Lycée
BAHUET
Réussir au cœur d'un campus nature

Lycée Bahuet
Brive

DEBIAN 11

Luka Grimaud, Clément Vincent, Ethan Quesney Florian Bessaguet

SOMMAIRE

I- Introduction

Dans un monde ou la Sécurité est primordial, les OS ne sont pas à négligé. Si on veut connaître un maximum d'information avec une sécurité optimal il nous faut d'abord choisir un OS de qualité dans notre cas nous avons choisie Debian 11 qui est optimisé pour ça facilité d'utilisation est son accessibilité avec un Wiki bien détaillé. On peut aussi noter que c'est le noyau de nombre distribution système Linux, celui-ci est un open source.

II- Système de fichier Debian / linux

Debian possède deux architectures principales :

- La première est AMD64 pour les ordinateurs en 64bits
- Le second est le I386 pour les ordinateurs 32 bits.

Comme tout système d'exploitation Debian possède un système de fichier = le système Ext4 à base UNIX qui est un multi-utilisateur et un multitâche.
Ext4 est apparu pour la première fois à la version de Debian JESSIE soit Debian 8. Il est introduit dans le noyau 2.6.28. Il peut gérer des volumes d'une taille allant jusqu'à un exaoctet (260 octets).

III- Les mises à jour et les gestions paquets

Il y a plusieurs types de mises à jour :

- Celles de son système
- Celles de ses applications.

Pour faire cela, on va passer par différentes applications ou méthodes.

Pour faire les mises à jour il y a une méthode dite à commande où l'on exécute toutes les commandes dans un terminal : apt-get update permettant de mettre à jour le paquet que l'on veut. Nous pouvons aussi faire des installations de paquets.
Ensuite, il y a des applications de gestion de pack à interface graphique.
Le premier c'est Aptitude : c'est celui qui fournit les commandes apt-get en question plus haut. Il permet aussi d'accéder facilement aux différentes versions de paquet, de faire des recherches de logiciels obsolètes sous « Paquets obsolètes ou créés localement ».
Le second c'est Synaptic qui est un gestionnaire de paquets à interface graphique qui est très simple à utiliser sans ligne de commande.

IV- Gestion des Utilisateurs

1- Debian utilise plusieurs outils d'administration :

a) les outils standard:

gnome-system-tools

L'outil de gestion des utilisateurs est défini par défaut de Gnome Desktop (connu dans le projet original sous le nom user-admin)

adduser

Adduser est le principal outil en ligne de commande pour ajouter, retirer ou modifier les utilisateurs et les groupes dans Debian.

b) Autres outils :

rcconf

Il permet de choisir facilement quels démons doivent être activés au démarrage de l'ordinateur. Par exemple, si vous ne souhaitez pas être accueilli par un écran de connexion graphique, vous pouvez désactiver xdm, gdm ou kdm.

modconf

Modconf fournit une interface graphique pour installer et configurer les modules des pilotes des périphériques.

cfengine

Il fournit un "moteur de règle" où vous décrivez comment votre ou vos ordinateurs doivent être configurés et gérés, puis il applique ces règles et fait des mises à jour incrémentales de la configuration des systèmes. cfengine devient vraiment indispensable quand vous passez de la gestion d'une douzaine de serveurs à plusieurs centaines. Dans une telle situation, devoir se connecter à chaque système pour modifier une configuration de DNS n'est vraiment plus possible : vous avez

2- Gestion des utilisateurs sous Debian

Toutes les commandes ci-dessous doivent être effectuées avec les droits **sudo**.

Ajouter un utilisateur :

adduser nom_du_nouvel_utilisateur

Changer le mot de passe d'un utilisateur :

passwd nom_utilisateur

Supprimer un utilisateur :

Supprimer l'utilisateur et son répertoire home

deluser --remove-home nom_utilisateur_a_supprimer

Créer un groupe :

addgroup nom_du_nouveau_groupe

Modifier un utilisateur :

Changer son groupe primaire :

usermod -g nom_du_groupe nom_utilisateur

Ajouter un groupe à un utilisateur :

usermod -aG nom_du_groupe nom_utilisateur

Ou, à condition d'avoir déjà crée l'utilisateur et le groupe :

adduser nom_utilisateur nom_groupe

Affecter plusieurs groupes en même temps à un utilisateur :

usermod -G groupe 1, groupe 2, groupe 3 nom_utilisateur

Renommer un utilisateur :

Attention, cela ne renomme pas son répertoire home

usermod -1 nouveau_nom ancien_nom

Modifier (et afficher) les informations d'un utilisateur :

chfn nom_utilisateur

Éventuellement, pour modifier ou afficher un seul paramètre :

chfn [option] nom_utilisateur

Afficher les groupes auxquels appartient un utilisateur :

groups nom_utilisateur

Afficher la liste des utilisateurs :

cat /etc/passwd | awk -F: '{print $ 1}'

Ou

cat /etc/passwd

Ou simplement aller lister les dossiers dans /home, mais les homes des différents utilisateurs n'ont pas forcément le nom de leur utilisateur si l'utilisateur a été renommé.

