

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from the bar, containing the date.

27/11/2017

TOIP avec Asterisk

Version 1.0 : Version Initiale

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

Fabien MAUHOURLAT
[NOM DE LA SOCIETE]

Gestion de la Téléphonie sur IP avec Asterisk

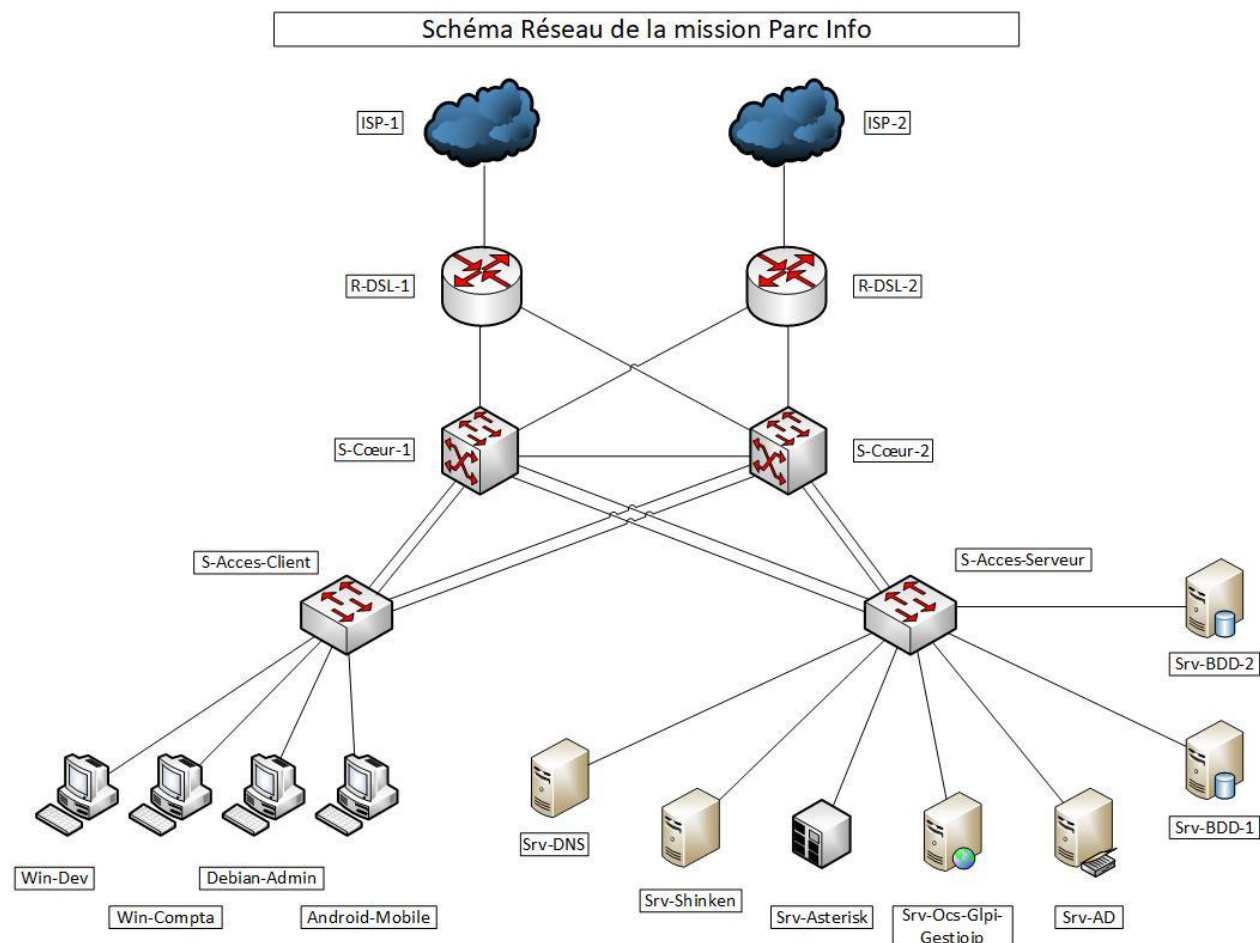
Contexte :

Pour permettre une communication à moindre frais dans l'hôtel Bora-Bora l'utilisation de la téléphonie sur IP a été choisie. Le serveur de téléphonie sera hébergé par une machine linux sous Debian 9. Le logiciel Asterisk sera utilisé dans le cadre de cette documentation.

Les nombreux client avec des systèmes d'exploitation différents amène la configuration de softphone sur Windows et Linux.

La confidentialité des échanges devra notamment être intégré grâce à la couche de sécurisation TLS.

Voici l'architecture mise en place :



Sommaire

- I. Présentation et choix des solutions
- II. Prérequis
- III. Installation d'Asterisk
- IV. Configuration d'Asterisk
- V. Configuration du client blink pour Windows et ekiga pour Linux
- VI. Mise en place d'une écoute clandestine des communications avec Wireshark
- VII. Mise en place du SIP over TLS et du SRTP
- VIII. Annexes

I. Présentation et choix des solutions

II. Prérequis

1. Système d'exploitation :

Le système d'exploitation utilisé dans cette note technique est la version 9.2 de Debian (Stretch) et la version du noyau utilisé est le 4.9.0.

```
fabien@debian-template:/etc/shinken/brokers$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 9.2 (stretch)
Release:       9.2
Codename:      stretch
fabien@debian-template:/etc/shinken/brokers$ uname -a
Linux debian-template 4.9.0-3-amd64 #1 SMP Debian 4.9.30-2+deb9u5 (2017-09-19) x86_64 GNU/Linux
fabien@debian-template:/etc/shinken/brokers$
```

III. Installation d'Asterisk

- Pour bénéficier de la dernière version du logiciel asterisk il est préférable de l'installer à partir des sources.
- Mise à jour de la liste des paquets
 - `apt-get update && apt-get upgrade`
- Installation des prérequis pour permettre la compilation du logiciel et de ses dépendances comme les headers la commande make, le compilateur gcc et bien d'autres :
 - `apt-get install build-essential libxml2-dev libncurses5-dev linux-headers-`uname -r` libsqlite3-dev libssl-dev aptitude-common libboost-filesystem1.62.0 libboost-iostreams1.62.0 libboost-system1.62.0 libcgi-fast-perl libcgi-pm-perl libclass-accessor-perl libcwidget3v5 libfcgi-perl libio-string-perl libparse-debianchangelog-perl libsigc++-2.0-0v5 libsub-name-perl libjansson-dev uuid-dev`
- Pour installer asterisk il est nécessaire de télécharger ces trois composants :
 - DAHDI : `wget https://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/dahdi-linux-complete-current.tar.gz`
 - LIBPRI : `wget https://downloads.asterisk.org/pub/telephony/libpri/libpri-current.tar.gz`
 - Asterisk 15 : `wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-15-current.tar.gz`
- Une fois les composants téléchargés il faut les décompresser :
 - `tar -xzf asterisk-15-current.tar.gz`
 - `tar -xzf dahdi-linux-complete-current.tar.gz`
 - `tar -xzf libpri-current.tar.gz`

- Une fois les manipulations effectuées :

```
fabien@debian-template:~$ ls
asterisk-15-current.tar.gz  dahdi-linux-complete-current.tar.gz  libpri-current.tar.gz
fabien@debian-template:~$ tar -xzf asterisk-15-current.tar.gz
fabien@debian-template:~$ tar -xzf dahdi-linux-complete-current.tar.gz
fabien@debian-template:~$ tar -xzf libpri-current.tar.gz
fabien@debian-template:~$ ls
asterisk-15.0.0  asterisk-15-current.tar.gz  dahdi-linux-complete-2.11.1+2.11.1  dahdi-1
fabien@debian-template:~$ ls -al
total 35756
drwxr-xr-x  6 fabien fabien    4096 nov.   4 13:02 .
drwxr-xr-x  4 root  root     4096 oct.  22 17:36 ..
drwxr-xr-x 31 fabien fabien    4096 sept. 29 08:19 asterisk-15.0.0
-rw-r--r--  1 fabien fabien 27701379 oct.  15 14:47 asterisk-15-current.tar.gz
-rw-----  1 fabien fabien   1165 nov.   3 13:37 .bash_history
-rw-r--r--  1 fabien fabien    220 oct.  22 17:36 .bash_logout
-rw-r--r--  1 fabien fabien   3526 oct.  22 17:36 .bashrc
drwxr-xr-x  4 fabien fabien    4096 mars   2 2016 dahdi-linux-complete-2.11.1+2.11.1
-rw-r--r--  1 fabien fabien 8517162 oct.  15 14:44 dahdi-linux-complete-current.tar.gz
drwxr-xr-x  4 fabien fabien    4096 janv. 28 2017 libpri-1.6.0
-rw-r--r--  1 fabien fabien 340578 oct.  15 14:42 libpri-current.tar.gz
-rw-r--r--  1 fabien fabien    675 oct.  22 17:36 .profile
drwxr-xr-x 12 fabien fabien    4096 oct.  22 19:52 UnixAgent
-rw-----  1 fabien fabien    988 oct.  22 19:05 .viminfo
fabien@debian-template:~$
```

- Installation du premier module DAHDI :
- cd dahdi-linux-complete-2.11.1+2.11.1/
 - sudo make
 - sudo make install

```
#####
###
### DAHDI installed successfully.
### If you have not done so before, install the package
### dahdi-tools.
###
#####
```

- Une fois l'installation terminée il faut exécuter la commande :
- sudo make config

```
#####
###
### DAHDI tools installed successfully.
### If you have not done so before, install init scripts with:
###
### make config
###
#####
```



Information

Le module DAHDI est une dépendance du module Libpri.
Assurez-vous de l'avoir installé au préalable.

