

Manual de Asterisk y otras hierbas.



Fabián Sellés Rosa

Este documento se halla bajo la licencia FDL (Free Documentation License). Según estipula la licencia, se muestra aquí el aviso de copyright. Se ha usado la versión inglesa de la licencia, al ser la única reconocida oficialmente por la FSF (Free Software Foundation).

Copyright ©2009 Fabián Sellés Rosa. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Índice general

1. Introducción	9
1.1. Definiciones, acrónimos y abreviaturas	11
2. Instalación y configuración de Asterisk	13
2.1. Introducción	13
2.2. Compilando e instalando <i>Asterisk</i>	13
2.2.1. Descarga de las fuentes	13
2.2.2. Colocando los archivos	13
2.2.3. Preparación del entorno: configure y menuselect	14
2.2.4. Compilando	14
2.2.5. Ejemplos y documentación	14
2.3. Arquitectura, módulos y archivos de configuración	14
2.3.1. Directorios importantes de <i>Asterisk</i>	15
2.4. Ayuda para la configuración	15
2.5. Configurando el dialplan: extensions.conf	16
2.5.1. contexto [general] de extensions.conf	16
2.5.2. Contexto [globals] de <i>extensions.conf</i>	16
2.5.3. Sintaxis para construir dialplans	17
2.5.4. Hoja de referencia rápida para <i>extensions.conf</i>	20
2.6. Audio en <i>Asterisk</i> , reproducción y grabación	23
2.6.1. Reproducción	23
2.6.2. Grabación	25
2.7. Aplicaciones para la grabación	25
2.8. <i>Festival</i>	25
2.8.1. Instalación de <i>Festival</i>	25
2.8.2. Utilización de <i>Festival</i> en Asterisk	27
2.8.3. Música en espera: musiconhold.conf	27
2.9. Canales, protocolos y red.	27
2.9.1. Breve visión de funcionamiento de los canales en Asterisk	27
2.9.2. Canal SIP: <i>sip.conf</i>	28
2.9.3. Canal IAX: <i>iax.conf</i>	34
2.10. Bases de datos y <i>Asterisk</i>	36
3. Herramientas de análisis y monitorización de redes	41
3.1. <i>tcpdump</i>	41
3.1.1. Utilidad de <i>tcpdump</i>	41

Índice general

3.1.2. Creando expresiones, filtros BPF	41
3.1.3. Formato de salida	43
3.1.4. Parámetros de tcpdump	43
3.2. ngrep	47
3.2.1. descripción	47
3.3. <i>Wireshark</i>	49
4. GNU Free Documentation License	51
1. APPLICABILITY AND DEFINITIONS	51
2. VERBATIM COPYING	53
3. COPYING IN QUANTITY	53
4. MODIFICATIONS	54
5. COMBINING DOCUMENTS	55
6. COLLECTIONS OF DOCUMENTS	56
7. AGGREGATION WITH INDEPENDENT WORKS	56
8. TRANSLATION	56
9. TERMINATION	57
10. FUTURE REVISIONS OF THIS LICENSE	57

Índice de figuras

1.1. PBX manual, <i>Fuente: Wikipedia</i>	9
2.1. aspecto general de un trozo de <i>extensions.conf</i>	17
2.2. Hola Mundo en <i>Asterisk</i>	17
2.3. Extensiones lógicas y físicas	18
2.4. Prioridades	18
2.5. Prioridades	19
2.6. include y switch	19
2.7. Playback	24
2.8. Playback	25
2.9. Archivo <i>musiconhold.conf</i>	27
2.10. Canales en Asterisk	28
2.11. roles en Asterisk	31
2.12. Parámetros de usuario	31
2.13. Parámetros de cliente	32
2.14. Ejemplo de autentificación	33
3.1. Número de puntos de una dirección y subred asociada	42
3.2. Captura de <i>Wireshark</i>	49
3.3. <i>Wireshark</i> en acción	50

Índice de figuras

1 Introducción

La telefonía(del griego $\tau\eta\lambda\epsilon\varsigma$, lejos y $\varphi\omicron\nu\omicron\varsigma$, sonido) nace a finales del siglo XIX con la invención del telefono. La propia invención del telefono es objeto de controversia, tradicionalmente atribuida a [Alexander Graham Bell](#) por ser el primero en patentarla, se considera a otros como [Antonio Meucci](#) como sus inventores.

El despliegue de la red telefónica no fue ordenado. La red telefónica comenzó como una simple agrupación de conexiones entre clientes. El crecimiento de número de clientes y la necesidad de abaratar los costes de cableado, llevan a la creación de centralitas como *puntos de intercambio telefónico*. Poco a poco, el desarrollo y la innovación telefónica dejan de estar a cargo de los científicos o particulares y pasan a las grandes compañías telefónicas como Bell Labs y AT&T en EEUU, o las compañías telefónicas públicas

Estas primeras centralitas telefónicas no eran automáticas sino que estaban controladas por un operador humano. Para poder llamar a un abonado se descolgaba el teléfono y se solicitaba al operador la llamada, éste pinchaba la clavija de comunicación en su panel si la llamada era local, en caso de que llamada excediese la demarcación de la centralita. El operador debía contactar a otro operador que continuase con la petición para poder realizar la llamada.

Comienzan la adopción de PBX en las empresas y organizaciones, una PBX no es más que un panel de conexiones (veasé 1.1) que se instalaba en la empresa a cargo de un operador humano.

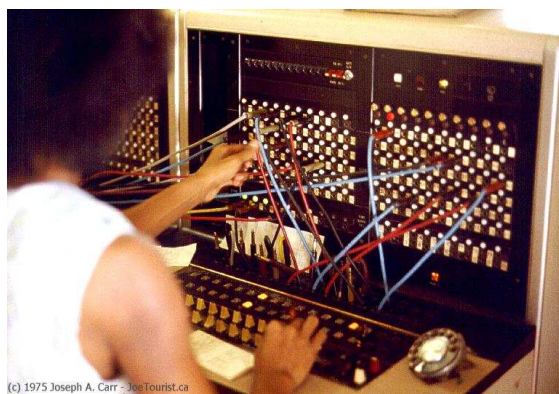


Figura 1.1: PBX manual, *Fuente: Wikipedia*

En 1960 comienzan a aparecer las primeras centralitas automáticas electrónicas analógicas que realizan la conmutación mediante relés. Tras la invención del transistor y el desarrollo de la electrónica digital se comienza a implantar la telefonía digital, sobre todo

1 Introducción

para *trunking*¹. Las centralitas se implementan con circuitos digitales y se vuelven más complejas y aparecen los primeros ordenadores comerciales.

La popularización de Internet y de las redes locales dan un nuevo sentido a la red telefónica. Ya no sólo sirve para transmitir voz entre abonados, se desarrollan los modems para transmitir datos digitales tratándolos como señales analógicas. Al hacerlo utilizan todo el ancho de banda asignado por lo que si se utilizan es imposible transmitir la voz a la vez.

Paralelamente comienza la implantación de una red telefónica completamente digital que permita la conexión simultánea de voz y datos, la RDSI² o ISDN³.

La explosión del número de accesos a Internet proporciona a la sociedad nuevos métodos de comunicación: mensajería instantánea, correo electrónico, foros, etc. Aunque, el servicio telefónico sigue fuertemente implantado en la sociedad para acceso a servicios críticos y comerciales (emergencias, servicio a domicilio, atención al cliente...) pero éste es *paralelo* a estas nuevas formas de comunicación.

Las PBX comienzan a implantarse mediante *software*, específico, diseñado del fabricante. Empiezan a ofrecer características avanzadas como *buzón de voz, conferencias, llamada en espera, música en espera...* La PBX se vende como una caja negra, a la que la organización conecta sus teléfonos y una línea externa, y dicha PBX los interconecta y además ofrece algunas de esas características avanzadas.

Aunque estas PBX se implementen mediante software, no existía antes de la VoIP una manera efectiva de integrar o que utilizara estas PBX. No era posible, por ejemplo, realizar llamadas utilizando un ordenador cuando se tenía al alcance de un click de ratón galerías de fotos, clips de películas, ...

En éste sentido el software libre (vease [Sta04]) tiene mucho que decir. Al devolver o entregar al usuario por primera vez la capacidad de crear su propia telefonía, de hacerla flexible, de explotarla y llevarla más allá para luego compartirla con una comunidad de usuarios.

Los programas son complejos y los errores siempre existen, un programa libre, que nos permite ver el código fuente, es un programa más depurado. Pues citando a Eric S. Raymond en su libro [Ray01]:

Dados los suficientes ojos, todos los errores aparecen

Lo que explica en parte la calidad del software libre, sometido a la revisión constante por pares.

La telefonía IP nos va a permitir unir dos mundos distintos, el de la telefonía y la informática, a través de un lenguaje común de comunicación: el protocolo IP. La telefonía IP basada en software libre nos va a permitir controlar todos los aspectos de nuestra instalación, haciendo que la centralita se amolde a nuestras necesidades y no nuestras necesidades a lo que puede hacer la centralita.

¹Llamada entre centralitas, canal de señalización compartido.

²Red Digital de Servicios Integrados.

³Integrated Services Digital Network.

1.1. Definiciones, acrónimos y abreviaturas

B2BUA (Back to Back user agent) *Agente de usuario de extremo a extremo*, es una entidad del protocolo SIP, según el RFC de SIP ([RSC+02]), un B2BUA tiene las siguientes funciones:

- Gestión de llamadas (facturación, desconexión automática ...)
- Interconexión de red (adaptación de protocolos)
- Ocultar la estructura de la red (direcciones privadas, topologías de red)
- Transcodificación entre las dos partes de una llamada.

BBDD Bases de datos.

CDR (Call Detail Record) *registro en detalle de llamadas*, que almacena datos pertenecientes al origen, destino, duración y otros de una llamada para la tarificación o el control en la PBX.

Dialplan , *plan de marcado*. Conjunto de reglas que determinan las extensiones y servicios a las que se pueden acceder desde una PBX.

IVR (Interactive Voice Response), *respuesta de voz interactiva* sistemas que permiten realizar gestiones informándonos a través de la voz. Como en los buzones de voz de los móviles y fijos o la reserva de entradas.

Jitter pequeño retraso o latencia producida entre paquetes.

Jitter-buffer *buffer de compensación* que almacena los paquetes entrantes para corregir el *jitter*.

NAT (Network Address Translation) *Traducción de direcciones de red*. Práctica utilizada para evitar el agotamiento de direcciones IP públicas. En las que una o varias IP públicas son utilizadas por varios dispositivos con una IP privada.

PBX (Private Branch Exchange) literalmente rama privada de intercambio, dispositivo que permite la interconexión de teléfonos (en este contexto extensiones), que interconectan una o varias salidas a la PSTN con estas extensiones.

POTS (Plain Old telephony Service), antiguo servicio de telefonía plano.

PSTN (Public Switched Telephone Network), Red telefónica conmutada (RTC), red que surge como evolución de la red POTS o de la RTB

QoS (Quality of Service) la calidad de servicio son una serie de tecnologías de los equipos de red que garantizan que una cierta cantidad o un tipo de datos llegará en un tiempo dado.

RFC (Request For comments) Serie de documentos normativos y constituyentes de protocolos de red, buenas prácticas para administradores de red y otros. Hay diferentes tipos desde estándares aceptados, a borradores o a simples recomendaciones.

1 Introducción

Softphone Aplicación que implemente un terminal telefónico de uno o varios protocolos de VoIP en un PC.

SGBD Sistema Gestor de bases de datos.

