Vos serveurs sont-ils sécurisés ?

Gazette Linux n°111 — Février 2005

Blessen Cherian

blessen CHEZ poorman POINT com

Antonin Mellier - Traduction française Yvon Benoist - Relecture de la traduction française Joëlle Cornavin - Relecture et correction de la traduction française

Table des matières

En un mot, non. Aucune machine connectée à l'Internet n'est sécurisée à 100 %. Cela ne signifie pas que vous êtes désarmé. Vous pouvez prendre des mesures pour éviter les attaques, mais vous ne pouvez pas les empêcher complètement. C'est comme une maison — lorsque les fenêtres et les portes sont ouvertes, la probabilité qu'un voleur entre est grande, mais si les portes et les fenêtres sont fermées et verrouillées, la probabilité de se faire cambrioler est moindre, mais non nulle pour autant.

1. Qu'est ce que la sécurité de l'information ?

Dans le cadre de notre article, la sécurité de l'information désigne les méthodes que nous utilisons pour protéger les données sensibles contre les utilisateurs non autorisés.

2. Pourquoi avons nous besoin de sécurité de l'information ?

Le monde entier s'adapte rapidement aux technologies de l'information (IT, *Information Technologies*). Où que vous regardiez, la technologie informatique a revolutionné la façon dont les choses fonctionnent. On peut citer comme exemples les aéroports, les ports, les industries des télécommunications et la télévision, la prospérité économique de toutes ces activités résultant de l'utilisation des technologies de l'information. Les technologies de l'information sont partout.

Beaucoup d'informations sensibles transitent par l'Internet, comme les données de cartes de crédit, les mots de passe vitaux pour les serveurs et les fichiers importants. Il y a toujours un risque que quelqu'un voit et/ou modifie les données pendant qu'elles sont transmises. On ne compte plus les histoires horribles qui arrivent lorsqu'un individu obtient la carte de crédit ou les informations financières de quelqu'un. Il peut les utiliser absolument comme il le veut et pourrait même vous détruire ainsi que votre entreprise en s'emparant ou en détruisant tous vos avoirs. Pour éviter de telles situations critiques, il est conseillé d'avoir une bonne politique de sécurité et une bonne mise en œuvre de la sécurité.

3. Organigramme de sécurité

Voici une illustration de l'organigramme nécessaire pour mettre en œuvre une sécurité efficace :

Cet organigramme montre les étapes de base dans le cycle de vie de la sécurisation d'un système :

- Analyse des risques : porte sur le risque associé aux données présentes dans le serveur à sécuriser.
- Exigences de fonctionnement : étudie ce qui a trait aux exigences proprement dites pour diriger l'exploitation.

Ces deux composants couvrent les aspects de fonctionnement de la mise en œuvre de la sécurité.

- Politique de sécurité : couvre huit domaines spécifiques de la mise en œuvre de la sécurité elle est exposée de façon plus détaillée dans la section 4 ci-après
- Service de sécurité, mécanismes et objets : est véritablement la partie mise en œuvre de la sécurité.
- Gestion de la sécurité, surveillance, détection et réponse : est la face opérationnelle de la sécurité, où nous décrivons la manière de découvrir une brèche de sécurité et comment réagir si nous en trouvons une.

4. Politique de sécurité

La *Politique de sécurité* est un document qui aborde les domaines suivants :

