LES RECOMMANDATIONS

SECURISER UNIX

Septembre 1997

Commission Réseaux et Systèmes Ouverts

CLUB DE LA SECURITE DES SYSTEMES D'INFORMATION FRANÇAIS
30, Rue Pierre Sémard – 75009 Paris
Téléphone : 01 53 25 08 80 - Fax : 01.53 25 08 88

REMERCIEMENTS

Le CLUSIF tient à mettre ici à l'honneur les personnes qui ont rendu possible la réalisation de ce document, tout particulièrement :

Muriel	*COLLIGNON*	**AXONE**
Philippe	*CONVERSIN*	**DASSAULT ELECTRONIQUE**
Michèle	*COPITET*	**CAP GEMINI**
Marie-Hélène	*COURBIS*	**ABYSS ENGINEERING**
Jacques	*GONIK*	
Didier	*GRAS*	**XP CONSEIL**
Frédéric	*HUYNH*	**CLUSIF**
François	*NOZICK*	**ALCATEL TITN ANSWARE**
Lazaro	*PEJSACHOWICZ*	**BULL**
Paul	*RICHY*	**FRANCE TÉLÉCOM**
Serge	*SAGHROUNE*	**BULL**
Hervé	*SCHAUER*	**HERVÉ SCHAUER CONSULTANTS**
Bernard	*VALENTIN*	**CBV2A/SYSIACOM**

TABLE DES MATIERES

1. GENERALITES

1.1 Introduction

La configuration et le paramétrage du système d'exploitation sont des éléments qui interviennent de manière importante vis-à-vis de la sécurité et de la sûreté de fonctionnement des Systèmes d'Information.

Dans le cas présent de systèmes Unix, c'est principalement le compte "root", du super-utilisateur, qui est cible des attaques puisqu'il possède tous les privilèges sur le système. Dans un deuxième temps, si le niveau de sécurité à atteindre le justifie, il convient de prendre en considération les comptes utilisateur ayant des droits spécifiques, sur des applicatifs par exemple.

Les accès distants à travers un réseau interviennent également de façon importante sur le niveau de sécurité puisque dès lors que le système est relié, il peut être la cible de tentative de pénétration. Un individu peut tenter à distance d'exploiter les failles et les vulnérabilités du système pour se connecter sur la station.

1.2 Mise en garde

Les failles de sécurité ne sont pas l'apanage d'Unix, contrairement à une rumeur savamment entretenue. En fait, Unix est marqué par son image universitaire et scientifique, et fut utilisé dans des milieux où l'aspect sécurité n'était pas le souci majeur. Cependant des fonctions de sécurité existent, même sur les versions standard.

Assurer la sécurité de systèmes d'information reliés en réseau est une tâche complexe et fastidieuse quel que soit le type de système, et Unix n'échappe pas à la règle. Cette tâche nécessite en effet du temps et des procédures adaptées afin d'assurer la cohérence de mécanismes de sécurité chargés de contrôler plusieurs dizaines de milliers de fichiers par station !

Dans le cas où l'administrateur système n'est pas suffisamment sensible au problème de la sécurité ou bien s'il ne possède pas les moyens suffisants pour s'en occuper (temps, formation, outils logiciels adaptés...), des failles se créent. Ainsi quel que soit l'organisme soumis à un audit, la présence de failles de sécurité se révèle extrêmement fréquente.

1.3 Objet du document

1.3.1 Une double cible

L'objet de ce document est double :

- fournir au **RSSI (Responsable Sécurité des Systèmes d'Information)** les grands thèmes de réflexion qui lui permettront d'apprécier le niveau de sécurité d'une station ou d'un réseau de stations UNIX. Une synthèse de chacun des thèmes est fournie en début de chapitre.

- fournir à **l'Administrateur système** de la station ou du réseau une "checklist" d'actions, à entreprendre pour renforcer la sécurité des systèmes dont il a la charge.

Afin d'obtenir sur vos systèmes Unix le niveau de sécurité en adéquation avec votre politique de sécurité, des conseils ou recommandations sont proposés, ceux-ci sont répartis en trois catégories :

- l'aspect organisationnel : traite des préalables structurels à la mise en oeuvre des mesures techniques
- la sécurité locale du serveur : traite des mécanismes de sécurité à mettre en oeuvre même sur une station de travail isolée
- l'aspect réseau : traite des mécanismes de sécurité supplémentaires dans le cas d'une station reliée à un réseau

1.3.2 Lecture des tableaux de recommandations

Les recommandations fournies dans les tableaux s'adressent à l'administrateur sécurité (cf. §2.2). L'hypothèse de base du document est que ce dernier connaît parfaitement le système Unix. La structure du système de fichiers, les diverses commandes et spécificités du système sont supposées acquises. L'objectif du document n'est pas de décrire le fonctionnement d'Unix mais de recenser les différents mécanismes susceptibles de participer la sécurisation d'un serveur ou d'un réseau de serveurs Unix.

La pondération des critères est la suivante :

Importance	Signification
4	utile en toute circonstance
3	utile dès que c'est possible
2	utile pour des systèmes sensibles
1	utile pour des systèmes très sensibles
0	utile pour des cas spécifiques

1.4 Références

- UNIX Computer Security Checklist - AUSCERT (AUStralian Computer Emergency Response Team)
- Guide de sécurité des systèmes Unix - Ch. Pélissier - HERMES

1.5 Comparaison Unix - Windows NT

Unix, archétype du système "ouvert", est souvent comparé à Microsoft Windows NT. On peut dire que le niveau de confidentialité atteint par les deux systèmes est sensiblement voisin.

En effet NT, dont Microsoft nous dit qu'il satisfait aux différents critères requis par le niveau C2 des TCSEC (Orange book), est un système multi-utilisateurs. A ce titre il doit offrir, ce que fait n'importe quel Unix depuis longtemps, la possibilité de définir un compte par utilisateur, de placer

des droits d'accès sur tous les fichiers, d'éviter que les cryptogrammes des mots de passe ne soient accessibles par tous ou d'effacer la mémoire avant sa réutilisation.

La différence essentielle ne réside donc pas là mais dans le fait qu'Unix est un système ouvert standard et multi-constructeurs alors que NT est propriétaire et "fermé".

Un reproche fréquemment attribué à Unix par ses détracteurs, est que ses sources sont publics (à la différence de NT) et que, par conséquent, chacun peut les examiner à loisir pour en détecter les failles les plus insidieuses. Cependant la mise à disposition des sources est plutôt un gage de sécurité (ils sont requis pour les hauts niveaux d'évaluation ITSEC). Il est ainsi facile d'ajouter de nouvelles fonctions de sécurité complémentaires (ex : cartes à puces, calculettes...). Cette caractéristique permet aussi de faire jouer la concurrence, ce qui a toujours garanti une grande réactivité de la part des grands constructeurs (Sun, IBM, SCO, HP...) quant aux corrections de bogues, domaine dans lequel Microsoft doit encore faire ses preuves.

Face à cela, le principal inconvénient de NT est sa jeunesse et son opacité. Depuis qu'il est utilisé à grande échelle, de nombreux bogues impactant la sécurité ont également été trouvés.

En réalité, un système Unix correctement administré se révèle sûr quel que soit l'axe de la sécurité considéré (Disponibilité, Intégrité, Confidentialité), Unix bénéficie là de ses longues années d'expérience. En outre, on peut noter que pour les besoins de niveau de confidentialité élevé, la plupart des grands constructeurs proposent des versions "militarisées" d'Unix (ex : Trusted Solaris...).

2. ASPECT ORGANISATIONNEL

2.1 Synthèse pour le RSSI

Le **Responsable de la Sécurité des Systèmes d'Information** (RSSI) d'un organisme doit prendre en compte les deux tâches suivantes :

- définir une « **Politique de sécurité** » adaptée à l'organisme. La politique de sécurité doit correspondre aux orientations stratégiques définies par la Direction vis-à-vis des Systèmes d'Information et prendre en compte les aspects de la réglementation et du droit ainsi que la valeur du patrimoine informationnel.

La politique doit inclure également les aspects formation et sensibilisation des utilisateurs. Elle est indispensable et doit être diffusée au personnel de l'organisme au minimum par des consignes de sécurité. Cette activité est générale et dépasse le cadre des seuls systèmes Unix, aussi n'est elle pas décrite dans le présent document.

- identifier la fonction d'**Administration de la sécurité** au sein de l'organisation mise en place pour la gestion des Systèmes d'Information. Il est préférable, pour des raisons d'efficacité et d'objectivité, qu'une personne différente de l'administrateur système assure cette fonction.