Transcoding , convertir de un codec a otro, generalmente se decodifica para pasarlo a audio sin compresión y se vuelve a codificar, esto tiene un coste computacional de $\Theta_{transcoding}(f) = \Theta(DESC(f) + COD(f))$

VoIP (Voice over IP) , voz sobre IP. En el presente documento se considera un término sinónimo a telefonía IP, aunque la VoIP tenga otros usos.

2 Instalación y configuración de Asterisk

2.1. Introducción

En este anexo se detalla y explica la configuración de algunas partes de una centralita *Asterisk* 1.4. Creemos que es de interés debido a que otros manuales o referencias como [VMSM07] se limitan a presentar las funciones de Asterisk y su potencialidad sin entrar demasiado en su configuración. Este anexo es una recopilación de la documentación existente y una localización española de la misma.

2.2. Compilando e instalando Asterisk

2.2.1. Descarga de las fuentes

Lo primero es descargar las fuentes del repositorio de *Asterisk*¹, *Asterisk* es un proyecto bastante activo y es habitual encontrar nuevas versiones del proyecto en un espacio de días. Las versiones empaquetadas (en deb u otros formatos) a menudo contienen versiones relativamente antiguas de las liberadas.

Los cambios en cada versión son varios, como en cualquier servicio de red. Hay cambios que afectan a la seguridad y a la funcionalidad, es conveniente leer los cambios de la versión [Diga] para valorar las consecuencias de la instalación.

La descarga puede realizarse en los repositorios oficiales de Digium, donde existe un paquete *simbólico* que apunta a la última versión (concretamente [Dige]). Ese paquete contiene todo el código fuente y los makefiles necesarios para compilar *Asterisk*.

Si queremos utilizar una base de datos MySQL o reproducir/grabar en MP3, será necesario igualmente descargarse el fichero de añadidos aparte, debido a problemas de licencias, [Digd].

2.2.2. Colocando los archivos

Aunque podemos descargarlo y ubicarlo en cualquier lugar, si seguimos el LSB (Linux Standard Base, un estándar para distribuciones), su lugar natural parece */usr/src*. Sea como fuere, una vez descargadas las fuentes, las desarchivamos y descomprimos mediante:

```
tar -xvzf asterisk-1.4-current.tar.gz
```

```
tar -xvzf asterisk-addons-1.4-current.tar.gz
```

¹Nosotros nos centraremos en la versión 1.4, la versión 1.6 aún no es estable.

2.2.3. Preparación del entorno: configure y menuselect

Es conveniente renombrar el directorio a por ejemplo *Asterisk* y *asterisk-addons*.

Tras esto hay que configurar el entorno, la orden `configure` buscará el compilador y las librerías instaladas para poder realizar la compilación. Es importante haber instalado previamente mediante paquetería todas las dependencias externas que se pudieran necesitar (ej: soporte para bases de datos, ODBC, *Festival*, ...). Especialmente importante es instalar las bibliotecas de *ncurses* que se utilizan en el menú de compilación.

Una vez satisfechas estas dependencias, vamos a configurar las opciones de *Asterisk* que queramos. *Asterisk*, es por diseño, bastante modular y admite muchas configuraciones. Para poder elegir adecuadamente puedes ejecutar `make menuselect`, lo cual compilará y mostrará una *interfaz tipo ncurses* de selección. En ella, el espacio sirve para seleccionar o deseleccionar y mediante `x` puedes salir y salvar las opciones, pulsa `h` para la ayuda.

2.2.4. Compilando

Ya con los deberes hechos, debemos afrontar la compilación. Es posible que en este punto exista algún error, generalmente este se debe a que o bien las librerías instaladas son demasiado antiguas o no se han satisfecho algunas dependencias. Pon especial atención en leer el error y en determinar que es lo que falta, generalmente se puede instalar desde paquetería. ² Para compilar ejecuta `make` (veasé [[GNUa](#)] y [[Gar](#)]), si todo sale bien el programa terminara sin presentar errores. Durante el proceso veras las sucesivas llamadas a *gcc* con las diferentes unidades compilables y cabeceras.

Una vez compilado, sólo nos queda instalarlo, para ello escribe `make install`, análogamente puedes desinstalarlo mediante `make uninstall`.

2.2.5. Ejemplos y documentación

Para poder empezar a utilizar *Asterisk*, los ejemplos son **vitales**. Para tenerlos en `/etc/asterisk`, simplemente ejecuta `make samples`. En tu directorio *Asterisk*, donde has descargado las fuentes, encontrarás también un directorio `doc` donde encontraras documentación sobre algunos aspectos del mismo. Por supuesto, siempre puedes consultar las paginas de `man`, aunque estas se centran en como manipular el demonio.

2.3. Arquitectura, módulos y archivos de configuración

Asterisk es muy modular. La distribución oficial de Digium incluye varios módulos agrupados por categorías según su funcionalidad. Desde codecs, formatos, canales, aplicaciones, etc.

Cada modulo tiene su propia configuración, sin embargo, comparten ciertas características comunes. Suele haber una sección `[general]` para configurar las principales

²En sistemas Debian y derivados, se puede buscar en que paquete se encuentra *X* dependencia, así `dpkg -S X`, se pueden utilizar también expresiones regulares

opciones de ese modulo, seguida de secciones especiales que complementen o sobrescriben a esa configuración general.

Los archivos de configuración se encuentran en */etc/asterisk* y cada modulo tiene su correspondiente *modulo.conf*. Lo primero que se debe hacer es revisar *asterisk.conf* y cerciorarnos que los directorios que allí se indican están creados en el sistema y tienen los permisos adecuados. Acto seguido, editar el archivo *modules.conf* que especifica que módulos se cargaran en *Asterisk* al iniciarse, la configuración inicial simplemente carga cualquier modulo que encuentre.

2.3.1. Directorios importantes de Asterisk

/usr/lib/asterisk/modules Directorio donde se encuentran los módulos compilados en extensión *.so*. Si se activa *autoload* en el archivo *modules.conf* se cargaran todos los archivos de este directorio.

/usr/share/asterisk Directorio donde se guardan las claves RSA, los archivos para reproducir de musica en espera , los script AGI. en una instalación desde paquetería para distribuciones Debian y Ubuntu.

/var/lib/asterisk Directorio donde se guardan las claves, la musica en espera y los sonidos del sistema en una instalacion desde las fuentes.

/var/spool/asterisk Directorio donde se guardan las conferencias salvadas por *MeetMe()*, los textos que son leidos por *Dictate()* y los mensajes de voz si no utilizamos una BBDD.

/var/run/asterisk Donde se encuentra el archivo *PID*³ del *Asterisk* que se esta ejecutando.

/var/log/asterisk Los *logs* y notificaciones realizado mediante por ejemplo la aplicación *Verbose()* o *NoOp()* se encuentran aquí, aunque también son dirigidos al demonio *syslog*.

2.4. Ayuda para la configuración

En las siguientes secciones, vamos a ir repasando algunos de los archivos de configuración para módulos más usuales de *Asterisk*. Puedes encontrar más ayuda en los archivos de ejemplo de cada modulo, los cuales suelen estar en la carpeta *samples* de */etc/asterisk*, o pueden ser generados mediante *make samples* en el directorio de fuentes de *Asterisk*.

³Process ID

2.5. Configurando el dialplan: extensions.conf

El archivo *extensions.conf* es uno de los archivos principales de *Asterisk*, pues en él es donde se configura el dialplan. El dialplan consiste en una serie de *contextos*, en los cuales existen *extensiones* que tienen varias *prioridades*.

Las extensiones de *Asterisk* son diferentes a las extensiones tradicionales de una PBX, no solo incluyen números telefónicos, pueden ser de texto o ser alguna de las extensiones especiales o por defecto.

El archivo *extensions.conf* tiene un *contexto* [general] y otro [globals]. Establecen algunas opciones respecto a como se tratará el dialplan (en el primero) y en el segundo como variables globales que se quieran definir. Vamos a hablar de ellos más tranquilamente.

2.5.1. contexto [general] de extensions.conf

Algunas de las opciones que podremos configurar son:

static si se define como 'yes' permite salvar el dialplan desde la consola de *Asterisk*.

writeprotect protección frente a escritura, si se deja como 'no' comandos como 'save dialplan' modificarán los ficheros de configuración.

autofallthrough si está activada esta opción, cuando una extensión haya acabado de ejecutar sus prioridades o la lógica salte a una prioridad inexistente hará que la llamada se cuelgue, señalizándola como BUSY (ocupada), CONGESTION o HANGUP dependiendo de que sea ésta la mejor opción para *Asterisk*.

clearglobalvars con cada recarga de *extensions.conf* o de *extensions.ael* se recargarán las variables globales de *Asterisk*. Si se desactiva las variables globales permanecerán con el valor que tienen en memoria, hasta que se vuelva a reiniciar *Asterisk* y a recargar el *extensions.conf*. Esto puede dar un quebradero de cabeza si se usa el archivo *extensions.conf* y el *extensions.ael* ya que *extensions.ael* siempre se carga después.

priorityjumping activa el salto de prioridad como respuesta, hay aplicaciones que tras su ejecución devuelve una prioridad a la que se debe saltar. Para que esto suceda debe activarse este parámetro.

2.5.2. Contexto [globals] de extensions.conf

En este contexto se definen las variables globales al dialplan. Sin embargo, pueden establecerse también utilizando la función GLOBAL(), aprovechamos para explicar las diferencias entre aplicación y función en *Asterisk*.

- Aplicaciones

Una aplicación es más cercana a una *función* de un lenguaje de programación: tiene parámetros y realiza acciones y al terminar devuelve un resultado. Una función es más cercana al concepto de macro con variables de C, es un *atajo* para el programador de *Asterisk*, también activan banderas (como la función `GLOBAL()`). Solo las aplicaciones pueden existir en cada línea de una extensión, dentro de la aplicación podremos utilizar una o varias funciones, incluso encadenadas.

Después de estos dos contextos especiales, estarán aquellos contextos y extensiones que necesitemos para armar nuestro dialplan. Todos ellos tendrán este aspecto.

```
[contexto]; esto es un comentario
exten => extensión,prioridad,Aplicación(Funciones)
```

Figura 2.1: aspecto general de un trozo de *extensions.conf*

2.5.3. Sintaxis para construir dialplans

- Ejemplo de dialplan: Hola Mundo

```
[general]
...
[globals]
...
[contexto]
exten => hola,1,Answer()
exten => hola,n,Playback(hello-world)
exten => hola,n,Hangup(30)
```

Figura 2.2: Hola Mundo en *Asterisk*

Este pequeño dialplan, sólo permite llamar a la extensión 'hola'. Una de las cosas que pueden sorprender es llamar a un nombre. Hay que tener en cuenta que en el mundo de la VoIP ya no hablamos sólo de terminales telefónicos, sino que hemos de tener en cuenta a softphones y otros dispositivos. Un problema es que un terminal telefónico jamás podrá marcar esta extensión, esto puede utilizarse para 'proteger' ciertas extensiones del marcado en estos terminales⁴.

- Contextos

⁴Aunque no la recomiendo por ser poco elegante. Además, algunos terminales si pueden enviar caracteres.

2 Instalación y configuración de Asterisk

Un contexto tiene ciertas similitudes a un *conjunto matemático*: las extensiones pertenecen a un contexto, y estos pueden incluirse en otros contextos o saltar a otro contexto.

Cada extensión SIP, o cada nueva llamada entrante de la PSTN, estará asociado a un contexto. El contexto en concreto al que se le asigne depende de la configuración.

