ROYAUME DU MAROC

ADMINISTRATION DE LA DÉFENSE NATIONALE

DIRECTION GÉNÉRALE DE LA SÉCURITÉ DES SYSTÈMES D'INFORMATION

GUIDE TECHNIQUE

RELATIF À LA SÉCURITÉ DU SERVEUR LINUX

INFORMATIONS

AVERTISSEMENT
Destiné à vous assister dans l'adoption d'une démarche cohérente et homogène pour la mise en conformité de la sécurité de vos systèmes d'information avec les règles de sécurité édictées par la Directive Nationale de la Sécurité des Systèmes d'information (DNSSI), ce guide élaboré par la DGSSI traite la démarche de sécurisation des systèmes GNU/Linux. Il est destiné à évoluer avec les usages, mais aussi avec vos contributions et retours d'expérience. Les recommandations citées dans ce guide sont livrées en l'état et adaptées aux menaces au jour de leur publication. Au regard de la diversité des systèmes d'information, la DGSSI ne peut garantir que ces informations puissent être reprises sans adaptation sur les systèmes d'information cibles. Dans tous les cas, la pertinence de l'implémentation des éléments proposés par la DGSSI doit être soumise, au préalable, à la validation du Responsable de la Sécurité des Systèmes d'Information (RSSI) et de l'administrateur du système concerné.

PERSONNES AYANT CONTRIBUÉ À LA RÉDACTION DE CE DOCUMENT :

Rédigé par	Version	Date
DGSSI	1.0	12/12/2014

ÉVOLUTION DU DOCUMENT :

Version	Date	Nature des modifications
1.0	12/12/2014	Version initiale

PUBLIC CONCERNÉ PAR CE DOCUMENT :

Secrétariat Général /Direction Générale	
Direction SI	
RSSI	**X**
Maîtrise d'ouvrage	
Administrateur systèmes et réseaux	**X**
Service des Ressources Humaines	
Service des Moyens Généraux	
Utilisateur	

POUR TOUTE REMARQUE :

Contact	Email
DGSSI	contact@dgssi.gov.ma

Table des matières

Introduction

Le présent document spécifie les techniques de durcissement d'un serveur linux basé sur l'état de l'art. Il définit les bonnes pratiques que l'administrateur système doit implémenter pour renforcer la sécurité matérielle et logicielle du serveur.

Les aspects traités dans ce guide sont communs à tous les systèmes GNU/LINUX indépendamment de la distribution utilisée (Debian, RedHat, etc.) et de la fonction remplie (Messagerie, Web, DNS, etc.).

Les recommandations présentées dans ce document ne sauraient donc aucunement avoir un caractère exhaustif. Il s'agit simplement d'énoncer les principaux axes de durcissement à explorer afin de protéger un serveur linux contre les activités malveillantes.

Ce guide est structuré de la manière suivante :

Section 1 : « les risques » , traite d'une manière générale les risques relatifs à la sécurité des serveurs Linux.

Section 2 : « Sécurité relative à un système GNU/Linux» , contient les techniques de base permettant de sécuriser un serveur linux dès l'installation du système à savoir :le partitionnement du disque, la réduction de la surface d'attaque par l'application du principe de minimisation, ainsi que l'application des mises à jour et des correctifs.

Section 3 : « Sécurité locale du système » , définit l'ensemble des bonnes pratiques que l'administrateur doit implémenter afin de garantir une bonne gestion des comptes, des utilisateurs et des mots de passe.

Section 4 : « Gestion des accès » , examine la manière de sécuriser les accès aux ressources du serveur Linux à savoir : les accès physique au système via sa console, les permissions sur les fichiers et les accès distants au serveur. Aussi elle traite la manière de sécuriser des connexions et d'implémenter le filtrage pour éviter les intrusions réseaux.

Section 5 : « Sauvegarde et restauration » examine les bonnes pratiques à prendre en compte pour une bonne politique de sauvegarde des données.

Section 6 : « Chiffrement » , présente l'importance du chiffrement des données et des communications afin de sécuriser l'information.

Section 7 : « Supervision et audit » , examine l'importance de la supervision régulière des journaux, la vérification de l'intégrité des données, et l'évaluation de la sécurité par des audits.

1 Les Risques

Dans la majorité des cas, les serveurs disposent d'une grande partie de données sensibles et vitales des organismes. Une fois compromises, ces informations deviennent exposées au vol et à la manipulation par des personnes malveillantes. En effet, les attaques auxquelles ces serveurs peuvent faire face sont diverses à savoir : déni de service, altération ou perte de données, contamination par des virus, interception des communications, etc.

Parmi les menaces qui peuvent nuire à la sécurité des serveurs informatiques on peut citer :

* Les menaces physiques potentielles liées à la sécurité physique du serveur. Il s'agit des événements imprévisibles comme les pannes, les accidents ou encore les atteintes intentionnelles sur les matériels. Par exemple :
  - Dégâts des eaux (inondation, humidité) ;
  - Feu (Accidentel ou criminel) ;
  - Electricité (surtension, baisse de tension, coupure du courant) ;
  - Température ambiante (dysfonctionnement de la climatisation) ;
  - Intrusions physiques (circulation de personnes non autorisées dans les locaux, vol) ;
  - Etc.

* Les menaces liées à la sécurité des systèmes d'exploitation générées par l'exploitation possible des vulnérabilités et failles éventuellement présentes sur le système, notamment :
  - Services inutilisés et ports ouverts ;
  - Services sans correctifs ;
  - Mauvaise gestion des privilèges et des accès aux ressources du système ;
  - Mauvaise gestion des mots de passe ;
  - Applications vulnérables ;
  - Etc.

Ainsi, il est primordial de mettre en place les contrôles nécessaires (physiques, administratifs et techniques) permettant de se prémunir contre les risques et de garantir la confidentialité, l'intégrité et la disponibilité de l'information.

2 Sécurité relative à un système GNU/Linux

2.1 Installation

Le durcissement d'un serveur Linux commence dès la phase de l'installation du système. Généralement, Il faut éviter de procéder à une installation par défaut. Il faut tout d'abord décider quelle en sera l'utilisation (serveur Web, DNS, messagerie, etc.) avant de déterminer des règles à mettre en place pour le sécuriser. Ci-après les principaux aspects à prendre en considération lors de l'installation du système linux.