Afficher la liste des groupes :

cat /etc/group | awk -F: '{print $ 1}'

Ou

cat /etc/group

Prendre la place d'un user

su - nom_de_l_user

Ou

sudo -s -u nom_de_l_user

V- Gestion des droits

Linux est résolument multi-utilisateur ; il est donc nécessaire de prévoir un système de permissions contrôlant les opérations autorisées pour chacun sur les fichiers et répertoires, recouvrant toutes les ressources du sys
tème.

Chaque fichier ou répertoire dispose de permissions spécifiques pour trois catégories d'utilisateurs :

> Son propriétaire (symbolisé par **u** comme user) ;
> Son groupe propriétaire (symbolisé par **g** comme group) — représentant tous les utilisateurs membres du groupe ;
> Les autres (symbolisés par **o** comme other).

Trois types de droits de base peuvent être combinés :

> Lecture (symbolisé par **r** comme read) ;

> Écriture (ou modification, symbolisé par **w** comme write) ;
> Exécution (symbolisé par **x** comme exécute).

Trois commandes manipulent les permissions associées à un fichier:

> **chown *utilisateur fichier*** affecte un nouveau propriétaire à un fichier;
> **chgrp *groupe fichier*** opère sur son groupe propriétaire;
> **chmod *droits fichier*** intervient sur ses droits.

Il existe deux manières de présenter les droits ; parmi elles, la représentation symbolique, sans doute la plus simple à comprendre et mémoriser, met en jeu les lettres symboles déjà citées. Pour chaque catégorie d'utilisateurs (**u/g/o**), on peut définir les droits (**=**), en ajouter (**+**), ou en retrancher (**-**).

VI- La sécurité

"Debian prend les questions de sécurité très au sérieux. Nous traitons tous les problèmes de sécurité qui sont portés à notre attention et nous nous assurons qu'ils sont corrigés dans un délai raisonnable"

Cela nous montre bien l'investissement de Debian pour faire vivre leurs systèmes, ils sont à la recherche des failles de sécurité

Sécuriser un système Debian n'est pas différent de la sécurisation d'un autre système. Mais il y a quand même qu'elle que condition à suivre pour avoir une configuration d'OS optimal. Pour ce faire, je vais vous dire et décrire les conditions à suivre.

Changer le mot de passe du BIOS.

Avant d'installer un système d'exploitation sur l'ordinateur, créez un mot de passe pour le BIOS. Après l'installation (une fois que vous avez rendu possible un démarrage à partir du disque dur), retournez dans le BIOS et changez la séquence de démarrage afin de rendre impossible le démarrage à partir d'une disquette, d'un CD ou de tout autre périphérique. Sinon un pirate n'a besoin que d'un accès physique et d'une disquette de démarrage pour accéder au système complet.

Désactiver le démarrage sans mot de passe est une solution encore meilleure. Cela peut être très efficace pour un serveur car il n'est pas redémarré très souvent. L'inconvénient de cette méthode est que le redémarrage nécessite l'intervention d'une personne, ce qui peut poser des problèmes si la machine n'est pas facilement accessible.

Il faut créer une partition pour le système

Pendant le partitionnement du système, vous devez également décider du système de fichiers à utiliser. Le système de fichiers choisi par défaut pendant l'installation de Debian pour les partitions Linux est **ext3**, un système de fichiers journalisé. Vous devriez toujours utiliser un système de fichiers journalisé comme **ext3**, **reiserfs**, **jfs** ou **xfs** pour réduire les problèmes découlant d'un plantage système dans mon cas je préconiserais d'utiliser le système de fichiers **ext3**. La raison pour cela est qu'il est rétro compatible

avec **ext2**, donc s'il y a un quelconque problème avec la journalisation, vous pouvez la désactiver et toujours avoir un système de fichiers fonctionnel. De plus, si vous avez besoin de récupérer le système avec une disquette d'amorçage (ou un CD), vous n'avez pas besoin d'un noyau personnalisé.

Le système ne devrait pas être connecté à Internet pendant l'installation. Cela peut paraître stupide mais il faut savoir que l'installation par le réseau est une méthode d'installation habituelle. Étant donné que le système va installer et activer les services immédiatement, si le système est connecté à Internet et que les services ne sont pas configurés correctement, vous les exposez à des attaques.

Étant donné que l'installation et les mises à jour peuvent être faites par Internet, vous pourriez penser que c'est une bonne idée d'utiliser cette caractéristique lors de l'installation. Si le système va être connecté directement à Internet (et pas protégé par un pare-feu ou un NAT), il est plus judicieux de l'installer sans connexion à Internet et d'utiliser un miroir local de paquets contenant à la fois les paquets source et les mises à jour de sécurité. Vous pouvez mettre en place des miroirs de paquets en utilisant un autre système connecté à Internet et des outils spécifiques à Debian (si c'est un système Debian) tels que apt-move ou apt-proxy ou tout autre outil de miroir pour fournir l'archive aux systèmes installés.

Définir un mot de passe pour le superutilisateur

Définir un bon mot de passe est la condition de base pour avoir un système sécurisé.

