

Asterisk

Let freedom ring

Scopo della prova:

Realizzare una rete telefonica all'interno del laboratorio di telecomunicazione utilizzando il software PBX Asterisk.

Strumenti utilizzati:

| | |
|---|--|
| Asterisk 3.0.1 | Software usato per il progetto |
| Filezilla | Server FTP per salvare i nostri progressi |
| Solarwind | Software usato per le impostazioni tftp di Asterisk |
| Putty | Usato per lavorare in remoto sul server PBX |
| Asterisk, The Definitive Guide, 4th Edition | Manuale in cui è presente la guida per la configurazione del PBX |



Teoria Asterisk

(Storia di Asterisk + definizione di PBX)

In telecomunicazioni un private branch exchange o PBX è una centrale telefonica per uso privato. È principalmente usato nelle aziende per fornire una rete telefonica interna. Il termine PABX (Private Automatic Branch eXchange) indica una versione automatica di un PBX, ma oggi i due termini sono usati come sinonimi perché tutti i PBX in commercio sono completamente automatici.

Asterisk è una implementazione libera di un software per sistemi PBX che permette di ottenere le stesse funzionalità offerte da altri sistemi proprietari a fronte di una spesa decisamente inferiore ed una maggiore flessibilità; esso include infatti: segreteria telefonica, possibilità di creare conferenze, comandi vocali per i menù e trasferimento di chiamata automatico. Gli utenti possono creare nuove funzioni scrivendo script di **dial plan** in uno dei vari linguaggi supportati come C o implementando il suo stesso linguaggio.

Asterisk è stato uno dei primi software PBX e supporta numerosi protocolli VoIP ma quello che a noi maggiormente interessa è il **SIP** (Session Initiation Protocol), che sarà quello da noi utilizzato.

La prima stesura di Asterisk venne realizzata da Mark Spencer, un ingegnere informatico statunitense che, attorno all'anno 2000, fondò una società che sviluppa schede d'interfaccia FXS ed FXO (Un Foreign Exchange Station, in sigla FXS, è ogni dispositivo che fornisce il servizio telefonico agli apparecchi telefonici, chiamati in questo contesto Foreign Exchange Office (FXO)). Spencer sviluppò Asterisk per favorire la diffusione delle interfacce Digium e, permettendone la libera distribuzione, solleticò l'interesse di moltissimi utilizzatori professionali ed appassionati, diffondendone largamente così la conoscenza. Asterisk è oggi un punto di riferimento nel settore dei PBX: la completezza dei suoi contenuti e la sua affidabilità lo rendono infatti una piattaforma ideale per una vasta gamma di applicazioni, ad esempio come elemento portante per la realizzazione di mediatori RTG/IP (centralini telefonici in grado cioè di utilizzare sia le linee telefoniche tradizionali sia i canali IP), sistemi Centrex (PBX "virtuali" e centralizzati), applicazioni per la gestione di Call Center ed altro ancora. Il sistema operativo su cui si basa il programma è CentOS, una versione di Linux.



CentOS è una delle distribuzioni più utilizzate sui server web dell'Internet odierna. Il motivo è presto detto: è un sistema operativo estremamente robusto ed affidabile, poiché rivolto all'esigente pubblico aziendale. Installarlo su una macchina di prova è comunque molto semplice: basta il relativo DVD, un computer qualsiasi (anche virtuale) e qualche cliccata su "Avanti" all'interno della semplicissima procedura guidata grafica. CentOS è quindi una distribuzione ottimale per realizzare server aziendali e domestici di ogni tipo, dai database ai siti dinamici, dall'FTP alla condivisione dei file e delle stampanti passando per DHCP, DNS, VPN LDAP, backup e tutti gli altri servizi analoghi. Risulta discreto anche come piattaforma per la produttività da ufficio.

Cenni teorici

Con **VoIP** (Voice over IP) si intende una tecnologia che rende possibile effettuare una conversazione telefonica sfruttando una connessione Internet o una qualsiasi altra rete dedicata a commutazione di pacchetto che utilizzi il protocollo IP senza connessione per il trasporto dati.

Più specificamente si intende l'insieme dei protocolli di comunicazione di strato applicativo che rendono possibile tale tipo di comunicazione. Uno di questi è il protocollo **SIP** (**Session Initiation Protocol**) che si definisce come un protocollo di controllo del livello applicativo usato per creare, modificare, e terminare sessioni tra uno o più partecipanti. Queste sessioni includono chiamate telefoniche via Internet, distribuzioni multimediali, e conferenze multimediali. SIP viene comunemente usato per instaurare o terminare chiamate video o vocali e permette di modificare le caratteristiche di chiamate in corso come l'indirizzo e la porta, l'invito di ulteriori partecipanti e aggiungere o cancellare media stream.