Partitionnement du disque : Le partitionnement est une étape clef de l'installation de GNU/Linux et de la prise en compte des supports de stockage de données. Il est important de bien choisir le partitionnement à adopter en vue d'obtenir un niveau de sécurité plus élevé.
La méthode de partitionnement varie en fonction de l'utilisation du serveur. Tout dépendra des services à offrir par ce serveur, de l'espace disque disponible et des applicatifs nécessaires à son fonctionnement. Généralement, il convient de procéder au partitionnement suivant :

- Les arborescences de répertoires modifiables par un utilisateur, telles que */home*, */tmp* et */var/tmp*, doivent être sur des partitions distinctes ;
- Toute partition qui peut fluctuer, par exemple */var* (surtout */var/log*) devrait être également sur une partition distincte ;
- Toute partition qui peut contenir des installations des logiciels ne faisant pas partie de la distribution (des applications tierces) devrait être sur une partition distincte (par exemple */opt*) ;
- Le répertoire */boot* qui contient les fichiers indispensables au démarrage peut être mis dans une partition à part.

R 1	Implémenter un schéma de partitionnement permettant de : – Eviter la saturation ; – Simplifier la sauvegarde ; – Appliquer les options de montage.

En créant des partitions différentes pour les répertoires cités ci-dessus, les données peuvent être séparées et regroupées. Et dans le cas où un incident inattendu se produit, seules les données de la partition concernée seront endommagées.

Droits lors du montage des partitions : Les partitions peuvent être montées avec certaines options (par exemple : ro, nodev, noexec, nosuid, etc.) qui limitent les droits attribués aux fichiers systèmes. Les options de montage sont définies

dans le fichier '/etc/fstab'. Elles peuvent être la cible de certains comportements malveillants en cas de mauvaise configuration. Il est recommandé à cet effet d'appliquer les options de montage appropriées au niveau du fichier /etc/fstab dans les conditions suivantes :

- Les supports de stockage amovibles doivent être montés en *nodev*, *noexec*, *nosuid*, pour éviter à tout programme d'être exécuté à partir du périphérique externe.
- Les partitions de stockage temporaire comme */tmp*, */var/tmp* et */dev/shm* doivent être montés en *nodev, noexec, nosuid ;*
- Monter le répertoire */home* de préférence en nodev, noexec, nosuid ;
- Pour éviter toute modification non autorisée sur les fichiers du démarrage, la partition */boot* ne doit pas être montée par défaut.

R 2	Implémenter les options de montages nécessaires et suffisantes adaptées au contexte d'emploi.

Ajout et suppression des composants logiciels : Lors de l'installation d'un serveur linux, il est conseillé d'opter pour les composants logiciels appropriés sur la base de la fonction du serveur. Les outils inutiles installés sur la machine pourraient être exploités par des personnes malveillantes pour compromettre le système.

A titre d'exemple, la présence d'outils de développement (compilateurs) ou de langages interprétés pourrait faciliter la tâche d'un attaquant dans la compilation et l'exécution de codes malveillants sur le serveur. En général, il est conseillé d'éviter l'installation par défaut des groupes de paquetage 'Software Development' ainsi que ceux relatifs au 'web server'.

De même, il est généralement conseillé d'opter pour l'utilisation de l'invite de commande au lieu des outils graphiques. En effet, en plus des failles de X Windows, l'environnement graphique est souvent lourd en termes de ressources matérielles.

R 3	Lors de l'installation et quand le système le permet, il faut décocher les paquetages associés aux applications inutiles pour le serveur : – Compilateurs ; – Logiciels de développement ; – Serveurs non nécessaires (exemple : le Serveur X) ; – Tout environnement graphique.

R 4	Supprimer les programmes inutiles pour le contexte d'utilisation du serveur notamment les services générant une grande quantité de dépendances.

Ne pas se connecter à Internet avant la fin complète de la configuration : Certains services peuvent avoir des vulnérabilités non corrigées dans les paquets

utilisés pour l'installation (exemple : utilisation d'anciennes versions de CD d'installation). Dans ce cas, si le système est connecté à internet, il sera exposé à des attaques avant même la fin de l'installation. Il est recommandé à cet effet de consulter la rubrique support de la distribution utilisée, de télécharger les derniers paquetages corrigeant les failles de sécurité et de les appliquer avant de se connecter à internet.
Dans le cas où l'installation nécessite l'accès à internet, il est recommandé de mettre en place des règles de pare-feu pour limiter l'accès au système pendant l'installation.

R 5	Protéger le serveur par un Pare-feu si l'utilisation de l'internet s'avère nécessaire lors de l'installation du système linux.

2.2 Elimination des services inutiles

L'élimination des services inutiles en écoute sur le réseau permet de réduire la surface d'attaque et d'améliorer la sécurité globale du système. En outre cela offre plus d'espace en termes de mémoire et une optimisation des performances. Pour ce faire, il convient de lister tout d'abord les services (systèmes et réseaux) et les programmes installés :

Identifier les processus : Pour afficher tous les processus qui tournent sur la machine en temps réel, utiliser la commande :

```
#ps - aux
```

Identifier les ports réseaux utilisés : Afin d'identifier les différents ports ouverts, taper la commande :

```
#netstat -a
```

Identifier les services : Pour lister les services qui se lancent automatiquement avec le démarrage du système, utiliser la commande :

```
#chkconfig –list
```

L'équivalent de cette commande sous Debian/Ubuntu est :

```
#sysv-rc-conf –list
```

Après l'installation, il convient d'identifier les services et les programmes installés pour désactiver ceux qui sont inutiles :
Arrêter les services en exécution en utilisant la commande :

```
#service SERVICE stop
```

Arrêter le démarrage automatique des services en utilisant la commande :

```
# chkconfig –levels 2345 SERVICE off
ou
# systemctl disable SERVICE.service
```

R 6	Désactiver les services inutiles en écoute sur le réseau.