Faire une mise à jour de sécurité

Avant de vous connecter à tout réseau, particulièrement s'il s'agit d'un réseau public, vous devriez, au minimum, faire une mise à jour de sécurité Pour mettre à jour vous-même le système, ajoutez la ligne suivante dans le **sources.list** et vous recevrez les mises à jour de sécurité automatiquement quand vous mettrez à jour le système. Remplacez [NOM] par le nom de la version, par exemple : *squeeze*.

```
deb http://security.debian.org/ [NOM]/updates main contrib non-free
```

Après avoir fait cela, plusieurs outils vous permettent de mettre à niveau le système. S'il s'agit d'un ordinateur de bureau, une application appelée **update-notifier** permet de vérifier facilement si de nouvelles mises à niveau sont disponibles. En choisissant cela, vous pouvez faire les mises à niveau depuis le bureau (en utilisant **update-manager**).

Une fois que vous avez exécuté une mise à jour de sécurité, il se peut que vous deviez redémarrer certains des services système. Si vous ne faites pas cela, certains services pourraient encore être vulnérables après une mise à jour de sécurité. La raison pour cela est que les démons qui fonctionnent avec une mise à jour peuvent encore utiliser les anciennes bibliothèques après la mise à jour.

Restreindre les redémarrages système depuis la console

Si vous désirez restreindre cela, vous devez vérifier le fichier **/etc/inittab** pour que la ligne incluant **ctrlaltdel** appelle **shutdown** avec le paramètre **-a**.

La valeur par défaut dans Debian inclut ce paramètre :

```
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

Le paramètre **-a**, conformément à la description de la page de manuel shutdown(8), donne la possibilité de permettre à *certains* utilisateurs d'arrêter le système. Pour cela, le fichier

/etc/shutdown.allow doit être créé et inclure le nom des utilisateurs qui peuvent redémarrer le système. Quand la combinaison du *salut à trois doigts* est exécutée en console, le programme va vérifier si l'un des utilisateurs définis dans ce fichier est connecté. Si aucun d'entre eux ne l'est, **shutdown** ne va *pas* redémarrer le système.

Pour désactiver la combinaison Ctrl+Alt+Del, il suffit de commenter la ligne contenant la définition de *ctrlaltdel* dans **/etc/inittab**.

N'oubliez pas d'exécuter **init q** après toute modification du fichier **/etc/inittab** pour qu'elle prenne effet.

Utilisation de tcpwrappers

Pour des services configurés dans **/etc/inetd.conf**, cela comprend **telnet**, **ftp**, **netbios**, **swat** et **finger**, vous observerez que le fichier de configuration exécute avant tout **/usr/sbin/tcpd**. D'un autre côté, même si un service n'est pas lancé par le super démon **inetd**, il peut être compilé avec la prise en charge pour les règles d'encapsulation TCP. Les services suivant sont compilés avec prise en charge d'encapsulation TCP dans Debian : **ssh**, **portmap**, **in.talk**, **rpc.statd**, **rpc.mountd**, **gdm**, **oaf** (le démon d'activation GNOME), **nessus** et beaucoup d'autres.
Pour voir quels paquets utilisent tcpwrappers, essayez :

```
$ apt-cache rdepends libwrap0
```

À présent, voici une petite astuce et probablement le plus petit système de détection d'intrusions disponible. Généralement, vous devriez disposer d'une politique correcte concernant le pare-feu en première ligne, puis disposer de l'encapsulation TCP en seconde ligne de défense. Un petit truc est de mettre en place une commande SPAWN [34] dans /etc/hosts.deny qui enverra un courrier au superutilisateur quand un service refusé déclenche l'encapsulation :

```
ALL: ALL: SPAWN ( \
   echo -e "\n\
   Encapsulation TCP \: Connexion refusée\n\
   Par \: $(uname -n)\n\
   Processus \: %d (pid %p)\n\
   Utilisateur \: %u\n\
   Hôte \: %c\n\
   Date \: $(date)\n\
 " | /usr/bin/mail -s "Connexion à %d bloquée" root) &
```

Attention: L'exemple ci-dessus peut-être facilement sujet à une attaque par déni de service en soumettant énormément de connexions dans une période très courte. De nombreux courriers signifient de nombreuses E/S en envoyant uniquement quelques paquets.

L'importance des journaux et des alertes

Debian GNU/Linux fournit quelques outils pour effectuer des analyses de journaux, notamment **swatch**[35], **logcheck** ou **log-analysis** (tous ont besoin d'être personnalisés pour enlever les choses non nécessaires des comptes-rendus). Il peut être également utile, si le système est proche, d'avoir les journaux du système affichés sur une console virtuelle. C'est utile car vous pouvez (de loin) voir si le système se comporte correctement. Le fichier **/etc/syslog.conf** de Debian est fourni avec une configuration commentée par défaut; pour l'activer, décommenter les lignes et redémarrez **syslogd** (**/etc/init.d/syslogd restart**:

```
daemon,mail.*;\
      news.=crit;news.=err;news.=notice;\
      *.=debug;*.=info;\
      *.=notice;*.=warn       /dev/tty8
```

Utiliser et personnaliser logcheck

Le paquet **logcheck** dans Debian est divisé en trois paquets **logcheck** (le programme principal), **logcheck-database** (une base de données d'expressions rationnelles pour le programme) et **logtail** (affiche les lignes du journal qui n'ont pas encore été lues). Le comportement par défaut sous Debian (dans **/etc/cron.d/logcheck**) est que **logcheck** est exécuté toutes les heures et une fois après le démarrage.
Cet outil peut être assez utile s'il est personnalisé correctement pour alerter l'administrateur d'événements système inhabituels. **logcheck** peut être complètement personnalisé pour envoyer des courriers selon les événements récupérés des journaux et qui sont dignes d'attention. Le meilleur moyen de configurer **logcheck** est d'éditer son fichier de configuration principal **/etc/logcheck/logcheck.conf** après l'avoir installé. Vous devriez également y positionner le niveau de compte-rendu. **logcheck-database** a trois niveaux de compte-rendu de verbosité croissante: station de travail, serveur, paranoïaque. «serveur» étant le niveau par défaut, «paranoïaque» n'est recommandé que pour les machines de haute sécurité ne faisant fonctionner qu'aussi peu de services que possible et «station de travail» est pour les machines relativement protégés et non critiques. Si vous désirez ajouter de nouveaux fichiers journaux, ajoutez-les simplement à **/etc/logcheck/logcheck.logfiles**. Celui-ci est configuré pour une installation de syslog par défaut.

Sécurisation des transferts de fichiers

Pendant l'administration normale du système, il est habituellement nécessaire de transférer des fichiers à partir et vers le système installé. La copie des fichiers de façon sécurisée d'un hôte vers un autre peut être effectuée en utilisant le paquet serveur **ssh**. Une autre possibilité est d'utiliser **ftpd-ssl**, un serveur FTP qui utilise *Secure Socket Layer* pour chiffrer les transmissions.

Utilisation de quotas

Avoir une bonne politique relative aux quotas est important, vu qu'elle empêche les utilisateurs de remplir les disques durs.

Vous pouvez utiliser deux systèmes de quotas différents : les quotas utilisateur et les quotas groupe. Comme vous l'avez probablement deviné, les quotas utilisateur limitent la quantité d'espace qu'un utilisateur peut avoir, les quotas groupe quant à eux font la même chose pour les groupes. Retenez cela quand vous calculerez les tailles des quotas. L'activation des quotas pour des systèmes de fichiers différents est aussi facile que la modification du paramètre **defaults** en **defaults,usrquota** dans le fichier **/etc/fstab**. Si vous avez besoin des quotas par groupe, remplacez **usrquota** par **grpquota**. Vous pouvez également utiliser les deux. Ensuite, créez des fichiers vides quota.user et quota.group à la racine du système de fichiers sur lequel vous voulez utiliser les quotas :

touch
/home/quota.user /home/quota.group

Pour un système de fichiers **/home**)
Redémarrez **quota** en faisant:

/etc/init.d/quota stop;/etc/init.d/quota
start

Maintenant les quotas devraient être en fonction et leurs tailles peuvent être configurées.
L'édition de quotas pour un utilisateur spécifique peut être réalisée en faisant
edquota -u <user>
. Les quotas par groupes peuvent être modifiés avec :
edquota -g <group>

Sécurisation des accès réseau

FIXME : Besoin de plus de contenu (spécifique à Debian).

Configuration des options réseau du noyau

En entrant **/sbin/sysctl -A**, vous pouvez voir ce que vous pouvez configurer et quelles sont les options, elles peuvent être modifiées en exécutant :
/sbin/sysctl -w variable=valeur
(Consultez sysctl(8)). Vous aurez seulement en de rares occasions à éditer quelque chose ici, mais vous pouvez augmenter la sécurité de cette manière aussi. Par exemple :
net/ipv4/icmp_echo_ignore_broadcasts = 1

C'est un « émulateur Windows » parce que ça agit comme Windows sur les ping de broadcast si celui-ci est positionné à 1. C'est-à-dire que les requêtes d'echo ICMP envoyées à l'adresse broadcast seront ignorées. Sinon, cela ne fait rien.

Si vous voulez empêcher le système de répondre aux requêtes d'echo ICMP, activez cette option de configuration :
net/ipv4/icmp_echo_ignore_all = 1

Pour enregistrer les paquets avec des adresses impossibles (à cause de routes erronées) sur le réseau, utilisez :
/proc/sys/net/ipv4/conf/all/log_martians = 1

Configurer syncookies

Cette option est à double tranchant. D'un côté, elle protège le système contre le syn packet flooding; d'un autre côté, elle viole les standards définis (RFCs).
net/ipv4/tcp_syncookies = 1

Si vous voulez changer cette option à chaque fois que le noyau fonctionne, vous devez le faire dans **/etc/network/options** en positionnant **syncookies=yes**. Cela prendra effet à chaque fois que **/etc/init.d/networking** est exécuté (ce qui est habituellement fait lors du démarrage) tandis que la commande suivante aura un effet unique jusqu'au prochain redémarrage:
echo 1 > /proc/sys/net/ipv4/tcp_syncookies

Cette option n'est dispobile que si vous avez compilé le noyau avec **CONFIG_SYNCOOKIES**. Tous les noyaux Debian sont compilés avec cette option incluse, mais vous pouvez le vérifier en exécutant :
$ sysctl -A |grep syncookies
net/ipv4/tcp_syncookies = 1

Sécurisation du réseau pendant l'amorçage

Un exemple de fichier de configuration **/etc/sysctl.conf** qui sécurisera quelques options de réseau au niveau du noyau est présenté ci-dessous. Notez les commentaires dans ce fichier, **/etc/network/options** peut forcer certaines options si elles sont en contradiction avec celles de ce fichier lors de l'exécution de **/etc/init.d/networking** (ce qui a lieu après **procps** dans la séquence de démarrage).

