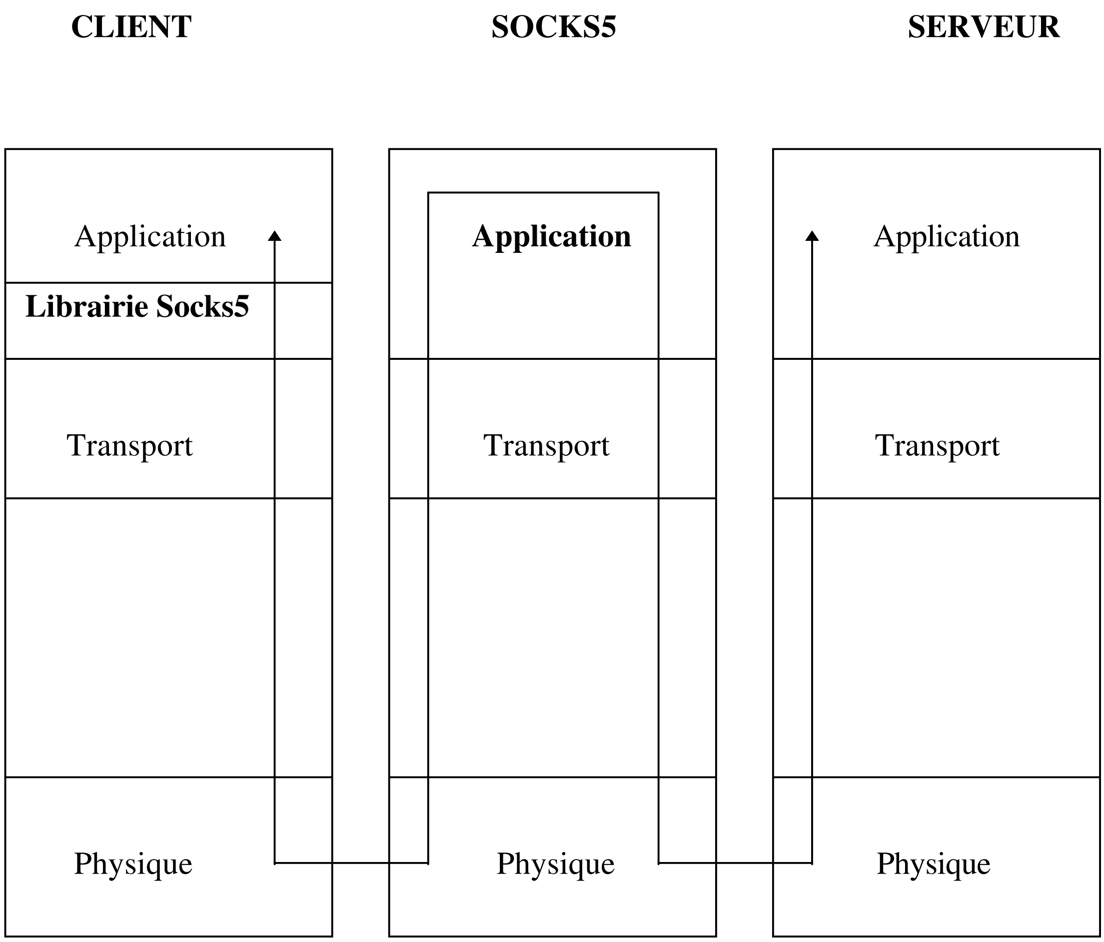
# *Présentation de*

# *SOCKS5*

*Lynda Sidi-Moussa*

*Stage Electricité de France - Direction Etudes et Recherches*

# *IMPLANTATION DE SOCKS5*



# Présentation de Socks5

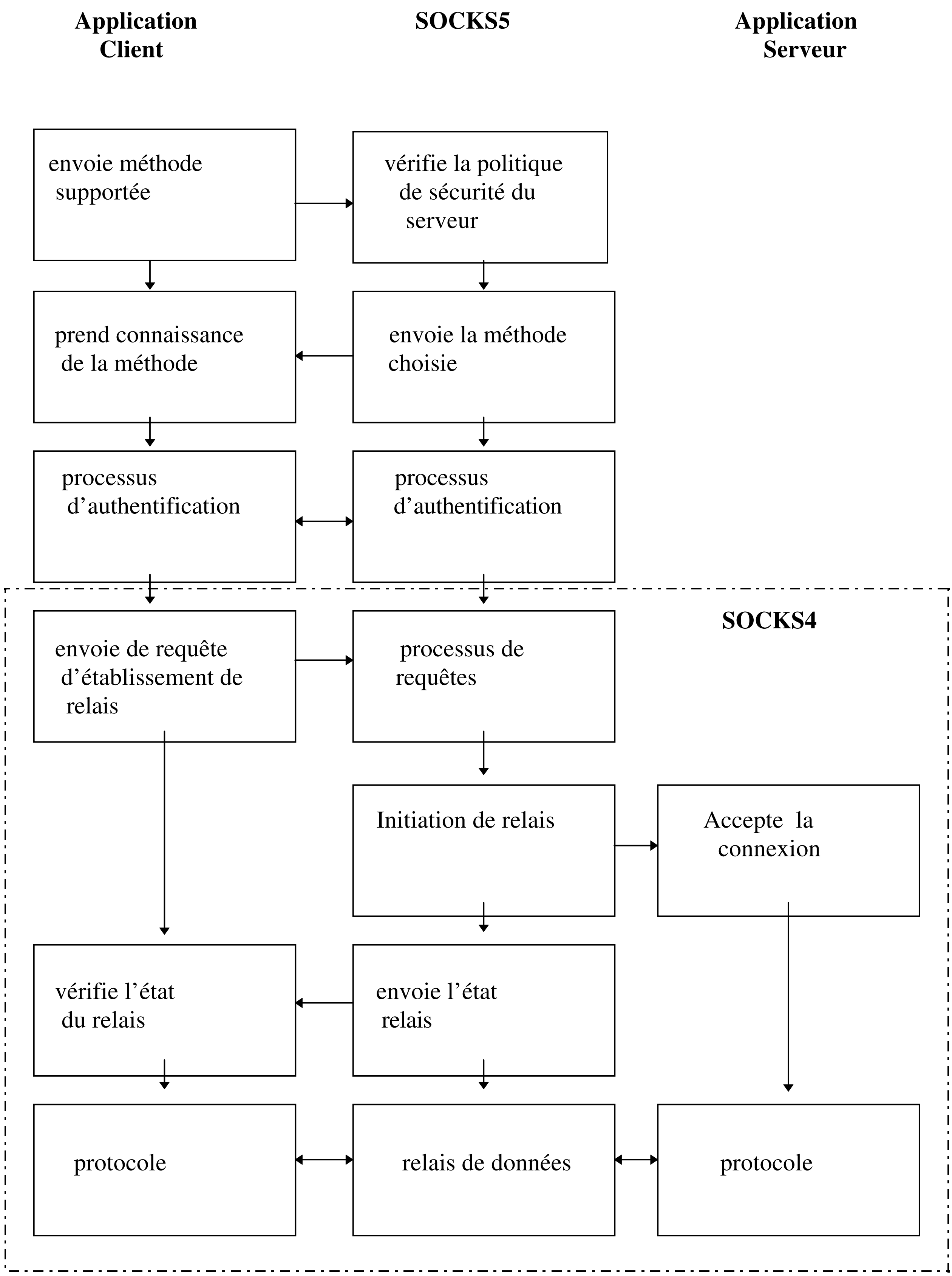
Les opérations de base réalisées par les serveurs Socks sont:

- requête de connexion,
- initiation d'un circuit proxy,
- relayage des données des applications.

L'objectif de Socks5 est de pallier toutes les insuffisances de Socks4:

- l'authentification, renforcée (authentifieur) ou non,
- la négociation pour le choix d'une méthode d'authentification,
- l'intégrité et la confidentialité des messages,
- un support pour les applications UDP.

# *Schéma de fonctionnement des Socks*



# *Implémentation de Socks5*

- les requêtes Socks4
- relayage de résolution d'adresse
- relayage de connexion serveur-à-serveur.