2.3 Patch et mise à niveau du système

L'application régulière des mises à jour est l'une des actions importantes pour sécuriser le système.
L'administrateur du serveur doit compléter son installation en téléchargeant au fur et à mesure les mises à jour les plus récentes de chacun des composants du système d'exploitation et de les appliquer. Un retard dans l'application d'un correctif est très souvent à l'origine de la compromission de la machine.
De même il convient de s'assurer que les applications installées sur le système sont à jour et de procéder à l'application des correctifs dans le cas échéant.

R 7	Une mise à jour régulière du système et des applications installées dessus est indispensable : – Créer, documenter et mettre en place une procédure de patch ; – Identifier régulièrement les vulnérabilités et les patches manquants ; – Installer les correctifs et les mises à jour à partir du site web officiel de la distribution utilisée ; – Si des correctifs ne sont pas encore disponibles, désactiver les services qui sont en relation avec la vulnérabilité si cela est possible.

2.4 Configuration des paramètres réseaux

Certains paramètres de la configuration réseau IP du système doivent être modifiés de manière à renforcer sa robustesse vis-à-vis des attaques potentielles. Comme c'est souvent le cas, les paramètres par défaut permettent de prendre nativement en charge beaucoup de fonctionnalités. Pour une configuration appropriée des paramétres réseaux, il convient de procéder comme suit :
Interface réseau : Il est fortement recommandé de désactiver toute interface réseau non utilisée. Lorsque le serveur comporte plusieurs interfaces réseau, il est conseillé de spécialiser celles-ci pour dissocier les différents type de flus métiers et les flux d'administration. Il convient alors d'imposer que les services soient en écoute uniquement sur les interfaces adaptées.
IPv6 : Il est fortement recommandé de désactiver le support d'IPv6 s'il n'est pas encore utilisé. Ceci peut être fait à partir du fichier /etc/sysctl.conf en ajoutant ces lignes en fin du fichier :

```
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
net.ipv6.conf.lo.disable_ipv6=1
```

Sécurisation du réseau pendant l'amorçage : Durant l'amorçage, le système lit et applique des paramètres du KERNEL trouvés dans le fichier /etc/sysctl.conf.

Il convient de le configurer afin de sécuriser quelques options du réseau au niveau du noyau. Cette configuration sera appliquée sur l'ensemble des interfaces activées. Voici quelques paramètres à configurer au niveau du fichier /etc/sysctl.conf :

```
# Ignorer les broadcasts ICMP
Net.ipv4.icmp.echo_ignore_broadcasts = 1
# Ignorer les erreurs ICMP erronées
Net.ipv4.icmp_ignore_bogus_error_responses = 1
# Ne pas accepter les redirections ICMP (empêche les attaques main in the middle)
Net.ipv4.conf.all.accept_redirects = 0
# N'accepter les redirections ICMP que pour les passerelles de la liste des passerelles par
# défaut (activé par défaut)
Net.ipv4.conf.all.secure_redirects = 1
#Ne pas accepter les redirections ICMP (ce n'est pas un routeur)
Net.ipv4.conf.all.send_redirects = 0
# Ne pas faire suivre les paquets IP (ce n'est pas un routeur)
# Remarque : assurez-vous que /etc/network/options contient « ip_forward=no »
Net.ipv4.conf.all.forwarding = 0
# Activer les TCP Syn Cookies : Remarque : assurez-vous que /etc/network/options contient
# « syncookies=yes »
Net.ipv4.tcp_syncookies = 1
# Enregistrer les paquets usurpés
Net.ipv4.conf.all.log_martians = 1
# Activer la vérification d'adresse source pour toutes les interfaces pour empêcher certaines
# attaques par usurpation
# Remarque : assurez-vous que /etc/network/options contient « spoofprotect=yes »
Net.ipv4.conf.all.rp_filter = 1
# Ne pas accepter les paquets de routage source IP (ce n'est pas un routeur)
Net.ipv4.conf.all.accept_source_route = 0
# Désactiver les touches magiques (magic-sysrq key)
Kernel.sysrq =0
# Diminuer la valeur de temps par défaut de « tcp_fin_timeout connection »
Net.ipv4.tcp_fin_timeout = 15
# Diminuer la valeur de temps par défaut de « tcp_keepalive_ time connection »
Net.ipv4.tcp_keepalive_time=1880
#Désactiver tcp_windows_scaling
Net.ipv4.tcp_windows_scaling=0
#Désactiver tcp_sack
Net.ipv4.tcp_sack=0
#Désactiver tcp_timestamps
Net.ipv4.tcp_timestamps=0
```

Note : il faut redémarrer le service réseau après chaque modification sur le fichier /etc/sysctl.conf

R 8	Désactiver toutes les interfaces inutiles au démarrage et n'activer que celles dont le besoin se pose.

R 9	Spécialiser les interfaces réseaux pour dissocier les flux métiers et les flux d'administration.

R 10	Désactiver le support d'IPv6 s'il n'est pas encore utilisé.

R 11	Configurer les paramètres du kernel afin de mettre en place toutes les options réseaux nécessaires.

3 Sécurité locale du système

3.1 Gestion des comptes et des utilisateurs

Linux est un système d'exploitation multi-utilisateurs, tous les utilisateurs doivent posséder un compte usager pour pouvoir y accéder. De plus ils doivent être identifiés afin d'assurer la confidentialité.

La gestion des comptes représente une partie primordiale dans la sécurité des systèmes. Avec une mauvaise gestion des utilisateurs et de leurs droits, de nombreux systèmes pourraient être corrompus. Il est donc crucial de mettre en place les techniques appropriées de gestion des comptes utilisateurs pour protéger l'accès au système.

Le super-utilisateur (root) : C'est le compte le plus important sur le système, son UID égal à 0. Ce compte dispose des droits d'accès administratifs. Il est recommandé de désactiver le compte root entièrement et d'ajouter des comptes d'administration nominatifs qui peuvent effectuer les tâches d'administrations en utilisant la commande sudo suivie d'une authentification.