➤ Installation du deuxième module LIBPRI :

- cd libpri-1.6.0
- sudo make
- sudo make install

```
fabien@debian-template:~/libpri-1.6.0$ sudo make install
mkdir -p /usr/lib
mkdir -p /usr/include
install -m 644 libpri.h /usr/include
install -m 755 libpri.so.1.4 /usr/lib
#if [ -x /usr/sbin/sestatus ] && ( /usr/sbin/sestatus | grep "SELinux status:"
( cd /usr/lib ; ln -sf libpri.so.1.4 libpri.so)
install -m 644 libpri.a /usr/lib
if test $(id -u) = 0; then /sbin/ldconfig -n /usr/lib; fi
fabien@debian-template:~/libpri-1.6.0$
```

➤ Installation de l'application asterisk:

- Cd asterisk-15.0.0
- cd contrib/scripts
- ./install_prereq install

➤ Pour vérifier que la configuration requise par asterisk est bonne il faut exécuter la commande :

- Sudo ./configure

[illegible]

```
Sudo make distclean
```

- Pour sélectionner les différentes options du logiciel asterisk il faut taper la commande :
 - o Make menuselect

```
*****
Asterisk Module and Build Option Selection
*****

    Press 'h' for help.

---> Add-ons (See README-addons.txt)
    Applications
    Bridging Modules
    Call Detail Recording
    Channel Event Logging
    Channel Drivers
    Codec Translators
    Format Interpreters
    Dialplan Functions
    PBX Modules
    Resource Modules
    Test Modules
    Compiler Flags
    Voicemail Build Options
    Utilities
    AGI Samples
    Core Sound Packages
    Music On Hold File Packages
    Extras Sound Packages
```


Nous allons en profiter pour installer **les sons français pour Asterisk au format μ -law**.

Dans **Core Sound Package** nous allons cocher la case **CORE-SOUNDS-FR-ULAW** avec la touche **Espace** puis appuyez sur **Echap** pour retourner à l'écran précédent.

Puis dans **Music On Hold File Packages** cochez **MOH-OPSOUND-ULAW** (Dechochez celui en WAV), appuyez sur Echap et enfin allez dans dans **Extras Sound Packages** et cochez **EXTRA-SOUNDS-FR-ULAW**.

Enfin appuyez sur Echap et une fois à l'écran principal refaites Echap et **appuyez sur S pour sauvegarder les changements**.

```

*****
Asterisk Module and Build Option Selection
*****

Press 'h' for help.

[ ] CORE-SOUNDS-ES-G729
[ ] CORE-SOUNDS-ES-G722
[ ] CORE-SOUNDS-ES-SLN16
[ ] CORE-SOUNDS-ES-SIREN7
[ ] CORE-SOUNDS-ES-SIREN14
[ ] CORE-SOUNDS-FR-WAV
[*] CORE-SOUNDS-FR-ULAW
[ ] CORE-SOUNDS-FR-ALAW

```

```

*****
Asterisk Module and Build Option Selection
*****

Press 'h' for help.

--- Core ---
[ ] MOH-OPSOUND-WAV
[*] MOH-OPSOUND-ULAW
[ ] MOH-OPSOUND-ALAW
[ ] MOH-OPSOUND-GSM
[ ] MOH-OPSOUND-G729
[ ] MOH-OPSOUND-G722
[ ] MOH-OPSOUND-SLN16
[ ] MOH-OPSOUND-SIREN7
[ ] MOH-OPSOUND-SIREN14

```

```

*****
Asterisk Module and Build Option Selection
*****

Press 'h' for help.

--- Core ---
[ ] EXTRA-SOUNDS-EN-WAV
[ ] EXTRA-SOUNDS-EN-ULAW
[ ] EXTRA-SOUNDS-EN-ALAW
[ ] EXTRA-SOUNDS-EN-GSM
[ ] EXTRA-SOUNDS-EN-G729
[ ] EXTRA-SOUNDS-EN-G722
[ ] EXTRA-SOUNDS-EN-SLN16
[ ] EXTRA-SOUNDS-EN-SIREN7
[ ] EXTRA-SOUNDS-EN-SIREN14
[ ] EXTRA-SOUNDS-EN_GB-WAV
[ ] EXTRA-SOUNDS-EN_GB-ULAW
[ ] EXTRA-SOUNDS-EN_GB-ALAW
[ ] EXTRA-SOUNDS-EN_GB-GSM
[ ] EXTRA-SOUNDS-EN_GB-G729
[ ] EXTRA-SOUNDS-EN_GB-G722
[ ] EXTRA-SOUNDS-EN_GB-SLN16
[ ] EXTRA-SOUNDS-EN_GB-SIREN7
[ ] EXTRA-SOUNDS-EN_GB-SIREN14
[ ] EXTRA-SOUNDS-FR-WAV
[*] EXTRA-SOUNDS-FR-ULAW
[ ] EXTRA-SOUNDS-FR-ALAW
[ ] EXTRA-SOUNDS-FR-GSM
[ ] EXTRA-SOUNDS-FR-G729
[ ] EXTRA-SOUNDS-FR-G722
[ ] EXTRA-SOUNDS-FR-SLN16
[ ] EXTRA-SOUNDS-FR-SIREN7
[ ] EXTRA-SOUNDS-FR-SIREN14

```

- Une fois les modifications effectuées il faut enregistrer la configuration en tapant la touche S :

```

*****
Asterisk Module and Build Option Selection
*****

Press 'h' for help.

ARE YOU SURE?
--- It appears you have made some changes, and
you have opted to Quit without saving these changes!

Please Enter Y to exit without saving;
Enter N to cancel your decision to quit,
and keep working in menuselect, or
Enter S to save your changes, and exit

```

- Ensuite il suffit de compiler le logiciel asterisk avec la commande make dans le répertoire de téléchargement :

```

+----- Asterisk Build Complete -----+
+ Asterisk has successfully been built, and +
+ can be installed by running:             +
+                                           +
+               make install                +
+-----+
fabien@debian-template:~/asterisk-15.0.0$

```

- Puis pour l'installer il faut taper la commande :

- o make install

```
+---- Asterisk Installation Complete -----+
+
+   YOU MUST READ THE SECURITY DOCUMENT   +
+
+ Asterisk has successfully been installed. +
+ If you would like to install the sample +
+ configuration files (overwriting any    +
+ existing config files), run:           +
+
+ For generic reference documentation:    +
+   make samples                         +
+
+ For a sample basic PBX:                 +
+   make basic-pbx                       +
+
+----- or -----+
+
+ You can go ahead and install the asterisk +
+ program documentation now or later run:   +
+
+           make progdocs                  +
+
+ **Note** This requires that you have     +
+ doxygen installed on your local system  +
+-----+
fabien@debian-template:~/asterisk-15.0.0$
```

- Ensuite pour générer les fichiers de configuration et les fichiers d'exemples il faut taper les commandes suivantes :
 - o make samples
 - o make config
- Il suffit ensuite de démarrer le logiciel asterisk puis de le mettre au démarrage :
 - o sudo systemctl start asterisk
 - o sudo systemctl enable asterisk
- Pour se connecter à la console astrisk il faut taper la commande ou les v designe la verbosité de la console, plus il y en a plus la console asterisk va afficher des informations :
 - o sudo asterisk -rvvvv

```
fabien@debian-template:~/asterisk-15.0.0$ sudo asterisk -rvvvv
Asterisk 15.0.0, Copyright (C) 1999 - 2016, Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 15.0.0 currently running on debian-template (pid = 55718)
debian-template*CLI>
```

