
Installation serveur asterisk (VOIP)

IP du serveur : 192.168.1.48

Je commence par installer les paquets nécessaires :

```
apt-get install asterisk dahdi libpri1.4
```

Pour configurer notre serveur Asterisk nous allons modifier les fichiers suivants:

Le fichier sip.conf : pour la configuration général d'Asterisk

Le fichier users.conf : pour la configuration des utilisateurs

Le fichier extensions.conf : pour la configuration du Dialplan

Je modifie le fichier « user.conf » en rajoutant 2 utilisateurs

```
[6000]
type=friend
host=dynamic
allow=ulaw
fullname= Sadek Adel
username=asadek
secret=test
context=work
[6001]
type=friend
host=dynamic
allow=ulaw
fullname=TEST
username=test
secret=test
context=work
```

Pour vérifier la création de nos utilisateurs il faut redémarrer le serveur et lancer dans le terminal cette commande « asterisk -c » pour avoir accès à l'invite de commande d'asterisk et interagir directement avec le serveur

En tapant la commande « show sip users » je peux vérifier si mes utilisateurs ont correctement été créés

```
kali*CLI> sip show users
Username          Secret          Accountcode     Def.Context     ACL  Forcerport
6001              test            work            work            No   No
6000              test            work            work            No   No
kali*CLI> █
```

Ensuite je vais modifier le fichier « extensions.conf » et définir le contexte « work » c'est comme une boîte où il y a nos utilisateurs et nos utilisateurs peuvent communiquer entre eux dedans et on peut définir différentes règles.

```
[work]
exten => _6XXX,1,Dial(SIP/${EXTEN},20)
exten => _6XXX,2,Hangup()
█
```

Dans ces trois dernières lignes nous allons voir deux choses, les contextes et les extensions.

[work] est le contexte c'est une sorte de conteneur dans lequel les utilisateurs faisant partie de ce contexte pourront communiquer entre eux.

Lors de la création de nos deux utilisateurs nous avons spécifié le contexte work.

exten => : déclare l'extension (on peut aussi simplement dire numéros)

_6XXX : Prend les extensions (ou numéros) de 6000 à 6999 le « _ » permet d'utiliser des regex

1 : Ordre de l'extension

Dial : application qui va être utilisée

SIP: Protocole qui va être utilisé

`\${EXTEN}` : variable de l'extension composée, si on appelle le 6001 la variable `\${EXTEN}` prendra comme valeur 6001

20: temps d'attente avant de passer à l'étape suivante.

Donc la ligne exten => _6XXX,1,Dial(SIP/\${EXTEN},20) se traduit par:

Quand on compose le numéro (par exemple) 6001, on appelle le numéro 6001 et si au bout de 20 secondes il n'y a pas de réponses on passe à la ligne du dessous.

Dans le cas du numéros 6001 la ligne devient comme ceci: exten => 6001,1,Dial(SIP/6001,20), mais l'avantage de la ligne précédente est qu'elle permet d'appeler les numéros de 6000 a 6999.

La seconde ligne : exten => _6XXX,2,Hangup() permet de raccrocher si il n'y a pas de réponses au bout des 20 secondes.

Maintenant, vous pouvez enregistrer votre fichier extensions.conf et faire un reload dans la console d'Asterisk.

Ensuite avant de me connecter depuis mon téléphone j'exécute cette commande pour avoir tout les logs :

Asterisk -cvvvvvvvvvvr

Je me connecte avant mon téléphone je met simplement l'IP de mon serveur VOIP l'identifiant et le mot de passe

Je valide et voila ce qui m'est affiché dans l'invite de commande

```
m 15-07-03:47] WARNING[7338]: db.c:348 ast_db_p  
-- Registered SIP '6001' at 192.168.1.5:60698  
m 15-07-03:48] WARNING[7228]: db.c:348 ast_db_p
```

Une connexion a bien été effectuée

Je me connecte avec un second téléphone

```
-- Registered SIP '6000' at 192.168.1.2:50168  
> Saved useragent "Zoiper v2.10.17.3" for peer 6000
```

Je vais tenter de lancer un appel du numéro 6000 au 6001

Un appel a été lancer

```
Connected to Asterisk 18.11.2-dfsg+~cs6.10.40431413-1 currently running on kali (pid = 3104)
= Using SIP RTP CoS mark 5
> 0x7f96580327f0 -- Strict RTP learning after remote address set to: 192.168.1.2:65435
-- Executing [6001@work:1] Dial("SIP/6000-00000004", "SIP/6001,20") in new stack
= Using SIP RTP CoS mark 5
-- Called SIP/6001
> 0x7f964c010020 -- Strict RTP learning after remote address set to: 192.168.1.5:55934
-- SIP/6001-00000005 answered SIP/6000-00000004
-- Channel SIP/6001-00000005 joined 'simple_bridge' basic-bridge <0e1bf382-9902-4481-9f56-cfd7fc1898c2>
-- Channel SIP/6000-00000004 joined 'simple_bridge' basic-bridge <0e1bf382-9902-4481-9f56-cfd7fc1898c2>
> Bridge 0e1bf382-9902-4481-9f56-cfd7fc1898c2: switching from simple_bridge technology to native_rtp
> Remotely bridged 'SIP/6000-00000004' and 'SIP/6001-00000005' - media will flow directly between them
> 0x7f964c010020 -- Strict RTP learning after remote address set to: 192.168.1.5:55934
> 0x7f96580327f0 -- Strict RTP switching to RTP target address 192.168.1.2:65435 as source
```

Cela fonctionne parfaitement