Il est impératif que quelqu'un gère la sécurité du système sinon des problèmes apparaîtront inévitablement à la suite soit d'erreurs soit de malveillance. Unix doit sa mauvaise réputation à cette absence de prise en compte car on a longtemps cru que ces systèmes multi-utilisateurs n'avaient pas besoin d'être gérés à la différence des sites centraux. Le problème sera identique pour NT.

2.2 L'administration de la sécurité

Cette fonction est réalisée par des administrateurs système maîtrisant parfaitement les spécificités du système d'exploitation. A partir d'un certain nombre de machines, l'un d'eux doit se spécialiser et assurer le rôle d'administrateur sécurité.

Cette fonction est essentielle pour l'application des mesures de sécurité définies par la politique de sécurité. Dans le cas de systèmes Unix, l'administrateur doit effectuer les actions précisées dans le tableau :

Enoncé	Importance
1. mettre en place des procédures de sauvegarde et d'archivage (comme pour tout système informatique)	4
2. se tenir informé (via les organismes tels que CERT, CIAC...décrits en annexe) des failles ou bogues détectés sur les systèmes qu'il gère	4
3. installer les patches afin de corriger les failles ou bogues des systèmes	4
4. consulter les documents et fiches de sécurité Unix (manuel sécurité constructeur, fiches disponibles sur les serveurs Web...)	4

Enoncé	Importance
5. activer les mécanismes de sécurité standard et si besoin installer des outils complémentaires	4

3. PROTECTION LOCALE DU SERVEUR

3.1 Synthèse pour le RSSI

La protection locale du serveur Unix, la qualité de son administration ou l'activation de ses fonctions de sécurité sont de la plus grande importance. Cela constitue la base de toute sécurisation.

Unix est un système multi-utilisateurs et multitâches, aussi s'apparente-t-il plus à un grand système de type site central qu'à un simple PC sous DOS, ce que l'on oublie parfois. Cependant, à la différence des grands systèmes centraux qui ne fonctionnent pas s'ils ne sont pas soigneusement gérés, Unix continue à rendre les services qu'on lui demande mais avec des failles de sécurité. Ainsi, il requiert des actions d'administration système et sécurité, et la qualité de la sécurité Unix réside donc dans la présence d'administrateurs rigoureux.

En ce qui concerne la protection des applications, il convient de garder à l'esprit que le niveau de sécurité d'un système ne peut se concevoir que globalement. Lors de la mise en place d'une application, la sécurité de celle-ci doit être étudiée en relation avec celle du système d'exploitation sous-jacent. Il ne sert en effet à rien d'avoir un contrôle d'accès renforcé au niveau d'une application si un intrus peut s'introduire par un compte Unix et éditer directement les fichiers de l'application.

Les principes pour obtenir un bon niveau de sécurité avec Unix sont classiques :

- **une gestion des comptes rigoureuse** : la gestion des mots de passe par l'administrateur est incontournable. Celui qui s'approprie le mot de passe d'un utilisateur, usurpe tous les privilèges de ce dernier. En outre cette gestion doit être complétée par une **sensibilisation des utilisateurs** car si l'un d'eux laisse une porte ouverte, c'est l'ensemble des utilisateurs qui devient vulnérable. Un seul compte utilisateur compromis peut mettre en danger l'ensemble des systèmes reliés en réseaux. En particulier, les comptes de maintenance ne doivent pas rester actifs en permanence.

- **une gestion des fichiers rigoureuse**, non seulement en terme de contenu mais aussi et surtout en terme de privilèges associés à chacun de ces fichiers. Une seule faille peut permettre à un utilisateur standard d'obtenir les privilèges de l'administrateur. L'administrateur doit notamment vérifier que les applications système possèdent les privilèges suffisants mais pas plus.

- **un audit régulier** : l'administrateur doit régulièrement surveiller les traces et les journaux fournis par le système et être vigilant à toutes les tentatives de connexion intempestives, de modifications de privilèges d'utilisateurs ou tous les événements susceptibles d'introduire des vulnérabilités.

Outre l'activation des fonctions de sécurité standard du système, l'ajout de logiciels supplémentaires est utile dans le cas de réseau à fortes contraintes de sécurité. Citons par exemple ceux permettant l'utilisation de mots de passe à usage unique (non rejouables) qui permet de s'affranchir du risque d'écoute (les mots de passe transitent généralement en clair sur les réseaux). Il faut toutefois veiller à ne pas ajouter de nouvelles vulnérabilités lors de l'intégration de ces produits.

3.2 Administration des comptes

3.2.1 Mots de passe

Enoncé	Importance
1. Vérifier que tous les comptes ont un mot de passe	4
2. Obliger l'utilisateur à saisir un mot de passe à sa première connexion (en général automatique sur les nouveaux Unix, pour les anciennes versions mettre ",.." dans le champ cryptogramme de /etc/passwd)	4
3. Contrôler la qualité des mots de passe (éviter prénoms et noms connus, insérer des caractères numériques et spéciaux) pour les comptes administrateur et utilisateur. Ainsi les versions d'Unix les plus récentes permettent de définir une longueur minimum, l'introduction d'au moins un caractère non alphabétique... Une alternative consiste à utiliser les outils tels que ANLPASSWD ou PASSWD+ pour les Unix plus anciens.	4
4. Contrôler le vieillissement des mots de passe en conformité avec la politique de sécurité	3
5. Ne pas mentionner dans le champ GCOS (commentaire sur l'utilisateur) d'informations susceptibles d'aider un individu malveillant, ex : sa fonction...)	2
6. Pour des besoins forts de sécurité, utiliser des mots de passe dynamiques, tels que les mots de passe à usage unique comme OPIE, les cartes à puces ou les calculettes ("token")	2
7. N'utiliser le champ mot de passe dans la définition des groupes (/etc/group) que dans des cas spécifiques.	0

3.2.2 Déport des mots de passe

Enoncé	Importance
1. Utiliser les possibilités de déporter (par ex : fonction C2conv sur SunOS) le fichier des cryptogrammes dans un fichier non lisible par tous les utilisateurs. Ce fichier peut s'appeler /etc/security/passwd.adjunct (SunOS) /etc/shadow (Solaris), /etc/security/passwd (AIX).	4
2. Vérifier périodiquement la cohérence entre le fichier /etc/passwd et ce fichier.	3

3.2.3 Compte super-utilisateur (root)

Enoncé	Importance
1. Limiter le nombre de personnes connaissant le mot de passe root au strict minimum (2 ou 3 et placer le mdp dans un coffre en prévision des congés des administrateurs)	4
2. Ne pas utiliser de ~root/.rhosts	4
3. Restreindre le login de root aux consoles reliées physiquement (ex : /dev/console) Le fichier de paramétrage peut être appelé /etc/ttys, /etc/default/login, /etc/security... S'assurer que ce fichier est bien la propriété de root et que ses permissions sont 600.	4
4. Vérifier que les fichiers ~root/.login, .profile, .cshrc, .exrc, . logout... ne contiennent pas d'appel à des fonctions n'appartenant pas à root. Renouveler le contrôle périodiquement.	4
5. Exécuter en tant que root uniquement des fichiers (binaires ou scripts)	4

Enoncé	Importance
protégés, c'est à dire modifiables seulement par root et dont le contenu est digne de confiance	
6. Vérifier que les fichiers "crontab" ne font pas appel à des fichiers non protégés	4
7. Positionner la variable « umask » à 077 (les fichiers créés posséderont par défaut les droits complémentaires 700 rwx------)	4
8. Activer les fonctions de verrouillage automatique d'écran (fonction standard sur certains Unix ou utiliser des logiciels tels que xautolock décrit en annexe)	4
9. Vérifier que le PATH ne contient pas des répertoires modifiables par d'autres utilisateurs.	4
10.Ne pas utiliser de "." dans le PATH	3
11.Utiliser les commandes en les nommant avec le chemin absolu (/bin/su, /bin/passwd...)	3
12.Ne pas changer la variable IFS (forcer IFS="\t\n" dans le .profile)	3
13.Ne pas se connecter root directement, mais utiliser de préférence la commande "su" à partir d'un compte non privilégié (les traces sont plus aisées)	3
14.Eviter les connexions au travers du réseau (le mot de passe circule en clair) ou utiliser des systèmes de connexion avec chiffrement comme SSH (avec les restrictions légales)	3