1. Authentification : cette section étudie quelles sont les méthodes utilisées pour déterminer si un utilisateur est réel ou non, quels sont les utilisateurs pouvant ou non accéder au système, la longueur minimale de mot de passe autorisée, combien de temps un utilisateur peut être inactif avant d'être déconnecté, etc.
2. Autorisation : cette section étudie la classification des niveaux utilisateur et ce que chaque niveau est autorisé à faire sur le système, quels utilisateurs peuvent devenir *root*, etc.
3. Protection des données : cette section étudie des détails, comme quelles sont des données devant être protégées et qui peut accéder à quels niveaux de données dans le système.
4. Accès Internet : cette partie étudie les informations sur les utilisateurs ayant accès à l'Internet et ce qu'ils peuvent y faire.
5. Services Internet : cette section étudie quels sont les services sur le serveur qui sont accessibles depuis Internet et ceux qui ne le sont pas.
6. Audit de sécurité : cette partie étudie comment un audit et une revue des secteurs et processus liés à la sécurité seront effectués.
7. Gestion des incidents : cette partie aborde les étapes à suivre et les mesures à prendre s'il y a une brèche de sécurité. Sont inclues également les étapes nécessaires pour identifer le coupable à proprement parler et les méthodes pour prévenir des incidents futurs.
8. Responsabilités : cette partie examine qui sera contacté à un stade donné quelconque d'un incident et les responsabilités de l'administrateur ou des administrateurs pendant et après l'incident. C'est une section très importante, puisque le déroulement du mécanisme de gestion des incidents en dépend.

5. Types de sécurité de l'information

Il y a deux types de sécurité :

1. Sécurité physique / sécurité hôte et
2. Sécurité réseau

Chacune de ces sections comporte trois parties :

1. Protection : ralentir ou arrêter les intrusions ou les dommages.
2. Détection : prévenir quelqu'un si une brèche (ou une tentative de brèche) de sécurité se produit, quantifier et qualifier le type de dommage apparu ou susceptible de s'être produit.
3. Récupération : resécuriser le système ou les données après la brèche ou les dommages et si possible, réparer tout dommage apparu.

5.1. Sécurité hôte / sécurité physique

Sécurité hôte / sécurité physique signifie sécuriser le serveur contre des accès non autorisés. Pour ce faire, on peut protéger la machine par mot de passe, via des étapes telles que la mise en place d'un mot de passe dans le BIOS, l'installation de l'ordinateur dans un local verrouillé auquel seuls des utilisateurs autorisés ont accès, l'application de correctifs de sécurité au niveau du système d'exploitation et la vérification régulière des journaux pour rechercher toute intrusion ou attaque. Au niveau Sécurité hôte, on vérifie et on corrige les permissions sur tous les fichiers liés au système d'exploitation.

5.2. Sécurité réseau

La sécurité du réseau est l'un des aspects les plus importants de la securité globale. Comme je l'ai mentionné précédemment, aucune machine connectée à l'Internet n'étant complètement sécurisée, les administrateurs de sécurité et les propriétaires de serveur doivent être vigilants et s'assurer qu'ils sont informés de tous les nouveaux bogues et exploits qui sont découverts. Ne pas vous tenir au courant de ceux-ci risque de vous laisser à la merci de n'importe quel pirate néophyte.

5.3. Quel système d'exploitation est le mieux sécurisé ?

Chaque système d'exploitation a ses partisans et ses détracteurs. Il existe des moyens de rendre Windows plus sécurisé, mais la mise en œuvre est assez coûteuse. Linux est stable et raisonablement sécurisé, mais beaucoup d'entreprises le perçoivent comme bénéficiant d'une prise en charge médiocre au niveau des fournisseurs. Mon vote pour le meilleurs système d'exploitation en termes de sécurité va à FreeBSD, un autre système d'exploitation de type Unix, mais peu de personnes sont au courant de son existence.

6. Un pare-feu est-il la solution ultime aux problèmes de sécurité réseau ?

Non, un pare-feu ne représente qu'une partie de l'implémentation de la sécurité. Encore une fois, nous utiliserons l'exemple d'une maison. Dans une maison, toutes les fenêtres et les portes peuvent être fermées, mais si le verrou de la porte d'entrée est si inefficace qu'il suffit à quiconque d'introduire n'importe quel objet ressemblant à une clé pour l'ouvrir, alors à quoi sert-il que toute la maison soit fermée ? De même, si nous avons une politique de pare-feu ferme, elle restreindra les accès non autorisés, mais si les logiciels fonctionnant sur la machine sont périmés ou remplis de bogues, les pirates peuvent l'utiliser pour s'introduire dans le serveur et obtenir l'accès *root*. Cela montre qu'un pare-feu n'est pas la solution ultime. Une implémentation de sécurité planifiée est la seule vraie solution de qualité à ce problème.