- On peut vérifier que le logiciel asterisk écoute sur les bons ports avec la commande
 - o Sudo netstat -ntplu

```
fabien@debian-template:~/asterisk-15.0.0$ sudo netstat -ntplu
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Adresse locale Adresse distante Etat PID/Program name
tcp 0 0 0.0.0.0:2000 0.0.0.0:* LISTEN 55718/asterisk
tcp 0 0 0.0.0.0:47000 0.0.0.0:* LISTEN 616/sshd
udp 0 0 0.0.0.0:161 0.0.0.0:* 621/snmpd
udp 0 0 0.0.0.0:2727 0.0.0.0:* 55718/asterisk
udp 0 0 0.0.0.0:39626 0.0.0.0:* 55718/asterisk
udp 0 0 0.0.0.0:5000 0.0.0.0:* 55718/asterisk
udp 0 0 0.0.0.0:5060 0.0.0.0:* 55718/asterisk
udp 0 0 0.0.0.0:4569 0.0.0.0:* 55718/asterisk
udp 0 0 0.0.0.0:68 0.0.0.0:* 841/dhclient
udp 0 0 0.0.0.0:40011 0.0.0.0:* 621/snmpd
udp6 0 0 :::54699 :::* 55718/asterisk
```

IV. Configuration d'Asterisk

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**

1. Création des utilisateurs

Fichier Users.conf

La configuration d'un template(modèle de paramètres) permet de définir un ensemble de paramètres qui seront communs à plusieurs utilisateurs. Le but étant de factoriser ces paramètres afin d'éviter de multiples saisies lors de la création des comptes.

[template](!)

type=friend -> type d'objet SIP, friend = utilisateur

host=dynamic -> Vous pouvez vous connecter a ce compte SIP a partir de n'importe quelle adresse IP

dtmfmode=rfc2833 -> type de rfc utilisé

disallow=all -> Désactivation de tous les codecs

allow=ulaw -> Activation du codec µlaw

```
[template](!) ;notre template s'appelle template. Le ! Indique qu'il s'agit d'un template.
type = friend ;type d'objet SIP.
host = dynamic ;l'utilisateur n'est pas associé à une IP fixe.
dtmfmode = rfc2833 ;mode DTMF.
disallow = all ;interdit tous les codecs.
allow = opus ;autorise le codec opus.
allow = g722 ;autorise le codec g722.
;allow = all
;transport=tls
;encryption=yes
```

Le utilisateurs crée peuvent faire référence au template :

```
[6001](template)
fullname = John DOE
username = jdoe
secret=secret
context=admin
```

```
[1001] (template)
fullname=Fabien Mauhourat
username=fmauhourat
secret=toor
context=compta

[1002] (template)
fullname=Jean Pierre
username=jpierre
secret=toor
context=compta

[2001] (template)
fullname=Jean Charles
username=jcharles
secret=toor
context=dev

[2002] (template)
fullname=Jean Parle
username=jparle
secret=toor
context=dev
```

- Après avoir effectué des modifications dans le fichier de configuration il faut taper la commande reload dans la console asterisk.
- Pour afficher les utilisateurs créer il faut taper la commande :
 - Sip show users

```
Srv-Asterisk*CLI> sip show users
Username      Secret      Accountcode  Def.Context  ACL  Forcerport
1001          toor        1001         compta       No   No
1002          toor        1002         compta       No   No
2002          toor        2002         dev          No   No
2001          toor        2001         dev          No   No
Srv-Asterisk*CLI> █
```

2. Création du Dialplan

Le plan d'appels (DialPlan) permet le routage des appels à travers le serveur. Le fichier concerné est

extensions.conf.

Il s'agit de déterminer le comportement du serveur en cas d'appels reçus et émis.

Tout d'abord, le DialPlan est agencé sous forme de contexte.

Chaque utilisateur appartient à un contexte, d'après le fichier **User.conf**.

Quand un utilisateur lance un appel, Asterisk va chercher, dans le contexte associé à l'utilisateur, les actions à effectuer.

Dans cet exemple il y a 2 contextes qui sont admin et dev.

[general]

static = yes ;le DialPlan est statique.

writetoprotect = yes ;On ne peut pas le modifier depuis le CLI.

clearglobalvars = yes ;les variables sont effacées et recalculées à chaque redémarrage d'Asterisk.

[admin]

exten => _2XXX,1,Goto(dev,\${EXTEN},1)

exten => _1XXX,1,Dial(SIP/\${EXTEN},20)

exten => _1XXX,2,Hangup()

[dev]

exten => _1XXX,1,Goto(admin,\${EXTEN},1)

exten => _2XXX,1,Dial(SIP/\${EXTEN},20)

exten => _2XXX,2,Hangup()

```
[general]
static = yes ;le DialPlan est statique.
writetoprotect = yes ;On ne peut pas le modifier depuis le CLI.
clearglobalvars = yes ;les variables sont effacées et recalculées à chaque redémarrage d'Asterisk.

[compta]
exten => _2XXX,1,Goto(dev,${EXTEN},1)
exten => _1XXX,1,Dial(SIP/${EXTEN},20)
exten => _1XXX,2,Hangup()

[dev]
exten => _1XXX,1,Goto(compta,${EXTEN},1)
exten => _2XXX,1,Dial(SIP/${EXTEN},20)
exten => _2XXX,2,Hangup()
```

V. Configuration du client Blink et Ekiga

Un **softphone** ([anglicisme](#)) est un type de [logiciel](#) utilisé pour faire de la [téléphonie par Internet](#) depuis un [ordinateur](#) plutôt qu'un [téléphone](#).

Les softphone disponible sous linux sont ekiga ou encore jitsi et bien d'autres. Dans cette note technique le softphone utilisé est ekiga pour le système linux.

Malheureusement le softphone ekiga ne support pas le chiffrement des communications donc nous utiliseront un deuxième softphone sur windows du nom de Blink qui supporte le SIP over TLS et le SRTP.

1. Configuration de Ekiga sur Linux

- Après avoir installer le logiciel ekiga avec la commande :
 - Apt install ekiga
- Il suffit de la lancer avec la commande :
 - Ekiga
- Entrer ensuite les informations suivantes :
 - Nom : ce que vous voulez
 - Registraire : Adresse IP du serveur asterisk
 - Utilisateur : Identifiant de l'utilisateur
 - ID Auth : Même que l'identifiant de l'utilisateur
 - MDP : Mot de passe définit dans le fichier users.conf



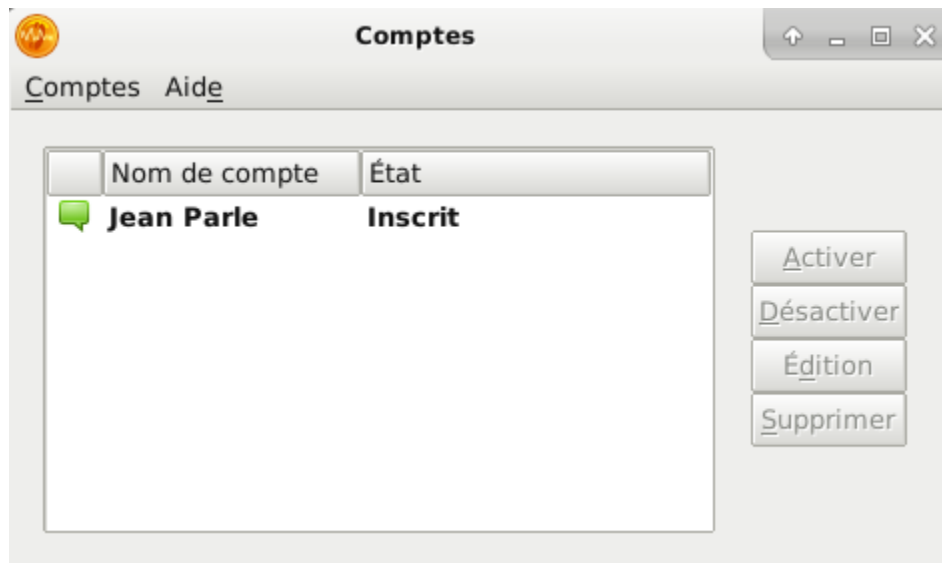
Modifier le compte

Veuillez mettre à jour les champs suivants :

Nom :	Jean Parle
Registraire :	192.168.10.245
Utilisateur :	2002
Identifiant d'authentification :	2002
Mot de passe :	●●●●
Délai :	3600

☒ Activer le compte

- Une fois l'utilisateur inscrit il est possible de passer les premiers appels :