3.2.4 Comptes utilisateur

Enoncé	Importance
1. Utiliser des comptes nominatifs individuels et ne pas utiliser de comptes partagés	4
2. Vérifier que seul le login de l'utilisateur est dans .rhosts (surtout pas de "+ +") et que les machines sont dans les domaines de l'entreprise.	4
3. Supprimer les fichiers .netrc de connexions automatiques (sauf pour la connexion aux serveurs ftp anonymes)	4
4. Vérifier que les fichiers .login, .profile, .cshrc, .exrc, .forward, .logout... ne sont modifiables que par leur propriétaire et que leur contenu est licite	4
5. Vérifier que les fichiers "crontab" appartiennent à l'identité du compte sous lequel les crontab sont lancés, et que ces fichiers ne contiennent pas des appels à d'autres fichiers non protégés.	4
6. Vérifier que le PATH ne contient pas des chemins non protégés. Vérifier que le "." n'est pas présent ou en dernière position.	4
7. Positionner la variable « umask » à 027 (les fichiers créés posséderont par défaut les droits complémentaires 750 rwxr-x---) voire si possible à 077.	3

3.2.5 Comptes spéciaux

Enoncé	Importance
1. Maîtriser les comptes qui possèdent l'UID = 0. Deux philosophies peuvent être adoptées, soit les administrateurs partagent l'usage du compte root, soit chaque administrateur possède son propre compte privilégié à UID = 0 (donc équivalent à root)	4
2. Désactiver (ou supprimer) les comptes « invité » (guest), fournisseur, préconfigurés (sync, sysadm)...	4
3. Limiter l'usage du "su" root (ex : avec le groupe wheel pour SunOS)	3
4. Pour chaque compte système devant être verrouillé (ex : lp, deamon...) mettre /bin/false comme shell (champ de /etc/passwd)	3

Enoncé	Importance
5. Désactiver les comptes inutilisés pendant une période donnée.	2

3.2.6 Distribution des bases d'administration (NIS, NIS+)

Enoncé	Importance
1. Il est préférable de ne pas utiliser NIS (Network Information Service appelé autrefois Yellow Pages). Le système ne supporte notamment pas le déport de mots de passe.	3
2. Utiliser les RPC sécurisées	3
3. Vérifier que seules les stations clientes et non pas le serveur NIS contiennent la ligne '+' dans leur fichier de mot de passe (/etc/passwd)	3
4. Activer si possible ypbind avec l'option -s dans les fichiers d'initialisation (ex : /etc/rc.local),	3
5. N'utiliser NIS+ que si des besoins de sécurité spécifiques le justifient.	0

3.3 Administration des fichiers

3.3.1 Gestion des permissions

Enoncé	Importance
1. Gérer les permissions au niveau des fichiers mais aussi des répertoires	4
2. Gérer les groupes et appliquer les permissions idoines en fonction des groupes et pas seulement des utilisateurs (cela évite souvent de devenir root pour des raisons futiles)	4
3. Gérer les permissions pour les progiciels (SGBD...)	4
4. Installer la version la plus récente de /usr/lib/expreserve (consulter les avis du CERT concernant la fonction)	3

3.3.2 Exemples de permissions pour les fichiers sensibles

Enoncé	Importance
1. Utiliser les permissions 644 (rw-r--r--) : le noyau /vmunix, /hp-ux... les fichiers /etc/utmp, /etc/state, /etc/motd, /etc/mtab, /etc/syslog.pid	4
2. Utiliser les permissions 1777 (drwxrwxrwt). pour les répertoires temporaires (/tmp, /var/tmp, /usr/tmp...)	4
3. S'assurer que le noyau et les répertoires sensibles (/etc, /usr/etc, /bin, /usr/bin, /sbin, /usr/sbin, /tmp et /var/tmp) appartiennent à root (plutôt qu'à un pseudo-compte tel que "bin")	3
4. Utiliser les permissions 2755 (rwxr-sr-x). pour /etc/sm and /etc/sm.bak	2

3.3.3 Partage de « file systems » distants (NFS)

(NFS : Network File System)

Enoncé	Importance
1. Monter les « file systems » distants avec les options no suid	4
2. Exporter si possible les « file systems » distants avec l'option lecture seule	3
3. Exporter les « file systems » en nommant explicitement la liste des machines	3

Enoncé	Importance
autorisées (option -access) à effectuer le montage	
4. Limiter les arborescences exportées: ne pas exporter tout le « file system » mais seulement les répertoires nécessaires	3

3.3.4 Procédures de démarrage et d'arrêt

Enoncé	Importance
1. Examiner le contenu des fichiers de démarrage et d'arrêt (ex : /etc/rc* /etc/shutdown). Ils ne doivent pas, entre autre, faire appel à des fichiers non protégés.	4
2. Vérifier notamment que les permissions conseillées pour le /etc/motd ne sont pas modifiées. Le « Message Of The Day » qui présente le message quotidien de l'administrateur à ses utilisateurs ne doit pas être modifiable par n'importe qui.	3

3.4 Audit

3.4.1 Les fonctions standard

Enoncé	Importance
1. Examiner périodiquement le contenu des logs ("wtmp", "sulog"...), échecs répétés de connexions	4
2. Utiliser le syslog pour gérer ses journaux et les niveaux d'alertes	4
3. Activer les fonctions d'« accounting » ou d'« audit trail » sur les serveurs sensibles. L'inconvénient est que ces fonctions génèrent beaucoup de traces et sont gourmandes en place disque.	3

3.4.2 Les contrôles périodiques

Enoncé	Importance
1. Les fournisseurs offrent (ou vendent) de plus en plus fréquemment des outils destinés à auditer le système ou à contrôler les dérives par rapport à une base de confiance (TCB). Les outils gratuits Tiger (ou COPS qui est plus ancien) et Tripwire permettent de réaliser ces fonctions d'audit. L'avantage est que l'on peut utiliser ces outils sur tout type d'Unix (Solaris, HP-UX, AIX...)	4
2. Contrôler périodiquement la qualité des mots de passe. Le contrôle des mots de passe faibles peut être effectué a posteriori par des outils tels que CRACK (mais préférer un contrôle a priori tel que décrit précédemment)	3
3. Examiner périodiquement le contenu des fichiers à exécution programmée (crontabs)	3
4. Vérifier que la liste des répertoires et des fichiers en écriture pour tous est licite.	3
5. Vérifier qu'il n'y a pas de fichiers spéciaux (block, caractère) en dehors du répertoire « /dev ».	2

3.4.3 Surveiller les fichiers SUID, SGID

Enoncé	Importance

Enoncé	Importance
1. Les fichiers possédant le bit SUID ou SGID constituent l'une des vulnérabilités majeures d'Unix, en particulier s'il s'agit de scripts. Leur contenu doit être écrit suivant les règles de l'art1 et les permissions soigneusement positionnées. Tout nouveau fichier de ce type est un cheval de Troie potentiel.	4
2. Ne pas positionner le bit SUID sur les scripts appartenant à root, préférer les binaires (les nouveaux Unix le vérifient automatiquement)	4
3. Examiner périodiquement la liste des fichiers possédant le bit SUID ou SGID	4

3.5 Recommandations générales

Enoncé	Importance
1. Placer les sauvegardes et archivages dans un lieu adéquat (coffre ignifugé...)	4
2. Sécuriser les locaux (accès, climatisation, protection contre les inondations, etc)2	3
3. Sur Sun, il faut donner un mot de passe au niveau de l' EEPROM et demander un mdp lors du boot en single user	3
4. Sur RS6000, retirer la clé permettant l'accès en mode maintenance	3
5. Ne pas laisser les disquettes de boot en libre accès près du serveur	3
6. Détruire les supports magnétiques inutiles	2

1 Notamment : appeler des commandes sûres par leur chemin absolu, positionner les variables d'environnement sensible (PATH, IFS..), etc.

2 Ces recommandations s'appliquent à tout système d'information. Plusieurs ouvrages font référence et traitent le sujet de manière plus complète.

4. ASPECT RESEAU

4.1 Synthèse pour le RSSI

Avec la généralisation des systèmes client/serveur, de la distribution des applications et l'augmentation considérable de la part des réseaux dans les systèmes d'information, il convient de compléter les mesures de sécurité relatives à la protection locale d'une machine.

Les attaques à travers le réseau sont les plus insidieuses puisque l'intrus peut accéder à toutes les données du système d'information sans même se déplacer. Ceci est d'ailleurs vrai pour tout type de machine et pas seulement pour les stations Unix.

