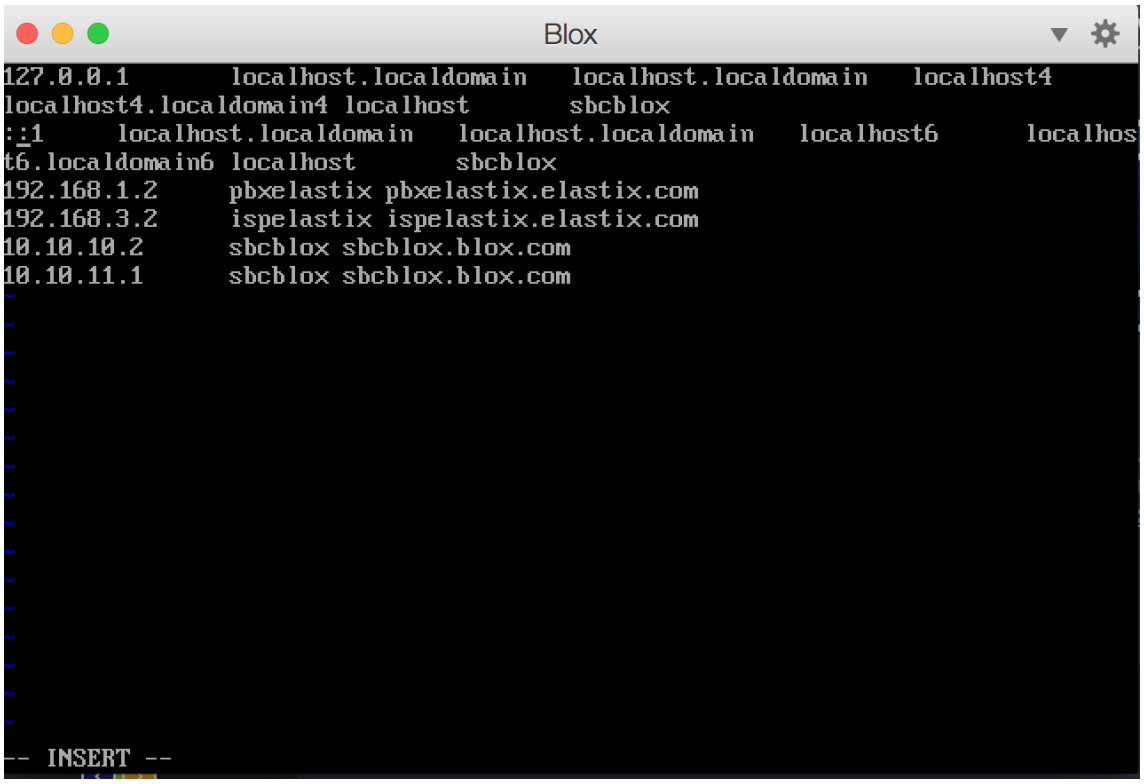


Figura 5. 52Configuración de /etc/hosts Blox

Al editar el archivo **/etc/hosts** se procede a realizar la configuración de los clientes como se muestran en las Figura 5.53 y Figura 5.54, tanto de la PBX como del ISP respectivamente.

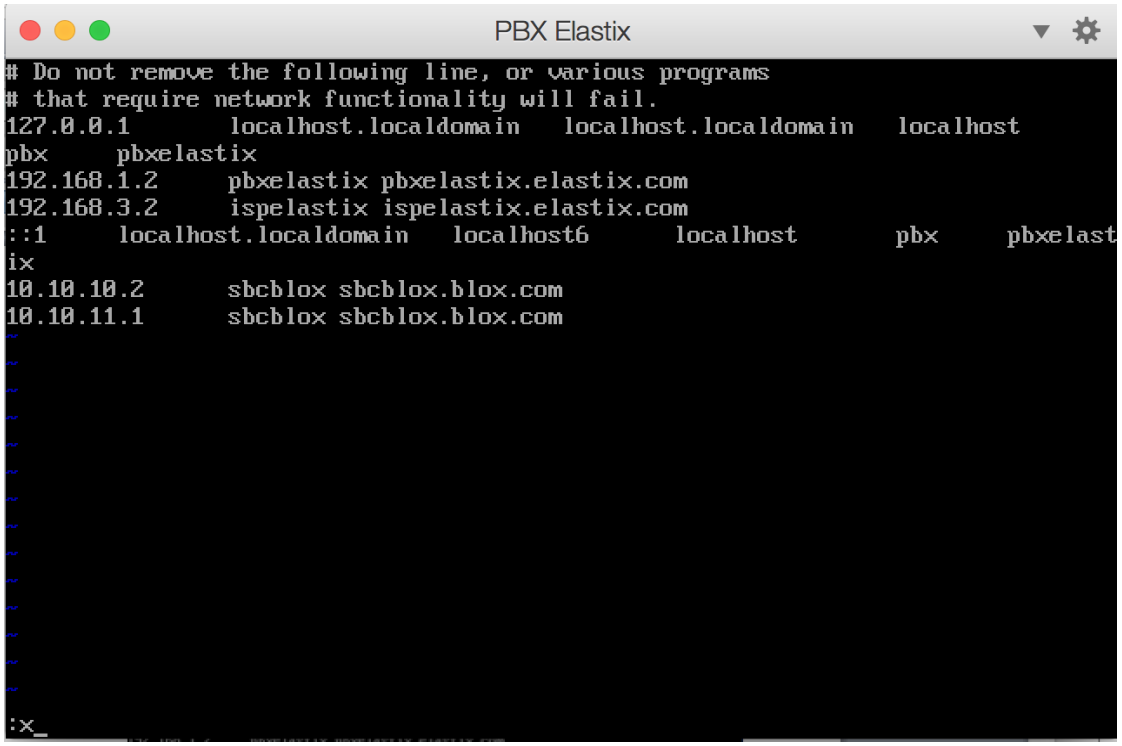


Figura 5. 53 Configuración de /etc/hosts/ PBX

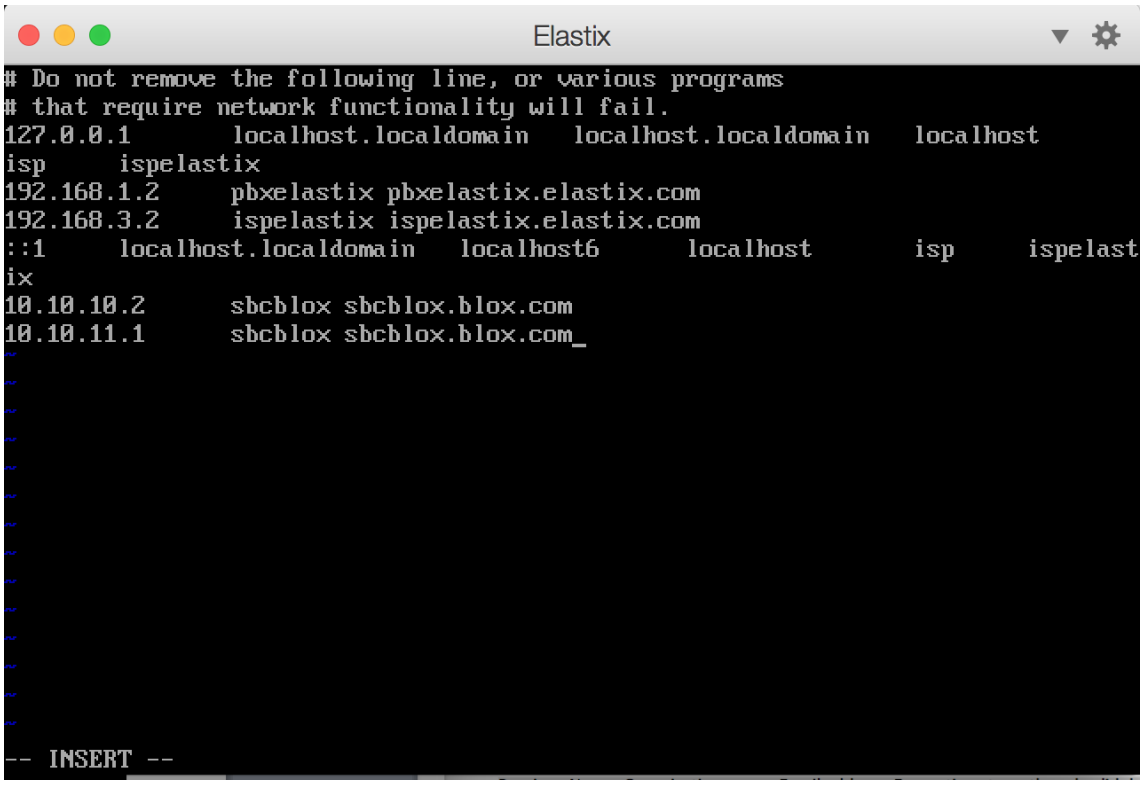


Figura 5. 54 Configuración /etc/hosts ISP

El usuario puede cargar un archivo CA o generar un certificado digital, como se indica en la Figura 5.55.

