Asterisk - Configuracion SIP

Introducción

Asterisk es una fuente abierta PBX que se ejecuta en Linux y muchos otros sistemas operativos. Fue creada en 1999 por Mark Spencer, fundador de Digium, que es una empresa privada con sede en Huntsville, Alabama. Entre otras cosas, Digium está especializada en el desarrollo de hardware para su uso con Asterisk. Como resultado, Asterisk puede no ser independiente del proveedor, pero sigue siendo el más popular PBX de código abierto.

El desarrollo de Asterisk fue significativo, ya que fue la primera vez que las organizaciones y los individuos pudieron crear su propia PBX sin “perder un brazo y una pierna”. En lugar de ello, el coste de un Asterisk PBX sólo consiste en el hardware y los teléfonos que se conectan a él; todos los cuales son estándar, fácilmente disponibles y, por tanto asequibles.

Como cualquier PBX, Asterisk es, básicamente, un router para las llamadas telefónicas entrantes y salientes. Se puede configurar para admitir una gama de conexiones externas utilizando diversos medios de comunicación y protocolos, así como un gran número de puntos finales: por lo general teléfonos que se conectan a través de la red Asterisk (o Internet) utilizando un protocolo u otro.

Esta página describe cómo instalar un sistema Asterisk y una mínima configuración SIP en Debian 5.0 (lenny) . El sistema operativo viene con Asterisk 1.4.21 y 1.4.11 Zaptel. En realidad, Debian ofrece dos paquetes de Zaptel: zaptel y zaptel-source, con un paquete zaptel-módulos que deben ser compilados a partir de este último.

En los procedimientos de instalación y de configuración siguientes se supone que un sistema de Debian Lenny ya están en funcionamiento, que un teléfono SIP está disponible y funcionando, posiblemente a través del uso de un adaptador de SIP, y que una cuenta SIP externa está disponible a través de un proveedor comercial de VoIP.

Instalar Asterisk

Lo primero es instalar los siguientes tres paquetes:

```
~# apt-get install asterisk zaptel zaptel-source
```