El siguiente ejemplo sirve para aclarar que una extensión en *Asterisk* no equivale a una extensión real.

```
[general]
...
[globals]
...
[contexto]
exten => 123,1,Dial(${TRONCAL_PSTN}/956222222)
$
```

Figura 2.3: Extensiones lógicas y físicas

En el anterior dialplan, vemos como un lenguaje de script accede a una variable mediante `${}`, la cual contiene el nombre del canal del troncal de salida de la PSTN. Es importante notar que la extensión 123 no equivale a marcar dicha extensión, marcar 123 equivale a empezar a ejecutar la aplicación de prioridad 1 (que puede ser llamar a esta aplicación).

■ Prioridades

Las prioridades marcan qué aplicación se llamará primero. Esto implica que no importa donde las coloquemos en *extensions.conf* siempre quedaran ordenadas.

```
[general]
...
[globals]
...
[contexto]
exten => ext1,2,Dial(${EXTEN})
exten => ext1,1,Set(VAR="Hola!")
$
```

Figura 2.4: Prioridades

En el ejemplo 2.4, según puede leerse podría parecer que primero se ejecuta `Dial()` (la variable `${EXTEN}` es una variable estándar y especial en *Asterisk* que almacena la extensión que se marco) y posteriormente se establece la variable. En realidad, el orden

de ejecución viene determinado por la prioridad, así que primero se establece la variable y posteriormente se realiza el Dial().

Hasta ahora los dialplans mostrados son muy cortos, y no parece muy molesto andar recordando la última prioridad. Pero no es difícil que una extensión tenga varias líneas dedicadas a ella, especialmente si estamos definiendo una macro, por este motivo en Asterisk hay una prioridad especial, la prioridad *n*.

```
exten => ext1,1,Dial(${EXTEN})
exten => ext1,n,Set(VAR="Hola!")
exten => ext1,n,Aplicacion3...
$
```

Figura 2.5: Prioridades

En el ejemplo de la figura 2.5 muestra cómo se utiliza. Es bastante más cómodo de escribir y de leer, pero el orden seguido por las líneas después de la prioridad 1 ahora sí marcan el orden de ejecución.

▪ Directivas de contextos include => y switch

En *extensions.conf* la directiva `include => contexto` sirve para incluir un nuevo contexto en el actual. Si se realiza una llamada a este contexto, primero se buscará en este y luego, si no lo encuentra, en sus incluidos.

Por contra, `switch` sirve para cambiar a otro contexto, generalmente en otro servidor *Asterisk* con el que se está enlazado (trunking⁵).

```
[general]
...
[globals]
...
[contexto]
include => otrocontexto
switch => IAX2/servidor2/entrantes
exten => ext1,2,Dial(${EXTEN})
exten => ext1,1,Set(VAR="Hola!")
$
```

Figura 2.6: include y switch

En el ejemplo 2.6, si se realiza una llamada a *ext2*, esta se buscará siguiendo el siguiente algoritmo:

⁵se denomina 'trunking' al enlace troncal entre centralitas.

2 Instalación y configuración de Asterisk

1. Busca la extensión en el contexto actual.
2. Si hay alguna directiva `switch` , síguela y busca en ese contexto y en aquellos que se incluyan en él.
3. Si no se ha encontrado, coge el siguiente contexto incluido y vuelve a 1.

▪ Directivas de ficheros de configuración #include y #exec

- La directiva `#include`, incorpora en ese punto el archivo que le indiquemos, hay que tener en cuenta que la inclusión se realiza desde el directorio donde se encuentra los archivos configuración (por lo general `/etc/asterisk`, consulta `asterisk.conf`).
- La directiva `#exec <script>` ejecuta el script y escribe la salida del script en el punto donde se le indiquemos. Para poder ejecutar el script el usuario que ejecuta *Asterisk* debe tener permisos para llegar hasta él y ejecutarlo. Además debes incluir en `asterisk.conf` dentro del apartado `[options]` , `execincludes=yes`.

2.5.4. Hoja de referencia rapida para extensions.conf

Acceso a variables

Para acceder a una variable, hace falta encerrarla entre llaves `${VARIABLE}`.para establecerla es necesario utilizar la aplicación `Set()`, si queremos que sea global se ha de usar la función `GLOBAL ()`.

Las variables y las aplicaciones o funciones no son *case sensitive*, es decir, 'VAR' y 'var' son la misma variable,aunque suelen escribirse en mayúsculas. No se admite el uso de guiones bajos '_' para declarar una variable.

Ámbito de las variables

`Set(GLOBAL(VAR)=2)`, *VAR es global.*

`Set(VAR=2)`, *VAR sólo existe en el canal actual.*

`Set(_VAR=2)`, *VAR existe en el canal actual, y se hereda al siguiente canal.*

`Set(..VAR=2)`, *VAR existe en el canal actual, y se hereda indefinidamente.*

Operadores

Se pueden utilizar operadores, dentro de cualquier función. Para utilizarlos es necesario encerrar la operación entre corchetes `$()`. Ejemplo:

2.5 Configurando el dialplan: extensions.conf

```
exten => 5000,1,Set(CONTADOR=0) ;Establecemos variables
exten => 5000,n,While($[${CONTADOR} < 10]) ;mientras contador sea menor que 10
exten => 5000,n,Set(CONTADOR=${[${CONTADOR} + 1]) ;suma 1
exten => 5000,n,Dial(5000) ; llama
exten => 5000,n,EndWhile
```

operadores lógicos

```
expr1 | expr2 (OR logico)
expr1 & expr2 (AND logico)
!expr (negación lógica)
expr1 = expr2
expr1 != expr2
expr1 < expr2
expr1 > expr2
expr1 <= expr2
expr1 >= expr2
```

operadores aritméticos

```
expr1 + expr2
expr1 - expr2
- expr
expr1 * expr2
expr1 / expr2
expr1 % expr2
```

Expresiones regulares

```
expr1 : regexp
```

devuelve " " si no hay concordancia, y la cadena resultante en otro caso.

Precedencia de operadores (de mayor a menor)

1. Paréntesis: (,)
2. Operadores unarios !, -
3. Comparación con expresiones regulares: :
4. Operadores aritméticos de multiplicación y división, modular: *, /, %
5. Operadores aritméticos suma y resta: +, -
6. Operadores comparaciones: =, !=, <, >, <=, >=
7. Operadores lógicos: |, &

utilización de patrones

Pueden utilizarse en cualquier extensión, siempre que sean precedidas con un guión bajo:

X	concuerta con cualquier dígito del 0-9
Z	concuerta con cualquier dígito del 1-9
N	concuerta con cualquier dígito del 2-9
[1237-9]	concuerta con 1,2,3,7,8,9
.	comodín, concuerta con uno o más caracteres (clausura positiva de Kleene).
!	comodín, concuerta con cero o más caracteres inmediatamente.

Ejemplo:

```
exten => _956XXXXXX,1,Dial(${EXTEN}) ; llama a cualquier numero de Cádiz
; (del 956000000 al 956999999 )
```

a	llamada cuando el usuario pulsa '*' durante la presentación del buzón de voz.
h	extensión de colgado
i	extensión invalida
o	extensión del operador, llamada cuando se pulsa 0 en el buzón de voz
s	extensión entrante en un contexto
t	extensión de tiempo de espera agotado
T	extensión llamada por <i>AbsoluteTimeout()</i>
failed	usada cuando un archivo .call de llamada automática falla
fax	usado para la detección de fax en canales Zaptel
talk	extensión usada por <i>BackgroundDetect</i>

Cuadro 2.1: Extensiones estándar

2.6. Audio en Asterisk, reproducción y grabación

2.6.1. Reproducción

El audio en *Asterisk* juega un papel importante. En el mundo de las PBX tradicionales es habitual que mientras el llamante no llega a su destino se le entretenga con música en espera. Además la reproducción de audio es necesaria para la construcción de sistemas IVR.

Asterisk, en su instalación, facilita algunos sonidos comunes para, por ejemplo, el buzón de voz, o para notificar fallos de la red (el usuario no se encuentra, etc), Aunque podemos llevar esto mucho más allá, mediante *Festival* y otras aplicaciones.

▪ Codecs y carga de CPU

En principio Asterisk puede reproducir de muchísimos codecs (GSM,G711,G729,MP3) y de archivos de audio sin codificar (WAV), siempre que nos sea posible deberíamos pasar cualquier archivo que queramos reproducir a WAV, pues decodificar es un trabajo que puede cargar mucho a la CPU con cada reproducción. Aunque también es cierto que un archivo codificado ocupa bastante menos que uno que si lo está, pero sacrificamos espacio por cantidad de llamadas que pueden ser atendidas.

Si quieres saber cuantos codecs soporta *Asterisk* escribe en la consola de *Asterisk*, `core show codecs audio`. Si además quieres conocer el coste de trasladar de un codec a otro puedes verlo con `core show translation`.

Asterisk no soporta por defecto la reproducción de mp3, puedes activarla compilando el paquete de addons y activando el soporte para MP3, [Digd], o instalando el paquete de tu distribución.

▪ Definir un idioma

Un idioma en *Asterisk* es un conjunto de sonidos colocados en el directorio de sonidos de *Asterisk* (revisa *asterisk.conf* para saber dónde). Se puede definir un lenguaje en el archivo de configuración del canal (por ejemplo `sip.conf` o `iax.conf`) o utilizando `Set(CHANNEL(language)=idioma)` que cambia el idioma para la llamada en curso.

Esto permite que aplicaciones que reproducen sonidos, sobre todo al usuario, como `Playback()`, `saydigit()`, `saynumber()` puedan hacerlo en el idioma del llamante. Es bastante común definir un idioma global (español , por ejemplo) y para ciertos contextos, (entrantes-internacionales) ponerlo a un idioma más apropiado como inglés. Incluso no es difícil presentar un menú de selección de idioma por usuario.

Playback() La aplicación `Playback` reproduce un archivo de audio que esté localizado en el directorio de sonidos de *Asterisk*. No es necesario incluir la extensión del archivo, de hecho es recomendable no hacerlo. *Asterisk* reproducirá el archivo que menos coste de conversión tenga.

```
[general]
...
[globals]
...
[contexto]
...
;Hay que responder el canal antes de la reproducción
exten => ext1,n,Answer()
exten => ext1,n,Wait(5)
; si el sonido es muy corto poner segundos
; de espera permite que el buffer de audio
; se llene y se escuche completamente
exten => ext1,n,Playback(archivo_de_audio1[&archivo_de_audio2...][j])
;Despues de la reproducción se debe colgar para liberar el canal.
exten => ext1,n,Hangup()
```

Figura 2.7: Playback

La sintaxis es la siguiente:

La opción **j** provoca que al acabar de reproducir el archivo se salta a la prioridad $n+101$. Una cosa a tener en cuenta es que mientras se esté reproduciendo el archivo no se admiten tonos de marcado.

Background() La aplicación `Background()` reproduce un audio al igual que `Playback()`, pero permite marcar mientras se escucha. En concreto, en el momento en que se comienza a marcar se deja de reproducir el sonido de audio. Puede ser útil para construir IVR.

SayDigits() Lee las cifras que se pasan como argumento. Ej: `SayDigits(1234)` lee "1", "2"...

SayNumber() Lee el numero que conforman todas las cifras del argumento. Ej: `SayNumber(1234)` leerá *mil doscientos treinta y cuatro*.

SayAlpha() deletrea la cadena pasada como argumento, en el caso de pasar sólo números el resultado es el mismo que `SayDigits()`.

Obviamente hay que grabar previamente el sonido que queramos reproducir. Esto puede llegar a ser un problema, ya que no todos los micrófonos dan la calidad necesaria para una buena reproducción. Si queremos hacerlo a nivel profesional, grabar algunos mensajes para nuestra centralita requerirá de un buen locutor, un espacio bien sonorizado y con instrumentación adecuada y de tiempo.