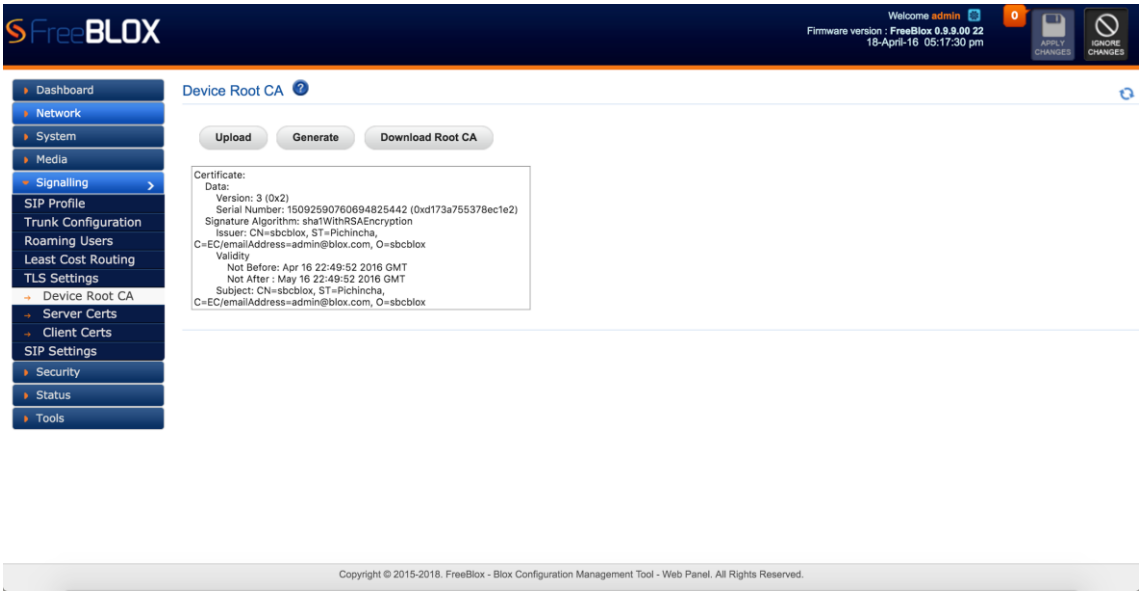


Figura 5. 55 Autoridad Certificadora (CA)

A continuación se muestra la configuración de CA, en el cual se ingresan parámetros como el Common Name que es el hostname que apunta al servidor Blox, configurado previamente, Country Name, en la cual se setea con 2 dígitos el identificador de país, Provincia, Organization Name, Email Address, Encryption Strength y Valid days que son los días de validez del certificado. Como se muestra en la Figura 5.56.

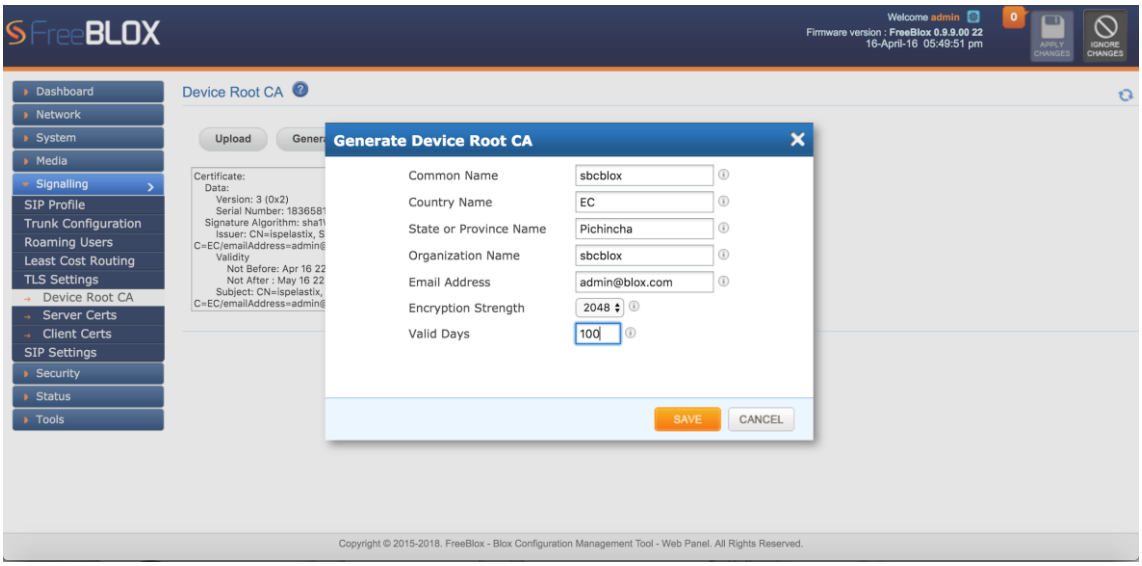


Figura 5. 56 Configuración de la Autoridad Certificadora.

- Certificado del servidor.

Al igual que se puede subir la CA, el usuario de igual manera puede subir el certificado del servidor sin embargo también se puede generar como se muestra en la Figura 5.57 y es almacenado como se observa en la Figura 5.58

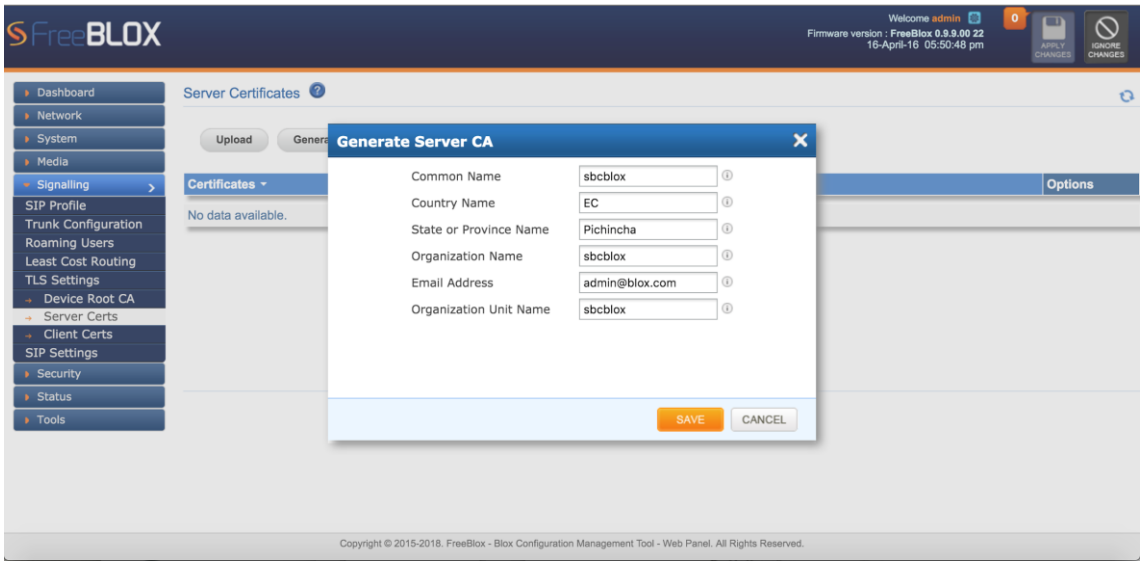


Figura 5. 57 Certificado del servidor.

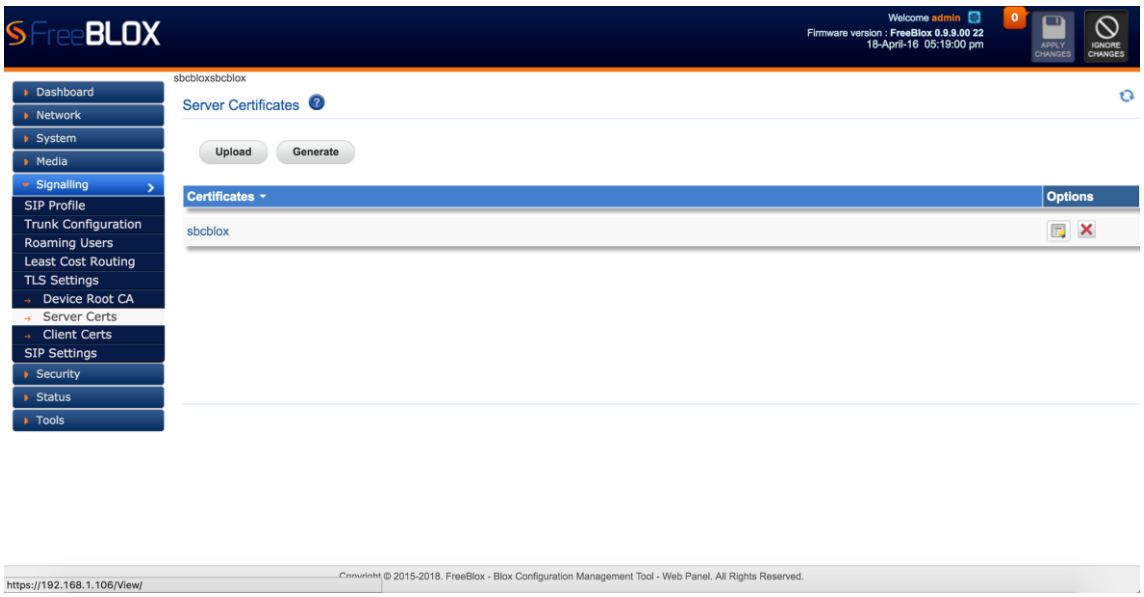


Figura 5. 58 Certificado del Servidor.

