GNU/Linux : empêcher le brute force

Author : pierrehirel

Contrairement à une idée reçue, les systèmes basés sur Linux ne sont pas inattaquables. Nous verrons ici comment se prémunir d'un type d'attaque, le brute force. Ce qui suit est valable sur de nombreux systèmes basés sur UNIX ou Linux, et devrait être considéré comme le B.A.-BA de la sécurité lorsqu'on installe un tel système.

Une façon basique (mais bourrin) de s'introduire dans un système Linux est le brute force, méthode par laquelle le pirate utilise un programme pour tester un très grand nombre d'identifiants, et finit par trouver les mots de passe nécessaires. Cette méthode est bien sûr encore plus redoutable si le pirate trouve le mot de passe du compte administrateur (« root »).

Qui est concerné ?

Cet article traite des vulnérabilités de serveurs. Si vous avez un serveur ssh ou http, il est important d'en connaître les vulnérabilités. Sur la plupart des distributions grand public (Ubuntu, Mandriva…) ces serveurs ne sont pas installés par défaut, cet article ne les concerne donc pas. Cependant il est courant d'activer ssh pour le contrôle à distance, ou Apache lorsqu'on veut faire du développement web. Les recommandations ci-dessous s'appliquent alors.

Comment un pirate peut-il trouver votre IP ?

L'une des méthodes pour localiser des PC vulnérables, c'est la requête « ping » : le pirate envoie ce genre de requête à plusieurs adresses IP, et relève celles qui répondent. Il obtient ainsi une liste d'adresses IP qu'il pourra ensuite attaquer. Ainsi un premier élément de sécurité est de désactiver la réponse au « ping », sur votre PC et/ou sur votre routeur Internet.

Une fois l'adresse localisée, le pirate peut chercher à scanner les ports, autrement dit vérifier quels ports sont ouverts. Les ports « classiques » (entendre, souvent recherchés et attaqués par des scripts de hacking) sont les ports 80 (serveur web), 21 (serveur FTP), 22 (accès ssh) et 5901 (serveur VNC). Il est donc conseillé, si vous utilisez de tels serveurs, de leur attribuer d'autres ports. Par exemple pour utiliser le port 2222 pour l'accès en « ssh », il faut modifier le fichier */etc/ssh/sshd_config* pour modifier la ligne indiquant le port à utiliser :

/etc/ssh/sshd

Port 2222

Le pare-feu et/ou le routeur doivent bien sûr être configurés en conséquence. Une autre possibilité serait, par exemple, de conserver le port 22 pour le démon ssh, de façon à toujours y accéder par le port par défaut en réseau local ; mais de fermer le port 22 sur le routeur, et de configurer un port forwarding, de sorte que les informations arrivant par le port 2222 du routeur (par exemple) soient toujours redirigées vers le port 22 du serveur.

Désactiver l'accès root via ssh

Ensuite, pour se connecter via ssh à un système Posix, deux informations sont nécessaires : le login et le mot de

passe. Un login qui existe sur tous les systèmes Posix est « root ». Le pirate peut donc tenter la simple commande :

ssh root@adresse

Si le PC cible répond en demandant le mot de passe, cela confirme qu'il s'agit bien d'un système Posix, et le pirate n'a plus qu'à trouver le mot de passe. Pour éviter cela, l'accès à l'utilisateur root doit être désactivé dans les options de sshd. Il faut pour cela éditer le fichier */etc/ssh/sshd_config* pour modifier la ligne :

/etc/ssh/sshd_config

PermitRootLogin no

Par la suite, pour administrer le serveur à distance, il faut s'identifier en tant qu'utilisateur normal, et ensuite seulement accéder aux droits root avec la commande *su*. Mais l'identification directement en root lors d'un accès ssh est à proscrire.

Se prémunir du « brute force » : installation de fail2ban

fail2ban est un programme qui protège votre PC contre le « brute force ». Son fonctionnement est simple : il compte le nombre de tentatives de connexion provenant d'une IP donnée ; au-delà d'un certain nombre de tentatives échouées (mauvais identifiants entrés), cette IP est bannie (ajout d'une entrée dans le pare-feu IPtables) et le serveur ne lui répond plus. Pour les plus fainéants, il suffit d'installer fail2ban, les paramètres par défaut suffisant à prévenir du « brute force ». Pour une configuration un peu plus fine, il est possible de paramétrer le nombre de tentatives autorisées, ainsi que le temps de bannissement. Il faut pour cela éditer le fichier */etc/fail2ban/jail.conf*. Les paramètres importants sont les suivants :

/etc/fail2ban/jail.conf

maxretry = 3

findtime = 300

bantime = 3600

- *maxretry* est le nombre d'échecs de connexion (mauvais login ou mot de passe) autorisés pendant le temps *findtime* (en secondes) ; au-delà de ce nombre d'échecs, l'IP d'où proviennent ces tentatives est bannie ;
- *bantime* est le temps de banissement de l'IP fautive (en secondes).

Ainsi dans l'exemple ci-dessus, au bout de 3 tentatives ratées en moins de 5 minutes (300 secondes), l'IP est bannie pendant une heure (3600 secondes).

Il est également possible d'ajouter des IP à la « liste blanche » (ces IP ne seront alors jamais bannies même en cas

d’erreur d’identification) en les ajoutant à la ligne *ignoreip*.

Attention à l’erreur du débutant, qui consiste à considérer que si une IP attaque le serveur il faut la bannir à vie, et donc à paramétrer un *bantime* infini (ce qui se fait en lui associant une valeur négative). Le problème est que si l’administrateur lui-même se trompe de mot de passe, il peut se retrouver banni à vie de son propre serveur… Il vaut donc mieux paramétrer un *bantime* raisonnable dans un premier temps, puis une fois que la solution est éprouvée, lui associer une valeur plus grande.

La vigilance de l’admin…

Aucun système n’étant infaillible, la vigilance de l’administrateur est toujours de mise : surveillez ce qui se passe dans */var/logs* de temps en temps…