Sería genial tener otra posibilidad, y la posibilidad existe. ***Festival*** es un programa libre de *text-to-speech* que sintetiza la voz de un texto leído (veasé [Fes]). No es perfecto

(y menos en otro idioma que no sea inglés), pero es suficiente para la mayoría de los casos.

2.6.2. Grabación

Para la grabación no hace falta ninguna aplicación especial, podríamos coger cualquier aplicación de audio como *Audacity* o una grabadora de sonidos. No obstante, *Asterisk* incorpora una serie de aplicaciones para la grabación en el sistema telefónico. Esto por sí mismo no es interesante, pero puede utilizarse para crear una macro que reproduzca un mensaje de bienvenida personalizado por el usuario.

2.7. Aplicaciones para la grabación

Record() La aplicación `Record()` graba en un archivo la voz del llamante.

```
exten => ext1,n,Record(archivo_de_audio1.formato[,silencio
,duracion_maxima,opciones])
```

Figura 2.8: Playback

Dial La aplicación `Dial` con el parámetro `w` o `W`, se activa al teclear `*1`.

Dictate `Dictate` nos permite grabar un mensaje y escucharlo después, funciona como un dictáfono.

2.8. Festival

2.8.1. Instalación de Festival

Aunque `festival` puede instalarse desde las fuentes, es recomendable instalar desde paquetería. En concreto en una `Ubuntu 8.04`, podemos proceder como `root` con:

```
apt-get install festival festvox-ellpc11k
```

Una vez instalados, debemos cambiar algunos ficheros de configuración que nos permitirán ejecutar `festival` desde *Asterisk*.

En concreto, ve al directorio `/usr/share/festival` y edita el fichero `festival.scm` debes añadir esto al final del fichero:

2 Instalación y configuración de Asterisk

```
;; Enable access to localhost (needed by debian users)
(set! server_access_list '("localhost\\.\localdomain" "localhost"))

;;; Command for Asterisk begin

(define (tts_textasterisk string mode)
  "(tts_textasterisk STRING MODE)
  Apply tts to STRING. This function is specifically designed for
  use in server mode so a single function call may synthesize the string.
  This function name may be added to the server safe functions."
  (let (wholeutt (utt.synth (eval (list 'Utterance 'Text string))))
    (utt.WAVE.resample wholeutt 8000)
    (utt.WAVE.rescale wholeutt 5)
    (utt.send.WAVE.client wholeutt)))

;;; Command for Asterisk end
```

Tras la instalación, *Festival* utilizará por defecto el idioma inglés o ninguno, vamos a cambiarlo para que utilice el español⁶. Para ello vamos a editar el archivo `/usr/share/festival/init.scm`, en el que debemos reemplazar

```
;;; Default voice (have to do something cute so autoloads still work)
(eval (list voice))
(provide 'init)
```

por:

```
;;; Default voice (have to do something cute so autoloads still work)
(eval (list voice_el_diphone))
(provide 'init)
```

Así le decimos a *Festival*, que utilice siempre la voz española que instalamos antes. Para probarlo, simplemente teclea `festival`, y una vez en la consola del programa:

```
festival>(SayText "Hola mundo hablado!")
```

Además podéis convertir cualquier fichero a cualquier formato de audio mediante la utilidad `text2WAVE`. Simplemente

```
text2WAVE entrada.txt -o salida.WAV
```

¡Cuidado! no hay que utilizar acentos, no los interpretará bien, y aunque las exclamaciones y las interrogaciones cambian el tono de la lectura, no es demasiado notorio en la versión española.

⁶vamos a hacerlo de manera global.

2.8.2. Utilización de Festival en Asterisk

```

exten => hola,1,Answer()
exten => hola,n,Festival(hola mundo)
exten => hola,n,Hangup(30)

```

7

2.8.3. Música en espera: musiconhold.conf

El archivo define las clases de música, que luego podrán ser reproducidas mediante la aplicación `MusicOnHold(clase)`. Si se llama a `MusicOnHold` sin parámetros se toma la clase que este configurada en el archivo de configuración del canal o la que esté en la variable `CHANNEL(musicclass)`.

```

[clase]
mode=[files,custom]
;files reproduce los archivos especificados en directorio
;custom permite definir una aplicación propia de la que extraer el audio

;al igual que playback,Asterisk reproducira el archivo que menos
; cueste transcodificar

;cuando se utiliza custom no es necesario definir un directorio
directory=/var/lib/asterisk/moh

;random=yes si se especifica en modo files, se activa reproducción aleatoria

;application=aplicación a ejecutar

; es buena idea ejecutar un script, y en el script llamar a una
; aplicación para reproducir streaming como xmms o mplayer

```

Figura 2.9: Archivo *musiconhold.conf*

2.9. Canales, protocolos y red.

2.9.1. Breve visión de funcionamiento de los canales en Asterisk

Asterisk es una centralita PBX, se dice también que es un B2BUA.⁸ Los terminales ven a Asterisk como un servidor y a diferencia de un proxy o de un router, Asterisk se encuentra en medio de toda la señalización.

⁷No hay que poner comillas dobles, se insertan después.

⁸*Back to back user agent*, agente de extremo a extremo podría ser una buena traducción.

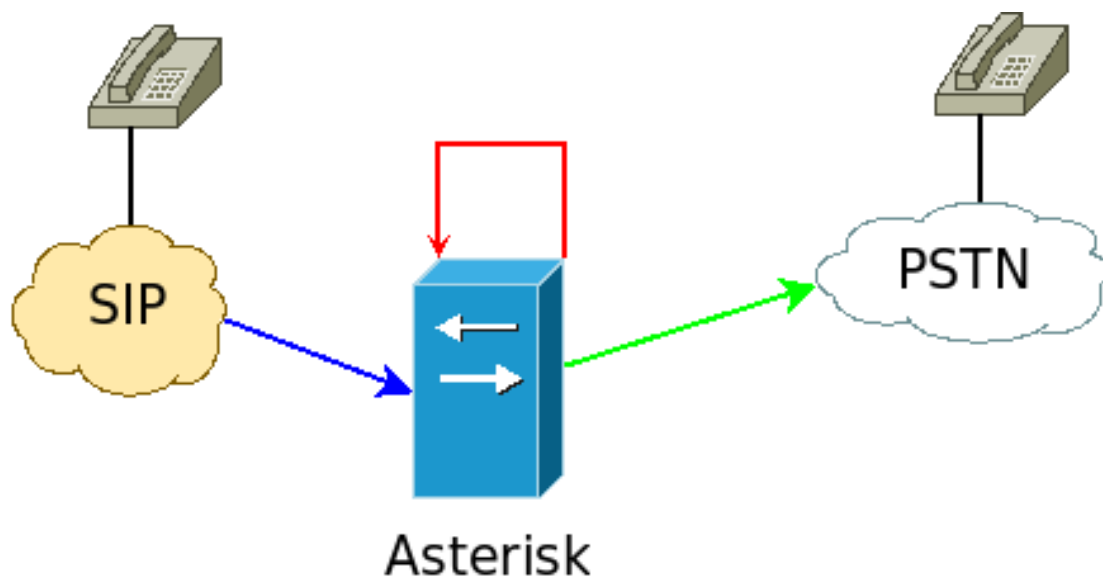


Figura 2.10: Canales en Asterisk

Una de las responsabilidades de un B2BUA es la de transcódecificar si cada extremo utiliza un codec distinto. Además Asterisk actúa como gateway⁹, y es capaz de convertir una llamada entrante en el protocolo *A* para que pueda recibirla el otro extremo que utiliza el protocolo *B*.

Ahora hablaremos de tres protocolos habituales en la VoIP, dos son estándares abiertos y uno no es un estándar pero es bastante utilizado en instalaciones de Asterisk. Además de estos existen otros canales para dar capacidades avanzadas a Asterisk como el canal de Gtalk, Zaptel/DAHDI, etc.

Para clarificar un poco todo esto, vease la figura 2.10. *Asterisk* convierte cada canal entrante (azul) en un canal local de audio (rojo). De este canal local lo vuelve a convertir al canal de destino (verde).

2.9.2. Canal SIP: sip.conf

El canal SIP de *Asterisk* sigue la misma sintaxis general de los archivos de configuración en *Asterisk*, tiene una parte general donde se configuran las opciones generales del canal. Así mismo, permite definir los clientes SIP que tendremos. Si hemos configurado una base de datos y la gestión en tiempo real de *Asterisk* (*extconfig.conf*) también podremos definir clientes SIP dinámicamente.

Una de las cosas que debemos tener en cuenta es que Asterisk **no es un servidor proxy** de acuerdo a [RSC+02]. *Asterisk* siempre se encuentra (a no ser que lo indiquemos explícitamente) en medio de la conversación entre terminales SIP, tanto a nivel de

⁹convertidor de medios

señalización como de datos. Otra cosa a tener en cuenta es que *Asterisk* no soporta conexiones SIP sobre TCP.

La sección `[general]` de *sip.conf* en el fichero de ejemplo, puede agruparse en subsecciones, hay opciones relativas a la configuración de la red (interfaz, puerto, etc), a como gestionar los terminales tras la NAT, el *jitter-buffer*¹⁰, o al propio comportamiento de *Asterisk* como terminal SIP. Tras las opciones generales vienen los clientes que queramos especificar, en la configuración del cliente debemos especificar al menos, tipo de cliente, campo host, y la contraseña o claves RSA utilizadas.

Comentamos ahora algunas de las opciones de configuración más usuales.

Opciones básicas del canal SIP, `[general]`

context define el contexto de las llamadas entrantes.

allowguest *yes/no* permite o no las llamadas de invitados.

realm nombre del reino de autenticación para el Digest. Debe ser un nombre único, por ejemplo el nombre de dominio. Si se especifica un nombre en *asterisk.conf* se utiliza ese.

bindport puerto donde escuchará *Asterisk* las conexiones entrantes SIP.

bindaddr dirección IP por la que escuchará *Asterisk*, por defecto todas (0.0.0.0).

srvlookup *yes/no* activa las búsquedas SRV en servidores DNS¹¹ para las llamadas salientes. *Nota: Asterisk sólo utiliza la primera entrada SRV.*

domain define el dominio por defecto, el dominio es la parte del SIP-URI que va después de la '@'. Ej: user@dominio.com. Si se configura, *Asterisk* sólo permitirá INVITE y REFER a éste dominio. Además puede asociarsele un contexto donde se dirigirán las llamadas de usuarios del dominio.

pedantic *yes/no* hace que *Asterisk* sea menos permisivo al interpretar SIP. (por defecto a no)¹².

disallow lista de codecs que no se permiten, se admite *all* para marca todos los codecs, o el codec en particular (*gsm, ulaw, alaw...*)

allow activa el codec. (Es muy habitual encontrar un `disallow=all`, seguido de los `allow` con los codecs que se permiten).

¹⁰Generalmente los protocolos son no orientados a conexión al ir sobre UDP. Por lo que muchos paquetes pueden perderse o llegar con retraso, la utilización de un *buffer de compensación* evita introduciendo un pequeño retraso los cortes en el audio.