- Certificado del Cliente

El certificado del cliente, se configura ingresando parámetros como el Common Name que es el hostname que apunta al ISP, configurado previamente, Country Name, en la cual se

setea con 2 dígitos el identificador de país, Provincia, Organization Name, Email Address, Encryption Strength y Valid days que son los días de validez del certificado. Como se muestra en la Figura 5.59 y es almacenado como se observa en la Figura 5.60

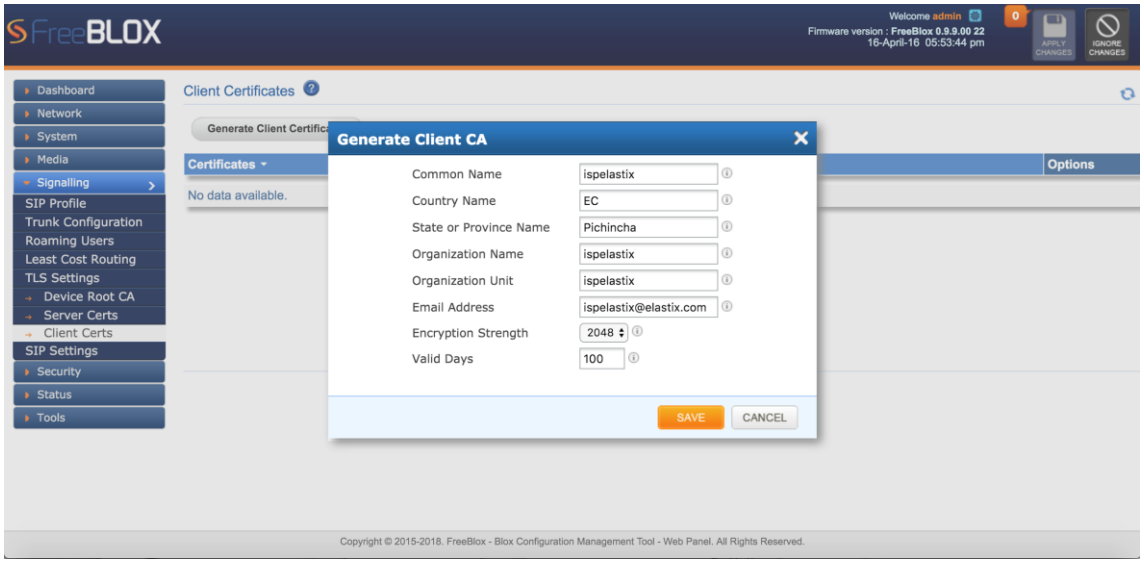


Figura 5. 59 Certificado Cliente.

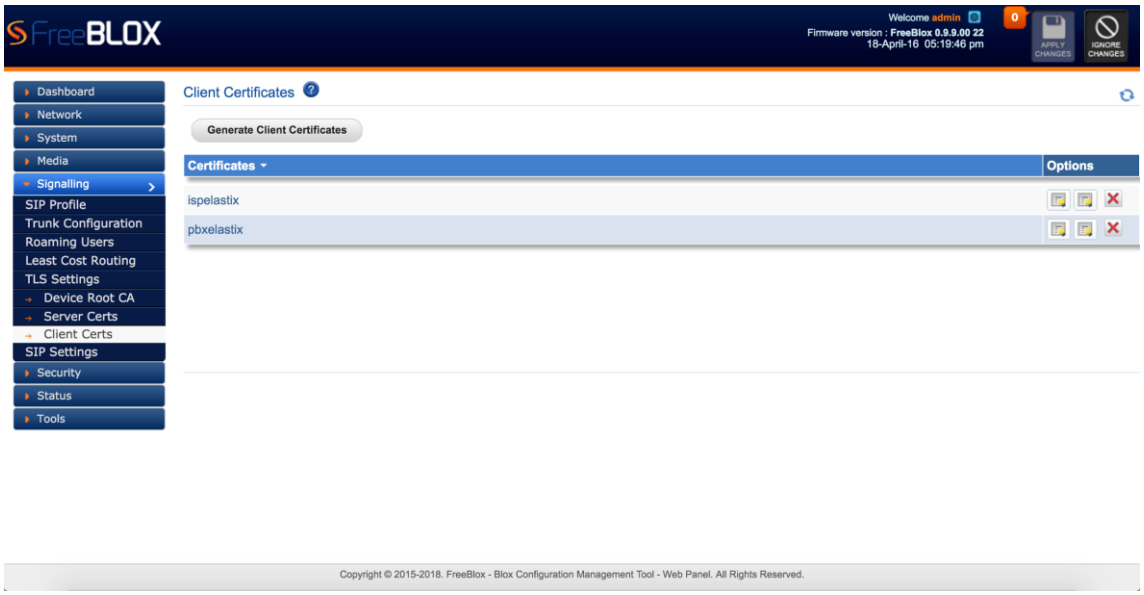


Figura 5. 60 Certificados generados para Clientes.

Se procede a configurar TLS v1.1 en el terminal softphone de Zoiper, como se muestra en la Figura 5.61, la opción TLS y seleccionar el certificado del cliente, para cifrar el enlace entre el Blox y el ISP.

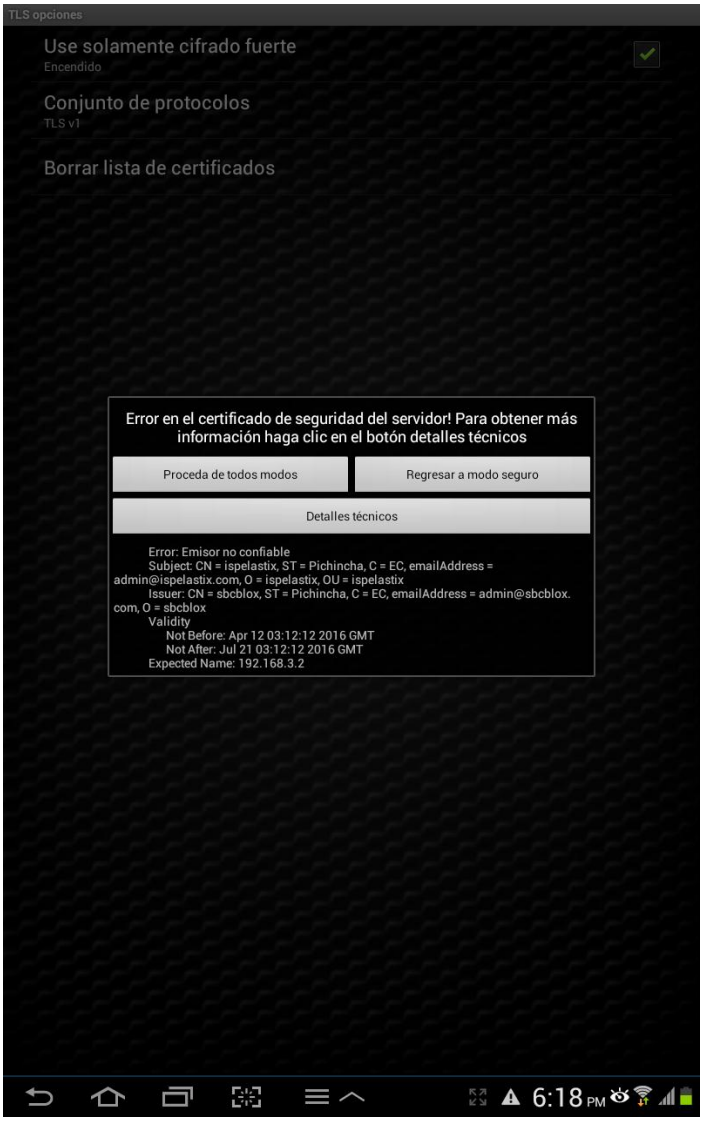


Figura 5. 61 Certificado TLS en el terminal Zoiper.

Para realizar la configuración completa se configura los perfiles SIP, por lo general sólo en la WAN agregando el protocolo, puerto TLS y el certificado del servidor previamente generado en este caso **sbcblox.** Como se observa en la Figura 5.62

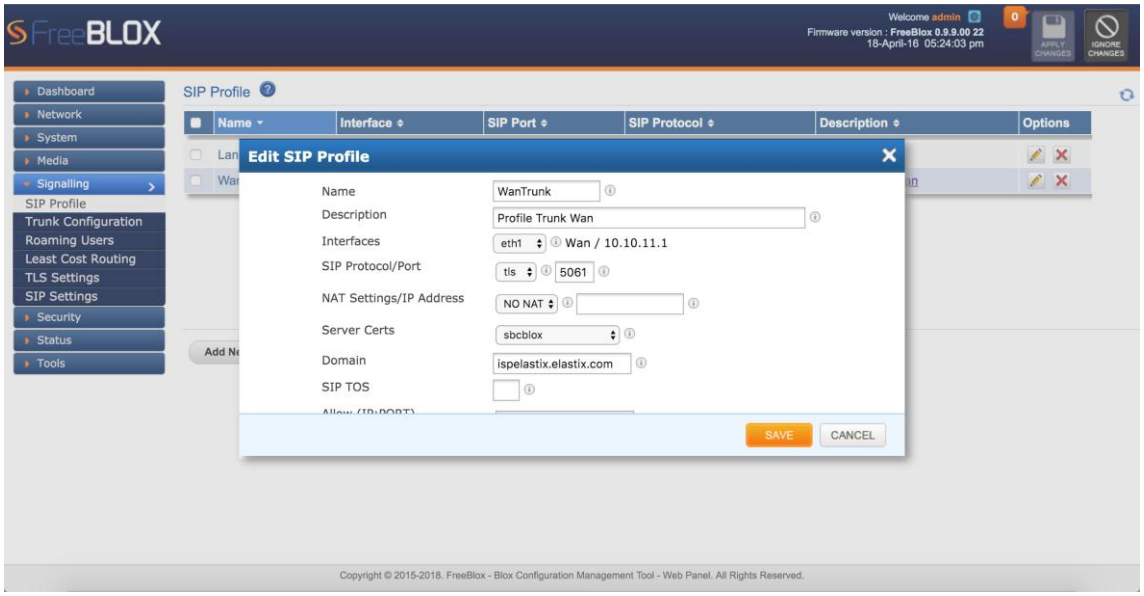


Figura 5. 62 Configuración perfil SIP con TLS

A continuación en la Figura 5.63 se observa los perfiles SIP tanto LAN como WAN con cifrado TLS, sin embargo es recomendable por temas de procesamiento que solo se cifre en enlace WAN.

