

Creación de un microlaboratorio VOIP configurando PBX con software libre

ARROYO-Jorge†, REYES-Matilde & GARCIA-Francisco

Recibido 5 de Abril, 2015; Aceptado 9 de Junio, 2015

Resumen

Como parte de los trabajos realizados por los integrantes del cuerpo académico denominado “Soluciones informáticas integrales”, se propuso la creación de un microlaboratorio de experimentación para comunicación por VOIP utilizando el software de telefonía IP Asterisk PBX. La finalidad de este proyecto es hacer pruebas de comunicación y posteriormente implementar esta tecnología en la Universidad Tecnológica de Xicotepec de Juárez, la cual cuenta en la actualidad con un servicio privado de VOIP. Actualmente, los principales fabricantes de telecomunicaciones crean sistemas caros, incompatibles, con un funcionamiento complicado, antiguos códigos de ingeniería y hardware obsoleto; Además de esto, no ofrecen la posibilidad de personalización y se tienen que pagar licencias por su uso. Asterisk PBX es un software de código abierto que permite descargarlo e instalarlo sin pedir permiso y sin cumplir requerimientos de licenciamiento establecidos por algún fabricante, por lo tanto la organización que lo implementa se ve beneficiada en la reducción de costos.

VOIP, PBX, Software

Citación: ARROYO-Jorge, REYES-Matilde & GARCIA-Francisco. Creación de un microlaboratorio VOIP configurando PBX con software libre. Revista de Tecnología e Innovación 2015, 2-3:502-509

Abstract

As part of the work done by the members of the faculty called "Integrated Solutions " the creation of a micro-laboratory for experimentation proposed VOIP communication using IP telephony software Asterisk PBX. The purpose of this project is to test communication and subsequently implement this technology at the Technical University of Xicotepec Juarez , which currently has a private VOIP service. Currently , the leading manufacturers of telecommunications create expensive , incompatible systems with a complicated operation, ancient codes and obsolete hardware engineering ; Besides this , they offer the possibility of customization and have to pay license fees for its use. Asterisk PBX is an open source software that allows to download and install without asking and without complying permisso licensing requirements set by a manufacturer , so the organization that implements it is benefiting in reducing costs.

VOIP, PBX, Software

† Investigador contribuyendo como primer autor.

Introducción

La revolución tecnológica, particularmente en Tecnologías de la Información y Comunicación, continúa creciendo a un paso acelerado. Actualmente existe una amplia variedad de sistemas y aplicaciones que se pueden implementar, lo cual puede ser abrumador para aquellos que no conocen los beneficios que cada herramienta tecnológica ofrece y no se elige la adecuada para solucionar las diferentes situaciones que pudieran presentarse en una empresa u organización.

El servicio de telefonía y comunicación en una empresa es de gran importancia para mantener la comunicación al exterior e interior de la misma, por ello se debe buscar la solución óptima para lograr buenos resultados y no generar gastos excesivos en licencias, soporte y hardware. Actualmente se escucha en la rama de las telecomunicaciones el término de PBX el cual por sus siglas significa Private Branch Exchange (Ramal Privado de Intercambio), y es un dispositivo que se encuentra conectado directamente a la red pública de telefonía por medio de líneas troncales para gestionar las llamadas internas, las entrantes y salientes con autonomía sobre cualquier otra central telefónica.

Por otra parte, VoIP es una tecnología que permite la comunicación de las personas mediante mensajes de voz a través de la red de datos y corresponde al proceso de dividir audio o video en pequeños paquetes, transmitir dichos paquetes sobre una red IP y reensamblarlos en el extremo receptor con el fin de lograr una comunicación multimodal, además de lo anterior VoIP ofrece la ventaja de la integración con la red telefónica actual.

VoIP tiene varias ventajas sobre los circuitos de conmutación. Por ejemplo, la Conmutación de Paquetes permite que varias llamadas telefónicas puedan ocupar la cantidad de espacio ocupado por una sola llamada en una Red de Circuitos Conmutados. Usando Red Telefonica Conmutada, una llamada telefónica de 10 minutos consume de lleno tiempo de transmisión a un costo de 128 Kbps. Con VoIP, la misma llamada ocuparía sólo tres minutos y medio en tiempo de transmisión a un costo de 64 Kbps, dejando otros 64 Kbps libres para esos mismos tres minutos y medio, más un adicional de 128 Kbps para el resto de seis minutos y medio sobrantes. Bajo esta simple estimación, otras tres o cuatro llamadas podían encajar fácilmente en el espacio utilizado por una sola llamada en el marco del sistema convencional [2].