Dès qu'une machine est connectée au sein d'un réseau, elle est accessible à distance, donc vulnérable. Les parades pour lutter efficacement contre les intrusions par le réseau consistent à définir des mesures de protection à tous les niveaux de l'architecture du réseau local. Ainsi convient-il de :

- **définir une configuration réseau sécurisé** pour les stations Unix : le système Unix permet de mettre en oeuvre un certain nombre de mécanismes afin de filtrer les accès et protéger les services et applications échangeant des données à travers le réseau.
- **filtrer les communications réseau** : une mesure complémentaire aux protections locales d'Unix consiste à mettre en oeuvre du filtrage et ainsi définir un cloisonnement au niveau du réseau. Rappelons que les trames réseau contiennent les mots de passe en clair. Le filtrage peut être réalisé par :

  ⇒ un équipement réseau de type routeur filtrant,

  ⇒ un équipement sécurité de type pare-feu

- **protéger les applications mises à disposition à travers le réseau**, telles que la messagerie, un serveur Web, un serveur ftp... Le choix entre un serveur ftp anonyme et un serveur ftp standard résulte de l'évaluation du risque entre la capture du mot de passe et l'intrusion sur le serveur. L'avantage d'un serveur ftp anonyme est qu'aucun mot de passe ne circule en clair sur le réseau. L'inconvénient est qu'il constitue une porte ouverte sur le système.

4.2 Configuration sécurisée

4.2.1 Fichiers de configuration réseau

Enoncé	Importance
1. **/etc/services** S'assurer que ce fichier est bien la propriété de root et que les permissions sur ce fichier sont 644.	4
2. **/etc/inetd.conf :** S'assurer que ce fichier est bien la propriété de root et que les permissions sur ce fichier sont 600. Supprimer tous les services qui ne sont pas vraiment utiles. Il est conseilloer de commencer par interdire tous les services en rajoutant le symbole '# ' en début de ligne. Ensuite pour autoriser les services au fur et à mesure supprimer	4

Enoncé	Importance
ce caractère. Ne pas oublier que pour qu'un changement prenne effet, il faut relancer inetd (kill -HUP). Pour certains systèmes (dont AIX), ces commandes ne sont pas suffisantes. Se référer à la documentation de vos vendeurs pour plus d'informations	
3. **/etc/hosts.lpd :** S'assurer que ce fichier est bien la propriété de root et que les permissions sur ce fichier sont 600 S'assurer que le premier caractère de ce fichier n'est pas '-'. S'assurer que l'on utilise pas dans ce fichier les caractères '!' et '#'.	4
4. Terminaux sécurisés : Le fichier de paramétrage peut être appelé **/etc/ttys, /etc/default/login, /etc/security...** Vérifier que l'option de sécurité est retirée de toutes les entrées qui n'ont pas besoin d'avoir des accès root, c'est à dire tout le monde sauf la console root. L'option sécurisée doit être retirée à partir de la console si l'on ne veut pas que les utilisateurs redémarrent en mode 'single user'.	3

4.2.2 Services sensibles

Enoncé	Importance
1. **rexd :** Supprimer ce service (dans inetd.conf). Le serveur rexd, associés à la commande « **on** », n'offre aucune sécurité dans sa mise en œuvre.	4
2. **Portmapper** : Désactiver tous les services qui sont démarrés à l'initialisation du système et qui ne sont pas nécessaires.	4
3. **fingerd :** Supprimer ce service (dans inetd.conf). Finger offre toute une série d'informations utiles pour monter des attaques. Tester la configuration de finger et vérifier la possibilité de restreindre celle-ci, sinon la remplacer par une version restreinte. Il est rappelé que des services comme rusers et netstat peuvent fournir des informations similaires. Sinon, utiliser une version récente et sécurisée.	3
4. **NIS** (Network Information Service / Yellow Pages) : Se reporter aux recommandations du chapitre précédent. En cas d'utilisation de NIS pour la distribution des bases d'administration, définir chaque netgroup de tel sorte qu'ils contiennent soit des noms de compte, soit des noms de host. Mais il faut éviter de mélanger les deux. L'utilisation de netgroups distincts rend plus simple l'administration. Le temps plus important requis par la multiplication des netgroups est compensé par la maîtrise plus simple des autorisations. En effet, on évite ainsi des autorisations croisées compliquées.	3

4.2.3 Sécurité du protocole UDP

Enoncé	Importance
1. Désactiver sur votre machine tous les ports UDP qui ne sont pas vraiment utiles. Pour cela, éditer le fichier **/etc/inetd.conf** et commenter les lignes concernant chaque service inutilisé (echo, chargen...) envoyer signal -HUP au processus inetd pour une prise en compte immédiate de la modification.	4
2. Désactiver en particulier les services chargen et echo. Des attaques existent qui exploitent les services **chargen** et **echo**, alors que ces deux services ne sont pas, à notre connaissance, généralement utilisés. Un exemple est fourni en **NB** après ce tableau.	3
3. Surveiller le réseau. Si vous fournissez des services UDP, il est utile de contrôler le débit du réseau, de savoir quels systèmes utilisent ces services ou	3

Enoncé	Importance
de surveiller les signes d'abus. Des utilitaires comme tcpdump (voir annexe) permettent de le faire.	
4. Bloquer sur votre pare-feu tous les ports inférieurs à 900 à l'exception des ports dont vous avez besoin, comme le DNS (port 53).	2

NB : Chaque service UDP accessible de l'extérieur, peut être l'objet d'une attaque portant atteinte à la disponibilité d'un système. Ainsi, lorsqu'une connexion est établie entre deux services UDP, chacun produisant du trafic, ces deux services peuvent engendrer assez de trafic pour entraîner un déni de service sur les serveurs. Un simple accès au réseau suffit.

Par exemple, en connectant le service chargen d'une machine au service echo de la même ou d'une autre machine, les deux machines affectées seront rendues inutilisables à cause du nombre excessif de paquets produits. Cela peut d'ailleurs congestionner tout le réseau.

Par contre, ce type d'attaque ne permet pas d'obtenir un quelconque autre accès aux serveurs.

4.2.4 Partage de fichiers à travers le réseau

Enoncé	Importance
1. Ne pas utiliser **NFS** (Network File System) entre un réseau local et un réseau non sûr.	4
2. Utiliser une version récente et appliquer tous les patches disponibles, un grand nombre de vulnérabilités de NFS a été corrigé.	4
3. Vérifier que les permissions du fichier **/etc/exports** sont 644 et propriété de root	4
4. Monter les « file systems » distants avec les options no suid	4
5. Exporter si possible les « file systems » distants avec l'option lecture seule	3
6. Exporter les file systems en nommant explicitement la liste des machines autorisées (option -access) à effectuer le montage	3
7. Limiter les arborescences exportées: ne pas exporter tout le file system mais seulement les répertoires nécessaires	3
8. Ne pas permettre que le fichier **/etc/exports** contienne une entrée « localhost ». Un serveur NFS ne doit pas faire référence à lui même : pas d'export vers le serveur NFS lui même, partiellement ou en totalité. S'assurer en particulier que le serveur NFS n'est pas contenu dans un netgroup faisant parti de ses fichiers exports.	3
9. Utiliser la commande ' showmount -e ' pour voir ce qui est exporté.	3
10. Filtrer le trafic NFS au niveau du routeur (TCP/UDP port 111 et 2049). Ceci empêchera des machines qui ne sont pas sur le réseau local de pouvoir accéder aux fichiers exportés.	3
11. S'assurer que les listes exportées n'excèdent pas 256 caractères. (Se référer à l'avis du CERT CA-94 :02).	3
12. S'assurer que le portmapper ou le rpcbind utilisé ne permet pas de remonter par l'intermédiaire des demandes des clients. Un client NFS malintentionné peut demander au démon portmapper du serveur de faire suivre les demandes au démon mountd. Le processus du démon mount effectue la demande comme si elle venait directement de portmapper. Si le « file system » est monté en mode partagé, il donnera au client des permissions non autorisées sur ce « file system ».	2

Rappel : Sur certains Unix, les changements dans **/etc/exports** ne prennent effet qu'après avoir exécuté **/usr/etc/exportfs**, sur d'autres il faut réinitialiser le démon rpc.mountd avec la commande 'kill -HUP'.

4.2.5 Multi-fenêtrage (X)

Enoncé	Importance
1. Gérer les autorisations par machine avec la commande « xhost ». Eviter l'option permissive « xhost + », préférer les autorisations par machine (ou liste de serveurs). La liste des serveurs autorisés à accéder au serveur X réside dans le fichier /etc/Xn.hosts. Supprimer la commande dans les fichiers Xsession et .xsession	4
2. Gérer les autorisations par utilisateur avec la commande « xauth ». La clé d'authentification (X magic cookie) est stockée dans le fichier « .Xauthority ».	4
3. Utiliser de préférence X11R6 qui corrige un grand nombre de problèmes relatifs à « xdm » détectés sur les versions précédentes. Attention à xdm qui court-circuite les fonctions getty et login et donc ignore certains mécanismes de sécurité (quota, propriété de /dev/console...)	3
4. Désactiver les droits d'exécution de la commande « xev » pour les utilisateurs.	3
5. Utiliser les possibilités de verrouillage d'écran (xlock, lockscreen).	3
6. Protéger le fichier /tmp/.X11-unix/X0 contre la destruction, sinon le serveur X devient inaccessible	3
7. Contrôler périodiquement le contenu des fichiers utilisateurs .xserverrc et .xinitrc	2

4.3 Filtrage réseau

4.3.1 Routeur filtrant

La fonction de filtrage au niveau réseau est une fonction indispensable de sécurité. Elle permet d'assurer le cloisonnement de votre réseau et d'en limiter les accès.