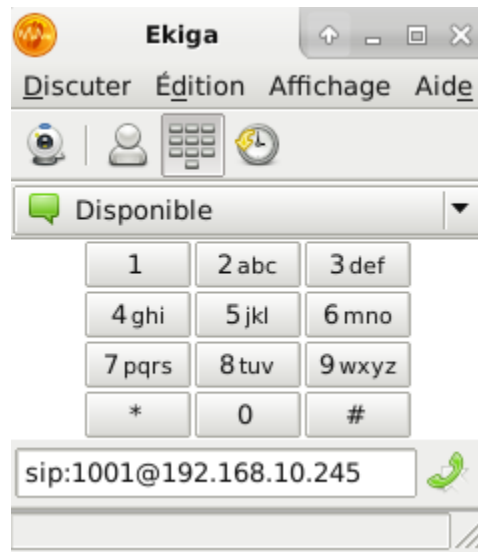
- Lorsqu'un utilisateur s'inscrit auprès du serveur un message apparaît dans la console :

```
-- Registered SIP '1002' at 192.168.40.1:53142
> Saved useragent "Blink 3.0.0 (Windows)" for peer 1002
Srv-Asterisk*CLI>
```

- Une fois les utilisateurs enregistrer il est possible de vérifier qu'elle utilisateurs est enregistrer sur que appareil avec la commande :
 - Sip show peers

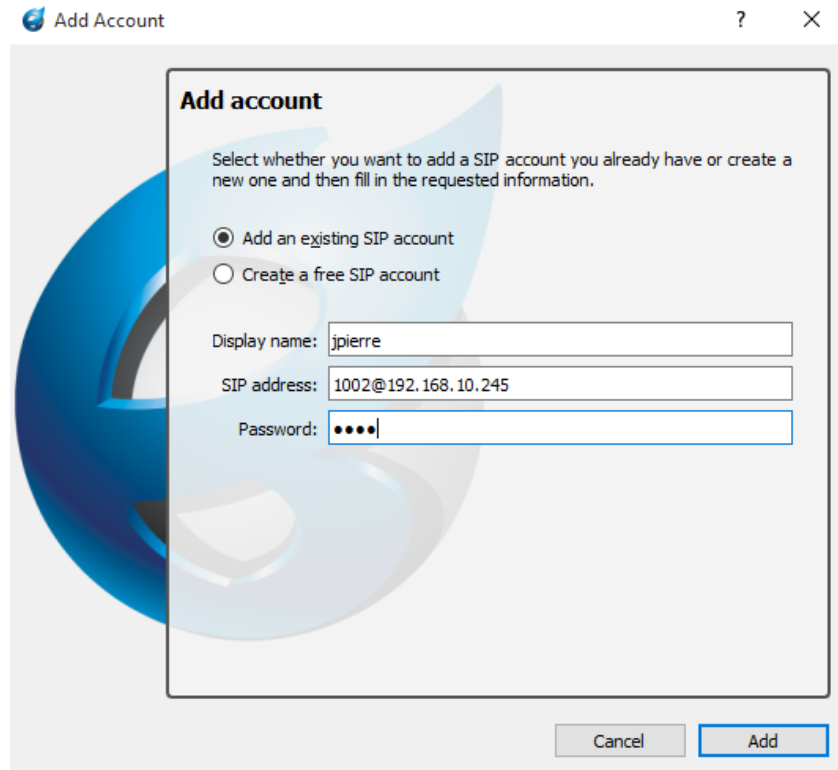
```
Srv-Asterisk*CLI> sip show peers
Name/username      Host                Dyn Forcerport Comedia  ACL Port
1001/fmauhourat    192.168.40.1       D Auto (No)  No      53142
1002/jpierre       192.168.40.1       D Auto (No)  No      53142
2001/jcharles      192.168.20.1       D Auto (No)  No      50939
2002/jparle        (Unspecified)      D Auto (No)  No      0
4 sip peers [Monitored: 0 online, 0 offline Unmonitored: 3 online, 1 offline]
Srv-Asterisk*CLI>
```

- Pour composer un numéro il faut taper le numéro de l'utilisateur suivi du signe @ puis de l'adresse IP du serveur asterisk :

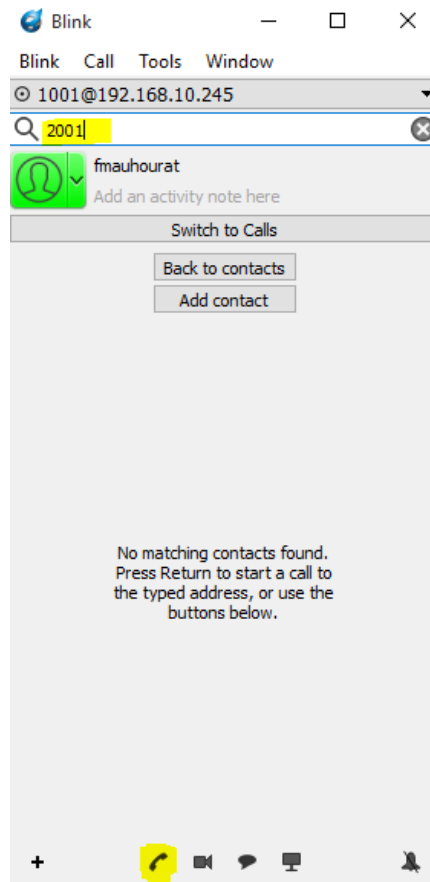


2. Configuration de Blink pour Windows

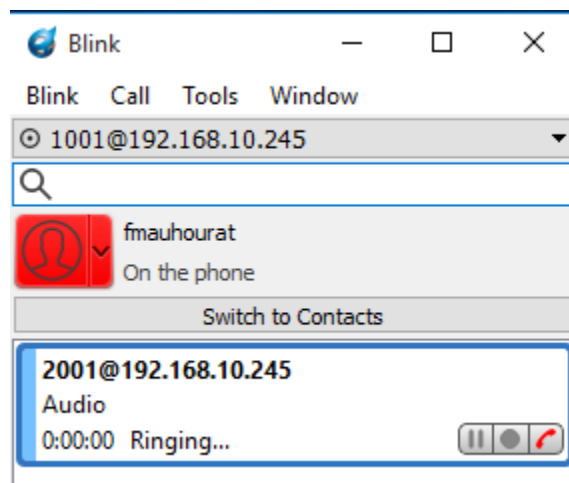
- Pour la configuration de blink il faut spécifier :
 - L'adresse de l'utilisateur au sens SIP : id user @ ip du serveur
 - Puis son mot de passe spécifier dans le users.conf



- Pour passer un appel il suffit de taper l'identifiant de l'utilisateur dans l'interface puis de cliquer sur le logo du téléphone :



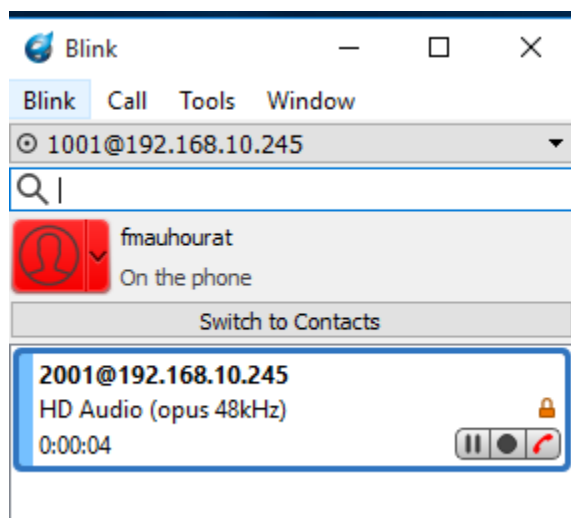
- Ensuite la sonnerie est déclenchée :



- Sur le deuxième poste la sonnerie apparaît il faut ensuite cliquer sur accepter :



- La communication est enfin établie :