La filosofía de Open Source es algo que ha impactado y dejado huella en la mayoría de los ámbitos tecnológicos, por ello el papel del software libre respecto a la telefonía IP ha sido y es importante en cuanto a las propuestas que la comunidad Open Source ha expuesto y continua desarrollando. Algunas de las más populares son:

- Asterisk©.- es la implementación de un IP-PBX4 híbrido basado totalmente en *software*, el cual involucra tanto con el mundo IP (redes de datos) como con redes de telefonía analógica convencional TDM. *Asterisk* fue creado por Mark Spencer, fundador de la empresa Digium Inc. en 1999; posteriormente, el *software* fue liberado bajo licencia GNU-GPL.

- OpenSER.- sus orígenes datan del proyecto SER (SIP Express Router) del Instituto de Investigación FhG FOKUS de Berlín, Alemania, del cual surgió la versión de código abierto OpenSER, en el año 2005.
- OpenH323.- es un proyecto cuyo desarrollo se basa en un solo protocolo; y aunque el protocolo H323 (ITU-T) no fue desarrollado exclusivamente para aplicaciones de telefonía IP, sino para videoconferencia, incluye los elementos necesarios para establecer sesiones multimedia (audio, video y datos).

Durante este artículo se centrará la atención al software de telefonía IP Asterisk PBX, el cual es una IP-PBX completa diseñada en software libre (*Open Source*), que funciona sobre el sistema operativo en *Linux* y proporciona todas las características que se esperan de una PBX al trabajar con voz sobre IP (VoIP- grupo de recursos que van a ser posible que la señal de voz viaje a través de Internet empleando un protocolo IP).

Conjuntamente con diversos protocolos (grupo de reglas que permiten el intercambio de información entre computadoras) tales como *SIP* e *IAX2*, además de que provee los siguientes servicios y facilidades: correo de voz, transferencia de llamada, respuesta interactiva de voz (IVR), música en espera, cuartos de conferencia, etcétera. La voz sobre IP no necesita de ningún *hardware* adicional, y para comunicarse con la red de telefonía pública se emplean dispositivos de costos bastante accesibles en comparación contra las PBXs propietarias.

Desarrollo

Es importante hacer un poco de historia antes de arrancar con el proceso de desarrollo comentando que la plataforma en la cual se construyó el microlaboratorio de Voz sobre IP es Asterisk, este tiene alrededor del mundo más de 1 millón de Servidores en producción, así como también se registran más de 1000 descargas diarias de código fuente dentro de las múltiples comunidades activas de usuarios como lo son <http://voip-info.org> y <http://asteriskmx.com>, también existen varias distribuciones de asterisk listas para ser usadas y demás cuentan con una interfaz más amigable pero con algunas limitantes, *elastix*, *Trixbox*, *AsteriskNow*, *PBXin a Flash*.

Asterisk es un software open source creado por Mark Spenser, como una alternativa a PBX propietarios y que puede convertir una PC ordinaria en un servidor de comunicaciones.

Algunas diferencias que podemos encontrar con un PBX convencional se mencionan a continuación:

- No se requiere Hardware Propietario.
- Sigue una arquitectura Universal de PC.
- Free Software: Tanto Libre como Gratis.
- Totalmente Programable y Configurable.
- Ofrece Servicios Adicionales de Datos, puede integrarse con cualquier sistema que pueda correr en una pc con Linux como Base de Datos, Consultas Web, etc.

- Puede interactuar con conmutadores Existentes.

Requisitos de Hardware

Lo mas importante para instalar asterisk y construir un PBX es la escalabilidad, por lo que se recomienda siempre usar Hardware de Marca, evitar equipos armados con componentes de dudosa procedencia y procurar siempre ofrecer soluciones de redundancia en los componentes que fallan más comúnmente como los discos duros y fuentes de poder.

El cpu necesario se mide según los requerimientos no existe una regla estricta en cuanto se necesita, es importante considerar la cantidad de llamadas concurrentes requeridas, asi como también cada acción que se requiera implementar al servidor de VOIP como la de Grabacion de Llamadas, Cunsultas de Base de Datos, reproducción de audio, etc.

Si el sistema es 100% IP, no necesitamos tarjetas adicionales, el hardware adicional es en el caso de que se requiera conectar directamente a la PSTN (Public Switched Telephone Network). Cualquier Pc es buena para pruebas podemos aprovechar el hardware viejo que tengamos.

Requisitos de Software

Se recomienda la utilización de sistemas operativos Linux Nativo FreeBSD, Centos o Debian, asi como también requiere paquetería adicional como son Gcc, Ncurses, Termcap, Newt, OpenSSL y LibUUID

Una vez especificado los requerimientos necesarios ahora es importante explicar la forma en cómo se implementa el microlaboratorio de VOIP utilizando Asterisk.

En el laboratorio de computo se encuentran instaladas 20 equipos de cómputo que cuentan con el sistema operativo de Linux CENTOS 6.5 o superior así como también Asterisk con todo el software necesario, ya que cada equipo actúa como una empresa individual y en cada una de ellas se estará configurando todo lo necesario para que pueda fungir como un PBX.