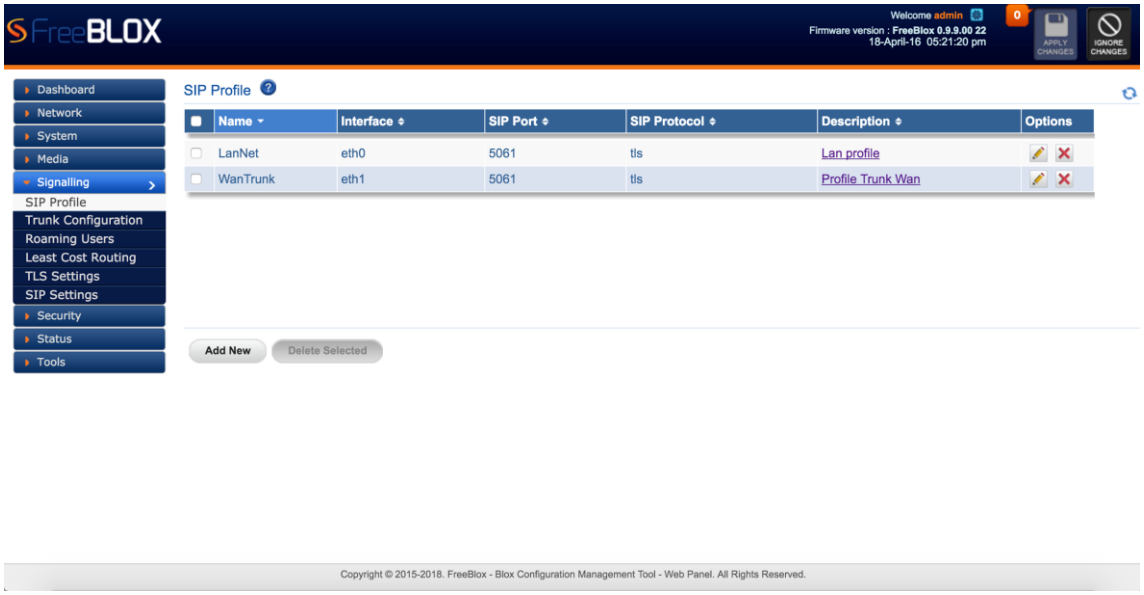


Figura 5. 63 Perfiles SIP con TLS.

Por último se comprueba mediante Wireshark la configuración TLS. Como se observa en la Figura 5.64, la cual presenta el establecimiento del protocolo TLS v 1.1.

TLS v1.1 se encuentra definido en el RFC 4346 [9].

Por tanto se demuestra que la comunicación se encuentra protegida con TLS, por tanto no se muestra las extensiones que levantan la conversación.

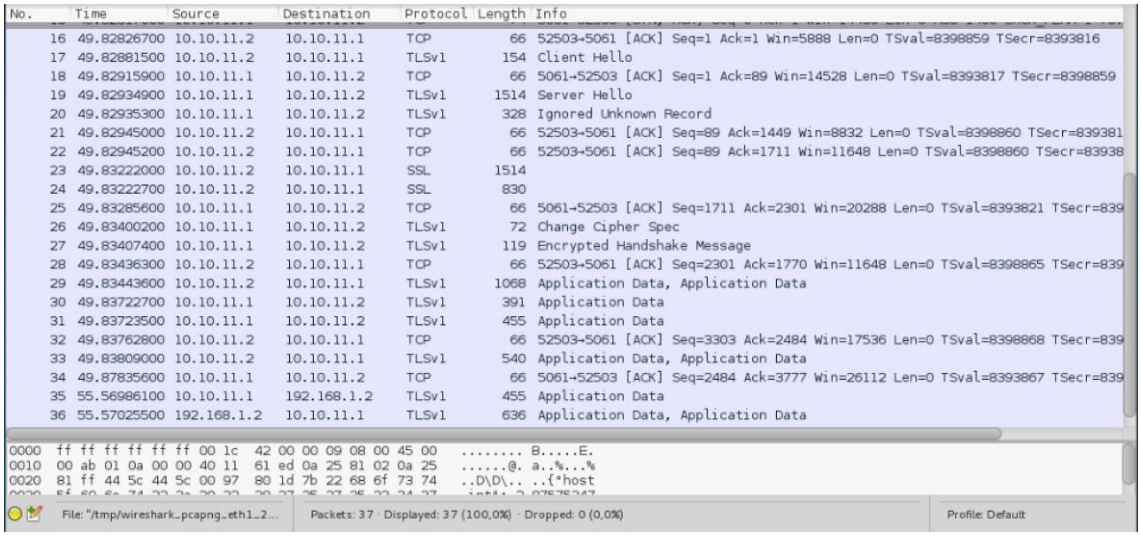


Figura 5. 64 Conversación cifrada con TLS

5.5.4.8 Seguridad.

- Detección de ataques SIP

Blox tiene un módulo de detección de ataques SIP, mediante **SIP ATTACKS DETECTION** en el módulo de **Security,** permite configurar reglas para inspección de paquetes SIP. El usuario puede activar o desactivar la inspección contra determinada categoría, de igual manera se puede tomar medidas al respecto dependiendo del ataque que esté siendo víctima la central VoIP.

Las posibles acciones que FreeBlox puede ejecutar se presentan en logs de alerta y bloqueo de paquetes que contienen un vector de ataque y la lista negra de la IP atacante para la duración dada.

La duración del bloqueo en la cual el atacante necesita ser bloqueado se realiza por categoría.

A continuación se muestra la configuración empleada para la detección de ataques SIP. Como se observa en la Figura 5.65 además se explican cada una de sus funcionalidades según se muestra en la Tabla 5.14.

Figura 5. 65 Ataques de detección SIP.

CATEGORIA	DESCRIPCIÓN	OPCIONES DE USUARIO CONFIGURABLE.
Reconnaissance Attacks	Puede considerarse como el primer paso de atacar a cualquier sistema o red. En este un hacker intenta aprender información acerca de la red, puertos, servicios, sistema operativo, versión de la aplicación dispositivos SIP.	-
SIP Devices Scanning	El atacante escaneará los puertos PBX para determinar los dispositivos que se encuentran en la misma. Con esta información puede aprovechar las vulnerabilidades **3rd Party Vendor Vulnerabilities**. FreeBlox no responderá a estas consultas.	-

SIP Extensions Discovery	El intruso solicitará a la PBX divulgar el rango de números de extensiones. Con esta información, se puede probar diferentes contraseñas para tomar el control de las extensiones. FreeBlox no responderá esa consulta.	**Invalid SIP User Registration Attempts/Duration**
Multiple Authentication Failures/ Brute Force Password Attempt	El intruso intentará conectarse con diferentes nombres de usuario y contraseñas por fuerza bruta. Una vez que lo logra tendrá el control de esa extensión. FreeBlox puede bloquear, mostrar logs o la IP en lista negra por un periodo de tiempo, si excede el número autorizado de pruebas.	**Failed Authentication Attempts/Duration**
Ghost calls Attempt	El intruso puede generar llamadas a una extensión y se verá llamadas procedentes de la misma extensión. Su objetivo es bloquear la PBX lo que puede provocar es una interrupción en la comunicación. FreeBlox puede bloquear, registrar o mostrar en lista negra la IP por un periodo de tiempo, si excede el número de intentos autorizados.	**No of Anonymous Invite**

SIP Protocol Compliance	Este tipo de ataques se refiere al uso de algún tipo de herramienta automatizada como SIPP para generar escritura falsa, donde algunos de los campos más importantes pueden ser modificados en función a la longitud, campos como cabecera SIP y cuerpo. También puede ser útil en el manejo correcto de la sesión entre puerto SIP y este protocolo.	
SIP Anomaly Attacks	El DPI (Deep Inspection Packet) es un inspeccionador de tráfico SIP. Las anomalías en las cabeceras SIP pueden dar lugar a fallas en el analizador SIP y paquetes con formato incorrecto dando lugar a SIP vulnerable a ataques. Los parámetros por defecto serán utilizados por DPI SIP para la identificación de anomalías de protocolo y tomar acciones configuradas por el administrador. Se recomienda utilizar la configuración por defecto para estos parámetros.	**-**
SIP Dos Attacks	Intentos de inundación utilizando diversos mensajes SIP.	**No of SIP Request Messages/Duration**

SIP DDos Attacks	Intentos distribuidos de inundación utilizando diversos mensajes SIP.	**No of SIP Response Messages/Duration**
SIP Cross site scripting Attacks	Cross Site Scripting (también conocido como XSS o CSS) es una de las técnicas de hacking a nivel de aplicación. En general, cross-site scripting se refiere a la técnica que aprovecha las vulnerabilidades en el código de una aplicación web permitiendo a un atacante enviar el contenido malicioso de un usuario final y recoger algún tipo de datos de la víctima. El uso de XSS podría poner en peligro la información privada, manipular o robar las cookies, crear solicitudes que pueden confundirse con los de un usuario válido, o ejecutar código malicioso en los sistemas de usuario final. Puede ser utilizado para robar datos sobre "FROM Header", "To Header" y "Call -ID", "CONTACTO", "contraseña de extensión" y otros datos confidenciales.	**-**
	Esto se refiere a tratar de manera ilegal para acceder a los recursos del dispositivo SIP	

Buffer overflow Attacks	al igual que su dirección de memoria para el que no tiene los permisos autenticados que conducen a daños en los datos.	**-**
3rd Party Vendor Vulnerabilities	Este ataque se refiere a cualquier actividad maliciosa de **3rd** como controlador DIGIUM de Asterisk, intento de DoS.	**-**
TCP Syn Flood	Es una especie de ataque DoS en la que se envió un gran número de paquetes TCP SYN a cada dispositivo de la víctima de estos paquetes tratará de establecer una nueva sesión, consumiendo así los recursos del dispositivo de la víctima. Tal ataque también se denomina conexión de medio abierto, como estas nuevas sesiones no se terminan y, finalmente, los usuarios legítimos están impedidos de hacer uso de los recursos del dispositivo	**No of TCP Syn Packet within specified duration**
TCP Flood	Esto se refiere a inundar el dispositivo con el paquete TCP sobre cualquier puerto donde los usuarios legítimos están impedidos de hacer uso de los recursos del dispositivo después de un cierto intervalo de tiempo.	**No of TCP Packet within specified duration**
	En un ataque DDoS TCP, el tráfico TCP entrante tiene	