¹¹una búsqueda DNS de tipo SRV no sólo es una consulta de DNS donde se especifica el nombre de dominio y el puerto donde debería estar escuchando el servicio. Más información en [\[GVE00\]](#)

¹²No hay razón alguna para desactivarlo, y sí alguna para activarlo (para ejemplo soportar los dominios con ñ)

2 Instalación y configuración de Asterisk

mohinterpret clase de música en espera que se reproducirá, vease 2.8.3 en la página 27.

language idioma por defecto para todos los usuarios.

useragent cadena que marca el agente de usuario , por defecto 'Asterisk PBX'

rtptimeout número de segundos de espera antes de que el canal se cuelgue por falta de actividad RTP.

rtpholdtimeout números de segundos de espera antes de que el canal se cuelgue por falta de actividad RTP mientras está en espera.

rtpkeepalive segundos en los que se mandará un paquete RTP para mantener la NAT activa.

sipdebug *yes/no* activa el modo de depuración SIP pegando en consola cualquier mensaje SIP. Puede activarse escribiendo en consola de asterisk `*CLI> sip set debug`.

allowsuscribe *yes/no* permite que los usuarios soliciten el método SIP SUSCRIBE, lo que permite mandar mensajes INFO ¹³ o para SIMPLE¹⁴

externip IP externa. Útil si *Asterisk* está tras un dispositivo que hace NAT y tiene por tanto una IP privada.

externhost a colación con lo anterior puede especificarse un nombre de dominio, con lo cual se harán consultas periódicas para conocer la IP.

externrefresh segundos para hacer la consulta anterior.

localnet Ej: 192.168.0.0/255.255.0.0 (o 192.168.0.0/16), red de area local privada en la que esta *Asterisk*, útil para detectar terminales tras la NAT. Por defecto se encuentran todas las redes privadas, definidas en [RMK+96].

nat *yes/no* activa el tratamiento para NAT de *Asterisk*.

canreinvite *yes/no/nonat/update* permite (si esta activado) que los datos vayan directamente de llamante a llamado, evitando que *Asterisk* este en medio.

- La opción **nonat** activa esta opción para dispositivos para los que no se detecte NAT ¹⁵.
- La opción **update** puede combinarse con **nonat**, para que *Asterisk* salga del camino de los datos en lugar de generar un nuevo INVITE manda un UPDA-TE.

autodomain genera dominios a los que atenderá *Asterisk* en base al nombre del sistema, IP externa y FQDN¹⁶.

¹³Usualmente para pasar tonos DMTF via SIP

¹⁴*SIP Instant Messaging and Presence*, protocolo para mensajería y presencia.

¹⁵Es decir que no cuadren con la opción localnet.

¹⁶Fully Qualified Domain Name, nombre de dominio

fromdomain actualiza la cabecera From para que aparezca un dominio y no la IP de salida.

■ Tipos de cliente SIP

Existen tres roles de usuarios en *Asterisk*, el rol *usuario* permite realizar llamadas **desde** *Asterisk*, el rol *cliente* permite realizar llamadas **hacia** *Asterisk*, y el rol *amigo* es a la vez usuario y cliente de *Asterisk*.

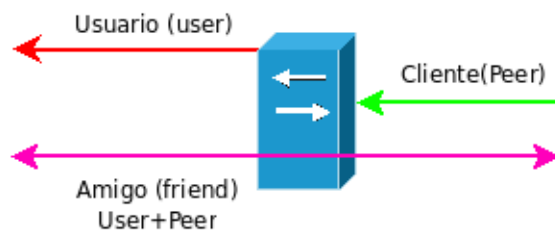


Figura 2.11: roles en Asterisk

En las tablas 2.12 y 2.13 pueden verse los parámetros configurables por cada *usuario* y *peer*. Si no se especifican se tomarán de la configuración general. Sin embargo, hay parámetros que sólo existen para los *roles* como **type**, que marca el rol.

- context	- nat	- progressinband	- allowoverlap
- callingpres	- callgroup	- promiscredirect	- allowsubscribe
- permit	- pickupgroup	- useclientcode	- allowtransfer
- deny	- language	- accountcode	- subscribecontext
- secret	- allow	- setvar	- videosupport
- md5secret	- disallow	- callerid	- maxcallbitrate
- dtmfmode	- insecure	- amaflags	- rfc2833compensate
- canreinvite	- trustpid	- call-limit	

Figura 2.12: Parámetros de usuario

■ Autenticación y registro

2 Instalación y configuración de Asterisk

- context	- allow	- allowoverlap	- host
- callingpres	- disallow	- allowsubscribe	- port
- permit	- insecure	- allowtransfer	- qualify
- deny	- trustpid	- subscribecontext	- defaultip
- secret	- progressinband	- videosupport	- rtptimeout
- md5secret	- promiscredir	- maxcallbitrate	- rtpholdtimeout
- dtmfmode	- useclientcode	- mailbox	- sendrpid
- canreinvite	- accountcode	- username	- outboundproxy
- nat	- setvar	- template	- rfc2833compensate
- callgroup	- callerid	- fromdomain	
- pickupgroup	- amaflags	- regexten	
- language	- call-limit	- fromuser	

Figura 2.13: Parámetros de cliente

Asterisk mantiene una tabla con los clientes SIP que tiene, entre otras cosas esta tabla sirve para conocer el estado del cliente y como llegar hasta él si esta activo. Además *Asterisk* a su vez puede ser cliente de un proxy SIP o de otros *Asterisk* que también tienen que conocer su estado y localización actual.

Asterisk permite tres métodos de autenticación:

Plana la contraseña se guarda en texto plano en el fichero de configuración, y se envía también en texto plano. La seguridad es casi nula.

MD5 o digest el servidor tras recibir la petición de registro por parte del cliente, envía una petición de autenticación con un *nonce*. El cliente toma ese *nonce* y la clave, los combina y devuelve un *hash* o resumen en formato alfanumérico, que el servidor *Asterisk* comprobará con el *hash* que tiene. Si concuerdan se acepta el registro.

Clave publica RSA , como se ha demostrado que MD5 es inseguro al encontrarse colisiones¹⁷, puede utilizarse un esquema de clave publica y privada basado en SHA-1. Los usuarios se asocian a una clave , clave que tendremos que distribuir a otros servidores *Asterisk*. De modo que cuando se reciba una llamada de estos se comprobara que la clave adjunta al registro coincide con la guardada.

¹⁷MD5, SHA-1 y otros son algoritmos de dispersión (del inglés *hash*) y son utilizados ampliamente en informática. Una de las propiedades de los algoritmos de dispersión es que son sobreyectivas(cada elemento de la imagen tiene un único origen) y es difícil hallarles la inversa.


```

;sip.conf
...

[usuario]
;autenticacion en texto plano clave 'blabla'
type=friend
secret=blabla

[usuario-md5]
;autenticacion con md5, clave MD5(blabla)
type=friend
md5secret=9b36b2e89df94bc458d629499d38cf86

[usuario-rsa]
;autenticacionf con claves.
type=user
;en inkeys ha de pasarse el nombre del fichero .PUB generado por astgenkey
inkeys=nombredelaclavepublica

[cliente-rsa]
;autenticacion con claves.
type=peer
;en outkey ha de pasarse el nombre del fichero .PUB
outkey=nombredelaclavepublica

```

Figura 2.14: Ejemplo de autenticacion

para crear las claves se puede utilizar el script `astgenkey`. una vez creadas pueden comprobarse que *Asterisk* las reconoce con el comando de consola `show keys`, las claves generadas deben colocarse en `/var/lib/asterisk/keys` si se ha compilado *Asterisk* desde las fuentes, o en `/usr/share/asterisk/keys` para la versión desde paquetería.

En la figura 2.14 puede observarse un ejemplo de uso.

- **Definiendo clientes**

Asterisk soporta dos maneras de especificar clientes, una manera es la *estática* que no es mas que añadir al final de `sip.conf` los clientes que queramos definir. Y la otra es una manera *dinámica* (realtime) para la cual es necesario haber definido un conector para BBDD y haber configurado `extconfig.conf`

Estático basta añadir al final del fichero de configuración lo siguiente:

```
[usuario]
type=valor
clave=valor
...
```

Donde usuario es el nombre de usuario (*usuario@dominio.com*) y el resto son parámetros configurables. Las claves que pueden utilizarse dependen del tipo de usuario que se defina. En las tablas 2.12 y 2.13 se listan todas las opciones, en 2.9.2 en la página 29 se listan algunas opciones generales del canal, algunas de estas pueden utilizarse en los clientes.

Si vamos a describir de manera estática los clientes, y los clientes que describimos son muy parecidos entre si (grupos de una oficina, o una clase etc) podemos utilizar las plantillas, por ejemplo:

```
[plantilla](!)
type=friend
disallow=all
allow=gsm
... mas opciones

[usuario](plantilla)
;define un usuario con las opciones de plantilla

[nueva-plantilla](!,plantilla)
allow=ulaw

[usuario-raro](nueva-plantilla)
;usuario con las opciones de plantilla + nueva-plantilla
```

Dinámico para ello ha de existir un conector ODBC y una base de datos, así como tener configurado *extconfig.conf*. En realidad su funcionamiento es relativamente sencillo: tenemos una tabla en la base de datos con todas las opciones que utilicemos. Así evitamos que la tabla de clientes SIP este en memoria, esto puede darnos problemas con aquellos usuarios que estén detrás de una NAT pues *Asterisk* al no tenerlos en memoria dejara de enviarles mensajes *keepalive*¹⁸.

2.9.3. Canal IAX: iax.conf

El canal IAX, sigue la misma estructura que el canal SIP: primero una sección general y luego los diferentes clientes IAX. IAX¹⁹ es un protocolo no estándar desarrollado por

¹⁸Mantener vivo, sirve para refrescar las redirecciones de dispositivos con NAT.

¹⁹*Inter Asterisk Exchange*, IAX2, la primera versión ya no existe

Digium, que transporta de manera binaria a la vez señalización y datos. Por tanto es un protocolo de voz *in-band*. Trabaja sobre UDP y a diferencia de SIP y H323, utiliza un único puerto, por defecto 4569, lo que lo hace mas manejable a la hora de trabajar con NAT ²⁰.

Al ser un protocolo no-estándar es muy difícil encontrar teléfonos que lo implementen, así que su utilización para terminales es escasa. Sin embargo, es interesante como protocolo de aplicación específica para *Asterisk*. Permite intercambiar dialplans e interconectar servidores *Asterisk*, ejecutar DUNDI... etc. Aunque todo lo anterior se supone que es *agnóstico* al protocolo utilizado, en la práctica se utiliza IAX. IAX es un protocolo útil pero no es un buen protocolo sobre el que construir todo el sistema de telefonía, es específico, no estándar y se establece en contra de la corriente de la telefonía tradicional que ha migrado a protocolos *out-band* (como RDSI, SIP o H323).

Opciones básicas del canal IAX, [general]

bindport puerto donde escuchará *Asterisk* las conexiones entrantes IAX, por defecto 4569 UDP.

bindaddr dirección IP por la que escuchará *Asterisk*, por defecto todas (0.0.0.0).

iaxcompat *yes/no* permite abrir nuevos hilos cuando se ejecuta una sentencia switch (sección 2.6 en la página 19).

disallow lista de codecs que no se permiten, se admite *all* para marca todos los codecs, o el codec en particular (*gsm, ulaw, alaw...*)

allow activa el codec. Es muy habitual encontrar un *disallow=all*, seguido de los *allow* con los codecs que se permiten.

mohinterpret clase de música en espera que se reproducirá, vease 2.8.3 en la página 27.

language idioma por defecto para todos los usuarios.

delayreject *yes/no* introduce un pequeño retraso en el dialogo de registro para evitar ataques por fuerza bruta.

trunkfreq cada cuantos segundos se manda un mensaje de tipo troncal.

iaxthreadcount numero de hilos que pueden lanzarse.

■ Definir clientes

La definición de los clientes se realiza de la misma manera que en SIP. sin embargo hay algunas opciones inherentes a un cliente IAX.