7. La sécurité est un processus permanent

Une sécurité ininterrompue est un processus continu. Les administrateurs de sécurité ne peuvent mener leur travail qu'en fonction des alertes et des corrections de bogues émises à la date de la mise en sécurité,

donc afin de tenir compte toutes les corrections des bogues les plus récents, le travail sur la sécurité doit être effectuée de façon régulière.

8. L'implémentation de la sécurité crée-t-elle du temps système et/ou reduit-elle les performances ?

Oui, l'implémentation de la sécurité crée une petite quantité de temps système, mais elle ne doit pas réduire l'ensemble des performances de façon drastique. Afin de s'en assurer, une implémentation de sécurité bien menée comporte une section optimisation où l'administration de la sécurité donne la priorité à la fois aux performances et à la sécurité. Lorsqu'on sécurise un logiciel quelconque, il faut le faire de telle façon qu'il fournisse des performances maximales.

9. Audits de sécurité — Ce qu'il faudrait vérifier

Un audit de sécurité est une partie de l'implémentation de la sécurité où l'on essaie de découvrir les vulnérabilités du système et de suggérer des actions pour améliorer la sécurité. Dans un audit normal, les points ci-dessous devront être vérifiés, et un rapport sur les résultats de cet audit devra être créé.

1. Vérifier la détection d'intrusion. Utilisez **chkrootkit** ou **rkhunter** à cette fin.
2. Rechercher les bogues connus dans les logiciels installés sur le serveur — le noyau, openssl, openssh, etc.).
3. Faire un balayage de tous les ports réseau et repérer quels sont les ports ouverts. Signaler ceux qui ne devraient pas l'être et quel programme écoutent dessus.
4. Vérifier si `/tmp` est sécurisé.
5. Rechercher les processus cachés.
6. Vérifier les mauvais blocs disque dans toutes les partitions (simplement pour être sûr que le système est raisonnablement sain).
7. Rechercher les permissions de fichiers dangereuses.
8. Vérifier si le noyau a une vulnérabilité **ptrace**.
9. Vérifier la mémoire (une autre vérification de la santé du système).
10. Vérifier si le serveur est un relais de messagerie électronique ouvert.
11. Vérifier si les partitions ont suffisament d'espace libre.
12. Vérifier la taille des fichiers journaux. Il est préférable que la taille des journaux reste en mégaoctets.

10. Comment savoir si vous êtes en train de vous faire pirater ?

Pour savoir si votre machine est compromise ou non, suivez ces étapes. Ce sont celles que j'ai l'habitude de suivre et qui sont utiles dans la plupart des situations.

1. Vérifiez votre machine pour voir si vos performances se sont dégradées ou si votre machine est surutilisée.

   Pour ce faire, utilisez les commandes suivantes :

   vmstat

   Affiche des informations sur la mémoire, le processeur et le disque.

   Exemple : bash# **vmstat** ***1 4*** (où 1 est le délai et 4 le nombre de requêtes).

   mpstat

   Affiche des statistiques sur l'utilisation du processeur, ce qui permet de savoir si votre processeur est surchargé ou non.

   Exemple : bash# **mpstat** ***1 4*** (où 1 est le délai et 4 le nombre de requêtes).

   iostat

   Cette commande affiche des statistiques sur le disque système.

   - Options utiles :
     - `-d` : donne le rapport d'utilisation du périphérique.
     - `-k` : affiche les statistiques en kilooctets par seconde.

   Exemple : bash# **iostat** ***-dk 1 4*** (où 1 est le délai et 4 le nombre de requêtes).

   sar

   Affiche l'ensemble des performances du système.