En plus des RFC 1929 et 1961, la version UNIX inclut aussi:

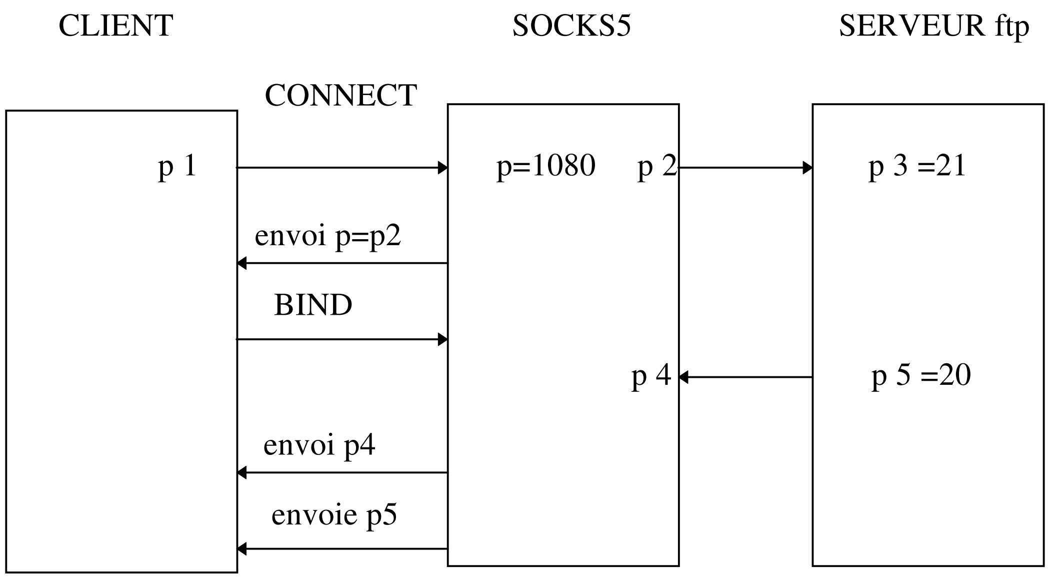
- « preforking server »,
- librairie client de Socks5,
- relayage de l'identité du client,
- relayage de ping et de traceroute.

# Protocole SOCKS5

## Procédures pour des clients TCP

- négociation pour le choix de la méthode d'authentification,
- requête du client, réponse du serveur,
- réponse du serveur à la requête du client.

*Exemple d'une requête CONNECT suivi de BIND*



## *Aspects sécurité du protocole Socks5*

La sécurité du protocole dépend d'abord de la première négociation entre le client et le serveur Socks5 parce que c'est dans cette phase que la méthode d'authentification est choisie et que à chaque méthode d'authentification est associé une sécurisation du protocole.

Rappelons qu'il existe actuellement deux méthodes d'authentification:

1. Avec username/password
2. GSS-API /Kerberos5.

La première est très peu sécurisé (mot de passe circulant en clair).

La deuxième méthode intègre en plus de l'authentification des méthodes d'encapsulation et confidetialité optionnelles qui peuvent être choisies selon les niveaux de protection désirée après le processus d'authentification. (Voire les RFC 1510 1961 et 1964)

Il est toutefois possible d'intégrer d'autres méthodes d'encapsulation basées ou non sur les GSS-API. Des améliorations sont prévues dans ce sens.

Socks5 assure la protection de la connexion entre le client et Socks5 aussi bien pour TCP que pour UDP mais pas entre Socks5 et le serveur d'application.

Dans le cas d'UDP, l'implémentation courante du serveur Socks5 essaye de limiter les attaques en rejetant tout paquet entrant provenant d'un serveur applicatif dont l'adresse n'a pas été mentionnée auparavant dans l'UDP ASSOCIATION.