[Imagen de Como estará organizado el MicroLaboratorio]

Procedimiento para la Implementación de PBX con Asterisk

- Instalación del Sistema Operativo
- Instalación de SSH

El servidor seguro, es una herramienta popular y poderosa, basada en la seguridad de la red. Cada vez que los datos se envían por una computadora a través de la red, SSH encripta automáticamente. Entonces, cuando los datos llegan a su destinatario SSH automáticamente lo descifra. El resultado es la encriptación transparente: los usuarios pueden trabajar con normalidad, sabiendo que sus comunicaciones están codificadas de manera segura.

SSH es un protocolo, no un producto, este es una especificación de cómo llevar a cabo comunicación segura a través de una red. Cubre la autenticación, el cifrado y la integridad de los datos transmitidos a través de la red.

Después de esta breve descripción de SSH y posteriormente a la instalación del Sistema Operativo es importante instalar OpenSSH el cual se ocupa para poder tener acceso al servidor Asterisk de forma remota.

Para instalar el paquete openssh-server se utiliza el comando yum:

```
yum install openssh-server
```

No basta con instalar el servicio, lo importante, es configurar el servidor ssh de forma segura. Para realizar esta tarea se edita el archivo `/etc/ssh/sshd_config` y hay varias cosas a tener en cuenta.

Cambiar el puerto de SSH

El puerto estándar es el TCP 22 y por lo general cualquier persona que intente tener acceso no autorizado al servidor va a probar este puerto, por lo que es importante que se configure a que escuche por otro puerto por ejemplo por el TCP 23 o 22 o el que se prefiera.

El archivo se edita `/etc/ssh/sshd_config` y cambia lo siguiente:

```
#Port 22
Port 23033
```

Otro punto importante es asegurarte que se utilice únicamente la versión 2 del protocolo con la siguiente línea:

```
#Protocol 2,1
Protocol 2
```

a) Se debe evitar ingresar al sistema con el usuario root, esto al igual que el puerto 22 es conocido por quienes intenten acceder al sistema y el nombre de usuario (conocido) y el mas importante a probar es root, ya que con este usuario se puede tener total control sobre el servidor. Para evitar eso niega el acceso mediante ssh como el usuario root.

```
#PermitRootLogion yes
PermitRootLogion no
```

Configurar interfaz de red

Muchos servidores tienen dos o más interfaces de red y se puede limitar que ssh escuche por una interfaz de red ssh así limitar el acceso al servidor.

Para hacer esto solo se tiene que indicar la siguiente directiva:

```
ListenAddress 192.168.1.10
```

Reiniciar ssh y aplicar los cambios

Para que los cambios que se han realizado se apliquen, se debe de reiniciar el servicio sshd.

```
# service sshd restart
Parando sshd: [ OK ]
Iniciando sshd: [ OK ]
```

Estos pasos son para configurar ssh de forma básica y mejorar la seguridad del servidor pero existen más opciones que puedes utilizar para tener un servicio ssh más sofisticado como autenticar a los usuarios por llaves públicas y privadas, permitir ciertos comandos al usuario root, limitar el tiempo y el número de conexiones a un usuario o solo permitir conectarse por ssh solo a algunos usuarios del sistema, entre otros.

Configuración de las Interfaces de Red

Para configurar las interfaces de Red se accede al archivo dentro de la ruta `/etc/sysconfig/network-scripts/` escribiendo la siguiente instrucción

```
nano /etc/sysconfig/network-
scripts/ifcfg-eth0
```

Se abrirá un editor de texto donde se busca la línea que dice `ONBOOT=no` y se cambia por `ONBOOT=yes`.

```
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.1.5
NETMASK=255.255.255.0
NETWORK=192.168.1.0
GATEWAY=192.168.1.254
DNS=8.8.8.8
```

Se guarda el archivo con Ctrl + O y se puede salir usando Ctrl + X. A continuación se aplican cambios:

```
/etc/init.d/network restart
```

Estos parámetros de red son de acuerdo a la configuración de la red que se este trabajando este es solo un ejemplo de cómo se debe hacer.

Una vez que la tarjeta de red está encendida en automático y corroborar que ya se tiene internet en el servidor continuaremos con la instalación de dependencias que se necesitaran más adelante.

Instalación de Dependencias

Las dependencias necesarias para la instalación de Asterisk ya se han mencionado anteriormente y la instrucción para su instalación es la siguiente:

```
yum install -y gcc ncurses-devel
libtermcap-devel libuuid-devel gcc-c++
openssl-devel newt-devel zlib-devel
unixODBC-devel libtool libtool-devel make
wget kernel kernel-devel kernel-headers sqlite-
devel libxml2-devel
```

Estos comandos comienzan a descargar las dependencias de Asterisk, pero no instalan Asterisk.