2. Vérifiez si votre serveur comporte des processus cachés en cours d'exécution.

   ps

   Affiche l'état de tous les processus connus.

lsof

Affiche la liste de tous les fichiers ouverts. Sous Linux, comme tout est considéré comme un fichier, vous serez en mesure de voir pratiquement la totalité de l'activité sur votre système avec cette commande.

3. Utilisez les outils de détection d'intrusion :
   - rkHunter : http://www.rootkit.nl/
   - chkrootkit : http://www.chkrootkit.org/
4. Vérifiez la durée de fonctionnement de votre machine.

   Si la durée de fonctionnement est inférieure à ce qu'elle devrait être, cela peut signifier que quelqu'un est en train d'utiliser les ressources de votre machine. Linux ne « plante » pas ou ne redémarre pas sous les conditions normales car c'est un système d'exploitation stable. Si votre machine a été redémarrée, essayez de découvrir de quoi il retourne vraiment.
5. Déterminez quels sont vos processus inconnus et ce qu'ils font.

   Utilisez des commandes comme celles qui suivent pour démonter les programmes inconnus :

readelf

Cette commande affichera ce que le programme de l'exécutable effectue.

ldd

Cette commande montrera les détails des bibliothèques utilisées par un exécutable.

string

Cette commande affichera les chaînes dans le fichier binaire.

strace

Cette commande affichera les appels système qu'un programme crée lorsqu'il tourne.

11. Méthodologie renforcée

1. Consultez tous les sites liés à la sécurité et tenez-vous au courant. C'est l'une des démarches les plus importantes qu'un administrateur de sécurité ou propriétaire de serveur(s) devrait faire. Les propriétaires de serveur(s) devraient être sensibilisés à la sécurité et à son importance. La formation en sécurité est une partie importante d'un paquetage de sécurité dans son ensemble.
2. Créez une bonne politique de sécurité. Menez des audits de sécurité en fonction de cette politique.
3. Gardez le système d'exploitation à jour en appliquant tous les correctifs.
4. Installez un noyau personnalisé en en retirant tous les services non souhaités et en le *patch*ant avec **grsecurity** ou **openwall**.

5. Désactivez tous les services non souhaités et renforcez les services laissés actifs ; changez les permissions des fichiers et des répertoires pour que la sécurité soit plus rigoureuse.
6. Installez un pare-feu et créez de bons ensembles de règles.
7. Testez et auditez le serveur de façon régulière.
8. Installez un système de détection d'intrusion, un moniteur de journaux, tous les modules de sécurité d'Apache, **bfd**, **faf** et **tmp monitor**. Sécurisez vos partitions.
9. Lancez un bon système de sauvegarde pour récupérer les données en cas d'intrusion, de « plantage » ou autre incident destructeur.
10. Installez un analyseur de journaux et les vérifiez pour rechercher d'éventuels éléments suspects.
11. Installez des scripts pour envoyer du courrier électronique ou déclencher l'envoi de notifications lorsqu'une brèche de sécurité se produit.
12. Après une brèche de sécurité, essayez de découvrir comment, quand et par où la brèche s'est produite. Lorsque vous avez trouvé comment la réparer, notez les détails pour référence ultérieure.

12. Résumé

Concluons à présent en nous intéressant aux principales étapes grâce auxquelles on peut sécuriser un serveur d'hébergement.

1. Déterminez les exigences de fonctionnement et les facteurs de risque qui sont applicables à ce système.
2. Élaborez une politique de sécurité en gardant à l'esprit les données ci-dessus. Obtenir l'approbation et l'autorisation de la direction concernant cette politique de sécurité.
3. Après approbation de la politique, effectuez un audit de sécurité sur tous systèmes existants pour déterminer les vulnérabilités actuelles et remettez un rapport à ce sujet à la direction.