Un simple routeur filtrant permet de réaliser le filtrage au niveau des adresses IP mais également au niveau des services (filtrage sur le numéro de port). Attention toutefois, l'association entre un numéro de port et un service est purement arbitraire. Elle est définie localement dans un fichier (**/etc/services**) configurable par l'administrateur. Toute exception aux conventions usuelles doit être documentée, justifiée et signalée au responsable sécurité.

En matière de filtrage, la règle classique de la sécurité s'applique : « tout ce qui n'est pas autorisé est interdit ». Une liste de services susceptibles d'être autorisés est proposée dans le tableau ci-dessous. Si votre réseau est connecté avec un réseau extérieur, utilisez ces services au travers de filtres.

archie	Recherche de logiciels sur Internet
datastar	Base de données
dns	Correspondance entre les noms de domaine et les adresses IP
finger	Demande d'information sur les utilisateurs
firstclass	Accès aux BBS utilisant le logiciel FirstClass
ftp	Transfert de fichiers
gopher	Interrogation des serveurs Gopher

http	Interrogation des serveurs WWW
https	Interrogation des serveurs WWW avec chiffrement
icmp	Messages de contrôle du réseau
ident	Identification d'un logiciel client
ipx	Encapsulation du protocole Novell
irc	Discussion en direct à plusieurs
logger	Journalisation des routeurs NSC
lpd	Impression à distance
netback	Sauvegarde
netbios	Encapsulation des services Netbios
nfs	Système de gestion de fichiers réparti
nntp	Lecture ou transfert des News Usenet
notes	Logiciel Lotus Notes
ntp	Fourniture du temps
patrol	Système de supervision d'applications
ping	Détection de présence de machines sur Internet
pop	Lecture de boites aux lettres à distance
radius	Serveur d'authentification
rcp	Copie de fichier à distance
realaudio	Transfert de son
routing	Protocoles de routage
rlogin	Connexion à distance
rsh	Exécution à distance
smtp	Messagerie électronique
sna	Encapsulation du protocole IBM SNA
snmp	Gestion de réseau
sqlnet	Oracle SQLNet
ssh	Exécution et connexion à distance avec chiffrement
syslog	Journalisation
tacacs	Serveur d'authentification
talk	Discussion en direct à deux
telnet	Connexion à distance
tftp	Transfert de fichier entre équipements
traceroute	Recherche des chemins d'accès à une machine
tuxedo	Moniteur transactionnel distribué
wais	Interrogation de base documentaires réparties
whois	Interrogation des bases de données de l'Internet
x11	Fenêtrage à distance
x400	Messagerie X400
x500	Annuaire X500

4.3.2 Pare-feu (firewall)

L'explosion de l'utilisation d'Internet, et ses inévitables problèmes de sécurité, a entraîné le développement de produits sécurisés de filtrage, les pare-feu (aussi appelés firewalls, garde-barrières...). Il s'agit d'équipements de sécurité et toute connexion de votre réseau avec un réseau non sûr (Internet n'est qu'un exemple) devrait être protégée par un pare-feu.

Une description des pare-feu est effectuée dans le document du CLUSIF : **A propos des pare-feu (oct. 96).**

Attention, l'installation d'un pare-feu ne dispense pas de mettre en oeuvre les fonctions de sécurité sur les machines de votre réseau : aucun n'est efficace à 100%.

4.3.3 tcp-wrappers (nommé aussi log_tcp)

Enoncé	Importance
1. L'installation du logiciel (gratuit) tcp-wrappers est vivement conseillée. Ce logiciel décrit en annexe est si utile qu'on se demande pourquoi il n'est pas livré en standard.	3
2. Interdire au départ à toute machine de se connecter vers son système (en mettant 'all : all' dans **/etc/hosts.deny)** puis indiquer explicitement les noms des machines autorisées (dans **/etc/hosts.allow**). Déclarer tous les services autorisés.	3
3. Désactiver le mode PARANOID.	2
4. Utiliser tcp-wrappers avec l'option RFC931.	2
5. « Wrapper » tous les services désactivés dans **/etc/inetd.conf**.	2

4.4 Connexions à distance

Enoncé	Importance
1. Eviter les connexions entrantes (**telnet** et surtout **rlogin**). Les services entrants entraînent un certain nombre de risques (détournement, espionnage de paquets, fausse authentification). Par contre le Telnet sortant peut être autorisé sans risque à travers un filtreur de paquets.	4
2. Si les données auxquelles on accède via Telnet sont sensibles, envisager l'emploi d'une version permettant le chiffrement.	2

4.4.1 Mécanisme de confiance et « r-commandes »

Enoncé	Importance
1. Eviter d'utiliser les "r" commandes (**rlogin, rsh, rcp, rdump, rdist**). Leur vulnérabilité réside dans le fait qu'elles s'appuient sur un mécanisme de confiance qui permet à un utilisateur de se connecter sans authentification. Désactiver ces commandes expose d'avantage à une attaque par écoute du mot de passe, mais les "r" commandes sont une source régulière d'insécurité. Choix à reconsidérer si l'écoute constitue le risque majeur.	4
2. En cas d'utilisation de ces commandes, se restreindre à une utilisation à l'intérieur du réseau local sûr. Bloquer alors les ports 512, 513 et 514 (TCP) à l'entrée de votre réseau, afin d'éviter des attaques classiques.	4
3. Vérifier régulièrement le contenu et les droits des fichiers **/etc/hosts.equiv** et **$HOME/.rhosts**	4
4. **/etc/hosts.equiv** Les droits du fichier sont 600 et il appartient à root. Le premier caractère n'est pas '-'. Eviter le symbole '+'. Le fichier ne contient pas les caractères '!' ou '#'.	4
5. Maintenir régulièrement la liste des machines autorisées (qui doit rester limitée).	4
6. Vérifier que vous n'autorisez que des machines se trouvant dans votre réseau local sûr.	4
7. **$HOME/.rhosts**	4

Enoncé	Importance
Eviter ces fichiers, qui peuvent être mis à jour par les utilisateurs non privilégiés, ou au moins contrôler régulièrement leur contenu. Vérifier que les droits du fichier sont 600 et qu'il appartient au propriétaire du compte. Vérifier que le premier caractère n'est pas '-'. Eviter le symbole '+'. Vérifier que le fichier ne contient pas les caractères '!' ou '#'. Il n'y a pas de commentaires sur ce type de fichier.	
8. Refaire les vérifications après l'installation d'un nouveau patch, où d'une modification de l'operating system de la machine.	4
9. Utiliser les versions les plus sécurisées de ces "r" commands. Il existe, en effet certaines versions qui permettent de consulter uniquement **/etc/hosts.equiv** et pas **$HOME/.rhosts**. Elles ont également une option pour invalider l'utilisation des caractères joker ('+').	3
10. Utiliser netgroups pour un management simplifié en cas d'emploi de NIS.	3
11. Utiliser tcp-wrappers pour fournir un meilleur contrôle et une meilleure journalisation sur ces services.	3

4.5 Courrier (Sendmail)

4.5.1 Configuration du sendmail

Enoncé	Importance
1. Utiliser la version de « sendmail » la plus récente (elle change fréquemment !) et corrigeant les dernières failles connues de sécurité. Elle est distribuée gratuitement (site officiel : ftp.cs.berkeley.edu, miroir : ftp.univ-lyon1.fr)	4
2. Si une version commerciale de sendmail distribuée par votre fournisseur est utilisée, s'assurez que tous les patches nécessaires sont installés. En particulier, effectuer le test suivant décrit dans ***NB*** après ce tableau.	4
3. Contrôler les options du fichier de configuration du sendmail (**sendmail.cf).** En particulier, supprimer toute ligne commençant par 'OW' (référence à Wizard).	4
4. Définir un niveau de log système (syslog) au minimum du niveau 'debug' mail.debug. Ecrivez dans **/dev/console** et dans **/var/adm/messages**.	3
5. Définir un niveau de log au minimum de niveau 9 (0-99). Cela aide à s'apercevoir d'une attaque si tel est le cas.	3
6. Prélever des messages d'erreur pour observer s'il existe des messages provenant ou partant d'un pipe (« | »), ou des messages de ou vers un utilisateur inconnu (exemple: toto, zorro, ...).	3
7. Utiliser smrsh si vous avez besoin d'une fonctionnalité de classement automatique du courrier (ex : « progmail ») Si cette fonctionnalité n'est pas utilisé, il faut interdire les mails vers des programmes, en mettant « /bin/false » dans ce champ dans le fichier de configuration de sendmail.	2

NB : test pour connaître la version utilisée de sendmail et d'éventuelles failles de sécurité

% telnet Machine 25

wiz

debug

kill

quit

%

Vous devriez avoir la réponse suivante « 500 Command unrecognized » après avoir émis les commandes ‘wiz’, ‘debug’ et ‘kill’. Si ce n’est pas le cas votre version de sendmail est vulnérable et devra être mise à jour.