TCP Distributed Flood	como tarea inundar a la víctima el cual se origina a partir de muchas fuentes diferentes potencialmente cientos de miles o más. Esto hace que sea imposible y efectiva para detener el ataque, simplemente mediante el bloqueo de una sola dirección IP; además, es muy difícil distinguir el tráfico de usuarios legítimos de los ataques al tráfico cuando se distribuyen a través de tantos puntos de origen.	**No of TCP Packet within specified duration**
UDP FLOOD	Esto se refiere a inundar el dispositivo con paquetes UDP en cualquier puerto donde los usuarios legítimos están impedidos de hacer uso de los recursos del dispositivo después de un cierto intervalo de tiempo.	**No of UDP Packet within specified duration**
UDP Distributied Flood	En un ataque DDoS UDP, el tráfico UDP entrante pretende inundar a la víctima y se origina a partir de muchas fuentes diferentes potencialmente cientos de miles o más. Esto hace que sea imposible efectiva para detener el ataque, simplemente mediante el bloqueo de una sola dirección IP; además, es muy difícil distinguir el tráfico de usuarios	**No of UDP Packet within specified duration**

	legítimos de los ataques al tráfico cuando se distribuyen a través de tantos puntos de origen.	
Generic Attacks	Algunos de los ataques comunes en esta categoría son: **Bye Teardown, Registration Hijack, Registration Adder, and Registration Eraser.** ● **Bye Teardown:** Este ataque interrumpe una llamada que está en sesión entre dos usuarios. ● **Registration Hijack:** Éste ataque es una suplantación de identidad, es decir de extensiones ya registradas con el fin de registrarse. ● **Registration Adder:** Este ataque permite enlazar otra dirección SIP y permitir llamadas tanto desde el atacante y desde el usuario autorizado. ● **Registration Eraser:** Esta herramienta pretende hacer efectivo un ataque de denegación de servicio, mediante el envío de un mensaje SIP REGISTER	

	falso para que el usuario no se encuentra disponible,	

Tabla 5. 14 Reglas o firmas de seguridad del DPI.

- Protocolo Compliance

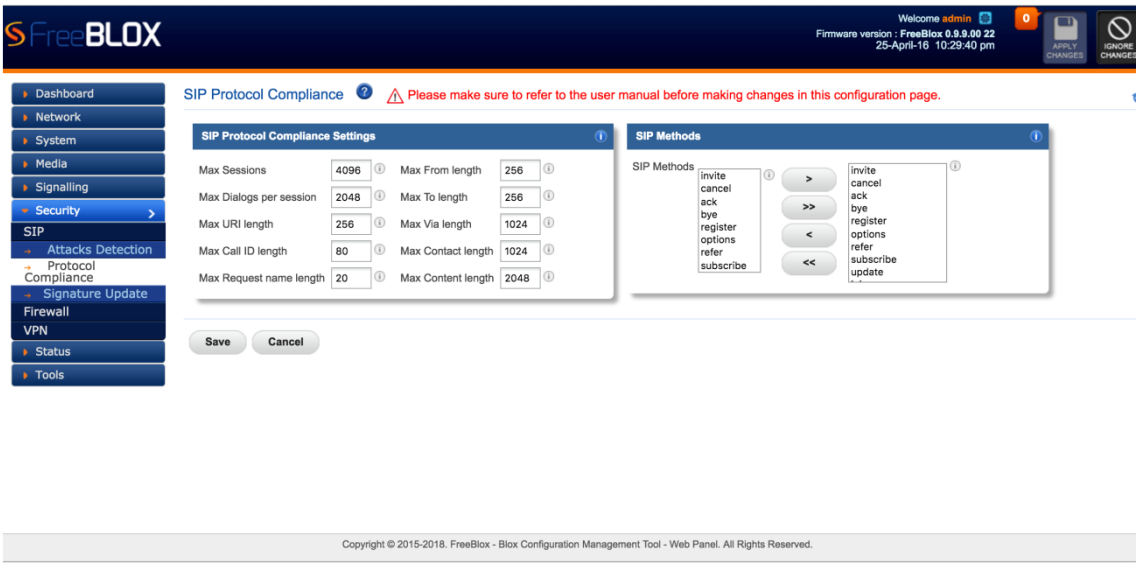


Figura 5. 66 SIP Compliance.

El DPI usado en Blox inspecciona el tráfico SIP con normas de seguridad. Las anomalías en las cabeceras SIP de mensajes pueden dar lugar a diversas condiciones erróneas, fallas en el análisis SIP y paquetes con formato incorrecto que dará lugar a aplicaciones SIP vulnerables a los ataques.

Los siguientes parámetros serán utilizados por el motor de inspección profunda de paquetes SIP DPI, para la identificación de las diferentes condiciones de anomalías de protocolo y para tomar acciones que son previamente configuradas por el administrador. Se recomienda utilizar la configuración por defecto para estos parámetros. Como se observa en la Figura 5.66.

- **Max Sessions**

Una sesión de SIP es la configuración de la conexión de nivel de aplicación que se crea entre el servidor SIP y cliente SIP, para el intercambio de los mensajes de audio / vídeo entre sí.

El parámetro **MAX SESSIONS** define el número máximo de sesiones que el SIP DPI puede analizar. El valor por defecto se ha establecido como 4096.

- **Max Dialogs per session**

Especifica el número máximo de mensajes SIP de transacciones que puede pasar entre el servidor y el cliente SIP.

- **Methods**

Este parámetro especifica qué métodos comprobar en cuanto a mensajes SIP se refiere. A continuación se presentan los mensajes SIP que el DPI puede identificar: (1) INVITE, (2) CANCEL, (3) ACK, (4) BYE, (5) REGISTER, (6) OPTIONS, (7) REFER, (8) SUBSCRIBE, (9) UPDATE (10) JOIN (11) INFO (12) MESSAGE (13) NOTIFY (14) PRACK.

- **Max URI length**

El campo **MAX URI LENGTH** identifica al usuario o servicio que se está abordando las peticiones SIP. **MAX URI LENGTH** especifica el tamaño máximo del campo URI de la solicitud. El valor predeterminado se establece en 256. El intervalo permitido para esta opción es de 1 - 65535.

- **Max call ID length**

El campo **MAX CALL ID LENGHT** en el mensaje SIP actúa como un identificador único y se refiere a la secuencia de mensajes intercambiados entre el cliente SIP y el servidor. **MAX CALL ID LENGHT** especifica el tamaño máximo del campo Call-ID. El valor predeterminado se establece en 256. El intervalo permitido para esta opción es de 1 - 65535.

- **Max Request name length**

Este campo especifica el tamaño máximo de nombre de la petición, que es parte de la ID CSeq. El valor predeterminado se establece en 20. El intervalo permitido para esta opción es de 1 – 65535

- **Max From length**

El campo de la **FROM HEADER** indica la identidad del iniciador de la solicitud SIP. **Max From length** especifica el tamaño máximo del campo. El intervalo permitido para esta opción es de 1 – 65535

- **Max To length**

El campo **TO HEADER** especifica el destinatario deseado de la petición SIP. **MAX TO LENGTH** especifica el máximo para el tamaño del campo. Por defecto se establece en 256. El intervalo permitido para esta opción es de 1 – 65535

- **Max Via length**

El campo de **VIA HEADER** indica el transporte utilizado para la transacción SIP e identifica la ubicación en la que la respuesta de SIP se va a enviar.

MAX VIA LENGTH especifica el máximo tamaño de campo. El valor predeterminado se establece en 1024. El intervalo permitido para esta opción es de 1 - 65535.

- **Max Contact length**

Identificador utilizado para ponerse en contacto con esa instancia específica del cliente / servidor SIP para posteriores requests. **MAX CONTACT LENGTH** especifica el tamaño máximo de contacto del campo. El valor predeterminado se establece en 256. El intervalo permitido para esta opción es de 1 - 65535.

- **Max Content length**

MAX CONTENT LENGTH especifica la longitud máxima de contenido del cuerpo del mensaje. El valor predeterminado se establece en 1024. El intervalo permitido para esta opción es de 1 - 65535.

5.5.4.9 Resultados

Esta sección indica los resultados en base a las configuraciones realizadas previamente.

- Profile Status

Como se puede observar en la Figura 5.67, se muestra la troncal configurada en base a los perfiles SIP, tanto LAN como WAN.

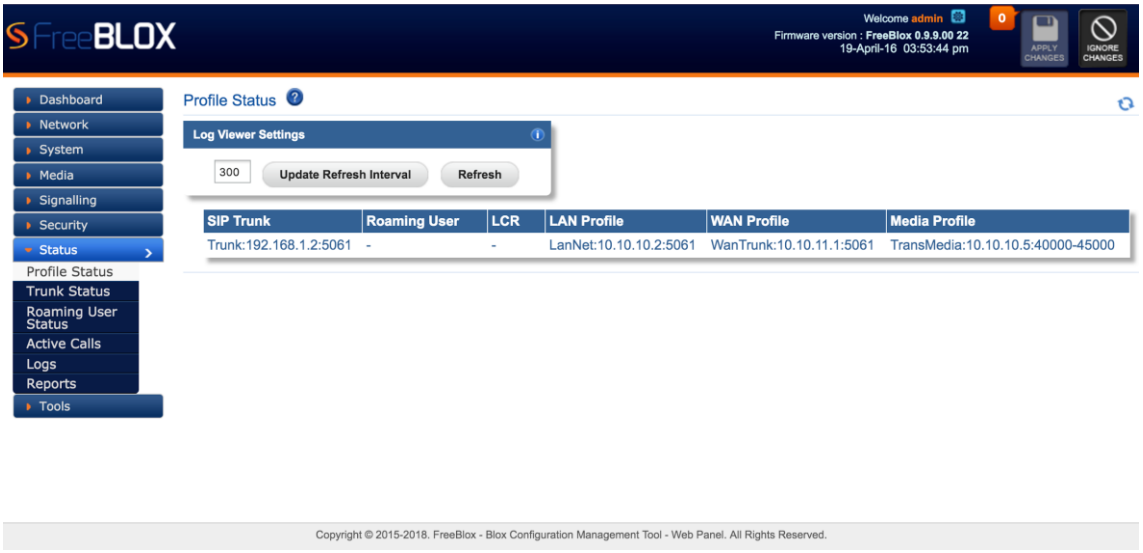


Figura 5. 67 Estado de Perfiles

- Trunk Status

A continuación en la Figura 5.68, se puede observar el estado del SIP TRUNK configurado el cual es registrado de manera exitosa.