# Authentification par username/passwd

Lorsque le client se connecte pour la première fois au serveur, il indique qu'il a choisi l'authentification avec username/passwd. Ceci débute la négociation d'authentification entre le serveur et le client. Le client envoie alors la requête constituée des champs suivants:

VER version de la négociation : X'01'

ULEN longueur du champ UNAME

UNAME username

PLEN longueur du mot de passe

PASSWD mot de passe

Après vérification du username et du mot de passe le serveur renvoie la réponse constituée des champs suivants:

VER version de la négociation : X'01'

STATUS indique si la connexion est acceptée ou non. Si la valeur du champ est différente de X'00' alors la connexion est immédiatement refermée.

**Remarque**

Cette authentification n'est pas conseillée dans les environnements où il est possible d'utiliser des sniffers, parce que le mot de passe est transmis en clair au serveur.

# *Installation de Socks5*

1. récupérer le source de Socks5 soit à partir de http://www.socks.nec.com/download.html
2. désarchiver et décompresser le source récupéré.
3. Construction de l'environnement de Socks5 et des fichiers de compilation.
4. Compilation du logiciel:
5. Installation dans le répertoire /usr/local

# *Configuration du serveur de proxy*

1. l'authentification,
2. le routage,
3. la définition des variables/flags,
4. la définition des relais,
5. les contrôles d'accès.

.

# *Les clients de socks5*

## *Configuration du client*

## *Socksification de l'application client:*

- soit : Recompilation avec la librairie de Socks5
- soit : Utiliser le script **runsocks** pour socksifier « dynamiquement » l'application (utilisation librairie dynamique Socks5)

# *EXEMPLES DE CONFIGURATION*

## <u>Sans authentification au niveau de Socks</u>

Côté serveur:

Le serveur est installé sur la station cli57af, donc le fichier configuration est /etc/socks5.conf:

auth - - -

# toutes les authentifications sont autorisées

set SOCKS_DEBUG 5

permit - - - - - -

# tous les accès sont permis.

1-1 côté client:

le fichier /etc/libsocks5.conf est vide

1-2 côté client

le fichier /etc/libsocks5.conf contient la ligne

socks5 - - - - cli57af

# Sans authentification au niveau de Socks5

Côté serveur:

Le serveur est installé sur la station cli57af, le fichier configuration /etc/socks5.conf contient:

auth - - -

# toutes les authentifications sont autorisées

set SOCKS_DEBUG 5

deny - b - - - -

#tous les bind sont interdits

permit - - - - - -

#toutes les connexions sont autorisées.

côté client

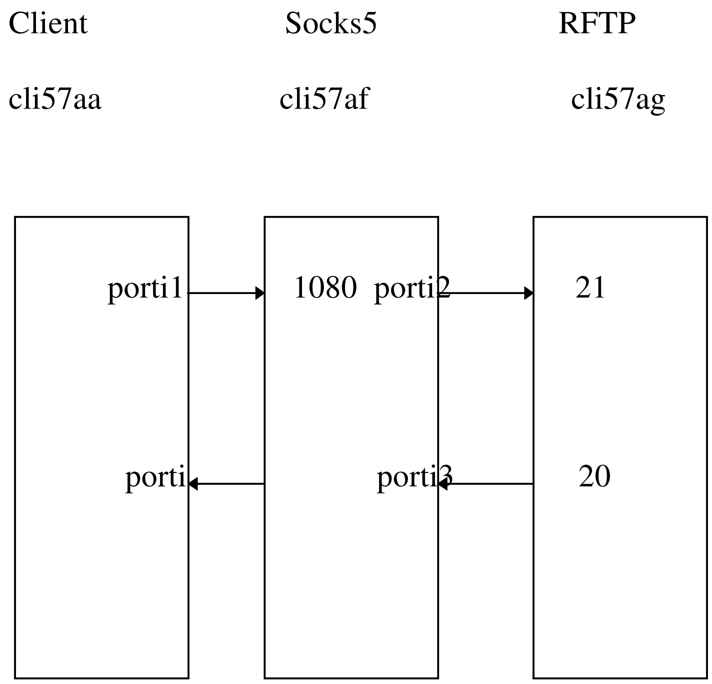
le fichier /etc/libsocks5.conf contient la ligne

socks5 - - - - cli57af

**Remarque** : si aucune ligne permit n'apparaît dans le fichier de configuration du serveur Socks5, toutes les connexions sont refusées.