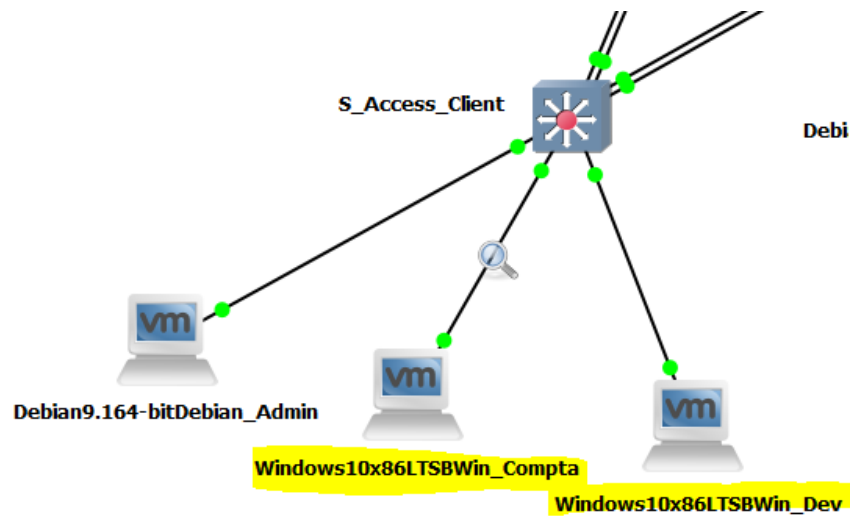


- Dans cette capture d'écran l'utilisateur 1001 du contexte admin appelle l'utilisateur 2001 du contexte dev. Ainsi il y a un saut de contexte qui s'effectue :

```
== Using SIP RTP CoS mark 5
> 0x7f3bc0036250 -- Strict RTP learning after remote address set to: 192.168.40.1:50014
-- Executing [2001@compta:1] Goto("SIP/1001-00000012", "dev,2001,1") in new stack
-- Goto (dev,2001,1)
-- Executing [2001@dev:1] Dial("SIP/1001-00000012", "SIP/2001,20") in new stack
== Using SIP RTP CoS mark 5
-- Called SIP/2001
-- SIP/2001-00000013 is ringing
> 0x7f3bc80067e0 -- Strict RTP learning after remote address set to: 192.168.20.1:50022
-- SIP/2001-00000013 answered SIP/1001-00000012
-- Channel SIP/2001-00000013 joined 'simple_bridge' basic-bridge <52831212-c785-4cc3-a826-3f1fd2d93757>
-- Channel SIP/1001-00000012 joined 'simple_bridge' basic-bridge <52831212-c785-4cc3-a826-3f1fd2d93757>
> Bridge 52831212-c785-4cc3-a826-3f1fd2d93757: switching from simple_bridge technology to native_rtp
> Remotely bridged 'SIP/1001-00000012' and 'SIP/2001-00000013' - media will flow directly between them
> 0x7f3bc80067e0 -- Strict RTP switching to RTP target address 192.168.20.1:50022 as source
> 0x7f3bc0036250 -- Strict RTP switching to RTP target address 192.168.40.1:50014 as source
> 0x7f3bc0036250 -- Strict RTP learning complete - Locking on source address 192.168.40.1:50014
-- Channel SIP/2001-00000013 left 'native_rtp' basic-bridge <52831212-c785-4cc3-a826-3f1fd2d93757>
-- Channel SIP/1001-00000012 left 'native_rtp' basic-bridge <52831212-c785-4cc3-a826-3f1fd2d93757>
== Spawn extension (dev, 2001, 1) exited non-zero on 'SIP/1001-00000012'
Srv-Asterisk*CLI>
```

VI. Mise en place d'une écoute clandestine des communiations avec Wireshark

- Pour réaliser l'écoute des conversations entre deux clients le logiciel GNS3 va être utilisé pour capturer les trames sur un lien d'un des clients :



- Cette capture de trame montre bien l'interception de la phase d'initialisation de l'appel vers l'utilisateur 1001 avec un filtre « sip » dans wireshark :

The screenshot shows a Wireshark capture of SIP traffic. The filter bar at the top is set to 'sip'. The packet list on the left shows a series of SIP messages between 192.168.40.1 and 192.168.10.245. The packet details pane on the right shows the structure of a SIP message, including the Request-Line: PUBLISH sip:1001@192.168.10.245 SIP/2.0. The packet bytes pane at the bottom shows the raw data of the captured frame.

No.	Time	Source	Destination	Protocol	Length	Info
15	16.904303	192.168.40.1	192.168.10.245	SIP/XML	861	Request: PUBLISH sip:1001@192.168.10.245
17	16.908196	192.168.40.1	192.168.10.245	SIP/XML	852	Request: PUBLISH sip:1002@192.168.10.245
18	16.959554	192.168.10.245	192.168.40.1	SIP	557	Status: 489 Bad Event
19	16.960412	192.168.10.245	192.168.40.1	SIP	551	Status: 489 Bad Event
21	17.074559	192.168.40.1	192.168.10.245	SIP/SDP	1043	Request: INVITE sip:2001@192.168.10.245
22	17.101021	192.168.10.245	192.168.40.1	SIP	625	Status: 401 Unauthorized
23	17.103252	192.168.40.1	192.168.10.245	SIP	434	Request: ACK sip:2001@192.168.10.245
24	17.103346	192.168.40.1	192.168.10.245	SIP/SDP	1209	Request: INVITE sip:2001@192.168.10.245
25	17.144882	192.168.10.245	192.168.40.1	SIP	569	Status: 100 Trying
30	20.981512	192.168.40.1	192.168.10.245	SIP	567	Request: REGISTER sip:192.168.10.245 (1 binding)
31	21.000907	192.168.10.245	192.168.40.1	SIP	631	Status: 401 Unauthorized
32	21.001708	192.168.40.1	192.168.10.245	SIP	728	Request: REGISTER sip:192.168.10.245 (1 binding)
33	21.025312	192.168.10.245	192.168.40.1	SIP	713	Request: NOTIFY sip:59842371@192.168.40.1:49571
34	21.028087	192.168.40.1	192.168.10.245	SIP	575	Status: 200 OK
36	21.510730	192.168.40.1	192.168.10.245	SIP	728	Request: REGISTER sip:192.168.10.245 (1 binding)
37	21.530012	192.168.10.245	192.168.40.1	SIP	643	Status: 401 Unauthorized
41	23.718364	192.168.10.245	192.168.40.1	SIP/SDP	884	Status: 200 OK
42	23.723931	192.168.40.1	192.168.10.245	SIP	439	Request: ACK sip:2001@192.168.10.245:5060
78	23.989619	192.168.40.1	192.168.10.245	SIP	573	Request: REGISTER sip:192.168.10.245 (1 binding)
112	24.324333	192.168.10.245	192.168.40.1	SIP/SDP	856	Request: INVITE sip:98075124@192.168.40.1:49571, in-dialog
113	24.329758	192.168.40.1	192.168.10.245	SIP/SDP	943	Status: 200 OK
132	24.504704	192.168.40.1	192.168.10.245	SIP	573	Request: REGISTER sip:192.168.10.245 (1 binding)
148	24.671871	192.168.10.245	192.168.40.1	SIP	637	Status: 401 Unauthorized
149	24.672887	192.168.40.1	192.168.10.245	SIP	734	Request: REGISTER sip:192.168.10.245 (1 binding)
157	24.754948	192.168.10.245	192.168.40.1	SIP	658	Status: 200 OK (1 binding)

Frame 15: 861 bytes on wire (6888 bits), 861 bytes captured (6888 bits) on interface 0
 Ethernet II, Src: Vmware_15:a7:a6 (00:0c:29:15:a7:a6), Dst: Cisco_00:01:01 (00:07:b4:00:01:01)
 Internet Protocol Version 4, Src: 192.168.40.1, Dst: 192.168.10.245
 User Datagram Protocol, Src Port: 49571, Dst Port: 5060
 Session Initiation Protocol (PUBLISH)
 Request-Line: PUBLISH sip:1001@192.168.10.245 SIP/2.0
 Message Header:
 c1 a3 13 c4 09 03 f9 3d 50 55 42 4c 49 53 48 20= PUBLISH
 73 69 70 3a 31 30 30 31 40 31 39 32 2e 31 36 38 sip:1001@192.168
 2e 31 30 2e 32 34 35 20 53 49 50 2f 32 2e 30 0d .10.245 SIP/2.0
 0a 56 69 61 3a 20 53 49 50 2f 32 2e 30 2f 55 44 .Via: SI P/2.0/UD
 50 20 31 39 32 2e 31 36 38 2e 34 30 2e 31 3a 34 P 192.16 8.40.1:4

Frame (861 bytes) Reassembled IPv4 (2307 bytes)
 SIP Request-Line (sip.Request-Line), 39 octets
 Paquets: 1113 · Affichés: 53 (4.8%) · Perdus: 0 (0.0%) · Profil: Defau