4.5.2 Les fichiers importants

Enoncé	Importance
1. Le fichier **aliases,** souvent placé dans /etc, contient la liste des alias. Mettre en commentaire l’alias ‘decode’ en mettant un « # » au début de la ligne. Pour que ce changement soit pris en compte faire tourner **/usr/bin/newaliases**. S’assurer que tous les programmes pouvant être exécutés par un alias appartiennent à root, avec la permission 755, et qu’ils résident dans un répertoire système comme par exemple **/usr/local/bin**.	4
2. Les fichiers **.forward** dans chaque compte utilisateur contiennent la liste des redirections des messages de l’utilisateur. Ils doivent être protégés (modifiables et lisibles seulement par l’utilisateur). Leur contenu doit être licite (vérifier les redirections, supprimer les caractères interprétés comme le « | ».	4

4.6 Transferts de fichiers

4.6.1 ftp

4.6.1.1 Configuration d’un serveur ftp

Enoncé	Importance
1. Utiliser une version de serveur ftp la plus récente et corrigeant les dernières failles connues de sécurité. Utiliser par exemple le washington university ftpd (wu-ftpd).	4
2. En cas d’utilisation de la version commerciale de ftp distribuée par votre fournisseur, s’assurer que tous les patches nécessaires sont installés.	4
3. Contrôler toutes les options de configuration par défaut de votre serveur ftp	4
4. Vérifier que votre serveur ftp ne possède pas la commande SITE EXEC	4
5. Vérifier qu’il existe un fichier /etc/ftpusers spécifiant les utilisateurs non autorisés à se connecter à votre serveur ftp. Celui-ci devra contenir au moins les entrées correspondant aux comptes système : root, bin, uucp, daemon, news, nobody et toutes les demandes des comptes de vendeurs	4

4.6.1.2 ftp anonyme

Enoncé	Importance
1. Contrôler toutes les options de configuration par défaut de votre serveur ftp.	4
2. Utiliser l’option de changement de racine afin que le serveur ftp travaille en environnement restreint (la plupart des Unix récents prennent cette option par	4

Enoncé	Importance
défaut).	
3. Toutes les versions de ftp ne sont pas configurables. Si la version utilisée est configurable (ex :wu-ftp) s'assurer que les options : All delete, overwrite, rename, chmod, unmask... ne sont pas autorisées aux invités et aux utilisateurs anonymes	4
4. Vérifier que la version n'inclut pas un interpréteur de commandes (shell, perl,...) dans ~ftp/bin qui puisse être exécuté par SITE EXEC. Seules quelques commandes comme uncompress gunzip résident à ces endroits mais il faut être conscient que la présence de ces commandes peut permettre à un utilisateur d'obtenir des accès non autorisés	4
5. Etre très prudent dans l'inclusion de commandes qui peuvent exécuter d'autres commandes. Par exemple, quelques versions de tar permettent d'exécuter un fichier quelconque.	4
6. Vérifier qu'il n'existe pas de copie du fichier /etc/passwd réel en tant que ~ftp/etc/passwd (idem pour /etc/group). Le fichier passwd ne doit contenir le nom d'aucun compte du fichier réel mais il doit contenir seulement root et ftp et des entrées fictives à mots de passe inactifs (Root:*:0:0:Ftp::).	4
7. Vérifier que les fichiers ~ftp/.rhosts et ~ftp/.forward n'existent pas	4
8. Installer le login shell du compte ftp à un shell inactif tel que /bin/false	4
9. Pour désactiver un ftp anonyme, changer ou supprimer tous les fichiers en ~ftp/ et changer le user ftp de votre fichier de mots de passe	4

4.6.1.3 Gestion des permissions

Enoncé	Importance
1. Vérifier que les droits du répertoire ~ftp sont 555 ainsi que les sous-répertoires, afin que les utilisateurs du ftp anonyme puissent uniquement lire les informations.	4
2. Vérifier que le propriétaire est root et non ftp.	4
3. Vérifier que les droits des répertoires ~ftp/etc et ~ftp/bin et fichiers ~ftp/bin/* sont 111, propriétaire root.	4
4. Vérifier qu'il y a un alias mail permettant de transférer le mail et vérifier que /usr/spool/mail/ftp est propriété de root, autorisation 400.	4
5. Si les utilisateurs doivent déposer des fichiers dans le serveur ftp, se limiter à un répertoire accessible en écriture qui devra être propriété de root, permission 1733.	4
6. Vérifier que le répertoire accessible en écriture ne l'est pas en lecture.	4
7. Mettre le répertoire accessible en écriture accessible en lecture dans une partition séparée.	4
8. Ne jamais monter un disque d'une autre machine dans la hiérarchie ftp sans qu'il soit uniquement configuré en lecture seule.	4
9. Vérifier que tous les fichiers et sous-répertoires appartiennent au compte ftp.	4

4.6.2 rcp

Consulter le paragraphe sur les commandes « r ».

4.6.3 Trivial ftp (tftp)

Enoncé	Importance
1. Si tftpd n'est pas nécessaire le mettre en commentaire dans le fichier inetd.conf et redémarrer le process inetd.	4
2. S'il est nécessaire, utiliser l'option permettant de limiter les répertoires accessibles (ex : tftpd -s).	4
3. Limiter les accès par du filtrage au niveau du routeur.	4

4.6.4 UUCP (Unix-to-Unix Copy Protocol)

Enoncé	Importance
1. Se référer à la documentation du fournisseur pour déterminer les options et les paramètres de sécurité. UUCP est encore utile notamment pour des échanges au dessus de liaisons séries ou X25.	4
2. Vérifier qu'aucun fichier uucp n'est autorisé à tout le monde en écriture.	4
3. Vérifier que le nombre de commandes que chaque uucp login peut exécuter est limité.	4
4. Retirer tous les fichiers .rhosts de la home directory du uucp.	4
5. Vérifier qu'il existe un login uucp différent pour chaque site ayant besoin de se connecter en uucp.	3
6. Considérer la possibilité de supprimer l'ensemble du sous-système uucp s'il n'est pas nécessaire et si l'administrateur n'en maîtrise pas le fonctionnement. Supprimer le compte uucp. Vérifier qu'il n'y a plus d'entrées dans les crontabs de uucp ou de root.	2

4.7 Web ou WWW (World Wide Web)

Enoncé	Importance
1. Pour faire tourner un serveur Web, utiliser un logiciel récent qui intègre la notion de sécurité.	4
2. Le propriétaire du démon serveur httpd doit être un utilisateur non privilégié créé spécialement, comme par exemple « httpd ». Grâce à cette méthode, si un pirate trouve une vulnérabilité dans le serveur, il n'obtiendra que les privilèges d'accès associé à cet utilisateur non privilégié.	4
3. Ne pas faire tourner le démon serveur en tant que root	4
4. Ne pas faire tourner les processus clients en tant que root	4
5. Faire tourner httpd dans un environnement restreint (chroot permet un changement de racine) afin de limiter les accès des clients http par rapport au reste du disque.	4
6. Etudier attentivement les options de configuration de son serveur	4
7. Utiliser les options de configuration sécurisée permettant par exemple à des fichiers d'être inclus dans les documents HTML (suppression de la propriété « include files »).	4
8. Eviter les programmes CGI (Common Gateway Interface) s'ils ne sont pas absolument nécessaires	4
9. Etre vigilant dans la conception de programmes CGI. Leurs failles sont fréquemment exploitées par les individus hostiles, car ils peuvent permettre l'exécution de commandes arbitraires sur le serveur. La plupart des vulnérabilités des serveurs Web sont issues de cette faiblesse.	4

Enoncé	Importance
10. S'assurer que le contenu, les permissions et le propriétaire des fichiers dans le répertoire des scripts CGI (ex : cgi-bin) sont sécurisés	4
11. Fournir les scripts CGI comme des binaires liés statiquement plutôt que comme des scripts interprétés. Cela évitera la nécessité d'un interpréteur de commandes à l'intérieur de l'environnement restreint.	3
12. Eviter le passage direct entre l'entrée utilisateur et les interpréteurs de commande comme Perl, Awk, les shells ou les programmes qui permettent d'encapsuler des commandes dans des messages sortants comme la commande mail.	3
13. Utiliser CGIWRAP	3
14. Filtrer l'entrée utilisateur pour supprimer les caractères potentiellement dangereux, avant qu'ils soient passés aux interpréteurs de commandes. Les caractères qui peuvent être dangereux sont les caractères suivants : \n \r (.,/;~!)>|^&$`< (se référer à l'avis du Cert CA-95 :04).	3

5. ANNEXE : COMPLEMENTS

5.1 Synthèse pour le RSSI

Les investigations et les recommandations proposées dans les chapitres précédents du présent document peuvent être mises en oeuvre manuellement avec les commandes standard Unix. Cependant certains logiciels complémentaires s'avèrent fort utiles.