# Analyse de rftp



## *Sans authentification au niveau de Socks5*

Côté serveur:

Le serveur est installé sur la station cli57af, le fichier configuration /etc/socks5.conf contient:

auth - - -

# toutes les authentifications sont autorisées

set SOCKS_DEBUG 5

deny - - - cli57aa - -

#interdit toute connexion vers la machine cli57af

permit - - - - - -

#toutes les connexions sont autorisées.

côté client

Un client de Socks5 pourra accéder à tous les hosts sauf le host cli57aa.

## Tests avec authentification username/password

Côté client

**setenv SOCKS5_USER username**

**setenv SOCKS5_PASSWD password,**

Côté serveur

Le fichier /etc/socks5.conf contient les lignes suivantes:

auth - - -

permit u - 192.9.200. - - -

#autorise à tous les utilisateurs d'accéder à toutes les machines à partir du sous-réseau 192.9.200.

deny u - - - - - - - sidimous

#interdit à l'utilisateur sidimous l'accès à toute machine

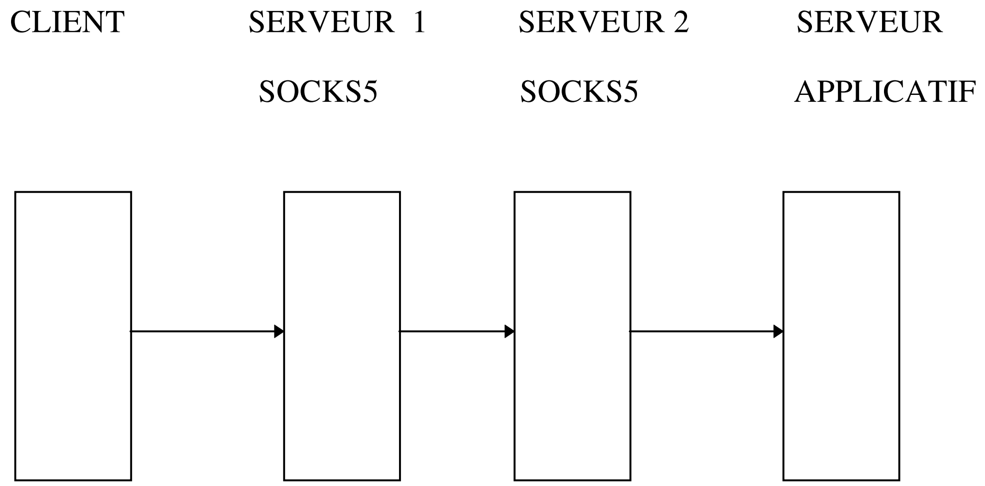
Côté client

le fichier /etc/libsocks5.conf contient la ligne suivante:

socks5 - - - - cli57af

# *Deuxième plateforme de tests : relais serveur-à-serveur*




Côté client

Le fichier /etc/libsocks5.conf sur la machine client contient la ligne suivante:

socks5 - - - - cli57aa

## ***Sans authentification user/passwd et sans authententification de serveur***

Côté serveur1 Socks5 : sur la machine cli57aa

auth - - -
socks5 - - cli57af 1080
#le deuxième serveur est sur la machine cli57af et le démon est à l'écoute sur le port 1080
permit - - - - - -

Côté serveur2 Socks5 : sur la machine cli57af
auth - - -
permit - - - - - -

## ***Avec authentification user/passwd et sans authentification de serveur***

Côté serveur1 Socks5 : sur la machine cli57aa

auth - - u

socks5 - - cli57af 1080

#le deuxième serveur est sur la machine cli57af et le démon est à l'écoute sur le port 1080