```
#
# /etc/sysctl.conf - Fichier de configuration pour positionner les
# variables système
# Consultez sysctl.conf(5) pour plus de renseignements. Consultez
# également les fichiers sous Documentation/sysctl/,
# Documentation/filesystems/proc.txt et
# Documentation/networking/ip-sysctl.txt dans les sources du noyau
# (/usr/src/kernel-$version si vous avez installé un paquet de noyau)
# pour plus d'informations sur les valeurs qui peuvent être définies ici.
```

```
#
# Attention : /etc/init.d/procps est exécuté pour positionner les
# variables suivantes. Cependant, après cela, /etc/init.d/networking
# positionne certaines options réseau avec des valeurs intrinsèques. Ces
# valeurs peuvent être forcées en utilisant /etc/network/options.
#
#kernel.domainname = example.com

# Paramètres supplémentaires - adapté du script fourni
# par Dariusz Puchala (voir ci-dessous)
# 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 en
# homme au milieu)
net/ipv4/conf/all/accept_redirects = 0
# _ou_
# N'accepter les redirections ICMP que pour les passerelles
# de notre liste de 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 »
anet/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 martiens
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
```

Pour utiliser le script, vous devez tout d'abord le créer, par exemple, dans **/etc/network/interface-secure** (le nom est donné comme exemple) et l'appeler à partir de **/etc/network/interfaces** comme ceci:

```
auto eth0
iface eth0 inet static
```

```
    address xxx.xxx.xxx.xxx
    netmask 255.255.255.xxx
    broadcast xxx.xxx.xxx.xxx
    gateway xxx.xxx.xxx.xxx
    pre-up /etc/network/interface-secure
```

Dans cet exemple, avant que l'interface eth0 ne soit activée, le script sera appelé pour sécuriser toutes les interfaces réseau comme montré ci-dessous.

```
#!/bin/sh -e
# Nom du script : /etc/network/interface-secure
#
# Modification de plusieurs comportements par défaut pour sécuriser contre
# certaines attaques et usurpations IP pour toutes les interfaces.
#
# Fourni par Dariusz Puchalak.
#

# Activation de la protection broadcast echo.
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Désactivation de l'IP forwarding.
echo 0 > /proc/sys/net/ipv4/conf/all/forwarding

# Activation de la protection TCP syn cookies.
echo 1 > /proc/sys/net/ipv4/tcp_syncookies


# Enregistrement des paquets avec des adresses impossibles
# (cela comprend les paquets usurpés (spoofed), les paquets routés
# source, les paquets redirigés), mais faites attention à cela
# sur les serveurs web très chargés.
echo 1 >/proc/sys/net/ipv4/conf/all/log_martians

# Activation de la protection sur les mauvais messages d'erreur.
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

# Protection d'usurpation IP.
echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter

# Désactivation des redirections ICMP.
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
echo 0 > /proc/sys/net/ipv4/conf/all/send_redirects

# Désactivation des paquets source routés.
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route

exit 0
```

Remarquez que vous pouvez en fait avoir des scripts par interface qui activeront différentes options réseau pour différentes interfaces (si vous en avez plus d'une), il vous suffit de changer la ligne pre-up en:

```
pre-up /etc/network/interface-secure $IFACE
```

Et utiliser un script qui n'applique les changements qu'à une interface spécifique et non à toutes les interfaces disponibles.

Vous pouvez également créer un script **init.d** et le faire exécuter au démarrage (en utilisant **update-rc.d** pour créer les liens **rc.d** appropriés).

Configuration des fonctionnalités de pare-feu

De façon à avoir des privilèges de pare-feu, soit pour protéger le système local ou d'autres *derrière* lui, le noyau doit être compilé avec les options correspondant aux pare-feu. Le noyau standard de Debian 3.0 (Linux 2.4) fournit lui le pare-feu **iptables** (netfilter).
Une configuration correcte de pare-feu serait donc une règle de refus par défaut, c'est-à-dire :

- les connexions entrantes ne sont autorisés que pour des services locaux par des machines autorisées ;
- les connexions sortantes ne sont autorisés que pour les services utilisés par votre système (DNS, navigation web, POP, courrier, etc.) ;
- la règle forward interdit tout (à moins que vous ne protégiez d'autres systèmes, voir ci-dessous) ;
- toutes les autres connexions entrantes et sortantes sont interdites.

Un pare-feu Debian peut aussi être installé de façon à protéger, selon des règles de filtrage, l'accès aux systèmes *derrière* lui, limitant leur exposition à Internet. Un pare-feu peut être configuré pour interdire l'accès de systèmes en dehors de votre réseau local à des services internes (ports) qui ne sont pas publics. Par exemple, sur un serveur de messagerie, seul le port 25 (où le service de courrier est fourni) doit être accessible depuis l'extérieur. Un pare-feu peut être configuré pour, même s'il y a d'autres services en plus des services publics, rejeter les paquets (c'est connu sous le nom de*filtrage*) dirigés vers eux.

Mettre en place un pare-feu

Bien sûr, la configuration du pare-feu dépend toujours du système et du réseau. Un administrateur doit connaître auparavant quelle est la disposition du réseau, les systèmes qu'il désire protéger et si d'autres considérations réseau (comme le NAT ou le routage) doivent être prises en compte ou non. Soyez prudent quand vous configurez votre pare-feu, comme le dit Laurence J. Lane dans son paquet **iptables** : *Les outils peuvent facilement être mal utilisés, entraînant d'énormes quantités de maux en paralysant complètement l'accès au réseau pour un système d'ordinateur. Il n'est pas très inhabituel pour un administrateur système de se bloquer lui-même en dehors du système situé à quelques centaines ou milliers de kilomètres de là. Il est même possible de se bloquer en dehors d'un ordinateur dont le clavier est sous ses doigts. Veuillez s'il vous plaît l'utiliser avec précaution.*
Rappelez-vous de cela : installer simplement le paquet **iptables** (ou l'ancien code de pare-feu) ne vous fournit pas de protection, mais seulement les logiciels. Pour avoir un pare-feu, vous devez le *configurer* !
Si vous ne savez pas comment configurer les règles de votre pare-feu manuellement, veuillez consulter le *Packet Filtering HOWTO* et le *NAT HOWTO* fournis par **iptables** pour une lecture hors ligne à **/usr/share/doc/iptables/html/**.
Configurer manuellement vos règles de pare-feu par un script init.d qui exécutera toutes les commandes **iptables**. Suivez les étapes ci-dessous.

- Consultez le script ci-dessous et adaptez-le à vos besoins.
- Testez le script et vérifiez les messages du journal système pour voir le trafic qui est rejeté. Si vous testez depuis le réseau, vous voudrez soit exécuter le script shell en exemple qui enlève le pare-feu (si vous ne tapez rien pendant 20 secondes), soit commenter les définitions de règle *default deny* (*-P INPUT DROP* et *-P OUTPUT DROP*) et vérifier que le système ne rejette pas de trafic légitime.
- Déplacez le script dans **/etc/init.d/parefeu**.

Voici l'exemple de script de pare-feu :

```
#!/bin/sh
# Exemple de configuration de pare-feu.
#
# Mises en garde
```

```
# - Cette configuration s'applique à toutes les interfaces réseau.
#   Si vous voulez ne restreindre cela qu'à une interface donnée,
#   utilisez « -i INTERFACE » dans les appels iptables ;
# - L'accès à distance pour les services TCP/UDP est accordé à tout
#   hôte, vous voudrez probablement restreindre cela en utilisant
#   « --source ».
#
# chkconfig : 2345 9 91
# description : activer ou désactiver le pare-feu au démarrage
#
# Vous pouvez tester ce script avant de l'appliquer avec l'extrait
# de script shell suivant, si vous ne tapez rien pendant
# 20 secondes, les règles de pare-feu seront effacées.
#-----------------------------------------------------------------
#  while true; do test=""; read  -t 20 -p "OK ? " test ; \
#  [ -z "$test" ] && /etc/init.d/parefeu clear ; done
#-----------------------------------------------------------------

PATH=/bin:/sbin:/usr/bin:/usr/sbin

# Services que le système offrira au réseau
TCP_SERVICES="22" # seulement SSH
UDP_SERVICES=""
# Services que le système utilisera du réseau
REMOTE_TCP_SERVICES="80" # navigation web
REMOTE_UDP_SERVICES="53" # DNS
# Réseau qui sera utilisé pour la gestion à distance
# (si non défini, aucune règle ne sera mise en place)
# NETWORK_MGMT=192.168.0.0/24
# Port utilisé pour le service SSH, à définir si vous avez configuré
# une gestion de réseau mais l'avez enlevé de TCP_SERVICES
# SSH_PORT="22"

if ! [ -x /sbin/iptables ]; then
   exit 0
fi

fw_start () {

 # Trafic d'entrée :
 /sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
 # Services
 if [ -n "$TCP_SERVICES" ] ; then
 for PORT in $TCP_SERVICES; do
  /sbin/iptables -A INPUT -p tcp --dport ${PORT} -j ACCEPT
 done
 fi
 if [ -n "$UDP_SERVICES" ] ; then
 for PORT in $UDP_SERVICES; do
  /sbin/iptables -A INPUT -p udp --dport ${PORT} -j ACCEPT
 done
 fi
 # Gestion à distance
 if [ -n "$NETWORK_MGMT" ] ; then
  /sbin/iptables -A INPUT -p tcp --src ${NETWORK_MGMT} --dport ${SSH_PORT} -j ACCEPT
```

```
 else
  /sbin/iptables -A INPUT -p tcp --dport ${SSH_PORT}  -j ACCEPT
 fi
 # Test à distance
 /sbin/iptables -A INPUT -p icmp -j ACCEPT
 /sbin/iptables -A INPUT -i lo -j ACCEPT
 /sbin/iptables -P INPUT DROP
 /sbin/iptables -A INPUT -j LOG

