

Note d'application

ENVOI D'UN MAIL AVEC PIECE JOINTE PAR UNE LIGNE DE
COMMANDE

Table des matières

Introduction.....	2
I/ Avant-propos	2
II/ Installation des logiciels	2
III/ Configuration des logiciels	3
IV/ Mise en œuvre.....	6
Conclusion	6
Annexes	7
Annexe 1 : envoi d'un mail dans le programme.....	7
Annexe 2 : contenu du fichier test 2	7

Tables des figures

Figure 1 : ligne de commande d'installation de ssmtp.....	2
Figure 2 : ligne de commande d'installation de mailutils	2
Figure 3 : ligne de commande d'installation de mutt	3
Figure 4 : ligne de commande d'ouverture du fichier de configuration	3
Figure 5 : fichier ssmtp.conf	4
Figure 6 : partie root	4
Figure 7 : partie TLS	5
Figure 8 : authentification	5
Figure 9 : domaine.....	5
Figure 10 : informations machine	5

Introduction

Dans le cadre du cursus de Polytech Clermont-Ferrand, chaque étudiant de dernière année réalise un projet. Mon projet est intitulé « Surveillances de zones dangereuses et protégées par vision ». Ce projet est proposé par ERDF Lyon. Le but de cette note d'application est d'expliquer comment envoyer un mail avec (ou sans) pièce jointe. Je déclinerai, durant ce travail, les outils à installer, leur configuration ainsi que leur mise en œuvre.

I/ Avant-propos

Lors de ce projet, j'ai été en charge de réaliser la création de la zone dangereuse (défini par un opérateur) ainsi que la détection d'un point dans cette zone. Lors du découpage du problème, nous avons une partie communication (non traitée lors du projet cette année en accord avec le client). Le but de cette partie était de définir la communication entre les caméras (s'il y avait lieu) mais aussi la communication avec un éventuel poste de sécurité. A terme, le projet veut pouvoir envoyer une image (voir une vidéo en direct) de la personne étant dans une zone dangereuse à un poste de sécurité voir à tous les véhicules du chantier. C'est dans cette optique que s'inscrit cette note d'application, envoyer une image par mail de la personne dans la zone. Ceci permettra au client d'avoir une ébauche de la partie communication du projet.

L'environnement de développement sera Linux (Ubuntu).

II/ Installation des logiciels

Afin de permettre à une application d'envoyer des mails, nous avons besoin d'un programme appelé ssmtp. Ce dernier va créer le server qui enverra les mails vers notre boîte mail. La commande pour installer ssmtp est :

A terminal window with a dark background and orange text. The prompt is 'nicolas@ubuntu: ~'. The command entered is 'sudo apt-get install ssmtp'.

```
nicolas@ubuntu: ~  
nicolas@ubuntu:~$ sudo apt-get install ssmtp
```

Figure 1 : ligne de commande d'installation de ssmtp

A présent, il faut installer le programme mailutils. Ce dernier sera appelé par la ligne de commande pour envoyer le mail. Pour installer mailutils il faut taper la ligne de commande suivante :

A terminal window with a dark background and orange text. The prompt is 'nicolas@ubuntu: ~'. The command entered is 'sudo apt-get install mailutils'.

```
nicolas@ubuntu: ~  
nicolas@ubuntu:~$ sudo apt-get install mailutils
```

Figure 2 : ligne de commande d'installation de mailutils

Pour envoyer des pièces jointes avec notre mail, nous allons avoir besoin de mutt. Mutt est un client de messagerie console permettant notamment d'envoyer des pièces jointes. La ligne de commande permettant d'installer mutt est la suivante :



```
nicolas@ubuntu:~$ sudo apt-get install mutt
```

Figure 3 : ligne de commande d'installation de mutt

Ici, j'ai fourni les 2 options possibles : envoyer un mail avec pièce jointe (mutt) ou sans pièce jointe (mailutils).

A présent nous avons tous les outils pour procéder à la configuration de notre server et envoyer des mails en ligne de commande.