Posteriormente se reinicia el sistema para actualizar el kernel que se acaba de hacer.

```
reboot
```

Instalación de Asterisk y Dahdi

Para la instalación de asterisk se debe de descargar, desempaquetar, compilar e instalar asterisk desde código fuente los paquetes es:

```
Asterisk-11.0.tar.gz
Dahdi-linux-complete-2.7.0.tar.gz
```

Se debe tener en cuenta la ruta en la cual se descargan los paquetes mencionados en el ejemplo se descargo en la ruta de

```
cd /usr/src
```

Se desempaquetan los archivos descargados

```
tar -zxf dahdi-linux-complete-
2.7.0.tar.gz
tar -zxf asterisk-11.0.tar.gz
```

Una vez hecho esto se crean 2 directorios que contendrán el código fuente:

```
/usr/src/dahdi-linux-complete-2.7.0
/usr/src/asterisk-11.0
```

Después de descomprimir DAHDI se compila e instala

```
cd /usr/src/src/dahdi-linux-complete-
2.7.0
```

```
make
make install
make config
```

Después de descomprimir Asterisk proseguiremos a compilar e instalarlo

```
cd /usr/src/src/asterisk-11.0
```

```
./configure
make menuselect
```

Antes de pasar al siguiente comando se debe escoger que es lo que se quiere compilar y lo que no

[*] Significa activado

[] Significa Desactivado

XXX Significa desactivado sin posibilidad de activarlo

Es recomendable que se realicen los siguientes cambios:

Add-ons: Se desactiva todo lo que se encuentra debajo de deprecated

Call Detail Recording: Se desactiva todo lo que se encuentra debajo de extended

Channel Event Logging: Desactiva todo

Chanel Drivers: Se desactiva todo lo que se encuentra debajo de deprecated y extended

PBX Modules: Se desactiva todo lo que se encuentra debajo de extended, excepto pbx_ael y pbx_realtime

Resources Modules: Se desactiva todo lo que se encuentra debajo de extended, excepto re_ael_share

Compiler	Flags:	Activamos
DONT_OPTIMIZE,		desactivamos
BUILD_NATIVE		

Core Sound Packages: Cambiamos EN-WAV por EN-GSM

Music On Hold File Packages: Cambiamos lo WAV por GSM

Extras Sound Packages: Seleccionamos EN-GSM

Una vez realizado todo esto se guarda y Sale (Save & Exit) y se continua con la compilación.

```
make
make install
make config
make samples
```

Centos trae el SELinux activado(es una característica de seguridad de Linux). Para correr Asterisk sin problemas, se debe desactivar editando el archivo /etc/selinux/config

```
nano /etc/selinux/config
Dentro del archivo se busca esta línea
SELINUX=enforcing
```

y se reemplaza por esta:

```
SELINUX=disable
```

En Centos, por default el firewall viene encendido. Se debe detener para evitar problemas de conexión más adelante

```
chkconfig iptables off
```

Con el cambio hecho, se reinicia el sistema operativo con

Reboot

Cuando el sistema reinicia ya se puede entrar a la consola de asterisk y ya se puede iniciar a configurar el servidor Asterisk y con ello implementar el Micro laboratorio de Telefonía Ip utilizando Asterisk.

asterisk –r

Resultados

Los logros más importantes en esta investigación fue inicialmente la implementación de la herramienta de asterisk en un laboratorio de cómputo donde cada equipo fungía como empresas individuales las cuales entablan comunicación y fueron los alumnos de la carrera de Tecnologías de la Información quienes ayudaron en la construcción.

Por otra parte el proyecto fue presentado a las autoridades de la Universidad Tecnológica de Xicotepec de Juárez donde se argumentaron las ventajas de utilizar Asterisk como una herramienta capaz de resolver los problemas de comunicación dentro del campus sin la necesidad de realizar gastos innecesarios.

El proyecto fue aprobado con presupuesto para ser implementado en nuestra Universidad, ya se an configurado extensiones con los teléfonos ip que se adquirieron y funcionan perfectamente en nuestra red.

Referencias

Madsen L. y Brian R. (2011). Asterisk Cookbook (e.Book), Gravenstein Highway North Sebastopol, C.A.: Editorial O' Reilly Media, Inc.

Madsen L., Brian R. y Meggelen J.V. (2011). Asterisk The Definitive Guide (e.Book), Gravenstein Highway North Sebastopol, C.A.: Editorial O' Reilly Media, Inc.

Madsen L., Brian R., Meggelen J.V. y Smith Jared (2007). Asterisk The Future Of Telephony (e.Book), Gravenstein Highway North Sebastopol, C.A.: Editorial O' Reilly Media, Inc.