Suponiendo que no existe nada más allá de un sistema básico en este punto, se instalará un total de 75 paquetes como resultado, incluyendo 72 dependencias:

```
asterisk                    1:1.4.21.2~dfsg-3+lenny1 Open Source Private
Branch Exchange (PBX)
asterisk-config             1:1.4.21.2~dfsg-3+lenny1 Configuration files
for Asterisk
asterisk-sounds-main        1:1.4.21.2~dfsg-3+lenny1 Core Sound files for
Asterisk (English)
binutils                    2.18.1~cvs20080103-7     The GNU assembler,
```

```
linker and binary utilities
build-essential               11.4                    Informational list of
build-essential packages
bzip2                         1.0.5-1                 high-quality block-
sorting file compressor - utilities
ca-certificates               20080809                Common CA
certificates
cpp                           4:4.3.2-2               The GNU C
preprocessor (cpp)
cpp-4.3                       4.3.2-1.1               The GNU C
preprocessor
debhelper                     7.0.15                  helper programs for
debian/rules
dpkg-dev                      1.14.29                 Debian package
development tools
fxload                        0.0.20020411-1.1        Firmware download to
EZ-USB devices
g++                           4:4.3.2-2               The GNU C++ compiler
g++-4.3                       4.3.2-1.1               The GNU C++ compiler
gcc                           4:4.3.2-2               The GNU C compiler
gcc-4.2-base                  4.2.4-6                 The GNU Compiler
Collection (base package)
gcc-4.3                       4.3.2-1.1               The GNU C compiler
gcc-4.3-base                  4.3.2-1.1               The GNU Compiler
Collection (base package)
gettext                       0.17-4                  GNU
Internationalization utilities
gettext-base                  0.17-4                  GNU
Internationalization utilities for the base system
html2text                     1.3.2a-5                advanced HTML to text
converter
intltool-debian               0.35.0+20060710.1       Help i18n of RFC822
compliant config files
libasound2                    1.0.16-2                ALSA library
libc-client2007b              7:2007b~dfsg-4+lenny3   c-client library for
mail protocols - library files
libc6-dev                     2.7-18lenny2            GNU C Library:
Development Libraries and Header Files
libcompress-raw-zlib-perl     2.012-1lenny1           low-level interface
to zlib compression library
libcompress-zlib-perl         2.012-1                Perl module for
creation and manipulation of gzip files
libcurl3                      7.18.2-8lenny4          Multi-protocol file
transfer library (OpenSSL)
libdigest-hmac-perl           1.01-7                  create standard
message integrity checks
libdigest-sha1-perl           2.11-2+b1               NIST SHA-1 message
digest algorithm
libfile-remove-perl           1.42-1                  remove files and
directories, accepts wildcards
```

```
libgcc1                       1:4.3.2-1.1             GCC support library
libgmp3c2                     2:4.2.2+dfsg-3          Multiprecision
arithmetic library
libgomp1                      4.3.2-1.1               GCC OpenMP (GOMP)
support library
libgsm1                       1.0.12-1                Shared libraries for
GSM speech compressor
libiksemel3                   1.2-4                   C library for the
Jabber IM platform
libio-compress-base-perl      2.012-1                 Base Class for
IO::Compress modules
libio-compress-zlib-perl      2.012-1                 Perl interface to
zlib
libio-stringy-perl            2.110-4                 Perl modules for IO
from scalars and arrays
liblocale-gettext-perl        1.05-4                  Using libc functions
for internationalization in Perl
libltdl3                      1.5.26-4+lenny1         A system independent
dlopen wrapper for GNU libtool
libmail-box-perl              2.082-2                 Manage a message-
folder
libmail-sendmail-perl         0.79-5                  Send email from a
perl script
libmailtools-perl             2.03-1                  Manipulate email in
perl programs
libmime-types-perl            1.24-1          Perl extension for
determining MIME types and Transfer Encodin
libmpfr1ldbl                  2.3.1.dfsg.1-2          multiple precision
floating-point computation
libobject-realize-later-perl  0.18-1                  Delayed creation of
objects
libogg0                       1.1.3-4                 Ogg Bitstream Library
libperl5.10                   5.10.0-19lenny2         Shared Perl library
libpq5                        8.3.9-0lenny1           PostgreSQL C client
library
libpri1.0                     1.4.3-2                 Primary Rate ISDN
specification library
libradiusclient-ng2           0.5.5-1                 Enhanced RADIUS
client library
libsensors3                   1:2.10.7-1              library to read
temperature/voltage/fan sensors
libsnmp-base                  5.4.1~dfsg-12    SNMP (Simple Network
Management Protocol) MIBs and documentati
libsnmp15                     5.4.1~dfsg-12           SNMP (Simple Network
Management Protocol) library
libspeex1                     1.2~rc1-1               The Speex codec
runtime library
libspeexdsp1                  1.2~rc1-1               The Speex extended
runtime library
libsqlite0                    2.8.17-4                SQLite shared library
libssh2-1                     0.18-1                  SSH2 client-side
```

```
library
libsys-hostname-long-perl     1.4-2                    Figure out the long
(fully-qualified) hostname
libsysfs2                     2.1.0-5                  interface library to
sysfs
libtimedate-perl              1.1600-9                 Time and date
functions for Perl
libtonezone1                  1:1.4.11~dfsg-3          tonezone library
(runtime)
liburi-perl                   1.35.dfsg.1-1            Manipulates and
accesses URI strings
libuser-identity-perl         0.92-2          manages different
identities/roles used by a physical person
libvorbis0a                   1.2.0.dfsg-3.1+lenny1    The Vorbis General
Audio Compression Codec
libvorbisenc2                 1.2.0.dfsg-3.1+lenny1    The Vorbis General
Audio Compression Codec
libvpb0                       4.2.38.1-1         Voicetronix telephony
hardware userspace interface library
linux-libc-dev                2.6.26-21lenny4          Linux support headers
for userspace development
make                          3.81-5                   The GNU version of
the "make" utility.
makedev                       2.3.1-88                 creates device files
in /dev
mlock                         7:2007b~dfsg-4+lenny3    mailbox locking
program
module-assistant              0.10.11.0                tool to make module
package creation easier
odbcinst1debian1              2.2.11-16       Support library and helper
program for accessing odbc ini file
openssl                      0.9.8g-15+lenny6  Secure Socket Layer (SSL)
binary and related cryptographic too
po-debconf                    1.0.15                   manage translated
Debconf templates files with gettext
unixodbc                      2.2.11-16                ODBC tools libraries
vpb-driver-source             4.2.38.1-1               Source for the
Voicetronix telephony hardware drivers
zaptel                        1:1.4.11~dfsg-3          zapata telephony
utilities
zaptel-source                 1:1.4.11~dfsg-3      Zapata telephony
interface (source code for kernel driver)
```

