

L3 Informatique
DEVOIR MAISON
Systèmes d'exploitation avancés et réseaux
à rendre le 6 décembre 2015 à 23h59 au plus tard
Serveur MTA (Mail Transfer Agent)

1 Présentation

Le but de ce devoir est de réaliser un serveur MTA (Mail Transfer Agent) permettant l'acheminement de mail. Vous aurez à réaliser un script ou un ensemble de scripts permettant cette opération sur les machines en libre-services du département informatique en implémentant le protocole SMTP.

Le principe de base du travail à réaliser est le suivant : une fois lancé, le serveur permet, au travers d'un client mail classique, d'envoyer un mail qui pourra ensuite être lu depuis un client pop3 ou imap.

Ce projet est à rédiger en bash ou en python en utilisant la commande netcat pour faire la partie gestion du serveur.

2 Rappels

2.1 Rappels généraux

Vous pourrez utiliser **netcat** (ou **nc**) pour programmer un mini serveur. Il est nécessaire de développer des programmes appelés par **netcat** côté serveur. Pour la réalisation du projet, on rappelle l'existence d'une option **-e** de l'utilitaire **nc.traditional**. Ainsi qu'on l'a vu en T.P., il existe plusieurs versions de netcat. Prenez soin de choisir une version qui accepte le transfert du contrôle à un programme extérieur (option **-e**).

Le descriptif détaillé de la première version du protocole SMTP est disponible dans le RFC821, référez vous à cette documentation pour réaliser ce projet :

<https://tools.ietf.org/html/rfc821>.

Pour tester votre serveur vous pouvez vous aider de la commande **telnet** en parlant directement en SMTP ou d'un gestionnaire de mail comme Thunderbird.

2.2 Acheminement des mails

3 Serveur

Le serveur doit être à l'écoute sur un port réservé à cet usage (25, 587 ou 465 pour le protocole smtp en fonction des modes de connexion). Votre serveur devra pouvoir être lancé sur n'importe quel port. En particulier, nous prendrons le port 4567 pour ce projet.

3.1 Connexion simple au serveur

Vous implémenterez tout d'abord les commandes permettant une connexion en mode anonyme au serveur. Ce type de connexion ne nécessite pas de login et de mots de passe pour pouvoir envoyer des emails. La version minimale (sans effet mais avec des codes de retour cohérents et valables) prendra en compte les commandes **HELO** et **QUIT**, initiant et mettant fin à la connexion.

3.2 Envoi d'un mail à une boîte mail sur le serveur

Nous avons vu en TP qu'il était possible d'installer le logiciel *Exim4* pour mettre en place un serveur smtp. Ce logiciel peut permettre de stocker les mails dans le répertoire */home/<utilisateur>/Maildir* ou dans le fichier */var/mail/<nom d'utilisateur>* qui peuvent être ensuite lu par un serveur pop tel que celui de *courier*.

Implémentez les commandes **MAIL FROM**, **RCPT TO**, **DATA** pour permettre l'envoi d'un mail. Le résultat devra être similaire à celui du logiciel *exim* et pourra ensuite être traité par le logiciel *courier*. Assurez vous bien de ce dernier point. Les adresses mails que gèrent votre serveur sont de type *<nom d'utilisateur>@<vos noms>.tp.info.unicaen.fr* . Pour vos tests vous pouvez utiliser une machine virtuelle mais vous n'aurez pas à rendre de fichier VirtualBox.

N'oubliez pas d'implémenter les messages d'erreur associés aux commandes. Référez vous au RFC pour savoir les différents scénarios possibles.

3.3 Autres commandes à implémenter

Ajoutez les commandes suivantes :

- **HELP** donnant au client les informations du serveur et notamment la liste des commandes implémentées.
- **RSET** permettant de réinitialiser la transaction en effaçant le mail en cours de rédaction. Néanmoins, la connexion n'est pas fermée. Les informations sur l'expéditeur, le ou les destinataires et le message sont effacées.
- **VRFY** permettant de vérifier que l'utilisateur destinataire existe et est disponible pour la réception de mail.
- **NOOP** cette commande ne fait rien. Elle permet juste de savoir si le serveur est toujours connecté.

- L'envoi d'une commande non valide doit être correctement traité par le serveur avec une réponse adaptée.
- Dans le bloc **DATA**, lorsque l'on veut écrire une ligne contenant uniquement un point, il est nécessaire d'envoyer au serveur un double point pour ne pas confondre avec la fin de la transmission. Gérez ce cas particulier dans le mail que vous enregistrez.

Conformément au chapitre 4.5.1 du RFC sur smtp. Votre serveur doit maintenant contenir l'implémentation minimale pour être fonctionnel selon la norme.

Vous pouvez tester les différents scénarios de l'appendix F du RFC.

3.4 Relai vers d'autre domaine

Votre serveur smtp permet actuellement de gérer uniquement des mails à destination du domaine <vos noms>.tp.info.unicaen.fr . Vous allez dans cette partie relayé les mails à destination du domaine tpmail.info.unicaen.fr . Pour cela votre serveur devra détecter que le destinataire est dans le domaine tpmail.info.unicaen.fr et établir une connexion smtp avec le serveur mail de ce domaine (cf. le TP 8). Vous transmettez alors le mail en question.

3.5 Authentification

On ajoutera un support de mot de passe lors du lancement du serveur via l'ajout d'une option **-pass** en ligne de commande. Par exemple, on veut pouvoir lancer le serveur via la commande **serveur.sh -pass login_pass.txt** afin que celui-ci n'autorise l'envoi de mail que pour les utilisateurs authentifiés dont le login et le mot de passe sont dans le fichier *login_pass.txt*.

4 Rendu

Le travail est à réaliser en binôme (ou seul). Vous devrez rendre votre travail en utilisant le système *Devoir* de moodle. Pour cela, vous devez déposer une archive targz dans moodle.

Rappel de la date limite de la remise : **6 décembre 2015 à 23h59**.

Dans le cas présent cette archive doit contenir au moins un fichier **noms.txt** contenant les noms des étudiants ou étudiantes composant le binôme. Chaque étudiant(e) doit déposer au minimum ce fichier, qu'il ou elle soit seul(e), ou en binôme. Les autres fichiers (liste ci-dessous) ne seront déposées qu'une seule fois par binôme.

- les sources de vos programmes, les scripts ;
- un fichier **README.txt** indiquant clairement les instructions pour lancer le serveur ;
- un rapport expliquant la manière dont vous avez résolu les problèmes, et donnant le mode d'emploi de votre réalisation ;
- tout autre fichier dont la présence est nécessaire pour faire fonctionner l'ensemble de votre réalisation.

5 Évaluation

Le non-fonctionnement du serveur sera fortement pénalisé, vous devrez donc respecter scrupuleusement le protocole et vous assurer du bon fonctionnement de votre serveur. Il est donc inutile d'essayer d'introduire des fonctionnalités si ce qui a déjà été fait ne marche pas correctement.

La qualité et le contenu du rapport ainsi que le respect des consignes seront également pris en compte dans la notation.