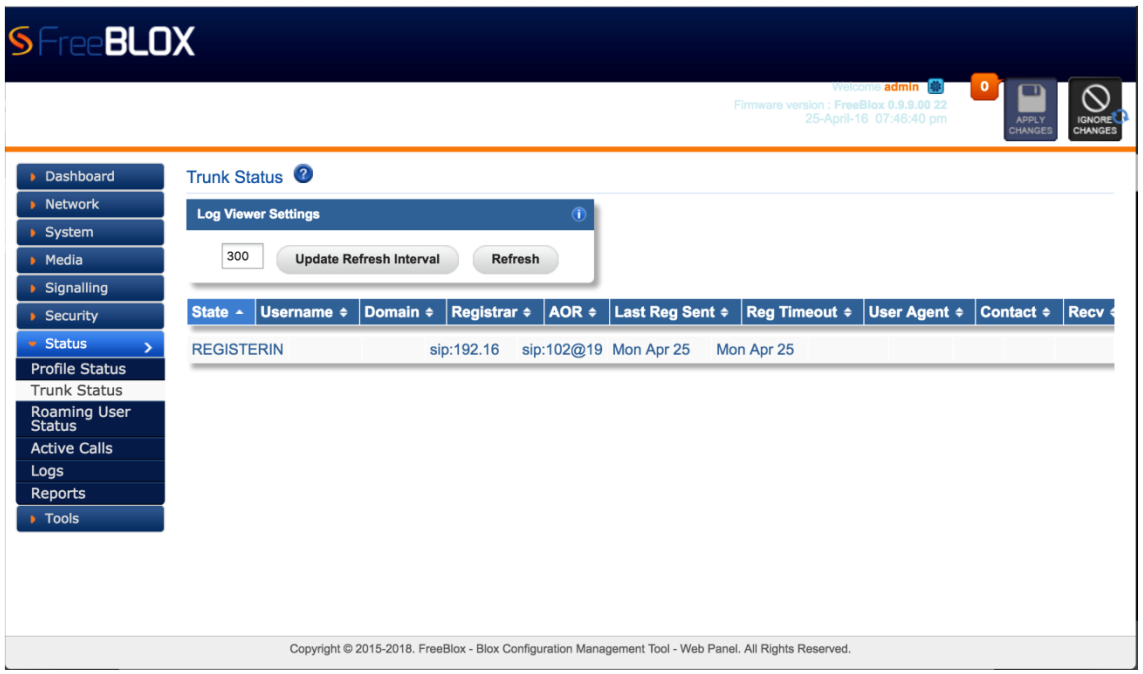


Figura 5. 68 Perfil de Trunk

- Llamadas Activas

Como se puede observar en la Figura 5.69, se despliegan todas las llamadas activas, desde la extensión que se realiza en este caso 102 y la segunda llamada en la extensión 101.

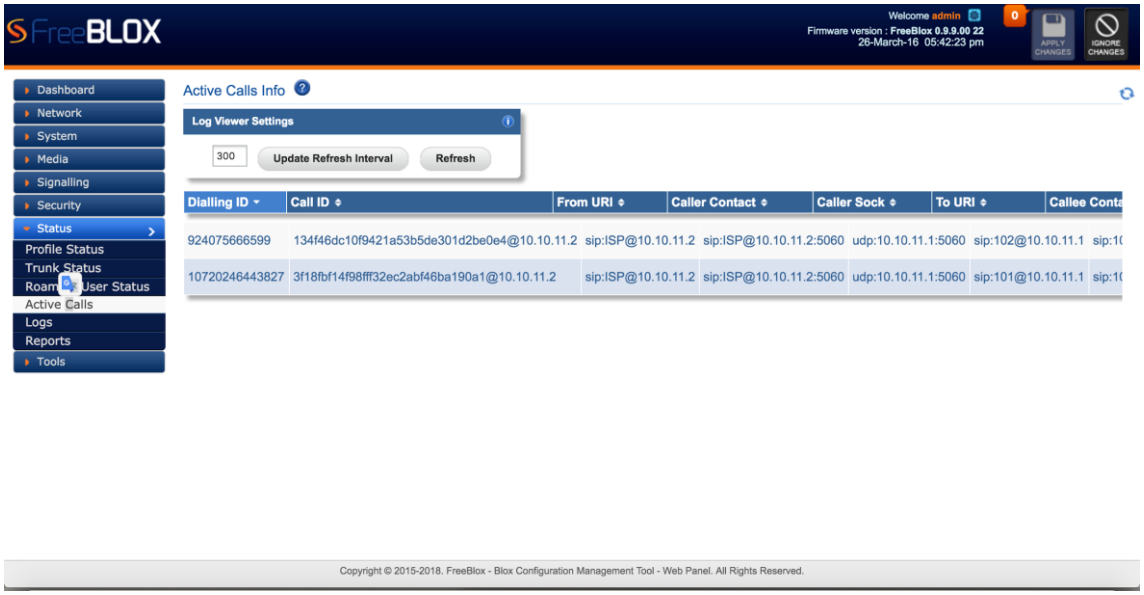


Figura 5. 69 Llamadas Activas.

- Logs
  - ✓ Logs de Señalización.

Mediante la Figura 5.70 se observa los logs de señalización, en este tipo de logs se puede obtener información acerca del intercambio de mensajes SIP.

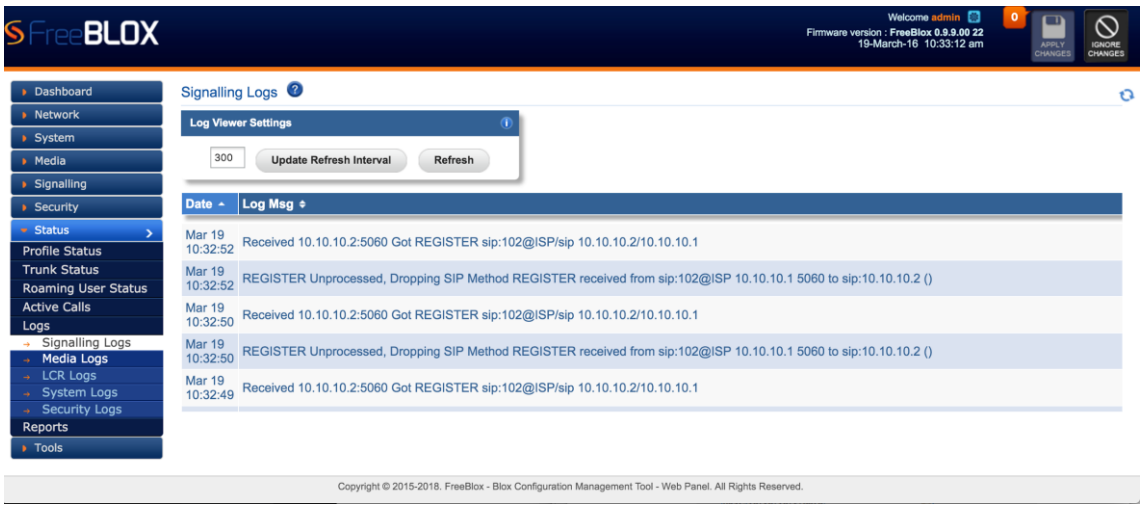


Figura 5. 70 Logs de Señalización

- ✓ Logs del sistema

Los Logs del sistema muestra reportes de todo lo concerniente a BLOX como sistema, así lo indica la Figura 5.71, por otro lado se observa en la Figura 5.72 la configuración del cliente NTP.

