Perspectives, un outil pour améliorer la sécurité de SSH et des protocoles équivalents

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 11 juin 2008

https://www.bortzmeyer.org/perspectives-ssh.html

Vous connaissez le TOFU ? Pas celui qui se mange. TOFU est un sigle qui veut dire "*Trust On First Use*" et qui désigne les applications réseau qui vérifient, via la cryptographie, que la machine à laquelle elles parlent est bien la même que la dernière fois, alors même qu'il n'y a en général pas de vérification sérieuse la première fois. SSH est l'exemple le plus connu.

À la première connexion SSH avec une machine, on reçoit un avertissement :

```
% slogin rebecca
The authenticity of host 'rebecca.generic-nic.net (192.134.7.252)' can't be established.
DSA key fingerprint is b8:28:72:4d:66:40:9e:40:47:55:a1:40:ad:4a:d8:30.
Are you sure you want to continue connecting (yes/no)?
```

Si on répond OUI, sans vérifier l'empreinte (ce que fait quasiment tout le monde), on est connectés. Mais si le méchant s'était glissé sur le chemin à ce moment là, on est fichus.

SSH est plus efficace si le méchant arrive par la suite :

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the DSA host key has just been changed.
The fingerprint for the DSA key sent by the remote host is
b8:28:72:4d:66:40:9e:40:46:55:a1:40:ad:4a:d8:30.
Please contact your system administrator.
Add correct host key in /home/bortzmeyer/.ssh/known_hosts to get rid of this message.
Offending key in /home/bortzmeyer/.ssh/known_hosts:11
DSA host key for rebecca.generic-nic.net has changed and you have requested strict checking.
Host key verification failed.
```

Par défaut, SSH bloque l'accès dans ce cas. Il reste donc la faiblesse de la première connexion. Une solution possible est d'utiliser des options « fascistes » comme `StrictHostKeyChecking yes` et de vérifier à la main les empreintes des clés. Très peu le font, c'est bien trop contraignant. Une autre approche a été choisie par le RFC 4255 [1], qui consiste à publier la clé SSH dans le DNS. Évidemment, cela implique que la zone soit signée par DNSSEC et que l'administrateur de la zone fasse bien son nouveau travail d'Autorité de certification.

Perspectives, présenté dans l'article "*Improving SSH-style Host Authentication with Multi-path Network Probing*" <`http://www.cs.cmu.edu/~perspectives/`> adopte une autre approche. Un réseau de machines situées un peu partout dans l'Internet se connectent aux machines qu'on leur a indiqué et, sur demande, indiquent la clé observée. Ce réseau fournit donc une **redondance spatiale** (les différentes machines observent depuis différents points du réseau, ce qui protège contre certaines attaques, par exemple celles utilisant BGP) et une **redondance temporelle** (les machines du réseau de vérification, les **notaires**, ont une mémoire des clés précédentes et peuvent détecter un changement).

Voyez l'article pour les détails, qui sont nombreux (il faut authentifier les notaires et qu'ils signent leurs messages, il faut pouvoir accepter plusieurs réseaux de notaires concurrents, il faut gérer le cas où, comme dans "*Minority Report*", ils ne sont pas d'accord entre eux, etc).

Ce système ne s'applique pas qu'à SSH, il peut concerner toutes les applications de type TOFU, ce qui est souvent le cas de HTTPS.

L'équipe de Perspectives a continué le travail et le tout est désormais présenté sur leur site officiel <`http://www.networknotary.org/`>. Des idées similaires ont été avancées pour le DNS, par exemple ConfiDNS <`http://www.usenix.org/event/worlds06/tech/prelim_papers/poole/poole_html/`>.

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc4255.txt`