Ainsi, il est beaucoup plus efficace de réaliser ces investigations à l'aide d'outils tels que les logiciels d'audit gratuits COPS ou TIGER. L'utilisation de ces outils apporte les avantages suivants :

- recherche exhaustive, pour chaque vulnérabilité traitée, sur l'ensemble du système (fichiers, comptes...) ;
- possibilité d'activation périodique et programmée de l'outil d'audit qui prend ainsi en compte les évolutions du système et génère un rapport sur les nouvelles vulnérabilités détectées.

Dans tous les cas, la correction de chaque vulnérabilité constatée reste (et doit rester) à la charge de l'administrateur du système.

Un grand nombre d'outils logiciels ont été écrits pour optimiser la sécurité des stations Unix et distribués gratuitement par leurs auteurs. L'intérêt de ces outils est indéniable et la liste proposée ci-après contient ceux jugés à priori les plus utiles. Certains sont redondants, c'est à chacun de choisir ceux qui sont le mieux adaptés à ses besoins. On peut noter qu'il existe également des produits commerciaux réalisants des fonctions analogues.

Pour les outils gratuits (appelé aussi « freeware »), il faut vérifier l'intégrité de la version récupérée, ce qui ne dispense pas de vérifier le contenu. Par prudence, n'utilisez des logiciels gratuits que si vous en possédez les sources.

Une raison supplémentaire d'utiliser de tels outils est que les pirates les ont également à leur disposition.

Certains organismes s'intéressent à la sécurité des systèmes Unix et des réseaux. Une liste est fournie ci-après. Ils diffusent des informations fort utiles sur les failles de sécurité des différents systèmes. Les logiciels de sécurité, mentionnés ci-dessus, peuvent être également récupérés sur leurs sites. Ils constituent l'une des sources d'information privilégiées concernant la sécurité des Réseaux et Systèmes Ouverts, que l'administrateur doit consulter régulièrement.

5.2 Les outils de sécurité les plus utiles

5.2.1 Outils généraux

1. **COPS**
   Computer Oracle and Password System. Cet ensemble d'outils passe en revue un certain nombre de failles localement sur un système et prévient l'utilisateur (administrateur) lorsqu'il rencontre un problème. ➔***ftp://coast.cs.purdue.edu/pub/tools/unix/scanners/cops***

2. **Tiger**
   Ensemble de procédures d'audit, comparable à COPS, mais plus à jour et plus simple à configurer et à utiliser. ➔***ftp://coast.cs.purdue.edu/pub/tools/unix/scanners/tiger***

3. **Tcp_wrappers**
   Appelé aussi log_tcp, ce programme permet de surveiller et de filtrer les requêtes entrantes pour les services réseau systat, finger, ftp, telnet, rlogin, rsh, exec, tftp, talk , ... Tcp_wrappers ne demande aucune modification des logiciels existants. Il journalise le nom de la machine client ainsi que le nom du service demandé, filtre suivant l'adresse ou le hostname ainsi que sur l'identité du demandeur, puis si les filtres l'autorisent, donne la main au serveur démon requis. Tcp_wrappers est transparent pour l'utilisateur. ➔***ftp ://coast.cs.purdue.edu/pub/tools/unix/netutils/tcp_wrappers/***

4. **Tripwire**
   Produit de contrôle d'intégrité de fichiers et répertoires qui repose sur différents éléments propres à chaque fichier: permissions, n° d'inode, nb de liens, uid et gid, dates, tailles ainsi que (et surtout) 2 signatures (par défaut MD5 et Snefru). Cet ensemble d'éléments entrant dans le calcul d'intégrité est configurable fichier par fichier.

   ➔***ftp ://coast.cs.purdue.edu/pub/tools/unix/ids/tripwire***

5. **xautolock**
   Logiciel de contrôle d'écran automatique sous X window. Il lance un programme au choix (en général une fonction de verrouillage telle que lockscreen, xlock...) après un certain temps (réglable) d'inactivité.

6. **tcpdump**
   Produit assimilable à l'etherfind de Sun. Il capture les paquets IP et les affiche. Un certain nombre de filtres permet à l'utilisateur de ne visualiser que ce qui l'intéresse. ➔***http ://www.tcpdump.org***

7. **cpm**
   Produit permettant de vérifier que l'interface réseau d'une station n'est pas en mode « promiscuous » (mode destiné à l'analyse de réseau). Ce logiciel est conçu pour SunOS 4.1.x et peut fonctionner sur de nombreux systèmes BSD.

   ➔***ftp://ftp.cerias.purdue.edu/pub/tools/unix/sysutils/cpm***

8. **smrsh**
   Logiciel permettant de restreindre les shell que peut lancer le démon de messagerie sendmail.

9. **smap**
   Logiciel frontal à sendmail destiné à corriger certaines failles de sécurité.

10. **ssh**
    Logiciel de connexion à distance sécurisée. Il utilise du chiffrement dont l'algorithme est paramétrable.

11. **PGP**
    Pretty Good Privacy est un produit de signature et de chiffrement de fichier. Son utilisation n'est pas encore autorisée en France.

12. **md5**
    Logiciel permettant de contrôler l'intégrité d'un fichier par la connaissance d'un sceau.

13. **anlpasswd**
    Logiciel de vérification des mots de passe. Il effectue une série de tests lorsque l'utilisateur choisit son mot de passe et le refuse s'il est jugé comme trop faible.

14. **OPIE**
    Logiciel permettant de s'authentifier de manière sûre vis-à-vis de l'écoute réseau. OPIE remplace les mots de passe standard par des mots de passe à usage unique. ➔***http ://inner.net/opie***

15. **swatch**
    Simple Watcher. Permet de fabriquer des alarmes (avertissement par e-mail, par pager, ...) et faire de la supervision. ➔***http://sourceforge.net/project/showfiles.php?group_id=68627***

16. **xinetd**
    Logiciel analogue à tcp_wrappers pour inetd. ➔***http://www.xinetd.org/***

17. **identd**
    Logiciel permettant l'identification inverse (du serveur vers le client) de l'utilisateur.

5.2.2 Outils d'audit

1. **Crack**
   Ce produit permet de rechercher des mots de passe à partir de mots contenus dans un dictionnaire. ➔ ***ftp://coast.cs.purdue.edu/pub/tools/unix/pwdutils/crack/***

2. **SATAN**
   Security Administrator Tool for Analyzing Networks. Ce logiciel scrute les différents système connectés sur le réseau et utilise les vulnérabilités les plus couramment exploitées portant sur ftp, nfs, NIS, rexd, tftp, sendmail et les serveurs X. Pour chaque faille détectée, le produit fournit un compte rendu et la solution à envisager.

   ➔***ftp://coast.cs.purdue.edu/pub/tools/unix/scanners/satan/***

3. **ISS**
   Internet Security Scanner. Logiciel analogue à Satan. Il sonde le système sur les bogues logiciels et les erreurs de configuration communément exploitées par les pirates.

   ➔ ***ftp://coast.cs.purdue.edu/pub/tools/unix/scanners/iss/***

Les versions les plus récentes sont commercialisées.

5.3 Les organismes de sécurité

Les organismes suivants, dont la liste ne se prétend pas exhaustive, permettent d'obtenir des informations intéressantes dans le domaine de la sécurité Unix. Certains sont spécialisés dans ce domaine, d'autres sont des généralistes mais la sécurité constitue cependant l'un des thèmes importants.