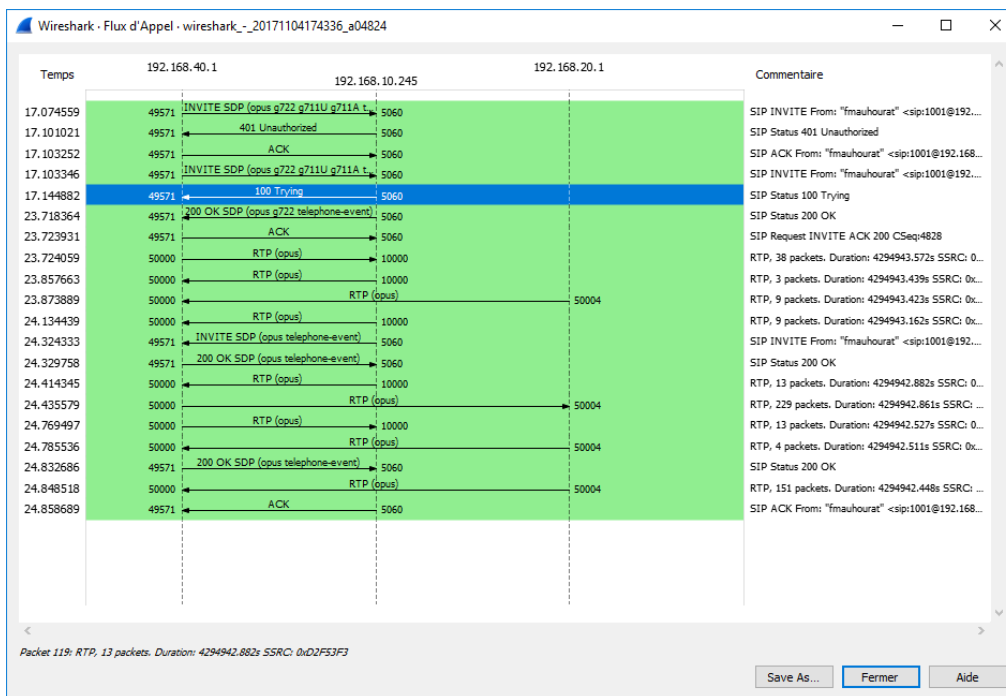
- Cependant avec WireShark l'interception de la communication en entier peut être effectué dans le menu Téléphonie puis appels VoIP :
 - Ainsi on peut voir que l'utilisateur 1001 appelle l'utilisateur 1001

The screenshot shows the 'Appels VoIP' menu in Wireshark. The table displays a call log entry for a call between 192.168.40.1 and 192.168.10.245.

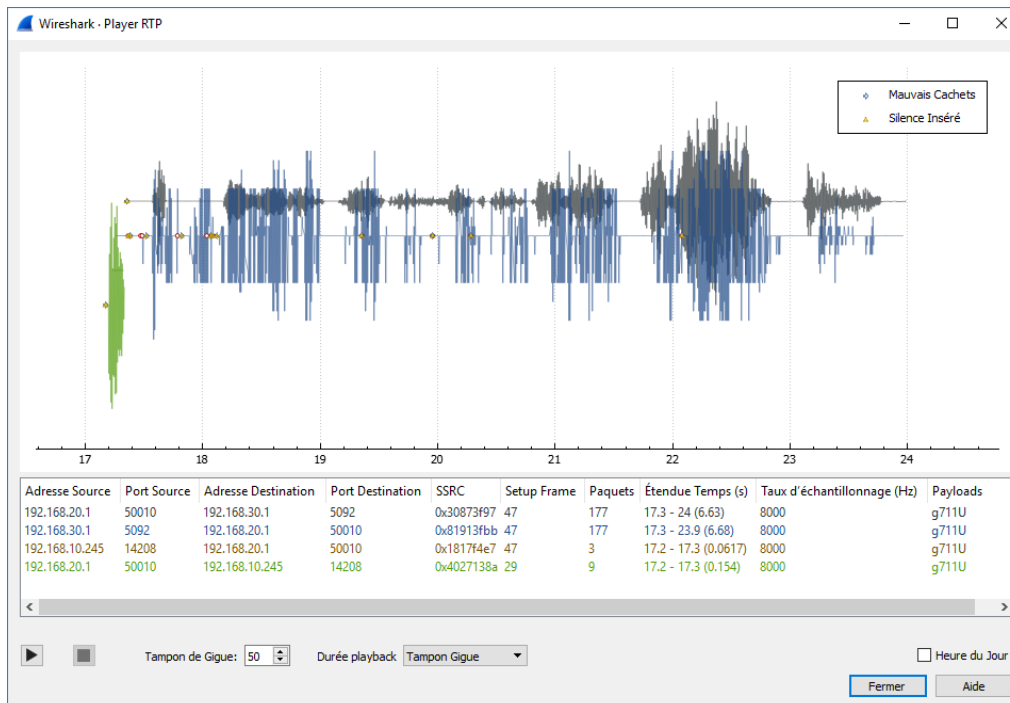
Heure de Début	Heure de Fin	Initial Speaker	De	À	Protocole	Paquets	État	Commentaires
17.074559	24.858689	192.168.40.1	"fmauhourat" <sip:1001@192.168.10.245	<sip:2001@192.168.10.245	SIP	11	IN CALL INVITE 401 200	

Buttons: OK, Annuler, Préparer Filtre, Séquence Flux, Jouer Flux, Copier, Aide

- On peut également observer la séquence de flux qui retrace toute la connexion des deux clients ToIP :



- Il est également possible d'écouter toute la conversation en jouant le flux :



VII. Mise en place du SIP over TLS et du SRTP

I. Configuration de l'autorité de certification et des certificats

Mise en place de la sécurisation des appel grâce au TLS :

```
sudo apt install libsrtplib2-1/stable libsrtplib0/stable libsrtplib0-dev/stable libsrtplib2-dev/stable
```

Création du certificat de l'autorité de certification :

L'autorité de certification devra signer les certificats générés.

- Création de la clé:
 - openssl genrsa -des3 -out ca.key 4096

Une passphrase est demandée lors de création du certificat.

- Création du certificat :
 - openssl req -new -x509 -days 365 -key ca.key -out ca.crt

```
fabien@Srv-Ocs-Glpi-Gestioip:/etc/apache2/ssl/CA$ ls -al
total 20
drwxr-xr-x 2 root root 4096 nov. 5 12:29 .
drwxr-xr-x 5 root root 4096 nov. 5 11:56 ..
-rw-r--r-- 1 root root 2090 nov. 5 11:58 ca.crt
-rw----- 1 root root 3311 nov. 5 11:57 ca.key
-rw-r--r-- 1 root root 2090 nov. 5 12:29 ca.pem
fabien@Srv-Ocs-Glpi-Gestioip:/etc/apache2/ssl/CA$
```

Création du certificat du serveur Asterisk :

- Création de la clé :
 - openssl genrsa -out key.pem 1024
- Création du fichier de demande de certificat :
 - openssl req -new -key key.pem -out req-srv.csr
- Création du certificate :

- openssl x509 -req -days 365 -in req-srv.csr -CA ../ca/ca.crt -CAkey ../ca/ca.key -set_serial 01 -out srv.pem
- Rassemble la clé privée et le certificat dans un fichier au format PEM :
 - cat key.pem > asterisk.pem
 - cat srv.crt >> asterisk.pem
- Voici les fichiers obtenue :

```
fabien@Srv-Ocs-Glpi-Gestioip:/etc/apache2/ssl/Srv$ ls -al
total 36
drwxr-xr-x 2 root root 4096 nov. 5 15:03 .
drwxr-xr-x 5 root root 4096 nov. 5 11:56 ..
-rw-r--r-- 1 root root 2306 nov. 5 15:30 asterisk.pem
-rw----- 1 root root 887 nov. 5 15:00 key-astersik.pem
-rw----- 1 root root 891 nov. 5 11:58 key.pem
-rw-r--r-- 1 root root 672 nov. 5 15:02 req-srv-asterisk.csr
-rw-r--r-- 1 root root 704 nov. 5 12:06 req-srv.csr
-rw-r--r-- 1 root root 1419 nov. 5 15:02 srv-asterisk.pem
-rw-r--r-- 1 root root 1448 nov. 5 12:06 srv.pem
fabien@Srv-Ocs-Glpi-Gestioip:/etc/apache2/ssl/Srv$
```

Création du certificat des clients (x2) :

- Création de la clé :
 - openssl genrsa -out key.pem 1024
- Création du fichier de demande de certificat :
 - openssl req -new -key key.pem -out req-client.csr
- Création du certificate :
 - openssl x509 -req -days 365 -in req-client.csr -CA ../ca/ca.crt -CAkey ../ca/ca.key -set_serial 01 -out client.pem
- Rassemble la clé privée et le certificat dans un fichier au format PEM :
 - cat key.pem > asterisk.pem
 - cat srv.crt >> asterisk.pem