Esto produce una instalación básica de Asterisk. Sin embargo, hay un mensaje de error que aparece casi al final del proceso de instalación:

```
Zaptel telephony kernel driver: FATAL: Module ztdummy not found.
```

El tema de esta falta del módulo Zaptel se aborda en el siguiente paso.

Módulos Zaptel

No hay una causa real de preocupación en relación con el mensaje de error anterior. Más bien, debe ser vista como un recordatorio de lo que se debe hacer a continuación, que es compilar e instalar los módulos Zaptel.

```
~# m-a a-i zaptel
```

El comando **m-a** es un enlace simbólico para el module-assistant, mientras que la opción **a-i** es la abreviatura de auto-instalación.

Antes de que comience el proceso de construcción actual, el comando anterior conllevará la instalación de seis nuevos paquetes, incluyendo tres que son kernel-specific:

```
cpp-4.1                         4.1.2-25                The GNU C
preprocessor
gcc-4.1                         4.1.2-25                The GNU C compiler
gcc-4.1-base                    4.1.2-25                The GNU Compiler
Collection (base package)
linux-headers-2.6.26-2-686    2.6.26-21lenny4         Header files for
Linux 2.6.26-2-686
linux-headers-2.6.26-2-common 2.6.26-21lenny4         Common header files
for Linux 2.6.26-2
linux-kbuild-2.6.26           2.6.26-3                Kbuild infrastructure
for Linux 2.6.26
```

El resultado final es que el paquete **zaptel-modules** es producido e instalado, incluyendo una serie de módulos para el núcleo en ejecución. Entre ellos se encuentra **ztdummy.ko**, que se hará cargo de el error mencionado. Este módulo proporciona la fuente de reloj que Asterisk utiliza como un mecanismo de tiempo, por ejemplo, para ayudar a mantener múltiples flujos de audio sincronizado mientras se mezclan entre sí.

Cargar el módulo ztdummy y reiniciar Asterisk

Se hace con los siguientes comandos:

```
~# modprobe ztdummy
~# /etc/init.d/asterisk restart
Stopping Asterisk PBX: asterisk.
Starting Asterisk PBX: asterisk.
~# _
```

Una revisión rápida con **lsmod** mostrará que, en realidad, se cargan como resultado un total de tres nuevos módulos:

```
~# lsmod
ztdummy                  3056  0
zaptel                 185060  1 ztdummy
```

```
crc_ccitt               2080  1 zaptel
```

De acuerdo con su producción modinfo, los otros dos proporcionan el Zapata Telephony Interface“ y “CRC-CCITT calculations.” Esta última, por cierto, no es parte del paquete de Zaptel.

Canal de configuración SIP

En este ejemplo, el protocolo SIP se utiliza tanto para la creación de un canal a la PSTN , utilizando una cuenta de un proveedor comercial de VoIP, y configurando un teléfono local para **testing puposes**.