²⁰Debido a que sólo hay que garantizar la apertura de un único puerto.

```
[servidor1]

;host=A.B.C.D definimos una IP del servidor estatica
;host_=dynamic el servidor tendra que registrarse como cliente.
;defaultip=A.B.C.D funciona junto a host=dynamic,
;si el equipo es dinamico pero
;no se ha registrado se tomara
;esta ip como la del servidor.
;transfer=yes|no|mediaonly. permite que Asterisk
;reinvite para que los terminales
; se conecten directamente o no
; o sólo para los datos , es similar a la opcion
; canreinvite del canal SIP.
host=192.168.1.2
type=friend
auth=rsa,md5,plaintext
inkey=servidor1
outkey=servidor1
;a diferencia del canal SIP. IAX
;permite definir prioridades de autentificacion
;y admite las tres maneras de autenticarse.
trunk=yes
;activa el modo troncal, cuando varias llamadas
;van al mismo destino
;(3 clientes a otro servidor,por ejemplo)
;en lugar de mandar una trama
;por cliente, se manda una trama para
;los 3 con el audio de los 3.
```

2.10. Bases de datos y Asterisk

Se pueden usar BBDD en *Asterisk* para los siguientes casos:

- Para configuración dinámica de *Asterisk* (Realtime)
- Para almacenar los registros de llamadas (CDR).
- Para almacenar mensajes de voz.
- Como una base de datos normal para su uso en el dialplan o con scripts.

Asterisk provee de conectores para bases de datos PostgreSQL y un conector ODBC²¹, también uno para MySQL aunque no se encuentra en la distribución estándar sino en el paquete de añadido [Digd].

▪ ¿ODBC o PostgreSQL?

Primero hay que elegir el sistema gestor de bases de datos que se va a utilizar. Las razones para escoger un SGBD pasan por las necesidades de eficiencia, integridad, escalabilidad o por otras razones técnicas u administrativas.

ODBC es una capa de abstracción entre el SGBD y la aplicación que permite utilizar cualquier SGBD por debajo, pero tener una capa intermedia tiene un coste tanto en el rendimiento como en las capacidades. Si vas a utilizar una BBDD PostgreSQL es mejor utilizar el conector directo que no pasa a través de ODBC, al igual que el de MySQL si se ha instalado.

Explicaré cómo configurar una conexión con una base de datos mediante ODBC, para una conexión directa basta configurar el fichero `res_{pgsql,mysql}.conf` o `cdr_{pgsql,mysql}.conf` y especificar los datos de la conexión a la BBDD (IP, puerto, usuario, contraseña, tabla utilizada) amen, por supuesto, de haber configurado y creado las tablas necesarias.

▪ Instalación y configuración de ODBC para *Asterisk*

Lo primero es compilar *Asterisk* con soporte para ODBC. En la versión de paquetería suele estar incluida, además debemos instalar unixODBC, que es el gestor de controladores para cada SGBD en entornos Unix o GNU/Linux.

los pasos que se siguen aquí son para una Debian Lenny o Ubuntu 8.04. En concreto configuraremos para utilizar una BBDD PostgreSQL. Para utilizar otro SGBD sólo hay que cambiar el conector, los pasos son los mismos.

1. Crea la base de datos que vayas a utilizar, define los usuarios y establece permisos.
2. Crea las tablas que se vayan a utilizar. Hay que crear una tabla para los siguientes supuestos:
 - Se va a almacenar CDR en BBDD, hay que crear una tabla con el esquema adecuado para el SGBD que utilicemos, hay que consultar la documentación de *Asterisk*, [Digc].
 - Se va a almacenar buzones de voz en BBDD, hay que crear una tabla con el esquema adecuado, veasé en la documentación oficial de *Asterisk* [Digb].
 - Se va a configurar *Asterisk* mediante BBDD.

²¹ *Open Data Base Connectivity*, es una estándar de acceso a base de datos que permite abstraerse del sistema de gestión de base de datos utilizado. Podemos utilizar para nuestra aplicación ODBC y luego enlazar una conexión ODBC a un motor en concreto.

2 Instalación y configuración de Asterisk

3. Edita el fichero `/etc/odbcinst.ini` para especificar los conectores disponibles.

```
#nombre del conector para el SGBD
[postgres]
Description = Conector para Postgresql de \emph{Asterisk}
#dependiente del sistema instalado
Driver = /usr/lib/odbc/psqlodbca.so
FileUsage = 1
```

4. Edita el fichero `/etc/odbc.ini` para especificar las fuentes de datos, puede haber mas de una para cada conector.

```
[pg-asterisk]
Description = Conexion de postgresQL a Asterisk, base de datos única
#definido en odbcinst.ini
Driver = postgres
#datos de la conexion
Database = asterisk
Servername = localhost
Username = asterisk
Password = password
Port = 5432
```

5. Si vas a almacenar los datos CDR mediante ODBC, modifica el archivo `cdr.conf` y habilita el registro mediante ODBC. Por defecto esta activado el registro en formato CSV²², se pueden definir varios métodos para almacenar CDR.

```
[csv]
usegmttime=yes ; almacena fecha y hora en GMT. por defecto es "no"
loguniqueid=yes ; almacena identificador unico de llamada.
loguserfield=yes ; almacena el registro de usuario.

;almacena mediante ODBC
;requiere que este configurado cdr_odbc.conf

[odbc]
loguniqueid=yes
loguserfield=yes
usegmttime=yes
```

Los ficheros `cdr_odbc.conf` y `cdr_pgsqldb.conf`, son muy parecidos , sólo se ha de especificar el conector ODBC configurado previamente y los datos de la conexión.

²² *comma-separated values*, fichero de valores separados por comas.

```
;  
; cdr_odbc.conf  
;  
  
[global]  
dsn=pg-asterisk  
username=asterisk  
password=password  
loguniqueid=yes  
dispositionstring=yes  
table=cdr           ;nombre de la tabla definida  
;usegmttime=no     ; ¿almacenar tiempo en GMT?
```

2 *Instalación y configuración de Asterisk*

3 Herramientas de análisis y monitorización de redes

3.1. tcpdump

tcpdump es un *sniffer* de red, un monitor que se encarga de capturar el tráfico de red y mostrárselo al usuario de manera apropiada. *tcpdump* filtra en base a reglas de la capa 3 y 4 del modelo OSI, si bien interpreta y decodifica algunos protocolos de capas superiores como SMB/CIFS.

En el presente mini tutorial, se utilizará *tcpdump* como analizador de paquetes TCP/UDP sobre IP y de tramas IP para redes Ethernet. Otros protocolos y redes son soportados, pero TCP/IP sobre Ethernet es sin duda lo más extendido.

3.1.1. Utilidad de tcpdump

Analizar la red es necesario para determinar si existen errores en la comunicación por red. La manera más sencilla que nos brinda *tcpdump* es un análisis de existencia, poder comprobar de manera directa si los paquetes llegan al destino y qué respuesta les da éste. Tras este análisis podemos determinar si el problema es físico (los paquetes no llegan, fallos de redundancia. . .) o lógico (no se sigue el protocolo de red, . . .).

El funcionamiento de *tcpdump* es simple: a través de consola se ejecuta *tcpdump -i interfaz de red | any expresión*. A partir de este momento, en tiempo real, si se recibe un paquete entrante en la red por el interfaz seleccionado (o por cualquiera si se escribe *any*) que cumpla con la expresión proporcionada, este paquete se registra y se muestra por la salida estándar. La información que nos da esta salida puede ser controlada mediante los propios parámetros del programa y según nuestros intereses.

3.1.2. Creando expresiones, filtros BPF

tcpdump utiliza para construir las expresiones filtros BPF¹. Los filtros BPF son un lenguaje regular que nos permite denotar que máquinas, puertos, tipos de paquetes y protocolos queremos escuchar. Una *expresión* esta formada por una o varias *primitivas* unidas por los operadores lógicos *not and or* y pudiendo ser agrupadas mediante '(' y ')'². Una *primitiva* suele consistir en un dato (nombre o número) precedido por uno o más *cualificadores*.

Los cualificadores pueden clasificarse según tres tipos.

¹Berkeley Packet Filter

²Para poder utilizarlos en Bash hay que escapar los paréntesis con \, ya que para Bash los paréntesis también tienen significado

▪ Cualificadores de tipo

host especifica un equipo. Por ejemplo *host google.es* , *host 10.1.1.3*

net especifica una red. Pueden omitirse los ceros de red *net 192.168.0.0* es igual que *net 192.168.0*; *net 172.26.0.0* es igual que *172.26* .La mascara de subred viene determinada por el número de puntos:

<i>Numero de puntos</i>	<i>mascara de subred</i>
4	255.255.255.255 (es un equipo no una red)
3	255.255.255.0
2	255.255.0.0
1	255.0.0.0

Figura 3.1: Número de puntos de una dirección y subred asociada

También puede especificarse la máscara en concreto con el cualificador **mask**. *net dirección de red mask mascara*

port especifica el puerto, *port puerto* .

portrange especifica un rango de puertos.

▪ Cualificadores de direccionalidad

src origen. *src host google.es (cuyo origen sea google.es)* .

dst destino.

src and dst mediante el operador **and** podemos formar una expresión compuesta de los cualificadores **src** y **dst**. Esto quiere decir que sea origen y destino. e.g. *src and dst host 192.168.1.1 (escuchar todos los paquetes que van destinados a un equipo de la red)* .

▪ Cualificadores de protocolo

ether red Ethernet.

tcp que utilice el protocolo TCP.

udp que utilice el protocolo UDP.

arp que utilice el protocolo ARP.

ip que sólo analice las cabeceras y muestre el protocolo IP.

- **Otros cualificadores**

less *tamaño en bytes*, concuerda si el tamaño del paquete es menor que el tamaño especificado.

greater igual que el anterior excepto que concuerda si el tamaño es mayor.

gateway *nombre de equipo*, concuerda si el paquete utiliza equipo como gateway. Hay que tener en cuenta que no se admiten direcciones IP (hay otros cualificadores para eso como `dst host IP`).

3.1.3. Formato de salida

`tcpdump` nos da cierta información cuando captura un paquete que cumple con la expresión especificada. Por lo general, la salida sigue el siguiente formato:

```
timestamp protocolo origen > destino: banderas
[numerodesecuencia ack window urgent options]
```

Donde,

- *timestamp* es la hora en que se capturó el paquete
- *protocolo* es el protocolo a nivel de red que se esta utilizando.
- *banderas* es una de las siguientes:
 - '.', si el paquete no tiene banderas activas.
 - *S,F,P,W,E*, o una combinación de las anteriores si el paquete tiene la bandera SYN(*S*), FIN(*F*), PUSH(*P*), reducción de la ventana (*W*), o E (*ECN eco* [\[Wik\]](#)).
- *urgent*, el paquete se marcó como urgente.
- *opciones*, donde se indica entre < y > las opciones incluidas.