 # Sortie :
 /sbin/iptables -A OUTPUT -j ACCEPT -o lo
 /sbin/iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
 # ICMP est permis :
 /sbin/iptables -A OUTPUT -p icmp -j ACCEPT
 # Ainsi que les mises à jour de sécurité :
 # Remarque : vous pouvez indiquer en dur l'adresse IP ici afin de prévenir une
 # usurpation DNS et configurer les règles même si le DNS ne fonctionne pas mais
 # dans ce cas vous ne « verrez » pas les modifications d'IP pour ce service :
 /sbin/iptables -A OUTPUT -p tcp -d security.debian.org --dport 80 -j ACCEPT
 # Ainsi que pour tous les services définis :
 if [ -n "$REMOTE_TCP_SERVICES" ] ; then
 for PORT in $REMOTE_TCP_SERVICES; do
  /sbin/iptables -A OUTPUT -p tcp --dport ${PORT} -j ACCEPT
 done
 fi
 if [ -n "$REMOTE_UDP_SERVICES" ] ; then
 for PORT in $REMOTE_UDP_SERVICES; do
  /sbin/iptables -A OUTPUT -p udp --dport ${PORT} -j ACCEPT
 done
 fi
 # Toutes les autres connexions sont enregistrées dans syslog
 /sbin/iptables -A OUTPUT -j LOG
 /sbin/iptables -A OUTPUT -j REJECT
 /sbin/iptables -P OUTPUT DROP
 # Autres protections réseau
 # (certaines ne fonctionneront que pour certaines versions de noyau)
 echo 1 > /proc/sys/net/ipv4/tcp_syncookies
 echo 0 > /proc/sys/net/ipv4/ip_forward
 echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
 echo 1 > /proc/sys/net/ipv4/conf/all/log_martians
 echo 1 > /proc/sys/net/ipv4/ip_always_defrag
 echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
 echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter
 echo 0 > /proc/sys/net/ipv4/conf/all/send_redirects
 echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route

}

fw_stop () {
 /sbin/iptables -F
 /sbin/iptables -t nat -F
 /sbin/iptables -t mangle -F
 /sbin/iptables -P INPUT DROP
 /sbin/iptables -P FORWARD DROP
 /sbin/iptables -P OUTPUT ACCEPT
```

```
}

fw_clear () {
 /sbin/iptables -F
 /sbin/iptables -t nat -F
 /sbin/iptables -t mangle -F
 /sbin/iptables -P INPUT ACCEPT
 /sbin/iptables -P FORWARD ACCEPT
 /sbin/iptables -P OUTPUT ACCEPT
}


case "$1" in
 start|restart)
  echo -n "Démarrage du pare-feu…"
  fw_stop
  fw_start
  echo "done."
  ;;
 stop)
  echo -n "Arrêt du pare-feu…"
  fw_stop
  echo "done."
  ;;
 clear)
  echo -n "Effacement des règles de pare-feu…"
  fw_clear
  echo "done."
  ;;
 *)
  echo "Utilisation : $0 {start|stop|restart|clear}"
  exit 1
  ;;
 esac
exit 0
```

Au lieu d'intégrer toutes les règles iptables dans un script init.d, vous pouvez utiliser le programme **iptables-restore** pour restaurer les règles sauvées avec **iptables-save**. Pour faire cela, vous devez configurer les règles et sauver le jeu de règles dans un endroit statique (comme **/etc/default/firewall**).

Tester la configuration de pare-feu

Tester la configuration de pare-feu est aussi facile et aussi dangereux que d'exécuter simplement le script de pare-feu (ou d'activer la configuration que vous avez définie dans l'application de configuration de pare-feu). Cependant, si vous n'êtes pas assez prudent et que vous configurez le pare-feu à distance (comme à travers une connexion SSH), vous pourriez vous enfermer dehors.
Plusieurs moyens permettent d'empêcher cela. L'un est d'exécuter un script dans un terminal séparé qui va enlever la configuration de pare-feu si vous ne faites pas d'entrée clavier. Un exemple de cela est :