Para ambos, los cambios primero se deben hacer en **/etc/asterisk/sip.conf**, que es el archivo de configuración de SIP. Inicialmente, este archivo contiene en su mayoría los comentarios, por lo que hay que renombrarlo por el momento:

```
~# mv /etc/asterisk/sip.conf /etc/asterisk/sip.conf-org
~# _
```

Después de eso, hay que crear una nueva versión vacía de este archivo y modifiquar su propiedad y los permisos:

```
~# touch /etc/asterisk/sip.conf
~# chown asterisk.asterisk /etc/asterisk/sip.conf
~# chmod 640 /etc/asterisk/sip.conf
~# _
```

A continuación, hay que editar el fichero nuevo y vacío en **/etc/asterisk/sip.conf** y añadir una serie de cosas. El fichero comienza con una sección **[general]**, las opciones en las que se aplicarán a todas las demás secciones de este archivo a menos que sean anulados en concreto:

```
 [general]
disallow=all
allow=ulaw
allow=alaw
allow=gsm
qualify=yes
canreinvite=no
```

Las explicaciones para estos ajustes son los siguientes:

disallow=all	Deshabilita el uso de todos los codecs . Se utiliza antes de codecs específicos están habilitadas en orden de preferencia.
allow=ulaw	Permite un codec basado en la μ-law, algoritmo que se utiliza principalmente en América del Norte y Japón. Proporciona rango ligeramente más dinámico que A-law . Se trata de un codec PCM (Pulse Code Modulation) de 64 kbps y una Compansión variante de la UIT-T G.711 estándar. Todos estos codecs, imponen una carga mínima en la CPU.

allow=alaw	Permite un códec basado en el algoritmo de A-law que se utiliza en Europa y en el resto del mundo. Requiere menos potencia de procesamiento de CPU que µ-law. En los EE.UU., se utiliza por convención para las conexiones internacionales, si al menos una de las partes utiliza. Otra variante G.711 de la UIT-T.
allow = gsm	Activa la GSM 06.10 codec, que es el códec preferido para Asterisk. Se opera a 13 kbps, emplea la compresión con pérdida del habla, no tiene requisitos de licencia y ofrece un excelente rendimiento relacionado con el CPU.
qualify=yes	Cuando está activado SIP NOTIFY, periódicamente se enviarán mensajes en la distancia entre pares para determinar tanto su disponibilidad y la latencia de las respuestas.
canreinvite=no	Evita que los dos puntos finales se conecten directamente entre sí, lo que es un comportamiento normal cuando se usa SIP. Esto obliga a Asterisk a permanecer en la ruta de transmisión, que es necesaria para detectar señales DTMF. Algunos proveedores SIP comerciales también hacen esto.

Además de lo anterior, tres más adiciones son necesarias antes de que sea posible realizar y recibir llamadas.

El primero es un registro SIP saliente que autentique este sistema para el proveedor de VoIP, hazle saber cuál es la dirección IP de este sistema y que está disponible. Tales declaraciones de registro tienen el siguiente formato:

```
register => user[:secret[:authuser]]@host[:port][/extension]
```

En el ejemplo de la declaración de registro a continuación, **jsmith** será el nombre de la cuenta remota, 1234 el **secret** (contraseña), **provider.example.com** el nombre del servidor del proveedor de VoIP y 0715551234 el destino de la llamada. La extensión , 0715551234, es descriptiva, que puede incluso ser alfanumérica. Basándose en esta información, la declaración de registro debe ser:

```
register => jsmith:1234@provider.example.com/0715551234
```

Hay que añadir esta declaración de registro al final del archivo bajo el contexto **[general]** mencionado anteriormente.

Manejo local de las llamadas entrantes y salientes

En la siguiente sección añadimos a **sip.conf** el manejo local de las llamadas entrantes y salientes entre el host y la mantenida por el proveedor de VoIP comercial.