Le rapport devrait également couvrir les méthodes nécessaires pour améliorer la sécurité existante. En voici une liste de contrôle rapide :

- Vulnérabilités des logiciels.
- Mises à niveau du noyau et vulnérabilités.
- Recherchez la présence de chevaux de Troie.
- Lancez **chkrootkit**.
- Vérifiez les ports.
- Recherchez d'éventuels processus cachés.
- Utilisez **audittools** pour vérifier le système.
- Vérifiez les journaux.
- Vérifiez les exécutables et les RPM.
- Vérifiez la présence de relais de messagerie ouverts.

- Recherchez les entrées **cron** malveillantes.
- Vérifiez les répertoires `/dev`, `/tmp` et `/var`.
- Vérifiez si les sauvegardes sont maintenues.
- Recherchez les utilisateurs, groupes, etc. non souhaités sur le système.
- Recherchez et désactivez tous services inutiles.
- Repérez les scripts malveillants.
- Vérifiez les fichiers journaux des requêtes DNS.
- Recherchez les scripts **suid** et les scripts anonymes.
- Vérifiez les scripts valides dans `/tmp`.
- Utilisez les outils de détection d'intrusion.
- Vérifiez les performances du système.
- Vérifiez les performances de la mémoire (lancer **memtest**).

4. Mettez en œuvre la politique de sécurité.

- Corrigez toutes les vulnérabilités connues des logiciels existants, soit en appliquant les correctifs, soit en mettant à niveau les logiciels.
- Mettez en œuvre la sécurité hôte.
  - Protégez vos systèmes avec des mots de passe.
  - Vérifiez les systèmes de fichiers et définissez correctement les permissions et appartenances pour tous les répertoires et fichiers :

    Utilisez **rpm -Va** pour savoir si un **rpm** est modifié.

  - Appliquez les correctifs de sécurité sur les logiciels vulnérables (c'est-à-dire **patch -p1 < patch *fichier***).
  - Supprimez tous les **tty** et les connexions de console inutiles en supprimant la ligne correspondante dans `/etc/securetty`.
  - Vérifiez les journaux système (par exemple `/var/log/messages`, `/var/log/secure`, etc.).
  - Définissez un mot de passe sur le chargeur d'amorçage (lilo et grub le permettent l'un et l'autre).
  - Surveillez le système (nagios ou big brother).

- Mettez en œvre la sécurité réseau.
  - Supprimez tous les utilisateurs et groupes non souhaités.

- Utilisez des scripts de sécurité personnalisés qui enverront des notifications en cas de connexion ssh en tant que *root* ou lors de la création d'un utilisateur avec un **uid** de 0, etc.
- Exigez des mots de passe comportant 16 caractères (ce peut être fait en effectuant des changements dans `login.def`).
- Désactivez les services non souhaités à l'aide de **tcpwrapper** (les services non souhaités peuvent également être désactivés grâce à `xinet.d` ou `xinetd.conf`).
- Mettez en place un délai d'attente d'inactivité pour que les utilisateurs inactifs soient déconnectés au bout d'un certain intervalle de temps.
- Désactivez tous les accès de programmes de console (par exemple **`rm-rf`** `/etc/security/console.app/` ***`<nom du service>`***).
- Autorisez l'option `nospoof` dans `/etc/host.conf`.
- Spécifiez l'ordre dans lequel les noms de domaine devraient être résolus (par exemple **order bind hosts**).
- Verrouillez le fichier `/etc/services` pour que personne ne puisse le modifier.
- Restreignez la connexion *root* directe (décommentez l'option `PermitRootLogin login` dans `sshd_config`).
- Restreignez **su** pour que seuls les utilisateurs sûrs puissent utiliser cette commande (vous pouvez utiliser **pam** ou désactiver les permissions des autres pour l'exécutable **su**).
- Limitez les ressources utilisateur (à l'aide de **pam**, spécifiez les limites pour chaque utilisateur dans `/etc/security/limit.conf`).
- Sécurisez `/tmp` (montez `/tmp` avec **noexec**, **nodev**, **nosuid**).
- Cachez les détails du serveur. Supprimez `/etc/issues` et `/etc/issues.net`.
- Désactivez les fichiers **suid** et **sgid** non souhaités (par exemple ***`find -type -perm -04000 -o perm 02000`***).