AFUU (Association Française des Utilisateurs d'Unix)

L'AFUU est un cadre indépendant structuré et opérationnel regroupant utilisateurs et professionnels, couvrant l'ensemble des aspects stratégiques et techniques des systèmes d'information modernes, et proposant : des échanges d'expériences avec d'autres utilisateurs, des rencontres de fournisseurs avec discussions techniques contradictoires et approfondies, des rencontres avec des experts, des pointeurs vers les utilisateurs et les fournisseurs les plus avisés pour répondre à nos besoins. (Ne traite pas fondamentalement des problèmes de sécurité sous Unix, mais fournit tout un ensemble de services en rapport avec Unix).

➔ ***http://www.afuu.fr/***

CERT-CC (Computer Emergency Response Team - Coordination Center)

Le CERT Coordination Center est l'organisation qui s'est développée à partir de l'équipe de réaction aux urgences formée par la Defense Advanced Research Projects Agency (DARPA). Son but est de travailler avec la communauté Internet pour améliorer sa capacité de réactions aux événements de sécurité informatique ; de mener une campagne de sensibilisation sur les problèmes de sécurité informatique ; enfin, de mener des recherches destinées à améliorer la sécurité des systèmes existants.

Les produits et les services du CERT-CC comprennent une assistance technique 24 heures sur 24 pour réagir à des incidents de sécurité, fournir une assistance sur les faiblesses des produits, fournir des documents et organiser des séminaires. De plus, l'équipe maintient un grand nombre de listes de diffusion et fournit un serveur FTP : ***ftp://info.cert.org***, où des documents relatifs à la sécurité, les précédents avis du CERT-CC et des outils sont archivés.

➔ ***http://www.cert.org/*** (**CERT-CC**). Pour tous les sites Internet.

➔ ***http://www.auscert.org.au/*** (**AUSCERT**). Pour les sites australiens.

➔ ***http://www.cert.dfn.de/eng/*** (**CERT-DFN**). Le CERT allemand regroupant les informations des CERT européens.

➔ ***http://www.renater.fr/Securite/CERT_Renater.htm*** (**CERT-RENATER**). Le CERT français regroupant les universités, le ministère de la recherche et de l'éducation, le CNRS, le CEA, l'INRIA, le CNES, l'INRA, l'IFREMER et l'EDF.

CIAC (Computer Incident Advisory Capability)

Le CIAC a été créé en 1989, il fournit des services dans le domaine de la sécurité informatique aux employés et aux sociétés travaillant pour le Département de l'énergie du Lawrence Livermore Laboratory des Etats-Unis.

➔ ***http://www.ciac.org/ciac/***

CLUSIF (CLUb de la Sécurité des systèmes d'Information Français)

Le CLUSIF a été fondé en 1984, pour offrir un cadre dans lequel les intervenants dans le domaine de la sécurité des systèmes d'information, responsables et prestataires de services peuvent se rencontrer, confronter leurs points de vue, travailler et progresser ensemble.

A ce jour, le CLUSIF rassemble plus de deux cent cinquante membres, appartenant à cent cinquante organismes ou sociétés, et a conservé, avec les mêmes finalités globales, cette particularité d'accueillir aussi

bien les utilisateurs que les offreurs, fondant sa culture sur une égale participation des uns et des autres et son équilibre sur une confrontation permanente de l'offre et de la demande.

➔ ***http://www.clusif.asso.fr/***

COAST (Computer Operations, Audit and Security Technology)

Il poursuit un ensemble de projets et d'enquêtes de recherche en sécurité informatique au Computer Science Department de l'Université de Purdue. Il est destiné à fonctionner en étroite liaison avec les chercheurs et les ingénieurs des grandes compagnies et des agences gouvernementales.

➔ ***http://www.cs.purdue.edu/coast/coast.html***

CRU (Comité Réseau des Universités)

Le CRU est constitué par des représentants de différents établissements d'enseignement supérieur (Universités, Grandes-Écoles), qui exercent une compétence et une responsabilité dans le domaine communication et réseau. Il apporte une aide pour l'élaboration des politiques «communication» au sein des établissements. Un correspondant CRU est désigné au niveau de chaque établissement. Un certain nombre de ses activités sont menées en collaboration avec l'UREC.

Il fédère les ressources humaines afin de pouvoir mener des activités dans les domaines suivants :

- évolution du réseau national de la recherche, nouvelles technologies, définition des besoins des établissements ;
- mise en oeuvre, suivi et évolution des services de base (DNS, NIC, News, messagerie...) ;
- introduction, validation des services documentaires (Archie, Gopher, Wais, WWW, Annuaires...) ;
- évaluation d'outils de travail collaboratif, vidéoconférence, enseignement à distance ;
- coordination des actions en matière de sécurité.

➔ ***http://www.cru.fr***

FIRST (Forum of Incident Response and Security Team)

Il est constitué d'une trentaine d'équipes intégrant des représentants de l'Administration, de l'industrie, des constructeurs de matériel informatique et des universités américaines ou d'obédience internationale. Le FIRST est chargé de mettre en place des unités de riposte individuelle (individual response teams), chargées d'établir des procédures de traitement des incidents dans leurs champs de compétence, elles doivent être en mesure de communiquer avec les autres services du FIRST.

➔ ***http://www.first.org/***

IN2P3 (Institut National de Physique Nucléaire et de Physique des Particules) - Sécurité Renater - Région P.A.C.A.

Créé en 1971, l'IN2P3 est un institut du CNRS dont la mission est de promouvoir et de fédérer les activités de recherche de ses laboratoires en physique nucléaire et en physique des particules. L'IN2P3 finance et coordonne l'activité de 18 laboratoires (dont trois sont des laboratoires mixtes CNRS-CEA) et un centre de calcul. Dans cet ensemble travaillent environ 2600 personnes.

➔ ***http://info.in2p3.fr/secur/home.html***

NIST (National Institute of Standards and Technology's) - Computer Security Resource Clearinghouse

Le NIST est un organisme d'état qui dépend du ministère du Commerce des Etats-Unis, et qui propose des normes et des recommandations (par exemple, le document « Mettre en place des capacités de riposte aux incidents de sécurité » - 800-3) auxquelles se conforment les autorités fédérales et le secteur privé. Son

principal intérêt est l'information sur la réaction à une crise, sur les menaces, les faiblesses et les solutions liées à la sécurité informatique.

➔ ***http://csrc.nist.gov/***

OSSIR (Observatoire de la Sécurité des Systèmes d'Information & des Réseaux) (ex-groupe SUR)

Le Groupe de Travail Sécurité a été créé en 1987 sous l'égide de l'AFUU (Association Française des Utilisateurs d'Unix et des Systèmes Ouverts) par Humberto Lucas (Gould/Encore, actuellement directeur de EUnet France) et Yvon Klein (Bull, actuellement au SCSSI). Il a été ensuite animé par Christian Pélissier (Thomson, puis ONERA), pour être actuellement géré par Hervé Schauer (HSC) et Daniel Azuelos (Institut Pasteur). En 1995, le groupe a quitté l'AFUU qui l'avait vu naître en restant indépendant des fournisseurs. Il se caractérise par une participation hétérogène, intégrant des fournisseurs et des utilisateurs de tous les horizons. Il traite principalement des sujets concernant de l'Internet, y compris ce qui se rapporte au protocole TCP/IP et au système d'exploitation Unix en général. -->

➔ ***http://www.ossir.org/ (http://www.sur.fr.net/)***

UREC (Unité Réseaux du CNRS)

L'Unité Réseaux du CNRS est une unité de service du CNRS dont la mission est d'organiser, de promouvoir et de développer les services de réseaux au CNRS et d'aider les utilisateurs à s'en servir. Pour cela, l'UREC participe étroitement au développement du réseau Renater qui est l'infrastructure privilégiée pour le développement des communications. L'UREC apporte une assistance technique aux campus et aux laboratoires qui le souhaite pour le choix et la mise en solution des solutions réseaux adaptées à leur environnement : câblage, interconnexion de réseaux locaux, accès à RENATER et à l'Internet, expertise technique etc ...

L'UREC diffuse une information sur les réseaux, les produits, les services et la sécurité. Cette diffusion est organisée sur son serveur d'information accessible par ftp anonyme, gopher et WWW.

➔ ***http://www.urec.fr/securite/***

USENIX

Depuis 1975, l'association USENIX a rassemblé la communauté des ingénieurs, des scientifiques et des techniciens travaillant à la pointe de l'informatique. Les conférences et les ateliers techniques sont devenus les fondations pour la présentation et la discussion des informations les plus avancées sur les développements de tous les aspects des systèmes informatiques.

➔ ***http://www.usenix.org/***