Esso gestisce in modo generale una sessione di comunicazione tra due o più entità, ovvero fornisce meccanismi per instaurare, modificare e terminare (rilasciare) una sessione. Attraverso il protocollo SIP possono essere trasferiti dati di diverso tipo (audio, video, messaggistica testuale, ecc). Inoltre, il SIP favorisce un'architettura modulare e scalabile, ovvero capace di crescere con il numero degli utilizzatori del servizio. Queste potenzialità hanno fatto sì che il SIP sia, oggi, il protocollo VoIP più diffuso nel mercato residenziale e business, sorpassando di molto altri protocolli quali H.323 ed MGCP.

Caratteristiche:

1. **Localizzazione dell'utente:** determinazione degli end system usati nella comunicazione;
2. **Disponibilità dell'utente:** identificazione della disponibilità delle parti ad impegnarsi in una comunicazione;
3. **Capacità dell'utente:** identificazione di media e parametri utilizzati;
4. **Impostazione della sessione:** avviso, instaurazione dei parametri di una sessione su chiamate;
5. **Gestione della sessione:** trasferimento e terminazione di una sessione, modifica dei parametri della sessione, e invocazione dei servizi.

Dial plan

Il Dialplan è il cuore pulsante del sistema Asterisk, definisce come vengono gestite le chiamate in ingresso e in uscita dal sistema. Nel dialplan è possibile associare a ogni chiamata una serie di istruzioni da eseguire, in modo simile a un linguaggio di scripting.

Questo dial plan è contenuto nel file extensions.conf ed è suddiviso principalmente in quattro parti:

- context;
- extension;
- priorities;
- applications.

Context:

I contesti, definiti nel file extensions.conf come [nomecontesto], nel nostro caso [labtele], sono un modo di suddividere i vari insiemi di istruzioni-comportamenti in Asterisk. Asterisk sceglie il contesto in base alla configurazione dei vari utenti o linee tradizionali, ogni chiamata da una particolare utenza verrà gestita dalle istruzioni del relativo contesto.

Extension:

Le estensioni non sono nient'altro che le azioni da eseguire in risposta alla chiamata ad un determinato numero composto.

Per la loro definizione si utilizza la keyword "exten":

exten => destinazione,priorità,application(args)

A ogni destinazione possibile (o estensione) vengono associate una o più azioni, definite dalle application e dai parametri passati. La priorità serve per stabilire un ordine di esecuzione delle application, anche perché più application possono essere associate a una destinazione.

Application:

Chiama le risorse specificate e, in caso di mancata risposta dopo il timeout, prosegue l'esecuzione alla priorità successiva.

Qui sotto potete vedere come abbiamo configurato il nostro primo telefono di prova nell'`extensions_custom.conf`:

```
[labtele]
exten => 344,1,Dial(SIP/telpc4,5)
same => n, Playback(pbx-transfer)
same => n, Wait(1)
same => n, Playback(transfer)
same => n, Wait(1)
same => n, Dial(SIP/telpc6,5)
same => n,Answer()
same => n, Dial(SIP/telpc7,5)
same => n,Answer()
same => n, Playback(tt-allbusy)
same => n, Wait(1)
same => n, Playback(vm-savemessage)
same => n, Wait(1)
same => n,VoiceMail(344@lab-tele)
same => n,Hangup
```

I comandi da noi usati sono i seguenti:

* `Dial(SIP/telefono da chiamare)`: cerca di stabilire una nuova connessione al canale e di collegarla a un canale esistente.

* `Answer()`: questo comando permette di rispondere alla chiamata sulla linea se il telefono sta squillando. se il telefono non squilla la funzione non ha effetto.

* `Hangup()`: funzione che chiude la connessione a un canale.

* `Playback(nome file)`: esegue un file audio o video (è necessario omettere l'estensione del file) questa funzione è soggetta alla lingua (funzione `Language` nel sip) cioè il file eseguito cambia in base alla lingua selezionata.

* `VoiceMail(numero della segreteria a cui mandare il messaggio@rete)`: funzione che permette di mandare un messaggio alla segreteria telefonica(unico implementato da noi per ora), richiedere la password, e-mail di notifica del messaggio in segreteria.

* Wait(numero di secondi): la funzione wait può essere utilizzata per definire il tempo di utile per rispondere alla chiamata.