Pour des distributions comme Ubuntu/Debian l'accès direct au compte root est désactivé par défaut, l'utilisateur devrait utiliser la commande sudo pour effectuer toute tâche administrative. En revanche, pour des distributions comme Fedora/Redhat, il est toujours possible de s'authentifier en tant que root. Il convient à cet effet de créer un autre compte et d'ajouter cet utilisateur au groupe wheel par la commande :

```
#usermod -G wheel Nom_du_compte
```

Ensuite décommenter dans le fichier /etc/sudoers la ligne qui contient :

```
%wheel ALL=(ALL) ALL
```

Finalement, il faut se connecter avec le nouveau compte et désactiver le compte root à l'aide de la commande sudo :

```
# su - Nom_du_compte
$ sudo usermode -L root
```

Les comptes systèmes : On trouve sur le système une série de comptes génériques (par exemple : bin, daemon, sync, apache, etc.). Les UIDs compris entre 1 et 499 sont généralement utilisés pour ces comptes. Il convient de bloquer l'exécution du Shell à partir de ces comptes. Pour ce faire, il faut tout d'abord lister les

comptes avec leurs UIDs ainsi que leurs Shells en utilisant la commande :

```
# awk -F : 'print $1 " :" $3 " :" $7' /etc/passwd
```

Ensuite, identifier ceux possédant un UID inférieur à 500 et différent de 0 et puis désactiver leur accès au Shell :

```
Sous RHEL/Fedora : # usermod -s /sbin/nologin Nom_du_compte
Sous Ubuntu/Debian : # usermod -s /bin/false Nom_du_compte
```

Les comptes ordinaires : Ce sont les comptes permettant à des utilisateurs standards de se connecter au système. L'UID de ces comptes sera un nombre supérieur ou égal à 500. Il convient de vérifier périodiquement que tous les comptes utilisateurs possèdent un mot de passe. Pour les comptes non utilisés, il convient de les désactiver à l'aide de la commande :

```
#usermod –L Nom_du_compte
```

D'une manière générale, il est fortement recommandé de :

R 12	Désactiver tous les comptes non utilisés.

R 13	S'assurer que tous les comptes possèdent un mot de passe non vide.

R 14	Supprimer tous les comptes non root avec le UID = 0, puisqu'avec un tel UID, le propriétaire du compte a les mêmes droits que le compte root.

R 15	Désactiver l'exécution du Shell pour tous les comptes non Root et ne l'activer qu'en cas de besoin justifié.

R 16	Eviter l'utilisation des comptes ayant les droits root dans des activités autre que l'administration du système.

R 17	Mettre à la disposition des utilisateurs des comptes nominatifs et uniques et attribuer les droits d'accès selon le principe du moindre privilège.

3.2 Sécurité des mots de passe

Une bonne gestion des mots de passe permet un accès sécurisé au système. Le mot de passe ne doit pas être vide et doit systématiquement respecter une politique de complexité.

Sous Linux, le système est configuré de manière à ce que l'algorithme MD5 et les mots de passe masqués soient utilisés. Il est fortement recommandé de ne

pas modifier ces paramètres.L'utilisation de l'option des mots de passe masqués permet de stocker ces derniers dans le fichier */etc/shadow* lisible uniquement par le root, et non pas dans le fichier */etc/passwd* accessible en lecture pour tous les utilisateurs.

Pour fixer le nombre minimum de caractères que le mot de passe doit contenir ainsi que la période de sa validité. Ajouter au fichier */etc/login.defs* :

```
PASS_MIN_LEN 12
PASS_MAX_DAYS 90
```

Le cryptage MD5 impose des mots de passe de plus de 8 caractères, contrairement à DES (data encryption standard), ancien format de chiffrement qui limite les mots de passe à huit caractères et donc génère des mots de passe faibles.

Utilisation de l'algorithme SHA-512 pour le hachage : Lorsque cela est possible et pour certaines distributions de linux, il convient de renforcer davantage la sécurité des mots de passe en utilisant le SHA-512 au lieu de MD5. Pour ce faire ajouter au fichier */etc/login.defs* :

```
MD5_CRYPT_ENAB no
ENCRYPT_METHOD SHA512
```

Sécurité des mots de passe dans PAM : L'administrateur peut améliorer l'authentification des utilisateurs sur le système par la configuration de PAM (Pluggable Authentication Modules). Le module *pam_cracklib* permet d'accepter ou de rejeter un mot de passe si celui-ci se trouve dans un dictionnaire (/usr/lib/cracklib.pwd). Il permet aussi de vérifier que le mot de passe n'est pas réutilisé. Pour paramétrer ce module ajouter la ligne suivante dans le fichier :

- Sous RHEL/ Fedora :/etc/pam.d/system-auth
- Sous Ubuntu /Debian : /etc/pam.d/commonpassword

```
password required pam_cracklib.so retry=3 minlen=12 difok=3 lcredit=-1
ucredit=-2 dcredit=-2 ocredit=-1
```

retry : Le nombre de tentative ; *minlen :* La longueur imposée ; *difok :* Le nombre de caractères existant dans l'ancien mot de passe et que l'on ne peut pas retrouver dans le nouveau ;
lcredit, *ucredit*, *dcredit* et *ocredit* correspondent respectivement au nombre de caractères minuscules, majuscules, numériques et autres. La valeur négative signifie le nombre minimum de caractère requit. La valeur positive en revanche signifie le nombre maximum de caractères. L'activation du module PAM *cracklib* oblige les utilisateurs à utiliser des mots de passe forts.

R 18	Définir les règles de choix des mots de passe. Le mot de passe doit : – Comprendre au moins 12 caractères ; – Ne pas contenir des informations personnelles (le nom d'utilisateur, la date de naissance, le nom de la société, etc.) ; – Ne pas contenir des mots du dictionnaire ; – Etre complètement différent des mots de passe précédents ; – Inclure une combinaison de lettres majuscules et minuscules, des caractères spéciaux et des chiffres ; – Doit être changé régulièrement (entre 60 et 90 jours).

R 19	Utiliser des mots de passe différents pour les comptes d'administration pour chaque hôte.

Note : En plus de l'utilisation du module PAM cracklib pour la création des mots de passe forts, il est toujours conseillé de recourir à des programmes de craquage de mot de passe pour s'assurer au maximum de son efficacité et sa résistance aux attaques contre les mots de passes.