- Voici les fichiers obtenue :

```
fabien@Srv-Ocs-Glpi-Gestioip:/etc/apache2/ssl/Client$ ls -al
total 40
drwxr-xr-x 2 root root 4096 nov. 5 15:08 .
drwxr-xr-x 5 root root 4096 nov. 5 11:56 ..
-rw-r--r-- 1 root root 1415 nov. 5 15:06 client-compta.pem
-rw-r--r-- 1 root root 1411 nov. 5 15:08 client-dev.pem
-rw-r--r-- 1 root root 2306 nov. 5 15:31 compta.pem
-rw-r--r-- 1 root root 2302 nov. 5 15:31 dev.pem
-rw----- 1 root root 891 nov. 5 15:04 key-compta.pem
-rw----- 1 root root 891 nov. 5 15:07 key-dev.pem
-rw-r--r-- 1 root root 672 nov. 5 15:05 req-client-compta.csr
-rw-r--r-- 1 root root 668 nov. 5 15:07 req-client-dev.csr
fabien@Srv-Ocs-Glpi-Gestioip:/etc/apache2/ssl/Client$
```

II. Modification de la configuration d'Asterisk

- Il faut ensuite modifier le fichier de configuration sip.conf :
- Modifier le transport en tsl
 - Activer le module TLS
 - Spécifier le chemin du certificat de l'autorité de certification et le certificat du serveur

```
[general]
transport=tlsv1
tlsv1enable=yes
tlsv1bindaddr=0.0.0.0
tlscertfile=/etc/asterisk/TLS/asterisk.pem
tlscacertfile=/etc/asterisk/TLS/ca.pem
tlscipher=ALL
tlsv1clientmethod=tlsv1
tlsv1dontverifyserver=yes
context=public ; Default context
;allowguest=no ; Allow or reject
```

- Désactiver le mode de transport en UDP et activer le mode TCP :

```
;
; Note that the TCP and TLS support is
; experimental. Since it is not
; subject to change in any release,
; be reflected in this sample configuration.
;
tcpenable=yes
tcpbindaddr=0.0.0.0
```

- Modifier ensuite le fichier users.conf :
 - Spécifier le nouveau mode de transport dans le template en TLS
 - Et activer le chiffrement des communications avec la mention encryption=yes

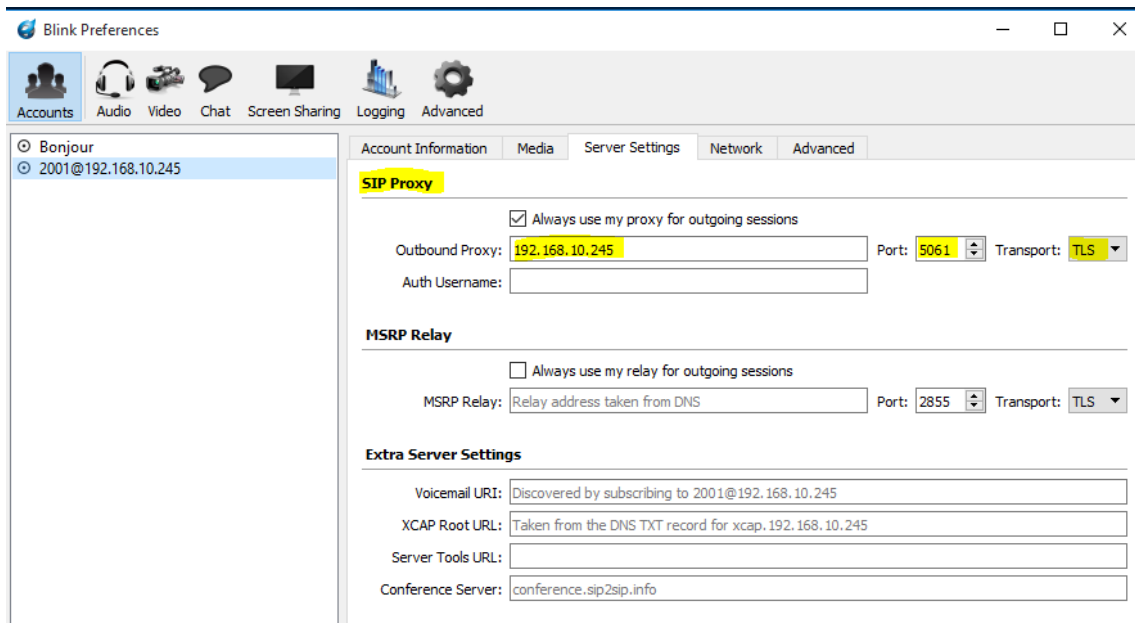
```
[template](!) ;notre template s'appelle template. Le ! Indique qu'il s'agit d'un template.
type = friend ;type d'objet SIP.
host = dynamic ;l'utilisateur n'est pas associé à une IP fixe.
dtmfmode = rfc2833 ;mode DTMF.
;disallow = all ;interdit tous les codecs.
allow = opus ;autorise le codec opus.
allow = g722 ;autorise le codec g722.
allow = all
transport=tls
encryption=yes
```

- Il est possible de vérifier que le logiciel asterisk écoute bien sur le port 5061 qui est le port tls :
 - Sudo netstat -petulan | grep asterisk

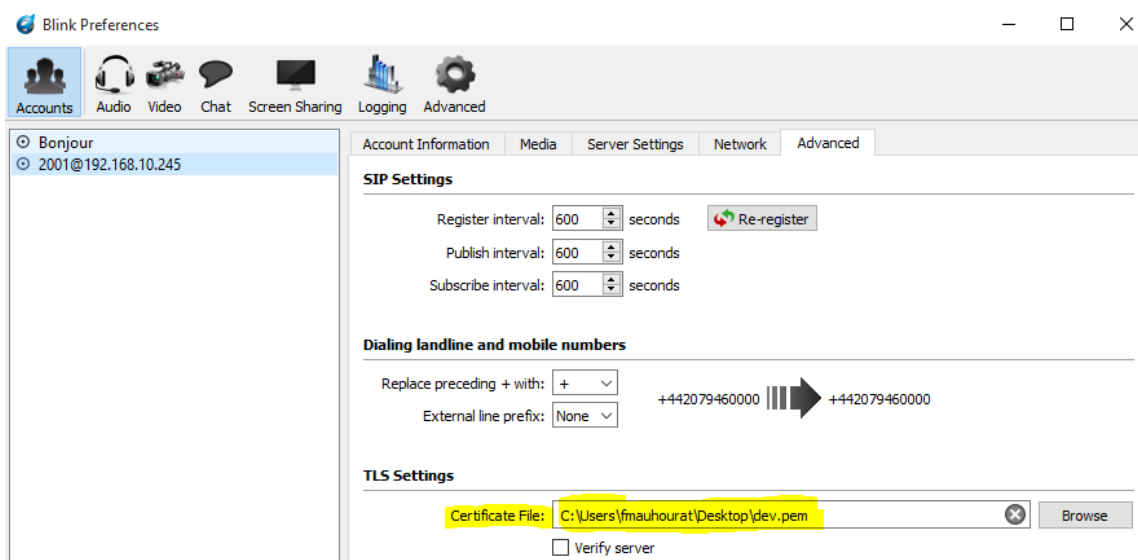
```
fabien@Srv-Asterisk:~$ sudo netstat -petulan | grep asterisk
tcp        0      0 0.0.0.0:2000        0.0.0.0:*           LISTEN      0          15780       701/asterisk
tcp        0      0 0.0.0.0:5060        0.0.0.0:*           LISTEN      0          15786       701/asterisk
tcp        0      0 0.0.0.0:5061        0.0.0.0:*           LISTEN      0          15787       701/asterisk
tcp        0      0 192.168.10.245:5061 192.168.40.1:49496  ESTABLISHED 0          17201       701/asterisk
tcp        0      0 192.168.10.245:5061 192.168.20.1:49499  ESTABLISHED 0          17156       701/asterisk
udp        0      0 0.0.0.0:2727        0.0.0.0:*           0           0          15782       701/asterisk
udp        0      0 0.0.0.0:5000        0.0.0.0:*           0           0          15795       701/asterisk
udp        0      0 0.0.0.0:5060        0.0.0.0:*           0           0          15785       701/asterisk
udp        0      0 0.0.0.0:4569        0.0.0.0:*           0           0          15784       701/asterisk
udp        0      0 0.0.0.0:53837       0.0.0.0:*           0           0          15777       701/asterisk
udp6       0      0 :::39015            :::*                0           0          15778       701/asterisk
fabien@Srv-Asterisk:~$
```