Così abbiamo poi configurato il file sip_custom.conf (con esempio del primo telefono) :

```
[general]

allowguest=no ; determina se chiamanti anonimi sono autorizzati
                ; a effettuare chiamate ad Asterisk.

videosupport= yes ;abilita la web-cam

srvlookup=yes ;controlla lo status del PBX

udpbinaddr=0.0.0.0

tcpenable=no ; esegue Asterisk su TCP invece che su UDP

language=it ; usa la lingua italiana

[office-phone](!)

type=friend

context=labtele ;contesto usato in extensions_custom.conf

host=dynamic

secret=!mAjo987! ;password per autenticazione da Soft-phone

allow=all

;+++++
;Telefoni 3CX (softphone utilizzato)
;+++++

[telpc1](office-phone)

mailbox=*341 ;numero da chiamare per ascoltare la propria segreteria
```

E così invece il voicemail.conf:

```
[lab-tele]

341 => !mAjo987!,office-phone,office-phone@lab.com
;la configurazione per la segreteria del primo telefono virtuale
;numero_telefono => secret, identificativo@rete
```

Conclusioni

Gli obiettivi che il docente ci ha chiesto di raggiungere sono

1. Effettuare una chiamata vocale
2. Effettuare una videochiamata
3. Impostare la segreteria telefonica
4. Impostare la lingua italiana
5. Implementare il trasferimento di chiamata
6. Implementare un sistema di registrazione dei messaggi.

I primi due punti sono stati abbastanza facili da configurare in quanto non abbiamo dovuto fare altro che seguire delle indicazioni scritte sulla guida per configurare i primi due telefoni e aggiungere una voce nel file sip per consentire la chiamata video.

Con il terzo punto della lista si sono presentati vari problemi per la configurazione del file voicemail e per comprendere appieno la sintassi della configurazione. Con l'aiuto del docente siamo riusciti a risalire al nostro errore, il numero da chiamare per ascoltare la propria segreteria coincideva con il numero per la chiamata e quindi, per un motivo di priorità di Asterisk, la chiamata alla segreteria veniva ignorata anche se partiva dal terminale configurato con quel numero.

Per la quarta consegna siamo stati ancora aiutati dal professore in quanto ha inserito nella directory `var/lib/asterisk/sounds/` una cartella chiamata `it` con tutti i file audio in lingua italiana; per abilitare poi questa è stato sufficiente l'aggiunta di un'istruzione nel file `sip_custom.conf`.

Per implementare il trasferimento di chiamata non era più necessario usare il manuale poiché siamo stati costretti a lavorare di logica; essendo il linguaggio di Asterisk composto di istruzioni sequenziali, cioè che avvengono una dopo l'altra, non abbiamo dovuto usare istruzioni di controllo (`if`) ma è bastato predisporre un altro comando `Dial()` dopo l'`Answer()` del precedente, quindi in caso di mancata risposta dell'utente precedente, la chiamata sarebbe stata poi reindirizzata al numero successivo. Dopo aver capito bene il funzionamento dell'inoltro abbiamo deciso di provare ad aggiungere anche dei messaggi preregistrati che informano l'utenza del processo che sta affrontando, quindi per esempio la mancata risposta di un utente o, in caso che tutti i numeri siano occupati, l'avvio della segreteria telefonica.

Il sesto punto sarà invece il nostro prossimo obiettivo che speriamo vivamente di portare a termine per inserirlo poi in una futura relazione insieme ad altri traguardi come per esempio l'utilizzo della funzione TTS (Text To Speech) o altre funzioni a noi ancora ignote.

Dato che abbiamo appreso le funzioni basilari di Asterisk il docente ora ci ha affidato un altro incarico: estendere la rete creata con Asterisk a tutti i computer del laboratorio per metterli tutti in comunicazione. Questo non sarà certamente un lavoro facile ma credo che alla fine potrebbe risultare altamente gratificante in quanto, nel nostro piccolo, pur essendo partiti da zero, saremmo riusciti in pochi mesi a creare una piccola rete privata estesa a tutti i membri del nostro laboratorio. Abbiamo già iniziato a creare una

tabella in cui sono presenti tutti i computer con varie informazioni:
numero PC, indirizzo IP, extension 3CX ed extensions Xlite(altra soft-phone).

| Nome PC | Indirizzo IP | Interno (Ext 3CX) | Interno (Ext Xlite) |
|----------------|---------------------|--------------------------|----------------------------|
| 1 | 10.0.0.41 | 341 | 441 |
| 2 | 10.0.0.42 | 342 | 442 |
| 3 | 10.0.0.43 | 343 | 443 |
| 4 | 10.0.0.44 | 344 | 444 |
| 5 | 10.0.0.45 | 345 | 445 |
| 6 | 10.0.0.46 | 346 | 446 |
| 7 | 10.0.0.47 | 347 | 447 |
| 8 | 10.0.0.48 | 348 | 448 |
| 9 | 10.0.0.49 | 349 | 449 |
| 10 | 10.0.0.50 | 350 | 450 |
| 11 | 10.0.0.51 | 351 | 451 |
| 12 | 10.0.0.52 | 352 | 452 |

Gruppo:
Mattia Marta,
Simone Gentile,
Luciano gavoni.