  Exemples : **gpasswd**, **wall** et **traceroute**.

- À l'aide de **iptables**, autorisez seulement les **ping**s depuis des emplacements spécifiques (pour que les systèmes de surveillance fonctionnent).
- Prenez des mesures préventives contre les attaques de DoS (déni de service), le « ping de la mort », etc.
- Installez un pare-feu (par exemple apf et iptables) et autorisez seulement le fonctionnement des ports dont la machine a besoin pour ses fonctions normales ; bloquez tous les autres ports pour prévenir les mauvais tours.

  Liens : http://rfxnetworks.com/ et
  http://yolinux.com/TUTORIALS/LinuxTutorialIptablesNetworkGateway.html

- Installez une détection d'intrusion (par exemple, installez tripwire ou aide).

  Liens : http://www.cs.tut.fi/rammer/aide.html et
  http://redhat.com/docs/manuals/linux/RHL-9-Manual/ref-guide/ch-tripwire.html

- Installez sxid pour garder un œil sur les scripts **suid** et **sgid**.

  Lien : http://linux.cudeso.be/linuxdoc/sxid.php

- Restreignez ssh à certaines adresses IP et à des utilisateurs spécifiques (je suggère une authentification par clé à l'aide d'un mot de passe).
- Installez logcheck pour vérifier les journaux.
- Installez tmpwatch pour supprimer les fichiers non utilisés du répertoire `/tmp`.
- Installez et paramétrez portsentry puis configurez-le pour qu'il utilise **iptables** pour bloquer les IP.
- Installez mod_security et mod_dosevasive pour sécuriser apache.
- Supprimez les fichiers sans utilisateur et sans groupe.
- Supprimez les fichiers et dossiers non souhaités dans `htdocs` et désactivez l'indexation des répertoires.
- Recherchez les scripts non souhaités dans `/root`, `/usr/local`, `/var/spool/mbox`.
- Installez BFD et FAF pour apporter une sécurité supplémentaire.
- Désactivez le relais de messagerie ouvert.
- Remettez un rapport d'état à la direction en détaillant toutes les vulnérabilités découvertes et les corrections.

5. Phase de test

Utilisez des outils tels que nessus et nmap pour effectuer un test de pénétration et voir comme votre serveur est bien sécurisé. Faites également un essai sous contraintes.

La sécurité est de la plus haute importance pour un serveur, compromettre la sécurité revient à compromettre le serveur lui même. Par conséquent, une bonne compréhension de la sécurité est un prérequis pour la possession et l'administration de serveurs.

À propos de ce document...

Le document original a été généré à l'aide de la version2002 (1.62) de l'interpréteur LaTeX2HTML (http://www.latex2html.org/).

Je m'appelle Blessen, mais je préfère qu'on m'appelle Bless. Je me suis intéressé à Linux lorsque j'ai rejoint la société d'édition de logiciels Poornam Info Vision Pvt Ltd (http://poornam.com/) connue également sous le nom de Bobcares (http://bobcares.com/). Il m'ont fait découvrir Linux.

Je suis titulaire d'une licence de technologie en informatique du *College of Engineering* de Chengannur (Inde). Diplômé en 2001, je suis entré dans cette société la même année. Au cours de mon travail, je me suis passionné pour la sécurité sous Linux et j'espère m'améliorer dans ce domaine.

Mes passe-temps sont la navigation sur Internet, apprendre les nouvelles technologies et aider les autres. Pendant mon temps libre, je développe également des logiciels *open source* et l'un d'eux est une version allégée de formmail. Le projet s'appelle Smart Mail et est plus sécurisé que formmail.