III/ Configuration des logiciels

Afin de configurer notre server, nous devons nous intéresser au fichier *ssmtp.config*. Toute la configuration du server se passe sur ce fichier. Il faut avoir les droits d'administrateur afin de pouvoir le modifier. Voici la ligne de commande à taper afin d'ouvrir ce fichier :

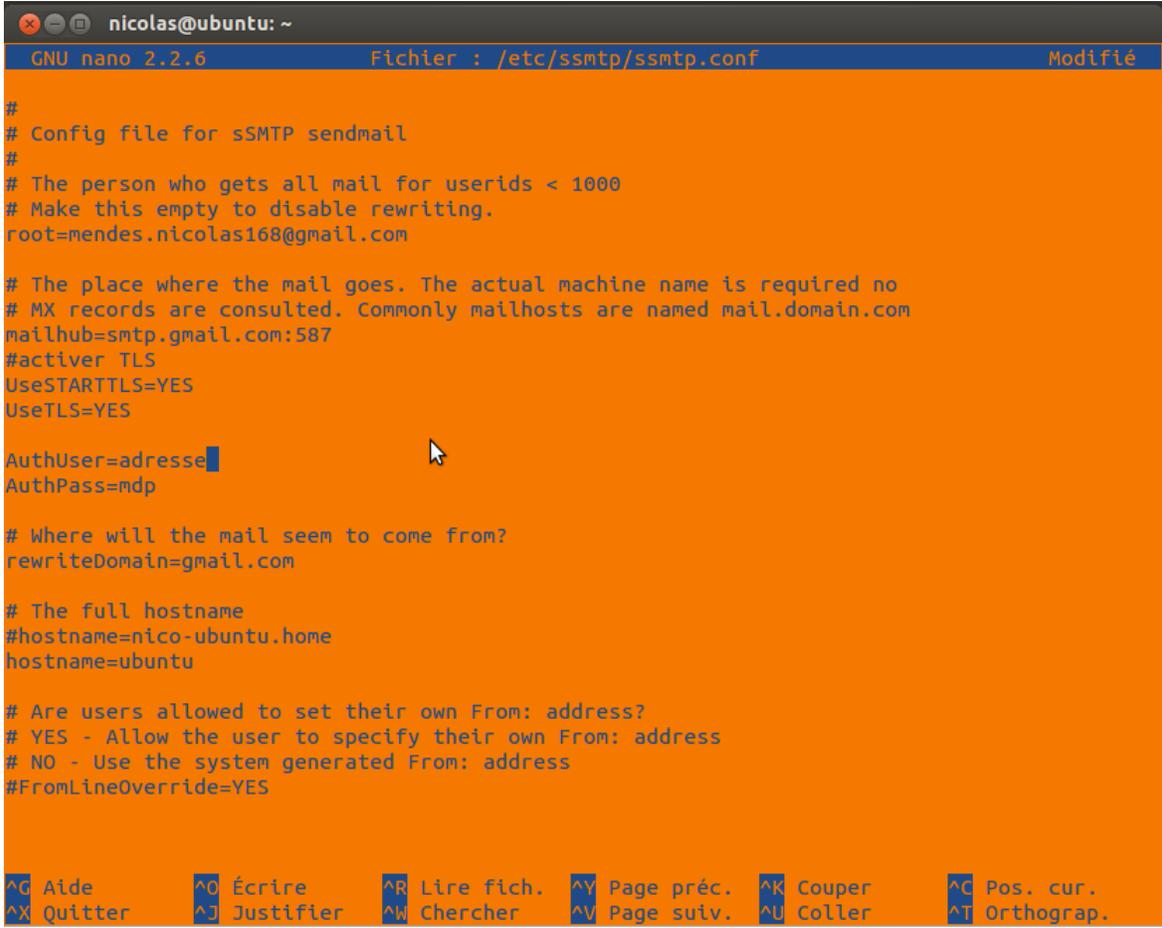


```
nicolas@ubuntu:~$ sudo nano /etc/ssmtp/ssmtp.conf
```

Figure 4 : ligne de commande d'ouverture du fichier de configuration

Il est important de respecter scrupuleusement l'orthographe et la syntaxe de cette ligne sinon un fichier vide s'ouvrira et en tapant les informations que je vais donner, vous allez créer un autre fichier à un autre emplacement et le fonctionnement ne sera pas garantis. Une fois ouvert, pour enregistrer les changements il faut appuyer sur CTRL+X puis sur o (pour oui) et entrée.