Añadimos al final del archivo, por debajo de la declaración de registro:

```
 [provider]
type=peer
context=incoming
host=provider.example.com
username=jsmith
fromuser=jsmith
secret=1234
```

Explicación:

[provider]	Título de la sección. Todas las líneas entre el título de la sección y la siguiente se aplican sólo a esta sección. Este nombre se refiere el plan de marcación para establecer las llamadas salientes.
type=peer	
context=incoming	El nombre especificado aquí, que es arbitraria, determinará donde las llamadas entrantes entrarán en el plan de marcación cuando llegan en el canal asociado con esta sección.
host=provider.example.com	Establece el nombre de la máquina remota a la que este host debe conectarse. En este caso, el valor debe ser un nombre de dominio completo.
username=jsmith	Establece el nombre de usuario con el que Asterisk se autentica en un par, así como el nombre de usuario para el interlocutor para la autenticación de Asterisk. Esto anula el nombre en el título de la sección (entre corchetes) que normalmente se utiliza para este propósito. También permite el registro con un compañero antes de que los compañeros se ha registrado con Asterisk. Esta opción también puede ser requerido por otras características, como la marcación de salida de correo de voz.
fromuser=jsmith	Otro método para especificar el nombre de usuario para la autenticación con un compañero para anular el nombre en el título de la sección. Con algunos proveedores SIP, esta opción puede ser necesaria para la definición de canal para trabajar.
secret=1234	Establece la contraseña. Se utiliza para la autenticación junto con el nombre (título) de esta sección, que en este caso es anulado por fromuser = jsmith.

Teléfono que se va a unir al nuevo servidor

La tercera nueva sección es para el teléfono que se va a unir al nuevo servidor. Su nombre ([sip-phone]) y la contraseña (5678) son básicamente arbitraria, pero deben coincidir con los utilizados en la configuración del software cliente SIP del teléfono. Una vez más, añadirlo a la final del archivo:

```
 [sip-phone]
type=friend
context=outgoing
host=dynamic
secret=5678
```

Explicación:

[Sip-phone]	Título de la sección. Todas las líneas entre el título de la sección y la siguiente se aplican sólo a esta sección. Junto con el secreto, este nombre se utilizará para la autenticación del cliente SIP que se hace referencia en el plan de marcado de llamadas entrantes cuando necesitan ser enrutados a este teléfono.
type=friend	Definición del tipo de conexión. Tipo amigo es una combinación de ambos tipos de usuario y los compañeros, ya que el host remoto se puede conectar a este host, así como a la inversa.
context=outgoing	El nombre especificado aquí, que es arbitraria, determinará dónde llamadas salientes entrarán en el plan de marcado cuando se hacen con el teléfono asociado a esta sección.

host=dynamic	Configura el host al que este host es conectar, aunque dinámico se utiliza para indicar que el host de conexión usa una dirección IP dinámica.
secret=5678	Establece la contraseña. Se utiliza para la autenticación junto con el nombre (título) de esta sección.

Después de guardar estos cambios, ejecutar este comando:

```
~# asterisk -rx "sip reload"
~# _
```

En este punto, la idea es configurar el software cliente SIP del teléfono para autenticarse en Asterisk. Establecer el método para enviar información DTMF de señalización para rfc2833, que se recomienda para la señalización dentro de la banda y es el predeterminado para Asterisk.

El teléfono también debe intentar autenticarse en la dirección IP o nombre de dominio completo de la nueva sede de Asterisk usando el puerto SIP (5060), con un nombre y la contraseña **5678** de **sip-phone**. Si tiene éxito, una entrada similar a la siguiente aparecerá en /var/log/asterisk/messages:

Apr 6 00:53:59] NOTICE[2781] chan_sip.c: Peer 'sip-phone' is now Reachable. (16ms / 2000ms)

Dial Plan

En el corazón de cada PBX es su plan de marcado : la lógica de que, en función del número y el patrón de los dígitos marcados, determina qué se hacen las conexiones de cualquiera y todas las llamadas entrantes y salientes. El plan de marcado se guarda en **/etc/asterisk/extensions.conf**.