4 Gestion des accès

4.1 Accès physique au système

La sécurité des accès physiques au système est une étape primordiale pour protéger la configuration physique d'un serveur Linux..

La sécurité des accès physiques au système repose en premier lieu sur l'emplacement et l'environnement physique où est installé le serveur. Cela permet d'empêcher l'accès non autorisé ainsi que les dommages de tout genre pouvant affecter le serveur..
Les mesures suivantes permettent de contrôler les accès au système : .

BIOS : C'est le premier programme qui s'exécute au démarrage du système, il permet le contrôle des éléments matériels. Il convient de sécuriser l'accès au BIOS par un mot de passe afin d'empêcher toute modification de ses paramètres (par exemple : changer la configuration du BIOS de manière à démarrer à partir d'un CD-ROM ou une clé USB).


Note : Les méthodes utilisées pour sécuriser le Bios par mot de passe varient selon les fabricants des serveurs. Il est conseillé de consulter le manuel du serveur pour obtenir les instructions appropriées.

GRUB : Il est recommandé de protéger la configuration du chargeur de démarrage par un mot de passe pour empêcher toute tentative de connexion avec le mode single ou bien le changement des paramètres pendant le démarrage. Pour ce faire, ajouter une directive de mot de passe dans le fichier de configuration du GRUB :

Tout d'abord il faut générer un hachage MD5 du mot de passe (Les commandes doivent être adaptées à la distribution de linux et à la version du GRUB utilisées). Par exemple sous Ubuntu/Debian taper les commandes :

```
#grub
> md5crypt
```

Puis éditer le fichier de configuration du GRUB et ajouter la ligne suivante en dessous de la ligne timeout :

```
Password – md5 passwd_hashé
```

- Sous RHEL/ Fedora : /boot/grub/grub.conf.
- Sous Ubuntu /Debian : /boot/grub/menu.lst (à partir de la version 9.10 les modifications doivent être faites au niveau du fichier /etc/default/grub).

Note : Bien que GRUB accepte également les mots de passe en texte clair, il est recommandé d'utiliser un hachage md5 ou sha-512 pour une meilleure sécurité.

Authentification pour le single mode : Il est recommandé d'activer l'authentification pour le single mode, pour cela éditer le fichier */etc/inittab* et ajouter la ligne suivante :

```
su :S :wait :/sbin/sulogin
```

Fermeture des sessions du Shell inactif : Il est recommandé de fermer les sessions Shell au bout d'un certain temps d'inactivité. Par exemple certains Shells Linux offrent la possibilité de définir la variable d'environnement TMOUT qui permet de déconnecter automatiquement les utilisateurs après une période d'inactivité. Afin d'éviter la modification de cette variable par les utilisateurs, il est d'usage de la définir dans le fichier */etc/profil*e et de lui appliquer la restriction "*ro*"..

```
if [ "$EUID" = "0" ]||["USER" = "root" ] ; then
TMOUT=900
else
TMOUT=3600
fi
readonly TMOUT
export TMOUT
```

Verrouiller l'accès à la console : *Vlock* (Virtual console lock program) est un programme qui permet de verrouiller le terminal et de demander un mot de passe pour être débloqué. Le paquet Vlock est présent dans les dépôts des principales distributions GNU/Linux. Pour l'installer utiliser la commande :

```
Sous RHEL/ Fedora : yum install vlock
Sous debian/ubuntu : apt-get install vlock
```

Pour verrouiller la session courante, utiliser la commande :

```
vlock -c
```

L'effet de la combinaison CTR-ALT-DEL : Dans la plupart des distributions linux l'utilisation de la combinaison ctrl-alt-del conduit au redémarrage du système. Il convient de désactiver l'effet de cette option surtout pour les serveurs de production. Pour ce faire, dé-commenter la ligne contenant ctrl-alt-del dans le fichier */etc/inittab* comme suit :

```
Trap CTRL-ALT-DELETE
ca : :ctrlaltdel :/sbin/shutdown -t3 -r now
```

Pour les versions les plus récentes de linux, cette configuration doit être faite au niveau du fichier */etc/init/control-alt-delete.conf*.

Blocage des Supports USB : Il est recommandé de désactiver le support USB au niveau du serveur sauf en cas de besoin. Pour cela, il est possible de modifier les paramètres du GRUB en ajoutant *'nousb'* dans le fichier de configuration du GRUB et en redémarrant le système par la suite :

- Sous RHEL/Fedora : */boot/grub/grub.conf*
- Sous ubuntu /Debian : */boot/grub/grub.cfg*

Pour sécuriser l'accès physique au système il convient de :

R 20	Placer le serveur dans une salle dédiée (salle serveur, salle machine, etc.

R 21	Définir un mot de passe pour le Bios.

R 22	Définir un mot de passe pour le chargeur de démarrage GRUB.

R 23	Activer l'authentification pour le single mode.

R 24	Verrouiller le Shell après un certain temps d'inactivité.

R 25	Désactiver l'effet de la combinaison CTR-ALT-DEL.

R 26	Bloquer le support du Storage USB.

4.2 Droits d'accès aux fichiers

Les droits d'accès aux fichiers constituent un élément essentiel du système linux. En effet, ils permettent de définir des droits différents (lecture, écriture, exécution) sur un même fichier selon la catégorie d'utilisateurs (propriétaire, groupe, autres). Ci-après des restrictions d'autorisation importantes qui doivent être vérifiées régulièrement.