deny u - - - - sidimous

permit u - - - - -

Côté serveur2 Socks5 : sur la machine cli57af

auth - - -

permit - - - - - -

## **Avec authentification de serveur**

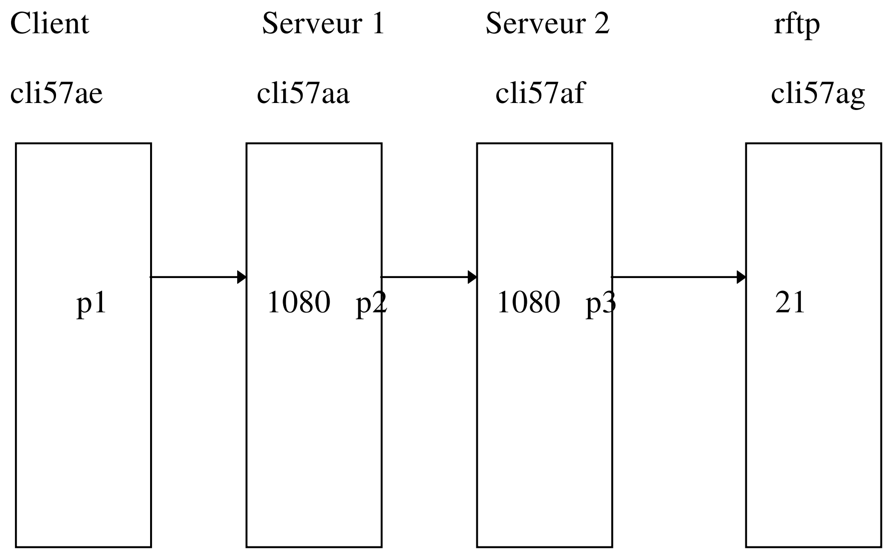
 

sur le deuxième serveur

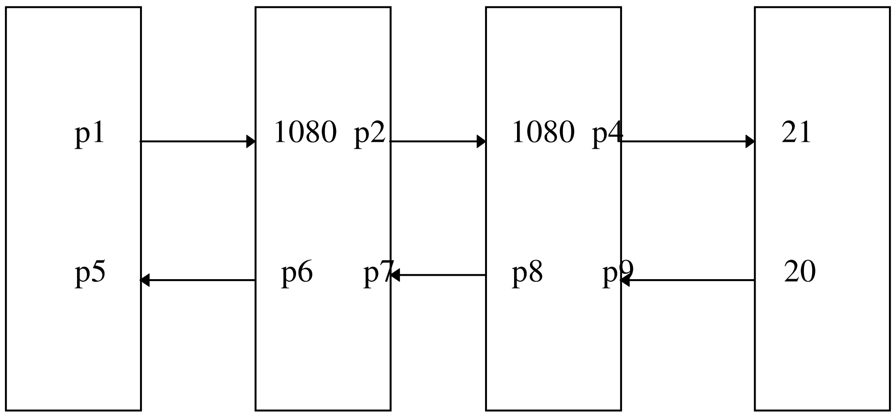
auth   cli57aa  -  k
permit -  -  -  -  -  -
permit k  -  cli57aa

# Analyse d'un rftp




Après un get:




Les dernières connexions sont supprimées dès la fin du transfert.

# *Authentification renforcée sur Socks5*

•pas implémentée actuellement. Pour l'impémenter, il faudrait :

•ajouter un nouveau code d'authentification « r » par exemple qu'il faudra enregistrer auprès de l'IANA,

•modifier la fonction HandleS5Connection dans le fichier proxy.c,

•ajouter un appel vers le module assurant l'authentification,

# *Authentification*

- Authentification avec username/password,
- Authentification avec GSS-API Kerberos 5.

# *Sécurité du procole*

- Socks assure seulement la protection de la connexion client / Serveur Socks5,
- Pour l'authentification avec username/password le mot de passe circule en clair,
- Pour l'authentification avec GSS-API : possibilité d'intégrité et de confidentialité des données.

# Conclusion

- Socks5 est un proxy standard facile à mettre en oeuvre
- Socks5 est d'utilisation simple pour les clients
- Socks5 permet de créer un réseau virtuel privé sur Internet avec des mesures de protections

Il reste à reste cependant à:

- Implémenter l'authentification de serveur dans le relayage serveur à serveur
- Implémenter l'authentification renforcée,
- Améliorer la sécurité du protocole