```
$  while true; do test=""; read  -t 20 -p "OK? " test ; \
 [ -z "$test" ] && /etc/init.d/firewall clear ; done
```

Un autre moyen est d'introduire une porte dérobée dans le système par un mécanisme alternatif qui vous permet soit d'enlever le système de pare-feu, soit de percer un trou dedans si quelque chose déraille. Pour cela, vous pouvez utiliser **knockd** et le configurer pour qu'une tentative de connexion sur un certain port enlève le pare-feu (ou ajoute une règle temporaire). Bien que les paquets soient rejetés par le pare-feu, comme **knockd** se lie à l'interface et les *voit*, vous pourrez contourner le problème.

Sécurisation de SSH

SSH devrait être utilisé pour toutes les connexions distantes
Vous devriez éviter de vous connecter au système en utilisant SSH en tant que superutilisateur et préférer l'utilisation de méthodes alternatives pour devenir superutilisateur comme **su** ou **sudo**. Enfin, le fichier **sshd_config**, dans **/etc/ssh**, devrait être modifié comme suit pour accroître la sécurité.

- Ne faîtes écouter SSH que sur une interface donnée, juste au cas où vous en ayez plus d'une (et ne voulez pas que SSH y soit disponible) ou si vous ajoutez, dans le futur, une nouvelle carte réseau (et ne voulez pas de connexions SSH dessus).
- Essayez autant que possible de ne pas autoriser de connexion en tant que superutilisateur. Si quelqu'un veut devenir superutilisateur par SSH, deux connexions sont maintenant nécessaires et le mot de passe du superutilisateur ne peut être attaqué par force brute par SSH.
- **Port 666** ou **ListenAddress 192.168.0.1:666** change le port d'écoute, ainsi l'intrus ne peut être complètement sûr de l'exécution d'un démon sshd (soyez prévenus, c'est de la sécurité par l'obscurité).
- Autorise seulement certains utilisateurs à avoir accès par SSH à cette machine. **user@host** peut également être utilisé pour n'autoriser l'accès qu'à un utilisateur donné depuis un hôte donné.
- Désactiver toute forme d'autorisation dont vous n'avez pas réellement besoin si vous n'utilisez pas, par exemple, **RhostsRSAAuthentication**, **HostbasedAuthentication**, **KerberosAuthentication** ou **RhostsAuthentication**, vous devriez les désactiver même s'ils le sont déjà par défaut (consultez la page de manuel sshd_config(5)).

Vous pouvez également restreindre l'accès au serveur ssh en utilisant **pam_listfile** ou **pam_wheel** dans le fichier de contrôle PAM. Par exemple, vous pourriez bloquer tous les utilisateurs qui ne sont pas dans **/etc/loginusers** en ajoutant cette ligne à **/etc/pam.d/ssh** :
auth required pam_listfile.so sense=allow onerr=fail item=user file=/etc/loginusers
Pour finir, soyez conscient que ces directives proviennent d'un fichier de configuration OpenSSH. Actuellement, trois démons SSH sont couramment utilisés, ssh1, ssh2, et OpenSSH par les gens d'OpenBSD. ssh1 était le premier démon SSH disponible et est toujours le plus couramment utilisé (il y a même des rumeurs à propos d'un portage pour Windows). ssh2 a beaucoup d'avantages par rapport à ssh1 sauf qu'il est diffusé sous une licence non libre. OpenSSH est un démon SSH complètement libre, qui gère à la fois ssh1 et ssh2. OpenSSH est la version installée sur Debian quand le paquet **ssh** est choisi.

Clients SSH

Si vous utilisez un client SSH pour se connecter au serveur SSH, vous devez vous assurer qu'il prend en charge les mêmes protocoles que ceux utilisés sur le serveur. Par exemple, si vous utilisez le paquet **mindterm**, il ne prend en charge que le protocole version 1. Cependant, le serveur sshd est, par défaut, configuré pour n'accepter que la version 2 (pour des raisons de sécurité).

Interdire les transferts de fichiers

Si vous ne voulez *pas* que les utilisateurs transfèrent des fichiers depuis et vers le serveur ssh, vous devez restreindre l'accès au **sftp-server** *et* l'accès **scp**. Vous pouvez restreindre **sftp-server** en configurant le bon **Subsystem** dans **/etc/ssh/sshd_config**.

VII- Configuration minimale

Type d'installation	RAM (minimum)	RAM (recommandée)	Disque dur
Sans bureau	128 mégaoctets	512 mégaoctets	2 gigaoctets
Avec bureau	256 mégaoctets	1 gigaoctet	10 gigaoctets

La quantité de mémoire minimale réellement nécessaire est inférieure à celle donnée dans le tableau. Selon l'architecture, il est possible d'installer Debian avec 20 Mo (sur s390) ou 60 Mo (sur amd64).

Il n'est pas possible d'indiquer des valeurs minimales concernant la mémoire et l'espace disques nécessaires à un serveur. Tout dépend d'utilisation qui en sera faite.

VIII- **Les avis utilisateur**

Les utilisations:	**Advantages:**	**Inconvenient:**
Nabin Secteurs d'activité : Logiciel Utilisation toutes les semaines depuis plus de 2 ans	- Système d'exploitation gratuit et open source - nouvelle version de Debian publiée toute les 2 ans	- Debian ne peut pas toujours être à jour avec les correctifs de sécurité - assistance limitée au niveau de l'entreprise
Eric Logiciel Utilisation tous les jours pendant 2 ans	- équilibre parfait entre disponibilité des nouveaux paquets et la stabilité du système - Debian = meilleur gestionnaire de paquet	- inférieur à d'autre distribution en tant que distributeurs de bureau (exemple de Ubuntu).
Santiago Divertissement Utilisation toutes les semaines pendant 6 à 12 mois	- très adapté aux différents matériels, personnalisable et sûr.	- mises à jour plus difficiles à installer que les autres distributions.
Ronald Hôpitaux et soins de santé Utilisation tous les jours pendant plus d'un an	- excellente interface utilisateur - distribution très bien orienté service = excellent laboratoire pour le scénario à tester	- besoin de nombreuse dépendance pour pouvoir installer n'importe quel autre outil au sein du système d'exploitation

IX- Conclusion

Pour conclure, Debian 11 peut remplacer Windows facilement dans une entreprise avec ses nombreuses fonctionnalités, il y a une bonne partie de la gestion administrative et des droits utilisateurs avec différentes commandes à disposition. Il dispose d'une bonne sécurité facile d'accès grâce à un bon wiki fait par la communauté. Il faut faire certaines recherches pour faire les mises à jour, dans une entreprise ce sont les administrateur système qui s'en occupent. En termes de performance Debian 11 est beaucoup moins gourmand que Windows.