Les fichiers /etc/passwd, /etc/shadow, /etc/group, /etc/gshadow : Ce sont les fichiers de configuration qui contiennent les informations concernant les utilisateurs et les mots de passe. Normalement linux attribue les droits suivants par défaut pour ces fichiers :

- rw-r–r– pour /etc/passwd et /etc/group
- r——— pour /etc/shadow et /etc/gshadow

Vu l'importance de ces fichiers, il convient de faire une vérification des droits de ceux-ci. Si les paramètres par défaut sont altérés, une investigation doit être menée pour préciser la source de ce changement et rétablir les droits susmentionnés en effectuant les commandes suivantes :

```
# cd /etc
# chown root :root passwd shadow group gshadow
# chmod 644 passwd group
# chmod 400 shadow gshadow
```

Le droit de *Sticky bit* : Le droit Sticky Bit (appelé aussi bit collant) est alloué à la catégorie "autres" d'un répertoire. Il permet d'interdire à tout utilisateur (sauf le root) de supprimer un fichier dont il n'est pas le propriétaire, quelque soient les droits du répertoire. Pour mettre en place cette option, il faut tout d'abord lister les répertoires ayant le droit d'écriture mais pas le Sticky bit. Ensuite, identifier les répertoires concernés et y ajouter le Sticky bit par la commande :

```
# find / -type d \( -perm -0002 -a ! -perm -1000 \) -print
# chmod +t /rep
```

Où rep représente le répertoire auquel vous désirez ajouter le Sticky bit.

Les fichiers avec droit d'écriture : Il convient de limiter les droits d'écriture sur les fichiers au stricte nécessaire. Tout d'abord il faut lister tous les fichiers ayant le droit d'écriture, puis enlever ce droit pour ceux dont le besoin n'est pas justifié :

```
# find / -type f -perm -0002 –print
# chmod o-w fichier
```

Les fichiers ayant le SUID et le GUID : Le droit SUID permet d'allouer temporairement à un utilisateur les droits du propriétaire du fichier, durant son exécution. De même, le droit GUID est similaire au droit SUID sauf qu'il donne à un utilisateur les droits du groupe auquel appartient le propriétaire du fichier et non pas les droits du propriétaire.
Il est fortement recommandé d'enlever ces droits sauf en cas de besoin majeur. Pour cela il faut identifier les fichiers ayant le SUID et le GUID, puis appliquer la commande suivante sur les fichiers trouvés :

```
# find / \( -perm -4000 -o -perm -2000 \) -type f -print
# chmod -s fichier
```

La valeur UMASK : Le masque de protection de fichier permet de définir les droits par défaut de tout fichier créé. Il convient de fixer la valeur de « UMASK » à 027. Ce qui signifie que tout fichier crée aura comme droits : *rwxr-x—*

R 27	Vérifier les permissions sur les fichiers /etc/passwd, /etc/shadow, /etc/group et /etc/gshadow.

R 28	Limiter les droits d'écriture sur les fichiers au strict minimum.

R 29	Identifier les fichiers ayant le SUID et le GUID, enlever ces droits si le besoin n'est pas justifié.

R 30	Fixer la valeur de UMASK à 027 pour protéger les fichiers.

R 31	Trouver et supprimer les fichiers sans propriétaires.

4.3 Accès à distance

L'administration des serveurs nécessite généralement une connexion à distance. Pour sécuriser les échanges entre la machine cliente et le serveur, il est recommandé d'utiliser le protocole SSH (Secure Shell) qui permet d'établir des connexions chiffrées. Afin d'augmenter le niveau de sécurité de l'utilisation de SSH, Il convient de configurer les paramètres ci-dessous au niveau du fichier */etc/ssh/sshd_config* :

Utiliser la version 2 du protocole SSH : Il existe deux versions différentes du protocole SSH (la version 1 et la version 2). Il est recommandé d'utiliser la version 2 du protocole puisque la 1ère version expose le serveur à une vulnérabilité qui permet à un attaquant d'insérer des données dans le flux de communication. Pour paramétrer l'utilisation de la deuxième version de SSH, ajouter au fichier :

```
Protocol 2
```

Limiter les utilisateurs qui peuvent utiliser l'accès au serveur via SSH : Pour cela, ajouter la ligne suivante :

```
AllowUsers USER1 USER2
```

Limiter l'accès par adresse IP : Il est recommandé de limiter l'accès SSH à des adresses IP spécifiques (par exemple, contrôler au niveau du Iptables) :

```
-A INPUT -p tcp -m tcp –dport 22 –source IPADDRESS -j ACCEPT
```

Désactiver l'authentification en tant que root :
Il convient de ne pas autoriser la connexion distante via SSH pour le compte root. Il est recommandé de se connecter tout d'abords par SSH en tant que simple utilisateur, puis utiliser la commande **su** pour se connecter en tant que root. Pour désactiver l'authentification en tant que root ajouter cette ligne au fichier de configuration */etc/ssh/sshd_config*

```
PermitRootLogin no
```

Authentification par clé RSA : Il convient aussi d'implémenter une authentification à base des clés publiques (RSA).

R 32	Utiliser une version récente de SSH pour les accès distants au serveur, afin d'assurer un échange de données sécurisé.

R 33	Paramétrer SSH pour augmenter son niveau de sécurité.

R 34	Surveiller les connexions en vérifiant régulièrement le fichier */var/log/auth.log*.

Note : Il est important de redémarrer le service SSH après ces modifications.

4.4 Mise en place d'un système de filtrage : Iptables

Pour contrôler les privilèges d'accès et limiter l'utilisation des ressources du réseau, il est important de mettre en place un mécanisme de filtrage. En effet, l'utilisation des filtres permet de contrôler le trafic entrant et le trafic sortant.

Le noyau Linux offre le module Netfilter qui intercepte et manipule les paquets IP avant et après le routage. Il est possible de le configurer via la commande iptables. La première étape lors de l'utilisation d'iptables est de démarrer le service iptables par la commande :

```
service iptables start
```

Pour que iptables soit lancé par défaut dès que le système est démarré, Il faut changer le statut du niveau d'exécution sur le service à l'aide de chkconfig :

```
chkconfig –level 345 iptables on
```

Une configuration optimale du pare-feu se base généralement sur une règle de refus par défaut. En effet, il est recommandé de bloquer tous les paquets entrants et sortants sur une passerelle réseau et de n'autoriser que les paquets spécifiques selon les cas.
Pour réinitialiser la configuration de Iptables si elle existe, utiliser la commande :

```
# iptables -F
# iptables -X
```

Pour bloquer tout trafic entrant ou sortant de la machine, utiliser les commandes suivantes :

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

Pour autoriser le trafic entrant d'une connexion déjà établie, taper la commande :

```
# iptables -A INPUT -m state –state ESTABLISHED, RELATED -j ACCEPT
```

