Faire passer XMPP au dessus de SSH (par exemple s'il est bloqué)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 4 avril 2013

https://www.bortzmeyer.org/xmpp-over-ssh.html

Si on veut faire de la messagerie instantanée en utilisant un protocole ouvert, avec du logiciel libre, sans serveur centralisé, la solution est le protocole XMPP, normalisé dans le RFC 6121 [1]. XMPP, comme le courrier électronique, repose sur le principe de fédération. Mais bien des réseaux bloquent (stupidement, mais c'est une autre histoire) XMPP en sortie. Si SSH passe, une solution possible est de faire passer XMPP sur SSH.

Cela m'est arrivé plusieurs fois : je suis sur un réseau a priori accueillant, je lance mon Pidgin pour bavarder ou discuter sérieusement et crac, pas moyen de se connecter à mon serveur. Les ports utilisés par XMPP sont bloqués par un pare-feu pénible (le port par défaut est 5222, mais l'utilisation des enregistrements SRV permet à un serveur XMPP d'être facilement joignable via un autre port.)

Il existe des tas de solutions à ce problème. Par exemple de configurer un VPN avec une machine externe et de faire passer XMPP par le VPN. Mais comme SSH marche à beaucoup d'endroits, est simple et offre une grande sécurité, j'utilise une solution basée sur SSH.

Je lance d'abord un relais Socks vers une machine de confiance, située en dehors du pare-feu pénible. Mettons le port 3022 :

```
% ssh -f -N -D 3022 moi@ma.machine.example
```

Le `-f` met SSH en arrière-plan, le `-N` dit de ne pas exécuter de commande distante, de juste faire le relais, et le `-D 3022` ouvre un relais Socks sur le port local 3022.

Ensuite, tout dépend du client XMPP utilisé. Dans Pidgin, choisir l'onglet Proxy, Proxy type SOCKS5, mettre comme Host la machine cliente SSH (ici `localhost`) et comme Port celui choisi (3022 dans l'exemple ci-dessus). Et voilà, on peut recommencer à communiquer.

Avec d'autres clients XMPP qui ne géreraient pas Socks, on peut utiliser tsocks <http://tsocks.sourceforge.net/> ou Dante <http://www.inet.no/dante/>.

Pour en apprendre plus sur les tunnels SSH + Socks, je recommande cet excellent article <http://artisan.karma-lab.net/faire-passer-trafic-tunnel-ssh>.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6121.txt