Voici alors le contenu du fichier :



```
nicolas@ubuntu: ~
GNU nano 2.2.6 Fichier : /etc/ssmtp/ssmtp.conf Modifié
#
# Config file for sSMTP sendmail
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
root=mendes.nicolas168@gmail.com

# The place where the mail goes. The actual machine name is required no
# MX records are consulted. Commonly mailhosts are named mail.domain.com
mailhub=smtp.gmail.com:587
#activer TLS
UseSTARTTLS=YES
UseTLS=YES

AuthUser=adresse
AuthPass=mdp

# Where will the mail seem to come from?
rewriteDomain=gmail.com

# The full hostname
#hostname=nico-ubuntu.home
hostname=ubuntu

# Are users allowed to set their own From: address?
# YES - Allow the user to specify their own From: address
# NO - Use the system generated From: address
#FromLineOverride=YES

^G Aide      ^O Écrire    ^R Lire fich. ^V Page préc. ^K Couper     ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher   ^V Page suiv. ^U Coller     ^T Orthograp.
```

Figure 5 : fichier ssmtp.conf

Il va falloir configurer ce fichier avec les éléments qui nous correspondent. Ici, ce fichier a été configuré pour Gmail. D'après mes recherches seul Gmail semble poser problème lors de la configuration de ce fichier. Je donnerais la configuration avec et sans adresse Gmail. Nous allons reprendre pas à pas la configuration de ce fichier. Toutes les lignes commençant par « # » sont des commentaires.



```
#
# Config file for sSMTP sendmail
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
root=mendes.nicolas168@gmail.com
```

Figure 6 : partie root

Ceci correspond au début du fichier de configuration ssmtp. Il est possible que cette partie soit absente du fichier, si c'est le cas ne pas hésiter à la rajouter. Il faut ensuite mettre à la suite de root votre adresse mail. C'est cette adresse qui apparaîtra comme expéditeur des mails.

```
# The place where the mail goes. The actual machine name is required no
# MX records are consulted. Commonly mailhosts are named mail.domain.com
mailhub=smtp.gmail.com:587
#activer TLS
UseSTARTTLS=YES
UseTLS=YES
```

Figure 7 : partie TLS

A présent, on arrive à une des spécificités de Gmail. Vous devez ici indiquer le serveur SMTP. Pour Gmail, il suffit de suivre ce que j'ai écrit. Le numéro de port peut être 465 ou 587. Pour ma part j'ai dû utiliser le 587 car dans le cas contraire je subissais un filtrage de la part du FAI. Afin de savoir quel serveur SMTP utiliser pour votre adresse mail, je vous invite à aller sur cette page :

<http://www.arobase.org/messageries/serveurs.htm>

Vous aurez accès à tous les serveurs POP, IMAP et SMTP de votre client de messagerie. Ne pas oublier de mettre « : » avant le numéro de port.

Concernant la partie TLS, il faut mettre YES aux 2. Cela permet d'envoyer des messages TLS à Gmail. Attention, ici il est possible qu'il y ait une erreur de base dans le fichier de configuration. Lorsque j'ai configuré ce fichier pour la première fois, j'avais « UseSTRATTLS =YES » au lieu de « UseSTARTTLS=YES ». Vous ne pourrez pas envoyer de mails si vous avez la même erreur que moi, il faut donc remplacer la ligne par la bonne forme (si vous avez cette erreur).

```
AuthUser=adresse
AuthPass=mdp
```