Pour interdire aux paquets externes d'utiliser l'adresse locale taper les commandes suivantes :

```
#iptables -A INPUT -i eth0 -s $LOOP -j DROP
#iptables -A FORWARD -i eth0 -s $LOOP -j DROP
#iptables -A INPUT -i eth0 -d $LOOP -j DROP
#iptables -A FORWARD -i eth0 -d $LOOP -j DROP
```

Exemple : pour autoriser les requêtes DNS, taper la commande :

```
#iptables -A OUTPUT -p udp -o eth0 –dport 53 –sport 1024 :65535 -j ACCEPT
```

Pour créer une chaîne pour journaliser tout le trafic rejeté :

```
# iptables -N LOGnDROP
# iptables -A LOGnDROP -j LOG –log-prefix 'DROP_LOG : '
# iptables -A LOGnDROP -j DROP
```

R 35	Appliquer une défense en profondeur en implémentant un pare-feu local : – Une configuration correcte de pare-feu se base sur une règle de refus par défaut ; – Les connexions entrantes ne doivent pas être autorisées que pour des services locaux par des machines autorisées ; – Les connexions sortantes ne doivent pas être autorisées que pour les services utilisés par le système (DNS, web, email, etc.) ; – Interdire la règle forward pour toutes les connexions (si le par feu ne protège pas d'autres systèmes) ; – Utiliser de préférence des actions telles que DROP plutôt que REJECT (qui envoie à l'émetteur un message indiquant que le port n'est pas ouvert) ; – Journaliser le trafic rejeté.

4.5 Contrôle des services réseaux : TCPwrapper

TCPwrapper permet de contrôler l'accès aux démons des services par hosts. Ceci est fait grâce aux deux fichiers de configuration à savoir :
• */etc/hosts.deny* (contient une liste des hôtes dont l'accès est interdit) ;
• */etc/hosts.allow* (contient une liste des hôtes dont l'accès est permit).

Le principe recommandé lors d'un filtrage et de tout rejeter et n'accepter que ce qui est utile. Donc ajouter au fichier */etc/hosts.deny :*

```
ALL :ALL
```

Ensuite autoriser seul le trafic utile dans le fichier **/etc/hosts.allow**
Par exemple, pour limiter le nombre d'hôtes ayant accès au service portmap puisqu'il n'est doté d'aucune forme d'authentification interne. Il faut ajouter les adresses IP des machines souhaitées dans le fichier : **/etc/hosts.allow**

```
Portmap : 10.0.0.0/255.255.255.0
```

Cette ligne indique que seul le réseau 10.0.0.0/24 est autorisé à utiliser le portmap.

TCPwrapper permet aussi d'envoyer des bannières de connexion, prévenir des attaques provenant d'hôtes particuliers et améliorer la fonctionnalité de journalisation.

R 36	Contrôler l'accès aux services réseaux par l'utilisation de TCPwrapper.

Note : Il convient d'utiliser les règles de pare-feu « Iptables » avec les « TCP-wrapper » pour créer une redondance dans les contrôles d'accès aux services.

5 Sauvegarde et restauration

Un système sécurisé doit absolument garantir un accès sûr aux données. Les différents incidents qui pourraient compromettre celles-ci ne sont pas prédictibles, et malheureusement malgré une politique très bien pensée et appliquée, ils ne peuvent pas être évités. C'est pourquoi une bonne politique de sauvegarde est nécessaire.

Les sauvegardes peuvent être simples (une simple copie, ou un archivage de base), ou évoluées (suivant un modèle client/serveur et/ou par l'utilisation d'outils de sauvegarde automatiques). Il existe différents outils de sauvegarde et le système Linux offre un ensemble de commandes. L'administrateur des systèmes peut choisir la méthode de sauvegarde la plus adaptée à son contexte applicatif. Par exemple :

La commande **tar** : permet de sauvegarder des fichiers et des arbres d'un utilisateur ou d'une application. Pour sauvegarder une liste de répertoires dans une archive **tar** unique, il suffit de lancer la commande **tar** suivie par la commande **gzip** pour compresser :

```
tar -cvf archive-name.tar dir1 dir2 dir3....
gzip -9 archive-name.tar
```

La commande dump permet de faire des sauvegardes incrémentales des systèmes de fichiers Ext2 et Ext3. Par exemple pour sauvegarder le système de fichiers /boot, utiliser la commande :

```
dump 0zf backup.boot /boot
```

La commande dd permet de réaliser des copies physiques, elle peut parfois être utilisée pour sauvegarder des disques et des systèmes de fichiers. Cependant l'utilisation de cette commande s'avère très dangereuse, donc il est recommandé d'éviter la sauvegarde par l'utilitaire dd.

Si les sauvegardes sont faites localement (comme le montrent les exemples ci-dessus), l'accès aux fichiers de sauvegardes doit être restreint.

R 37	Effectuer une sauvegarde de bas niveau à l'installation du système, avant sa mise en production, pour pouvoir rétablir le serveur dans son état initial en cas de problème majeur.

R 38	Refaire cette sauvegarde à chaque fois que le système est mis à jour.

R 39	S'assurer que le logiciel de sauvegarde utilisé exige une authentification entre le client et le serveur de sauvegarde.

R 40	Chiffrer les données sauvegardées selon leurs degrés de sensibilité.

R 41	Prévoir une durée minimale entre les opérations de sauvegarde afin de revenir vers un système propre si un incident intervient.

R 42	Surveiller régulièrement le bon déroulement des sauvegardes.

R 43	Vérifier les sauvegardes en restaurant régulièrement des éléments sauvegardés pour s'assurer du bon fonctionnement du mécanisme de restauration et éviter de se retrouver avec des sauvegardes inutilisables.

R 44	Vérifier que toutes les opérations de sauvegardes sont journalisées.

R 45	Restreindre l'accès aux fichiers de sauvegarde aux seules personnes autorisées.

R 46	Placer les sauvegardes dans un lieu sûr distinct du site source.