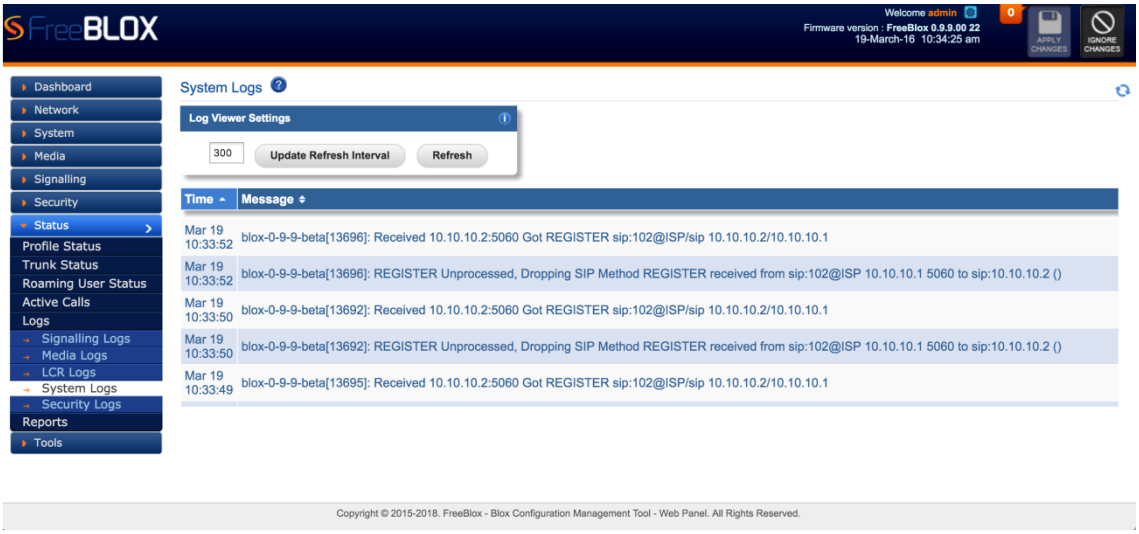


Figura 5. 71 Logs del Sistema

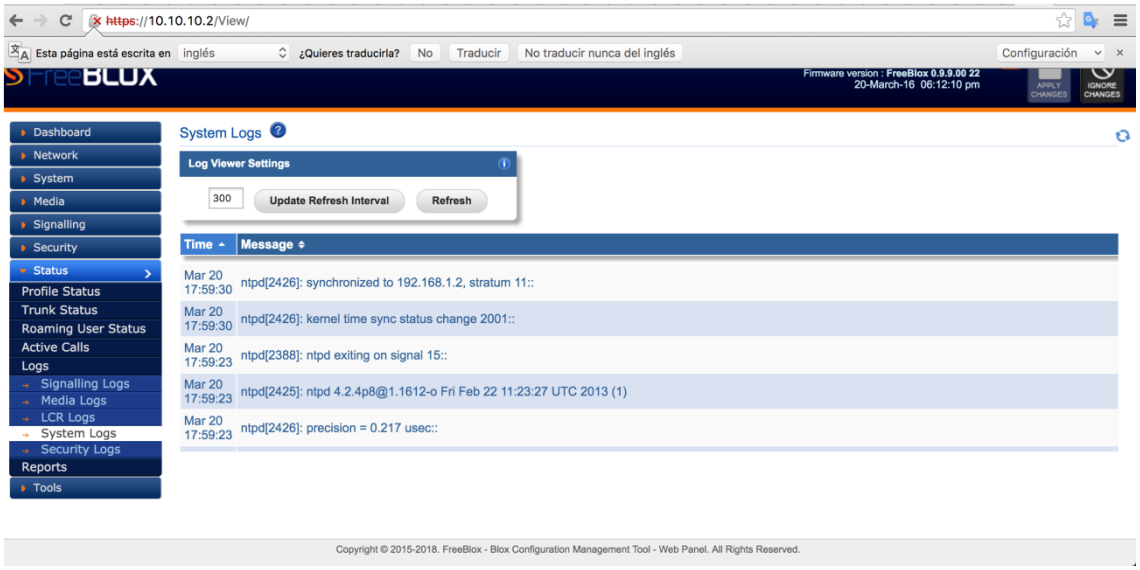


Figura 5. 72 Logs del Sistema respecto al servidor NTP

- ✓ Security Logs

En los logs de seguridad se puede observar mediante la Figura 5.73, que a través de Kali Linux se realiza el ataque de escaneo de dispositivos SIP mediante la herramienta **svmap,** la acción que toma Blox es el bloqueo al ataque.

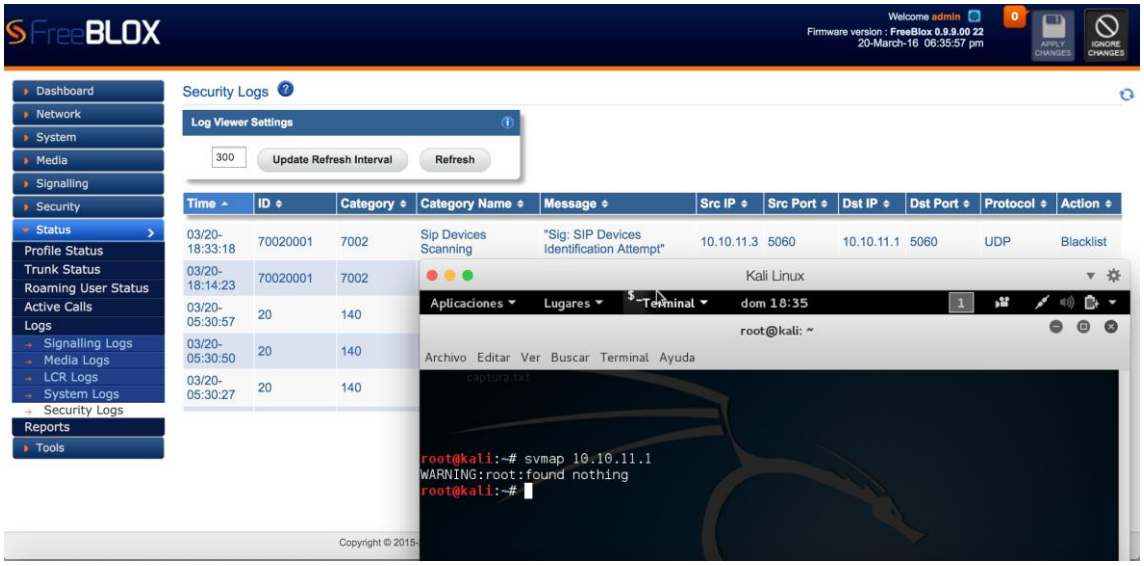


Figura 5. 73 Logs de seguridad escaneo de dispositivos SIP

Se realiza el ataque para obtención de extensiones mediante Kali y se observa en la Figura 5.74 que la acción de BLOX es el bloqueo, por lo que el atacante no tendría acceso a la información solicitada.

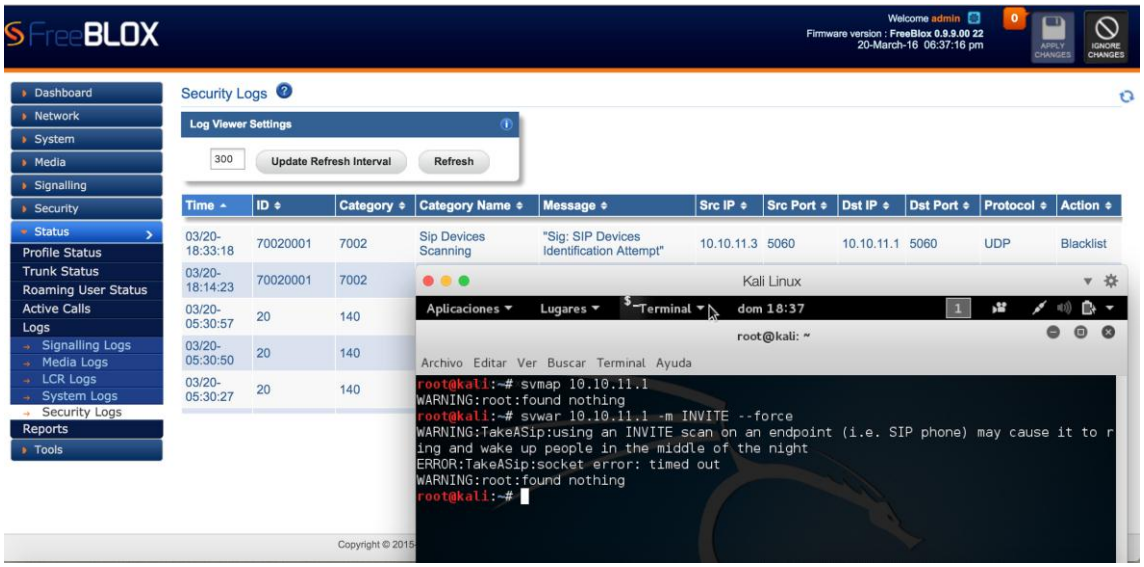


Figura 5. 74 Ataque para obtención de extensiones SIP

Por otro lado se muestra en la Figura 5.75 mediante el aplicativo svcrak, se realiza un ataque por fuerza bruta sin embargo BLOX bloquea el acceso a la información.

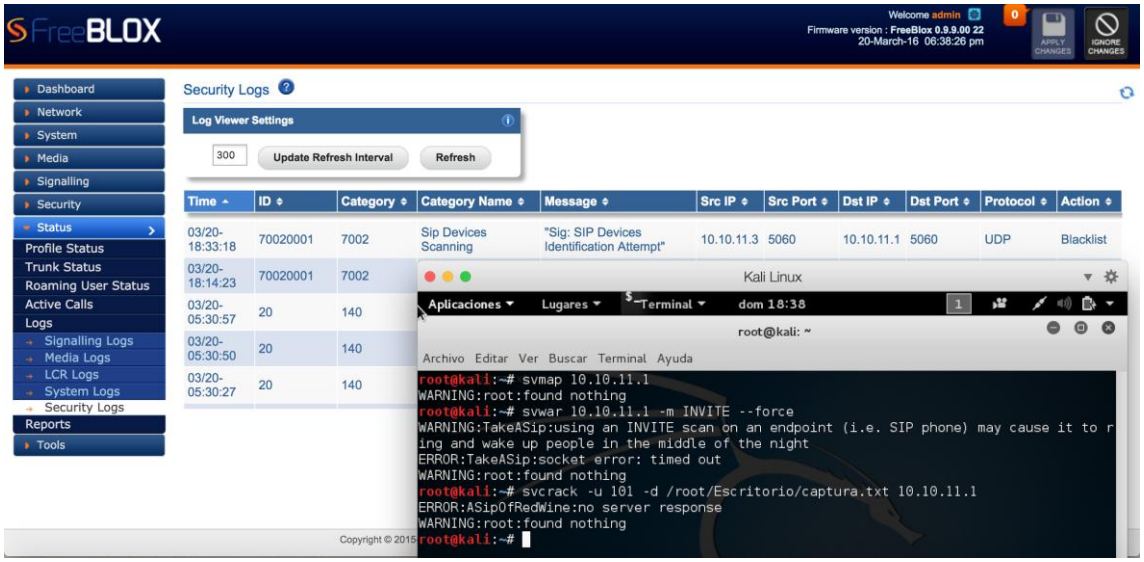


Figura 5. 75 Bloqueo de Ataque por fuerza Bruta por Blox

- Reportes

Blox también cuenta con un CDR (Call Detail Record) el cual sirve para dar detalles de las llamadas que se generan a través de BLOX. Como se observa en la Figura 5.76