Contiene muchas cosas interesantes para empezar, pero estos no son necesarios para este ejercicio, hay que renombrar el nombre del archivo:

```
~# mv /etc/asterisk/extensions.conf /etc/asterisk/extensions.conf-org
~# _
```

Después de eso, crear una nueva versión vacía de este archivo y modifique su propiedad y los permisos:

```
~# touch /etc/asterisk/extensions.conf
~# chown asterisk.asterisk /etc/asterisk/extensions.conf
~# chmod 640 /etc/asterisk/extensions.conf
~# _
```

A continuación, edite el nuevo archivo vacío en **/etc/asterisk/extensions.conf** y agregue el siguiente contenido:

```
 [incoming]
exten => 0715551234,1,Dial(SIP/sip-phone,60)
exten => 0715551234,n,Hangup()
```

```
 [outgoing]
exten => _X.,1,Dial(SIP/${EXTEN}@provider)
```

```
exten => _X.,n,Hangup()
```

Explicación:

[incoming]	Las llamadas entrantes desde el canal SIP, proveedor, se insertan en este punto, ya que el título de esta sección coincide con el contexto = instrucción de entrada en la sección [proveedor] de sip.conf. De lo contrario, el nombre de esta sección es arbitraria.
exten ⇒ 0715551234,1,Dial(SIP/sip-phone,60)	Cuando una llamada entrante llega a este punto, y coincide con la extensión número 0715551234, una secuencia de dos eventos se desencadena, comenzando con el Dial () de aplicación, que conecta entre sí todos los diferentes tipos de canales en Asterisk. Aquí, se conecta a un canal SIP, llamado sip-teléfono, que está representado por una sección llamada [sip-phone] en sip.conf, con un tiempo de espera de anillo de 60 segundos.
exten ⇒ 0715551234,n,Hangup()	Después de la llamada entrante ha terminado, o si se ha alcanzado el tiempo de espera del anillo, el segundo evento (n = n +1) que coincide con esta ampliación se llevará a cabo. Esta es la (aplicación) Colgar, que simplemente cuelga el canal actual incondicionalmente.
[outgoing]	Las llamadas salientes desde el canal SIP, SIP-phone, se insertan en este punto, ya que el título de esta sección coincide con el contexto = instrucción de salida en la sección [sip-phone] de sip.conf. De lo contrario, el nombre de esta sección es arbitraria.
exten ⇒ _X.,1,Dial(SIP/${EXTEN}@provider)	Este evento siempre se iniciará por las llamadas que llegan a este punto, ya que "_X." es un cajón de sastre que coincide con cualquier número y combinación de dígitos que se puede marcar. El Dial () la aplicación se conectará a un canal SIP, llamado proveedor, con ${EXTEN} representa el número marcado.
exten ⇒ _X.,n,Hangup()	Después de la llamada entrante ha terminado, el segundo caso (n = n +1) que coincide con esta ampliación se llevará a cabo y la aplicación Colgar () colgará el canal actual.

Una vez que el nuevo plan de marcado se ha guardado, presentar los cambios en el proceso de Asterisk con este comando:

```
~# asterisk -rx "dialplan reload"
Dialplan reloaded.
~# _
```

Resultado

En este punto, debería ser posible para realizar y recibir llamadas a través de este nuevo sistema

Asterisk usando el **SIP test phone**. Con sólo un teléfono y un único canal PSTN, esta es una configuración muy mínima. Puede que no sea capaz de todo lo que mucho todavía se espera, pero es una buena base para comenzar, y es una demostración razonable de cómo se enrutan las llamadas entrantes y salientes a través del plan de marcado.

Ver también

Asterisk: minimal SIP configuration

Asterisk: ISDN BRI support with a Cologne Chip HFC-S PCI card

Asterisk: ISDN PRI support with a Sangoma A108 PCI Express card

Localphone

Publicado por:

Juan Carlos Ballesteros