6 Chiffrement

Toutes les données transmises sur un réseau doivent être chiffrées, Il est fortement recommandé de ne rien faire transiter en clair, de manière à ce qu'aucune communication ne soit interceptée ou altérée. Donc la mise en œuvre systématique de solutions utilisant l'authentification et le chiffrement afin de protéger les données sensibles que ce soit celles transitant sur le réseau ou celles stockées sur le disque dur est primordiale. Par exemple il convient d'utiliser :

- SSH au lieu de TELNET, FTP, RSH ;
- IMAPS au lieu d'IMAP ;
- HTTPS au lieu de HTTP ;
- etc.

De même, il convient d'utiliser **GNU Privacy Guard (GPG ou GnuPG)** ou équivalent pour transmettre des messages signés et chiffrés afin de garantir l'authenticité et la confidentialité des données transmises.

R 47	Utiliser un moyen de chifferement approprié pour sécuriser les données sensibles transférées via le réseau (données d'authentification, emails, pièces jointes, documents sensibles, etc.).

7 Supervision et audit

7.1 Journalisation

Lorsque le système Linux démarre, fonctionne et effectue tout type d'action, ses actions et celles de la plupart de ses services sont tracées dans des fichiers divers. Deux démons sont spécialisés dans la réception des messages à écrire dans ces fichiers :

- *klogd* : kernel log daemon, chargé de la gestion des informations émises par le noyau.
- *syslogd* : system log daemon, chargé de la gestion des informations émises par tous types de services et éventuellement le noyau.

Il est recommandé de configurer *syslogd* afin de journaliser toutes les activités du système. En effet il faut définir les services, les niveaux et les destinations au niveau du fichier */etc/syslogd.conf*. Par exemple pour envoyer tous les événements système à un serveur de logs il faut ajouter la ligne suivante :

```
*.* @adresse_IP_serveur_log
```

Les logs systèmes sont situés dans le fichier /var/log. il est recommandé à l'administrateur de consulter ce qui suit :

- **/var/log/messages** : Ce fichier regroupe les messages des différents démons et services du système ;
- **/var/log/secure** : Ce fichier garde des traces sur les connexions de façon détaillées (adresse IP, service, port . . .). C'est également ici que le démon sshd stocke les tentatives de connexion ;
- **/var/log/auth.log** : Ce fichier enregistre tous les logins qui se connectent au système, ainsi que le mécanisme de connexion utilisé.

Il convient par la suite de mettre en place des outils d'analyse de journaux d'évènements, tels que :
- LogWatch ;
- Swatch, outil d'analyse des logs en Perl ;
- LogCheck, outil d'analyse basé sur le système Cron.
- Etc.

R 48	Définir et mettre en place une politique de journalisation d'événements.

R 49	Activer la journalisation du système et des services.

R 50	Rediriger les logs vers un serveur de logs dédié.

R 51	Centraliser les logs.

R 52	Archiver les logs.

R 53	Analyser les logs.

R 54	Restreindre l'accès aux fichiers de logs aux seules personnes autorisées.

R 55	Définir des rôles précis au niveau de l'outil de consultation des logs.

R 56	Afin de pouvoir confronter les journaux de plusieurs systèmes, il est nécessaire que leurs horloges soient synchronisées (NTP).

7.2 Vérification de l'intégrité du système

La vérification de l'intégrité du système linux permet de s'assurer de l'intégrité des répertoires et des fichiers importants du système en identifiant tous changements apportés à ces derniers. En effet, en cas de compromission, certains fichiers sont modifiés par l'attaquant pour masquer sa présence et installer une ou plusieurs portes dérobées (backdoors) sur le système.

Le principe de la vérification de l'intégrité du système consiste à prendre l'empreinte des fichiers pour créer une base de données de référence avec la signature de chacun des fichiers à surveiller. Cette signature est constituée de nombreux indicateurs garantissant l'unicité du fichier auquel elle se rapporte (propriétaire, date de création, date de dernière modification, taille, calcul d'empreinte, etc.). Cette empreinte sera ensuite régulièrement comparée à l'empreinte courante et l'administrateur sera avisé, en général par un courrier électronique, en cas de modification de l'intégrité d'un fichier.

De nombreux outils permettant la vérification de l'intégrité des fichiers peuvent être utilisés, notamment : Open source Tripwire, Advanced Intrusion Detection Environment (AIDE) et Another File Integrity Checker (AFICK), etc.

R 57	Utiliser des outils appropriés pour s'assurer de l'intégrité des fichiers du système.

- Il est recommandé de protéger la base de données des empreintes par mot de passe pour éviter qu'un attaquant compromettant la machine puisse aussi compromettre trivialement la base.

7.3 Audit

L'audit périodique de la sécurité de l'OS est un moyen essentiel pour identifier les vulnérabilités et veiller à ce que les mesures de sécurité existantes soient efficaces. Il existe différents outils commerciaux et open source que l'administrateur peut utiliser afin d'évaluer le niveau de sécurité de son système.

Lynis par exemple est un outil open source qui permet de réaliser un audit simple d'un serveur linux, en effet, il génère un rapport de sécurité récapitulatif et synthétique de l'état du serveur et ce, en analysant l'ensemble du système : le chargeur de démarrage (bootloader), les services, le kernel, la mémoire, les processus, les utilisateurs, les groupes et l'authentification, les shells, le système de fichiers, le stockage, la configuration réseau, etc.

L'administrateur pourra consulter le rapport complet, qui se trouve dans : */var/log/lynis.log*.

R 58	Effectuer régulièrement des audits du système afin d'analyser : – La configuration du serveur ; – Les performances du serveur ; – L'évolution du matériel ; – L'évolution des logiciels ; – Les mises à jour.

Références

1 National Security Agency, *"Guide to the Secure Configuration of Red Hat Enterprise Linux5"*,February 28, 2011.

2 SANS Institute, *Linux Security Checklist*.

3 Javier Fernández-Sanguino Peña, *"Securing Debian Manual"*, , 08 Apr 2012.

4 Agence nationale de la sécurité des systèmes d'information ANSSI-France, *"Recommandations de sécurité relatives à un système GNU/Linux"*,5 juillet 2012.

5 *http ://contrib.xarli.net/secure-gnulinux/index.html*.