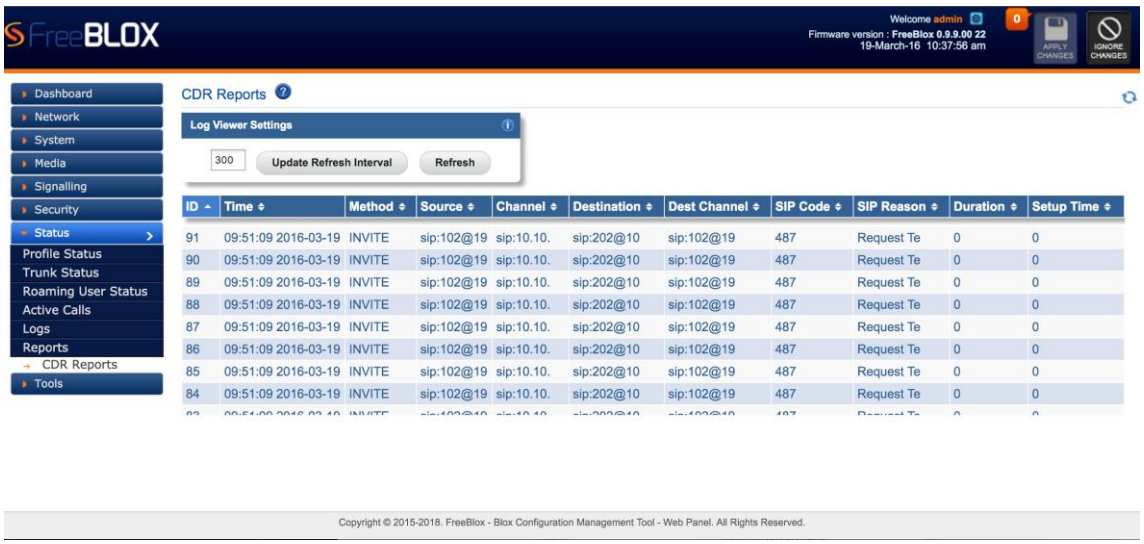


Figura 5. 76 Reportes de llamadas

6. Conclusiones

- Blox al ser un SBC de código abierto brinda muchas ventajas sobre SBC pagos o de marca, ya que no necesita licencias para el DPI, y no necesita licencias adicionales para el número de troncales por citar algunos ejemplos.
- Si bien es cierto la comunidad actual de SBC Blox no se encuentra muy difundida pero al ir investigando, y conjuntamente con el soporte de BLOX, se realizaron pruebas sobre el mismo, logrando obtener las principales funciones de un SBC pago de manera gratuita, tales como: configuración de troncales mediante perfiles SIP, a nivel de seguridad mediante el DPI se realizó bloqueos a ataques a dispositivos SIP, ataques de denegación de servicio, fuerza bruta para la obtención de contraseñas de extensiones SIP y la activación de TLS para evitar el EASVESDROPPING.
- En este trabajo, se ha realizado la funcionalidad de SBC instalándolo en una máquina virtual, sin embargo existe un limitante al realizar este ejercicio el cual es la tarjeta de transcodificación que permite un mejor desempeño en la comunicación al momento de tener diferentes códecs entre la PBX y el ISP, es ahí donde es necesario esta funcionalidad.
- SBC BLOX soporta una gran cantidad de códecs los mismos que dependerán del usuario al momento de elegir qué tipo de códec disponer de acuerdo a su necesidad basándose en su hardware, ancho de banda o calidad de voz, y obviamente que su PBX soporte.
- Es muy recomendable el uso de un SBC en la red VoIP ya que un firewall no es suficiente, SBC distingue el tráfico de voz al entrar en la red a diferencia del firewall que no reconoce este tipo de tráfico, los dos son muy necesarios en una red de datos sin embargo el SBC es propio para redes VoIP.

7. Recomendaciones.

- Para obtener mayor funcionalidad de Blox y que para este caso de estudio ha sido un limitante en sí, es el requerimiento de hardware, ya que se necesita interfaces tanto para la configuración de troncal, para realizar el roaming de usuario y LCR, sería necesario un hardware dedicado para estas funcionalidades y adicional que se cuente con una tarjeta de transcodificación en el caso de tener diferentes códecs de manera que este dispositivo garantice la comunicación.

- Se debe tener en cuenta la configuración previa de los hostname del servidor y cliente al momento de realizar cifrado mediante TLS para no tener errores al momento de emitir los certificados correspondientes.
- En el SIP TRUNK al momento de configurar TLS adicional a lo antes mencionado se debe setear el certificado para cada perfil SIP y comprobar que las centrales y terminales estén configuradas con el protocolo TLS.
- Es necesario que se configure un servidor NTP para la sincronización de los servidores y clientes, ya que permitirá el correcto funcionamiento en la comunicación.

Bibliografía

[1] Mark D. Collier, & Mark O'Brien. (18 de Febrero de 2014). *KALI TOOLS*. Obtenido de http://tools.kali.org/stress-testing/rtpflood

[2] A, Roach B. (June de 2002). *Session Initiation Protocol (SIP)-Specific Event Notification.* Obtenido de https://www.ietf.org/rfc/rfc3265.txt

[3] Angelos D. Keromytis, Symantec Research Labs Europe, & Sophia-Antipolis. (2009). *Voice over IP: Risks, Threats and Vulnerabilities.* Obtenido de http://www.cs.columbia.edu/~angelos/Papers/2009/cip.pdf

[4] Augustin López. (s.f.). *El portal de ISO 27000 en Español*. Obtenido de http://www.iso27000.es/glosario.html

[5] Blox.org. (2015). *Blox*. Obtenido de http://www.blox.org/downloads

[6] Campos, M. J. (2010). *Seguridad en voz sobre IP.* Valparaíso.

[7] Cisco. (19 de Mayo de 2008). *Cisco.com*. Obtenido de http://www.cisco.com/cisco/web/support/LA/7/73/73295_bwidth_consume.pdf

[8] Dierks, T. (Agosto de 2008). *The Transport Layer Security (TLS) Protocol Version 1.2*. Obtenido de https://tools.ietf.org/html/rfc5246

[9] Dierks, T.; E. Rescorla; RTFM, Inc.;. (Abril de 2006). *The Transport Layer Security (TLS) Protocol.* Obtenido de https://www.ietf.org/rfc/rfc4346.txt

[10] Duran, M., Martín, M., & Justel, V. (23 de Julio de 2013). *La seguridad de la información: perspectiva jurídica e informática*. Obtenido de http://www.movalen.com/la-seguridad-de-la-informacion-perspectiva-juridica-e-informatica/

[11] Elastixtech. (s.f.). *Codecs y Formatos en Telefonia IP*. Obtenido de http://elastixtech.com/codecs-y-formatos-en-telefonia-ip/

[12] Fernandez, Jose Pablo;. (Diciembre de 2013). Obtenido de http://eie.ucr.ac.cr/uploads/file/proybach/pb2013/pb2013_066.pdf

[13] García, Francisco Javier. (Diciembre de 2011). Obtenido de http://e-spacio.uned.es/fez/eserv.php?pid=tesisuned:GeoHis-Fjgarcia&dsID=Documento.pdf

[14] Gauci, Sandro. (18 de Febrero de 2014). *KALI TOOLS*. Obtenido de http://tools.kali.org/sniffingspoofing/sipvicious

[15] H, Schulzrinne; S, Casner; R, Frederick; V, Jacobson;. (Julio de 2003). *RTP: A Transport Protocol for Real-Time Applications.* Obtenido de https://www.ietf.org/rfc/rfc3550.txt

[16]J, Rosenberg; H, Schulzrine; G, Camarillo; A, Jhonston; J, Peterson; R, Sparks; M, Handley; E, Schooler. (Junio de 2002). *SIP: Session Initiation Protocol.* Obtenido de https://www.ietf.org/rfc/rfc3261.txt

[17]J. Hautakorpi, Ed.; G. Camarillo; Ericsson; R. Penfield; Acme Packet; A. Hawrylyshen; Skype, Inc.; M. Bhatia;. (Abril de 2010). *Requirements from Session Initiation Protocol (SIP) Session Border Control (SBC) Deployments*. Obtenido de https://tools.ietf.org/html/rfc5853

[18]Kali Linux. (2016). *Our Most Advanced Penetration Testing Distribution, Ever*. Obtenido de https://www.kali.org/

[19]*Kali Linux Official Documentation*. (s.f.). Obtenido de docs.kali.org/introduction/what-is-kali-linux

[20]M, B., D, M., Cisco Systems, I., M, N., & E, C. (Marzo de 2004). *The Secure Real-time Transport Protocol (SRTP).* Obtenido de https://www.ietf.org/rfc/rfc3711.txt

[21]M., Handley; V, Jacobson. (April de 1998). *SDP: Session Desciption Protocol.* Obtenido de https://www.ietf.org/rfc/rfc2327.txt

[22]Marin, Pedro. (06 de Abril de 2010). *Wireshark.org*. Obtenido de https://wiki.wireshark.org/Ejemplo_voip_calls_spanish

[23]Mark D. Collie, & Mark O'Brien. (18 de Febrero de 2014). *KALI TOOLS*. Obtenido de http://tools.kali.org/sniffingspoofing/inviteflood

[24]*OpenSIPS*. (s.f.). Obtenido de http://www.opensips.org/

[25]PaloSanto Solutions. (2006). *CARACTERÍSTICAS Y FUNCIONALIDADES DE ELASTIX*. Obtenido de http://www.elastix.org/caracteristicas/

[26]Platzi. (19 de Agosto de 2003). *Maestros de la Web*. Obtenido de http://www.maestrosdelweb.com/snort/

[27]Roberto Guitierrez Gil. (s.f.). *Seguridad en Voip: Ataques, Amenazas y Riesgos.* Valencia, España.

[28]*Session Border Controller.* (s.f.). Obtenido de allo.com/sbc.html

[29]Wireshark. (s.f.). *wikipedia.org*. Obtenido de https://es.wikipedia.org/wiki/Wireshark