III. Modification de la configuration des clients

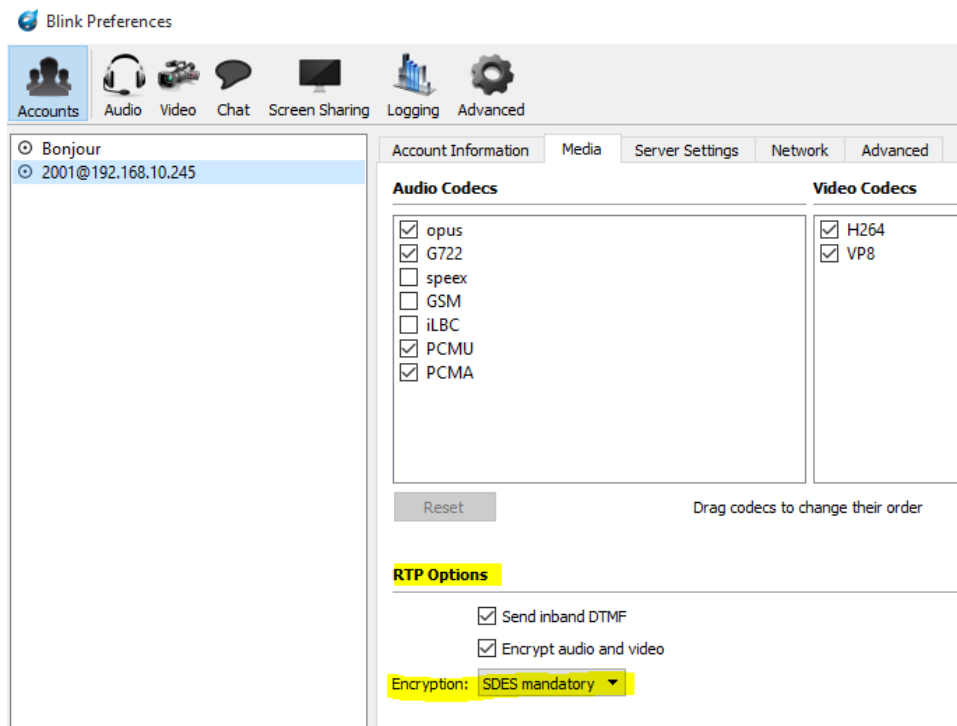
- Ensuite il faut configurer le client pour qu'il se connecte sur le port 5061 du serveur asterisk :
 - Dans les préférences de blink puis dans server settings



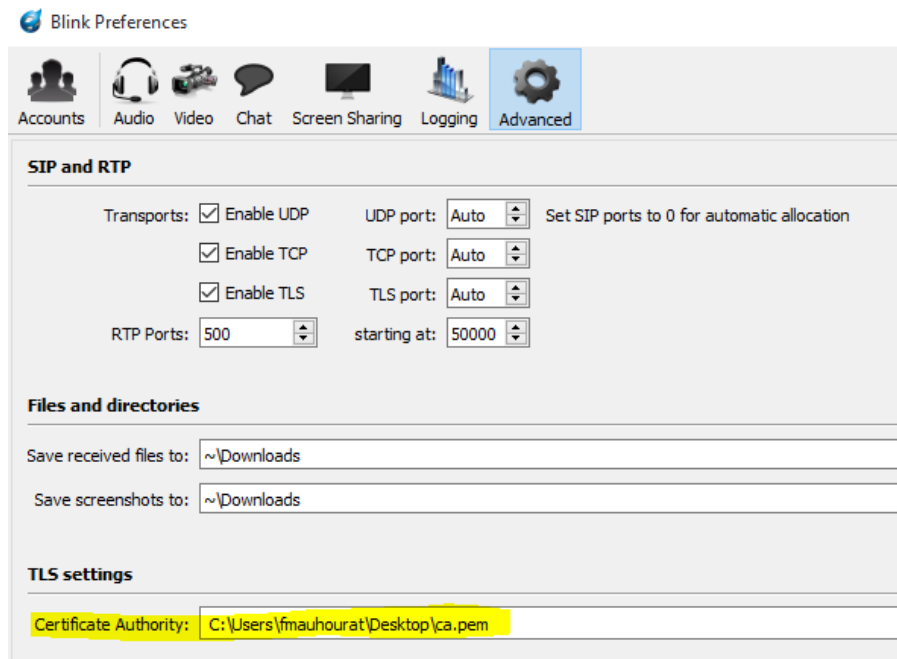
- Ensuite il faut spécifier le certificat du client dans l'onglet Advanced :



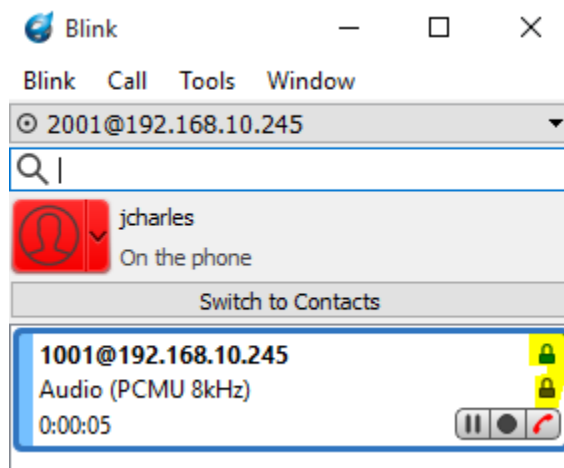
- Puis il faut active l'option SDES Mandatory dans la section RTP de l'onglet Media :



- Il faut ensuite dans l'onglet supérieure Advanced spécifier le certificat de l'autorité de certification :



- Lors de la phase d'appel les deux cadenas indiquent que l'initialisation de la connexion est sécurisée avec TLS et le deuxième cadenas indique que la communication est chiffrée :



- En réalisant une capture de trame avec le logiciel Wireshark on retrouve bien les protocoles TLS et TCP avec une connexion sur le port 5061 du serveur asterisk.

*Standard input [Windows10x86LTSBWin_Compta Ethernet0 to S_Access_Client Gi0/2]

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

tcp.port == 5061

No.	Time	Source	Destination	Protocol	Length	Info
70	17.317126	192.168.10.245	192.168.40.1	TCP	1514	[TCP segment of a reassembled PDU]
71	17.318342	192.168.10.245	192.168.40.1	TLSv1.2	444	Application Data
72	17.319839	192.168.40.1	192.168.10.245	TCP	54	49466 → 5061 [ACK] Seq=1 Ack=1851 Win=256 Len=0
73	17.323855	192.168.40.1	192.168.10.245	TLSv1.2	418	Application Data
74	17.391433	192.168.10.245	192.168.40.1	TCP	56	5061 → 49466 [ACK] Seq=1851 Ack=365 Win=602 Len=0
75	17.459937	192.168.40.1	192.168.10.245	TLSv1.2	597	Application Data
76	17.477731	192.168.10.245	192.168.40.1	TCP	56	5061 → 49466 [ACK] Seq=1851 Ack=908 Win=625 Len=0
83	20.167413	192.168.40.1	192.168.10.245	TCP	1514	[TCP segment of a reassembled PDU]
84	20.167541	192.168.40.1	192.168.10.245	TLSv1.2	928	Application Data
86	20.188936	192.168.10.245	192.168.40.1	TCP	56	5061 → 49466 [ACK] Seq=1851 Ack=2368 Win=648 Len=0
87	20.195691	192.168.10.245	192.168.40.1	TCP	56	5061 → 49466 [ACK] Seq=1851 Ack=3242 Win=670 Len=0
88	20.196841	192.168.10.245	192.168.40.1	TLSv1.2	604	Application Data
89	20.255277	192.168.40.1	192.168.10.245	TCP	54	49466 → 5061 [ACK] Seq=3242 Ack=2401 Win=254 Len=0
90	20.323025	192.168.40.1	192.168.10.245	TLSv1.2	994	Application Data
91	20.346172	192.168.10.245	192.168.40.1	TLSv1.2	561	Application Data
100	20.410771	192.168.40.1	192.168.10.245	TCP	54	49466 → 5061 [ACK] Seq=4182 Ack=2908 Win=252 Len=0

> Frame 72: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0

> Ethernet II, Src: Vmware_15:a7:a6 (00:0c:29:15:a7:a6), Dst: Cisco_00:01:02 (00:07:b4:00:01:02)

> Internet Protocol Version 4, Src: 192.168.40.1, Dst: 192.168.10.245

> Transmission Control Protocol, Src Port: 49466, Dst Port: 5061, Seq: 1, Ack: 1851, Len: 0

```

0000  00 07 b4 00 01 02 00 0c 29 15 a7 a6 08 00 45 00  ..... )....E.
0010  00 28 53 ac 40 00 80 06 f2 dc c0 a8 28 01 c0 a8  ..(S.@... ..(....
0020  0a f5 c1 3a 13 c5 ec c6 6a 14 d2 d8 b1 e1 50 10  ..:...j....P.
0030  01 00 49 f8 00 00  ..I...

```

IV. Annexes

<https://wiki.asterisk.org/wiki/display/AST/Installing+Asterisk+From+Source>

<http://denisrosenkranz.com/tuto-installer-et-configurer-asterisk-sous-debian-6-et-ubuntu/>