3.1.4. Parámetros de tcpdump

uso de `tcpdump`:

```
tcpdump [ -AdDeflLnNOpqRStuUvxX ] [ -c contador ]
          [ -C tamaño_archivo ] [ -F archivo ]
          [ -i interfaz ] [ -m modulo ] [ -M contraseña ]
          [ -r archivo ] [ -s snaplen ] [ -T tipo ] [ -w archivo ]
          [ -W maximo_numero_archivos ]
          [ -E spi@ipaddr algo:secret,... ]
          [ -y tipo_enlace_datos ] [ -Z usuario ]
          [ expresion ]
```

3 Herramientas de análisis de redes

- A** Imprimir cada paquete (menos la cabecera del nivel de enlace) en ASCII. Útil para capturar paginas web.
- c** *Contador* Acaba después recibir un numero de paquetes igual a *contador*.
- C** *Tamaño de archivo*. Se utiliza junto **-w**. Antes de escribir un nuevo paquete comprueba si añadiéndolo el fichero actual sobrepasa *tamaño de archivo*, si es así, cierra el fichero donde se esta guardando y abre uno nuevo. Los nuevos ficheros abiertos tendrán el nombre que se especifico con el parámetro **-w** seguidos de un número que se autoincrementará. Las unidades de *tamaño de archivo* están en millones de bytes y no en mega-bytes.
- d** Volcar el código de concordancia de paquetes compilado en una forma entendible por humanos a la salida estándar y acaba.³
- dd** Vuelca el código de concordancia de paquetes como un fragmento de código C.
- ddd** Vuelca el código de concordancia de paquetes como números decimales.
- D** Imprimir la lista de los interfaces de red disponibles en el sistema y en los cuales *tcpdump* puede capturar paquetes. Para cada interfaz de red, se imprime un número y un nombre de interfaz seguido posiblemente por una descripción textual de la interfaz. El nombre de la interfaz o el número puede ser proporcionados al parámetro **-i** para especificar la interfaz de escucha.
- e** Imprimir en cada linea de volcado la cabecera del nivel de enlace.
- E** Usa *spi@ipaddr algo:secret* para descifrar paquetes IPsec ESP que son dirigidos a *addr* , y contienen el valor *spi*.
- f** Imprimir las direcciones IPv4 'extrañas' numéricamente en lugar de simbólicamente. Es útil para capturar paquetes de un servidor NIS de Sun.
- F** Usa *archivo* como entrada para la *expresión* de filtrado. cualquier expresión adicional en la linea de comandos sera ignorada./
- i** Escuchar en *interfaz*, Si no se especifica, *tcpdump* buscara en la lista de interfaces de red del sistema aquella configurada y con el número más bajo, excluyendo la de *loopback*. En sistemas Linux con núcleos posteriores a la versión 2.2, puede escogerse *any* como parámetro de *interfaz* para capturar paquetes de todas las interfaces de red.
- I** Crea un buffer para la salida estándar. Útil si se quiere observar los datos mientras se captura. E.g.,

```
'\emph{tcpdump} -l | tee datos ' o '\emph{tcpdump} -l > dat & tail -f c
```

³Entendible por humanos quiere decir en ensamblador, que es muy para humanos...

- L Listar los tipos de enlace de datos conocidos para la interfaz y salir.
- m Carga las definiciones de modulo SMI MIB desde el archivo *modulo*.
- M Usar *contraseña* como una contraseña compartida para validar los desafíos encontrados en los segmentos TCP con la opción TCP-MD5 (RFC 2385[Hef98]), si se encuentran.
- n No convertir las direcciones (direcciones de equipos, números de puerto, etc.) a nombres.
- N No imprimir la cualificación del nombre de dominio para nombres de equipo. E.g., si se activa este parámetros entonces *tcpdump* imprimirá 'correo' en lugar de 'correo.organizacion.com'.
- O No ejecutar el optimizador de código para concordancia de paquetes. Esto sólo es útil cuando se sospecha de un fallo en el optimizador.
- p No poner la interfaz en modo promiscuo.⁴
- q Salida rápida. Imprime menos información que hace las líneas de salida sean mas cortas.
- R Asumir que los paquetes ESP/AH se basan en la especificación antigua (RFC1825 [Atk95c][Atk95a][Atk95b][MS95][KMS95]).
- r Leer los paquetes de *archivo* , donde *archivo* es de una captura anterior realizada con el parámetro *-w*. Se utiliza la entrada estándar si *archivo* es '-'.
 -S imprime los números absolutos, en lugar de los relativos, de los números de secuencia TCP.
- s observa *snaplen* bytes de datos por cada paquete en lugar de los 68 por defecto. 68 bytes es adecuado para IP, ICMP, TCP y UDP pero puede truncar información de protocolos como el DNS y paquetes NFS. Los paquetes truncados debido a una instantánea limitada son indicados en la salida con [proto], donde *proto* es el nombre del protocolo que se ha truncado. Nótese que tomar instantáneas mayores produce que se incremente la cantidad de tiempo necesaria para procesar paquetes y que además disminuya la cantidad de paquetes que se pueden guardar en el buffer. Esto puede causar perdida de paquetes. Se debe limitar *snaplen* al número de bytes menor que captura la información necesaria del protocolo en el que se esta interesado. Establecer *snaplen* a 0 significa que se capturara todo el paquete.
- T forzar que los paquetes concordantes con la *expresión* se interpreten como de un *tipo especificado*. Tipos conocidos actualmente son **aadv**, **cnfp**, **rpc**, **rtp**, **rtcp**, **snmp**, **tftp**, **vat** y **wb**.

⁴El modo promiscuo o modo monitor, hace que una interfaz dada acepte todo el tráfico que pase por ella y no sólo el trafico del cual es destinataria.

3 Herramientas de análisis de redes

- t** no imprimir la *marca de tiempo* en cada línea del volcado.
- tt** imprimir una marca de tiempo sin formato en cada línea.
- ttt** imprimir un delta (en micro segundos) entre la línea actual y la anterior en cada línea del volcado.
- tttt** imprimir una marca de tiempo en el formato por defecto producido por `date`.
- u** imprimir los manejadores NFS sin decodificar.
- U** hacer que el archivo de salida seleccionado mediante `-w` tenga un buffer de paquetes; i.e. , cada paquete que se almacene, se escribe en el fichero de salida, en lugar de esto se escribe sólo cuando el buffer de salida se llene.
- v** modo de salida detallada.
- vv** más detallada aún.
- vvv** incluso más salida detallada.
- w** escribe los paquetes en un *fichero* en lugar de comprobarlos y mostrarlos en la salida. Este fichero puede ser mostrado con la opción `-r`. la salida estándar se usa si *fichero* es `'.'`.
- W** usada en conjunción con la opción `-C`, esto limita el numero de archivos creados al *número* especificado, y comienza a sobrescribir archivos desde el principio, de manera que crea un buffer 'rotatorio'. Además, nombrara los archivos con 0 en el encabezado hasta llegar al numero máximo de archivos, permitiendo que estén ordenados de manera correcta.
- x** cuando se analice y se imprime, además de imprimirá las cabeceras de cada paquete, imprime los datos de cada paquete (menos la cabecera del nivel de enlace) en hexadecimal. La porción más pequeña del paquete o *snaplen* bytes serán mostrados. Notar que esto es para el paquete entero de la capa de enlace, así que para las capas de enlace que rellenen (e.g. Ethernet) los bytes de relleno serán imprimidos cuando el paquete de capa superior sea mas corto que el relleno requerido.
- xx** cuando se analice e imprime, además de imprimir las cabeceras de cada paquete , imprime los datos de cada paquete, incluyendo la cabecera del nivel de enlace, en hexadecimal.
- X** cuando se analice e imprima, además de imprimir las cabeceras de cada paquete, imprimir los datos de cada paquete (menos la cabecera del nivel de enlace) en hexadecimal y ASCII. Esto es muy útil para analizar nuevos protocolos.
- XX** igual que el anterior, pero incluyendo la cabecera de nivel de enlace.
- y** establece el tipo de datos de enlace a usar mientras se capturan paquetes a *tipo_-enlace_datos*, puede ser uno de los mostrados mediante la opción `-L`.

-Z quita privilegios (si es root) y cambia el id de usuario a *usuario* y el id de grupo al grupo primario de *usuario*.⁵

3.2. *ngrep*

3.2.1. descripción

ngrep es un *sniffer* o analizador de red como *tcpdump*, igual que este último admite filtros BPF y las *expresiones* que acepta siguen el formato del anterior. La diferencia fundamental con respecto a *tcpdump*, es que *ngrep* permite añadir un nuevo *cualificador*, la concordancia con un patrón en el formato habitual de *grep*.

En general el uso de *ngrep* es el siguiente:

```
ngrep <-hNXviwqpevx1DtTRM> <-IO pcap_dump > < -n num > < -d dev >
< -A num > < -s snaplen > < -S limitlen > < -W normal|byline|single|none >
< -c cols > < -P char > < -F archivo > < patron de busqueda >
< filtro bpf >
```

Donde filtro BPF son los mismos filtros expuestos para *tcpdump* en 3.1.2 en la página 41, y el patrón de búsqueda es una expresión regular (Veasé [GNUb]), que se intentará concordar con los datos (no en las cabeceras) del paquete. Esto nos permite afinar mucho más nuestra búsqueda de paquetes. En el caso de SIP podemos quedarnos con los paquetes que hayan pasado por el proxy proxy1.com basta filtrar por 'proxy1.com' o mejor por 'Via: proxy1.com' y obtendremos los resultados esperados.

parámetros de *ngrep*

- h** Muestra información de ayuda y uso.
- N** Muestra numero de sub-protocolo con un identificador de un carácter.
- X** Trata el *patrón de búsqueda* como una cadena hexadecimal.
- V** Muestra la información de la versión.
- i** Ignora la capitalización de la expresión regular.
- w** Tratar el *patrón de búsqueda* como una palabra.
- q** Modo silencioso; sólo se imprimen las cabeceras de los paquetes y su carga (si es relevante).
- p** No activar el modo promiscuo para la interfaz.
- e** Mostrar paquetes vacíos. Normalmente los paquetes vacíos son descartados debido a que no hay datos donde buscar. Si se especifica, los paquetes vacíos serán mostrado, sin tener en cuenta la expresión regular.

⁵En los sistemas Linux, sólo el usuario root tiene acceso a las tablas de ruteo del kernel. Por lo tanto un usuario no root sólo podrá analizar ficheros capturados anteriormente y salvados, no datos nuevos.

3 Herramientas de análisis de redes

- v Invierte el patrón; sólo muestra los paquetes que no concuerdan.
- x Volcar el contenido de los paquetes en hexadecimal en lugar de ASCII.
- l Crear un buffer para la salida estándar.
- D Cuando se leen archivos de volcado de pcap (*pcap_dump*), se muestra con el intervalo de tiempo grabado (simulación de tiempo real).
- t Imprimir una marca de tiempo con el formato YYYY/MM/DD HH:MM:SS.UUUUUU cada vez que un paquete concuerda.
- T Imprimir una marca de tiempo con el formato +S.UUUUUU, indicando el 'delta' entre paquetes.
- c *cols* Establecer el ancho de la salida de la consola a *cols*. Notar que esto es el ancho de la consola, y no el ancho total de lo que *ngrep* imprime como datos; dependiendo del modo de salida *ngrep* puede imprimir menos que *cols* bytes por línea (indentación).
- F *archivo* Lee el filtro BPF de *archivo*. Es una opción familiar para usuarios de *tcpdump*. Notar que especificar -F hará que se ignore cualquier filtro BPF escrito en la línea de comandos.
- P *char* Especificar un caracter alternativo para representar caracteres no imprimibles. Por defecto es '.'.
- W *normal—byline—single—none* Especificar una manera alternativa de mostrar paquetes, cuando no están en modo hexadecimal. El modo *byline*, corta el texto cuando se encuentra un salto de línea. El modo *none* no corta el texto en ninguna circunstancia (los datos del paquete se muestran en una línea). El modo *single* es similar al modo *none*, excepto que IP y la cabecera de información del origen y destino se encuentra en una línea. *normal* es el modo por defecto y está incluido sólo por completitud. La opción -W es incompatible con -x.
- s *snaplen* Establece la captura de cabeceras para los filtros bpf en *snaplen* bytes.
- S *limitlen* Establecer el límite superior del tamaño de un paquete en el que *ngrep* buscará. Útil para buscar en los primeros N bytes de los paquetes sin cambiar el tamaño de captura.
- l *pcap_dump* Leer archivo de volcado pcap en *ngrep*. Funciona con cualquier formato de volcado compatible con pcap. Esta opción es útil para buscar un amplio rango de patrones diferentes en el mismo flujo de datos.
- O *pcap_dump* Escribir los paquetes concordantes a un fichero de volcado compatible pcap. Este parámetro no interfiere con la impresión normal a salida estándar.
- n *num* captura *num* paquetes que concuerden con el patrón, entonces acaba el programa.