Figure 8 : authentification

Il faut ici indiquer ici votre identifiant Gmail dans « AuthUser » et votre mot de passe dans « AuthPass ».

```
# Where will the mail seem to come from?
rewriteDomain=gmail.com
```

Figure 9 : domaine

Ici, vous devez indiquer le domaine de votre client de messagerie. Dans mon cas c'était Gmail mais il faut modifier si vous utiliser un autre client de messagerie.

```
# The full hostname
#hostname=nico-ubuntu.home
hostname=ubuntu
```

Figure 10 : informations machine

Dans cette partie il s'agit de donner des informations relatives à la machine. Il faut indiquer le nom de la machine (dans mon cas, et par défaut, Ubuntu).

A présent, notre server est configuré. Mais il convient de protéger notre mot de passe afin qu'il ne soit pas visible de tous. Pour ce faire il suffit de taper la ligne suivante :

```
Sudo chmod 640 /etc/ssmtp/ssmtp.conf
```

On remplace par 640 un paramètre du fichier ssmtp.conf, qui entraîne l'invisibilité de notre mot de passe. A présent seul root peut envoyer des mails. Cependant il est possible d'ajouter des utilisateurs en procédant comme suit :

```
sudo chown root:mail /etc/ssmtp/ssmtp.conf
```

```
sudo gpasswd -a USERNAME mail
```

IV/ Mise en œuvre

Afin d'envoyer un mail sans pièce jointe il suffit de taper la ligne suivante

```
echo "objet" | mail -s "texte" adresse_mail
```

« objet » est l'objet du mail, mail est le programme qu'on utilise pour envoyer le mail, « texte » correspond au texte du mail et adresse mail est le destinataire du mail.

Afin d'envoyer un mail avec pièce jointe, il suffit de taper la ligne suivante :

```
echo "objet" | mutt -s "texte" -a fichier.xx -- adresse_mail
```

Ici, les éléments sont similaires au précédent sauf que mail est remplacé par mutt (on utilise le programme permettant d'envoyer des pièces jointes avec le mail) et fichier.xx est le fichier qu'on souhaite envoyer avec xx l'extension du fichier. Ce fichier doit être placé dans le build.

Ceci permet uniquement d'envoyer un mail. Afin de l'intégrer à mon programme, j'ai dû créer une variable que j'initialisais à 1. Lorsqu'une personne était dans la zone, je l'incrémentais. Lorsque cette variable valait 2 j'enregistrais l'image, puis l'envoyais. J'incrémentais ensuite ma variable pour quitter ma condition et n'envoyer que la première image. J'ai fourni en annexe 1 l'envoi d'un mail avec ou sans pièce jointe au sein de mon programme. En annexe 2 est fourni le script qui utilise la ligne de commande. Pour utiliser une ligne de commande au sein d'un programme, l'utilisation de la fonction system est nécessaire. Cette fonction est bloquante par conséquent on ne peut rien exécuter en même temps. En revanche l'ajout d'un « & » après le nom du script à exécuter peut résoudre ce problème.

Concernant le script, il doit avoir une extension .sh. La ligne commençant par « # » est **indispensable** au fonctionnement du programme.

Conclusion

Cette note d'application permet de configurer et d'effectuer l'envoi d'un mail avec ou sans pièce jointe. Elle permet aussi de protéger son mot de passe et d'ajouter des utilisateurs. Cette technique fonctionne avec tous les clients de messagerie, à condition de respecter les étapes fournies.

Annexes

ANNEXE 1 : ENVOI D'UN MAIL DANS LE PROGRAMME

```
if(colision(k, tab, pied))
    one_coll =true;
    int nb[] = {k};
    if (one_coll){
        message=system("bash test2.sh &"); //execution du script
    }
```

ANNEXE 2 : CONTENU DU FICHIER TEST 2

```
#avec objet
#!/bin/sh
echo "alerte" | mutt -s "intrusion dans la zone" -a capture.jpg -- test@gmail.com
exit 0
```

```
#sans objet
#!/bin/sh
echo "alerte" | mail -s "intrusion dans la zone" test@gmail.com
exit 0
```