- d dev** Por defecto ngrep seleccionará el interfaz por defecto donde escuchará. Usa esta opción para forzar a ngrep a escuchar en el interfaz *dev*.
- A num** Volcar *num* paquetes que sigan (aunque no concuerden) a un paquete concordante.
- K num** Acabar con las conexiones TCP concordantes (como *tcpkill*). *num* es el numero de segmentos RST que se enviarán.

patron de busqueda un patron de busqueda es una expresion regular extendida, o si la opción **-X** se especifica, una cadena que representa un valor hexadecimal. Una expresion regular extendida sigue las reglas de la implementación de la **GNU regex library**. Las expresiones hexadecimales pueden ser precedidas opcionalmente por '0x'. E.g., 'DEADBEEF', '0xDEADBEEF'.

expresion una *expresión* es un filtro BPF creado con las reglas explicadas en 3.1.2 en la página 41.

3.3. Wireshark

Wireshark es otro *sniffer* o analizador de redes, a diferencia de *tcpdump* y *ngrep* que se ejecutan en consola, *Wireshark* tiene una interfaz de usuario escrita en GTK. Al igual que *tcpdump* y *ngrep* admite filtros BPF, y es capaz de decodificar algunos de los protocolos conocidos.

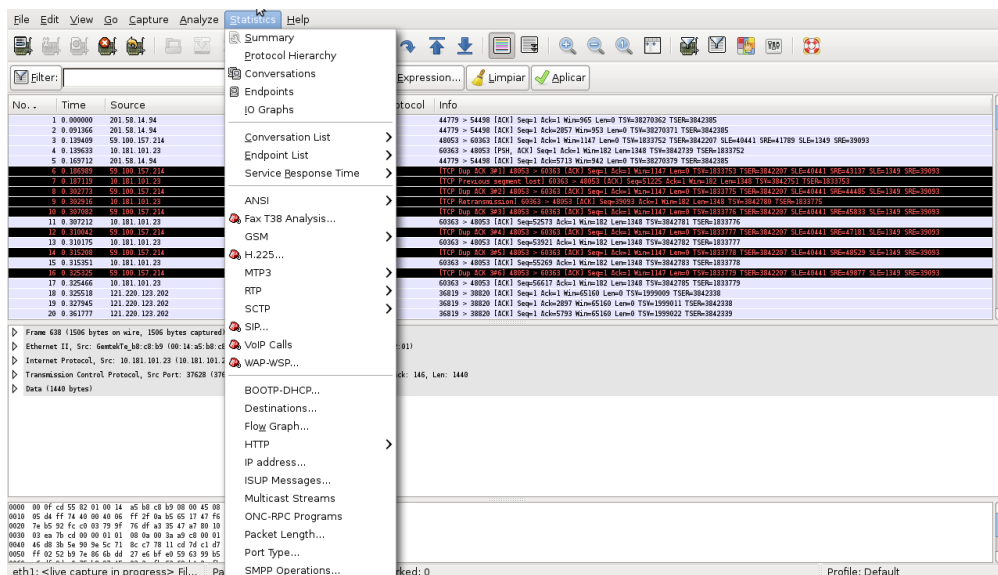


Figura 3.2: Captura de *Wireshark*

3 Herramientas de análisis de redes

Además de los filtros BPF, *Wireshark* tiene algunos filtros propios. Si se especifica el nombre de un protocolo (por ejemplo HTTP) sólo se mostrarán los paquetes HTTP de la captura. Existe una versión de *Wireshark* sin interfaz gráfica denominada *Tshark*.

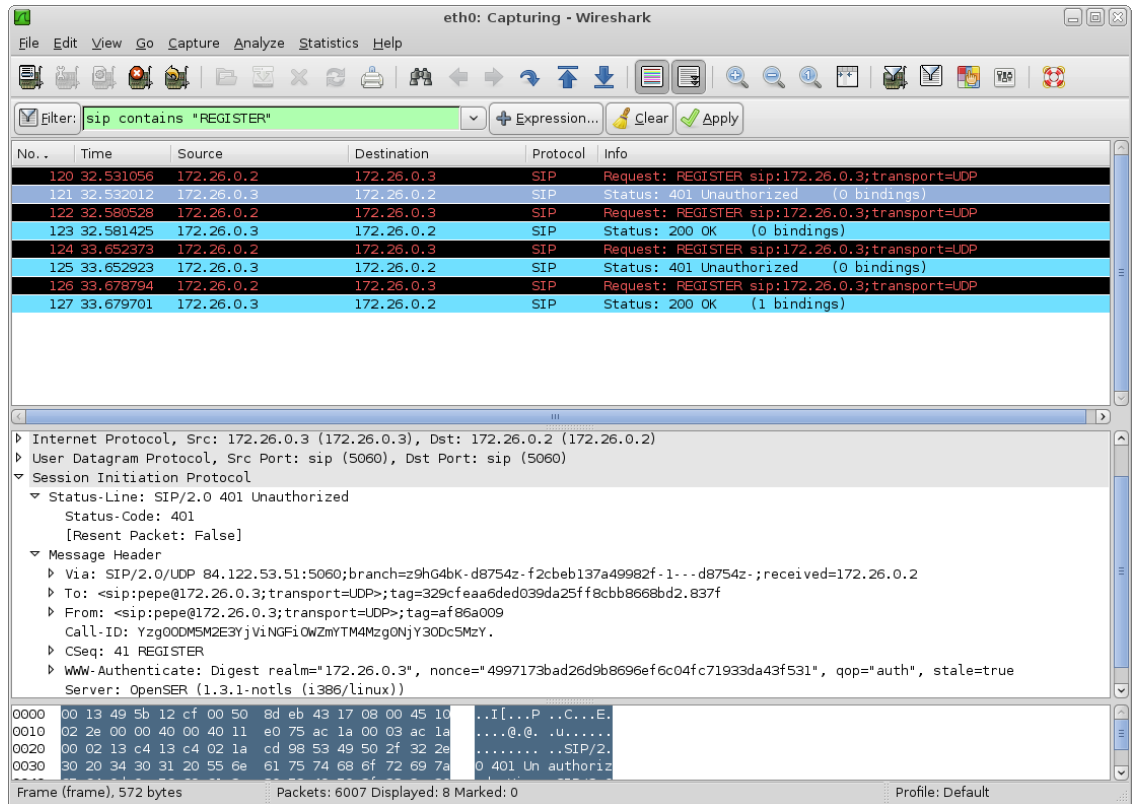


Figura 3.3: *Wireshark* en acción

4 GNU Free Documentation License

Version 1.2, November 2002

Copyright ©2000,2001,2002 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below,

such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute

an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

4 *GNU Free Documentation License*

Bibliografía

- [Atk95a] R. Atkinson. IP Authentication Header. RFC 1826 (Proposed Standard), August 1995. URL <http://www.ietf.org/rfc/rfc1826.txt>. Obsoleted by RFC 2402. 45
- [Atk95b] R. Atkinson. IP Encapsulating Security Payload (ESP). RFC 1827 (Proposed Standard), August 1995. URL <http://www.ietf.org/rfc/rfc1827.txt>. Obsoleted by RFC 2406. 45
- [Atk95c] R. Atkinson. Security Architecture for the Internet Protocol. RFC 1825 (Proposed Standard), August 1995. URL <http://www.ietf.org/rfc/rfc1825.txt>. Obsoleted by RFC 2401. 45
- [Diga] Digium. Cambios de la versión de Asterisk 1.4. <http://downloads.digium.com/pub/asterisk/ChangeLog-1.4-current>. 13
- [Digb] Digium. Documentación con el esquema para la creación de la tabla para la BBDD para almacenar buzones de voz . <http://svn.digium.com/view/asterisk/branches/1.4/doc/cdrdriver.txt?view=markup>. 37
- [Digc] Digium. Documentación con el esquema para la creación de la tabla para la BBDD para su uso en CDR. <http://svn.digium.com/view/asterisk/branches/1.4/doc/cdrdriver.txt?view=markup>. 37
- [Digd] Digium. Paquete de addons estable de Asterisk 1.4. <http://downloads.digium.com/pub/asterisk/asterisk-addons-1.4-current.tar.gz>. 13, 23, 37
- [Dige] Digium. Paquete de fuentes estable de Asterisk 1.4. <http://downloads.digium.com/pub/asterisk/asterisk-1.4-current.tar.gz>. 13
- [Fes] Proyecto Festival. Página principal del proyecto Festival. <http://www.cstr.ed.ac.uk/projects/festival/>. 24
- [Gar] Gerardo Aburrizaga García. Manual libre de Make en PDF, cortesía de Gerardo Aburrizaga (profesor de la Universidad de Cádiz), osluca. <http://www.uca.es/softwarelibre/publicaciones>. 14
- [GNUa] Proyecto GNU. Manual de GNU Make. <http://www.gnu.org/software/make/>. 14

Bibliografía

- [GNUb] Proyecto GNU. Página de man de grep, traducida por TDLP-ES. <http://es.tldp.org/Paginas-manual/man-pages-es-extra-0.8a/man1/grep.1.html>. 47
- [Gon08] Flavio E. Goncalves. *Building Telephony Systems with OpenSER: A step-by-step guide to building a high performance Telephony System*. Packt Publishing, 2008. ISBN 1847193730.
- [GVE00] A. Gulbrandsen, P. Vixie, and L. Esibov. A DNS RR for specifying the location of services (DNS SRV). RFC 2782 (Proposed Standard), February 2000. URL <http://www.ietf.org/rfc/rfc2782.txt>. 29
- [Hef98] A. Heffernan. Protection of BGP Sessions via the TCP MD5 Signature Option. RFC 2385 (Proposed Standard), August 1998. URL <http://www.ietf.org/rfc/rfc2385.txt>. 45
- [Kama] Proyecto Kamailio. Página de documentación de Kamailio. <http://www.kamailio.org/>.
- [Kamb] Proyecto Kamailio. Página de instalación de openSER. <http://www.kamailio.org/>.
- [KMS95] P. Karn, P. Metzger, and W. Simpson. The ESP DES-CBC Transform. RFC 1829 (Proposed Standard), August 1995. URL <http://www.ietf.org/rfc/rfc1829.txt>. 45
- [MS95] P. Metzger and W. Simpson. IP Authentication using Keyed MD5. RFC 1828 (Historic), August 1995. URL <http://www.ietf.org/rfc/rfc1828.txt>. 45
- [Ray01] Eric S. Raymond. *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly Media, Inc., 2001. ISBN 0596001088. 10
- [RMK⁺96] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918 (Best Current Practice), February 1996. URL <http://www.ietf.org/rfc/rfc1918.txt>. 30
- [RSC⁺02] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002. URL <http://www.ietf.org/rfc/rfc3261.txt>. Updated by RFCs 3265, 3853, 4320, 4916. 11, 28
- [Sta04] Richard M. Stallman. *Software libre para una sociedad libre*. Proyecto Editorial Traficantes de Sueños, 2004. ISBN 8493355518. 10
- [VMSM07] Jim Van Meggelen, Jared Smith, and Leif Madsen. *Asterisk: The Future of Telephony*. O'Reilly Media, Inc., 2nd edition, 2007. ISBN 0596009623. 13

- [wdpKao] wiki del proyecto Kamailio (antiguo openSER). Wiki de Kamailio. <http://www.kamailio.org/>.
- [Wik] Wikipedia. Explicit Congestion Notification. http://en.wikipedia.org/wiki/Explicit_Congestion_Notification. 43