Sécurité des systèmes UNIX/Linux et administration avancée

Sécurisation du système Linux Fedora

Dossier réalisé par Romain Quarré
Étudiant en 5ème année d'ingénierie en SR à l'ESGI
Dernière mise à jour le 29/05/09

Informations sur la machine

- Hostname de la machine sécurisée: machineTP
- Espace disque total : 8 Go
- Administrateur standard : admin
- Utilisateur standard : romain
- Adresse IP statique : 192.168.1.233

[admin@machineTP ~]$ uname –r
2.6.27.5-117.fc10.i686

Autre machine sur le réseau pour les tests:

- Fedora 10 (même noyau)
- roms@romain avec l'IP : 192.168.1.142

Installation automatique via le cd

- On peut récupérer la dernière version de Fedora sur le site web officiel que l'on installera sous Virtual Box:
  (NB : Je n'ai PAS trouvé de versions sans packages préinstallés)
  http://www.fedora-fr.org/

- Mot de passe root (demandé lors de l'installation automatique) = **aTg6;,88hytV+**
  A ce stade la première étape de sécurisation est de choisir un mot de passe complexe résistant aux attaques par brute force. (Suffisamment long et mélangeant caractères alphanumériques, chiffres et symboles)

- Lors du partitionnement du système, choisir « **create custom layout** » pour définir son propre plan de partitionnement.
  Le plan de partitionnement choisi pour le média **/dev/sda** est le suivant :

  - Une nouvelle partition au point de montage **/boot/** (200Mo, partition primaire)
  - Une nouvelle partition au point de montage **/var/** (1500Mo)
  - Une nouvelle partition au point de montage **/home/** (2Go)
  - Une nouvelle partition **swap** (2Go= taille mémoire vive)
  - Une nouvelle partition au point de montage **/** (Restant de l'espace disque)

/boot est la partition primaire et contient donc le noyau. Les autres partitions ont été créés pour prévenir d'un risque de crash (système rendu instable par exemple) (**/var** et **/home** étant souvent remplis de données) . Il y a également une partition swap dans le cas ou le système ne disposerait plus assez de mémoire vive.

- Mettre ensuite un mot de passe complexe au bootloader pour contrôler les autorisations de modification de boot: **fdxjèY6"/l/**

- Création d'un utilisateur « admin » ayant moins de droit que root mais servant pour l'administration

- Activer le protocole NTP pour que les logs et applications soient ordonnés sur une même base de temps entre autre

Délégation de droit à l'utilisateur 'admin'

- L'utilisateur « admin » est un utilisateur qui administre TOUT le système. Aussi, nous allons lui donner tous les droits d'administration dans le fichier sudoers (sans restriction de type service networking, software , etc..) en le rajoutant au groupe « wheel » . Nous allons donc insister sur la sécurisation de cet utilisateur par la suite, puisque nous lui accordons des droits importants d'administration.

1- ***Activer le groupe « wheel » dans /etc/sudoers***

```
[admin@machineTP ~]$ su –
Password:
[root@machineTP ~]#  visudo  -f   /etc/sudoers
```

On décommente la ligne pour activer le groupe.

```
…
## Allows people in group wheel to run all commands
 %wheel ALL=(ALL)       ALL
…
```

2- ***Ajouter l'utilisateur 'admin' dans le groupe wheel***

```
[root@machineTP ~]# vi  /etc/group
```

```
…
kmem:x:9:
wheel:x:10:root,admin
mail:x:12:mail
uucp:x:14:uucp
…
```

Configuration de base du système

- Mettre a jour le système

```
[admin@machineTP ~]$  sudo  yum  update kernel*
[admin@machineTP ~]$  sudo  yum  update
```

- Configuration réseau: comme l'OS va partager des ressources et que la disponibilité du système de doit pas dépendre d'un service DHCP (DHCP hors service ou changement de bail), je fais le choix de mettre une configuration réseau de type statique au système. Attention, puisque l'on va gérer le réseau manuellement il faut faire un 'sudo yum remove NetworkManager' pour supprimer ce programme car celui-ci écrase le fichier /etc/resolv.conf et on perd ainsi les DNS rentrés manuellement au démarrage suivant.

```
[admin@machineTP ~]$   cd   /etc/sysconfig/network-scripts/
[admin@machineTP network-scripts]$  sudo  vi   ifcfg-eth0
```

Ce fichier comportera les informations suivantes:

```
DEVICE=eth0
HWADDR=08:00:27:b0:4d:aa
ONBOOT=yes
BOOTPROTO=static
BROADCAST=192.168.1.255
IPADDR=192.168.1.233
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
USERCTL=no
NETWORK=192.168.1.0
```

Puis une fois le fichier sauvé, on relance le script de démarrage du réseau :

```
[admin@machineTP ~]$ sudo   /etc/init.d/network   restart
Shutting down interface eth0:                  [  OK  ]
Shutting down loopback interface:              [  OK  ]
Bringing up loopback interface:                [  OK  ]
Bringing up interface eth0:                    [  OK  ]
Bringing up interface pan0:                    [  OK  ]
[admin@machineTP ~]$
```

- Suppression des packages inutiles : Un OS qui contient un grand nombre de packages offre autant de failles potentielles pour un pirate. Il faut donc supprimer les packages inutiles à l'aide de la commande « yum remove <nom_du_package> » . **Dans le cadre de ce TP je n'ai pas trouvé comment installer une version de fedora sans packages préinstallés. De ce fait, mon OS contient un bon nombre de paquets inutiles. Il serait trop long de tous les désactiver un par un, aussi j'ai donné la manière de procéder de manière théorique et en pratique, il faudrait le faire (ou installer un système minimaliste puis les packages utiles un a un).**

- Configuration DNS de l'OS (mode serveur) : Le serveur doit être client DNS d'un serveur DNS de confiance pour déléguer la gestion des noms. En effet, en centralisant la gestion des noms sur un serveur DNS, on évite de configurer le fichier /etc/hosts de chaque serveur pour la traduction des noms. On vérifie dans le fichier /etc/nsswitch.conf que la ligne indiquant la manière de gérer la traduction n'est pas désactivée.

```
[admin@machineTP ~]$  more  /etc/nsswitch.conf  | grep  hosts
#hosts:    db  files  nisplus  nis  dns
hosts:     files  mdns4_minimal [NOTFOUND=return]  dns
[admin@machineTP ~]$
```

Pour ajouter un serveur de nom, on édite le fichier /etc/resolv.conf avec l'adresse IP du serveur DNS

```
[admin@machineTP ~]$  cat  /etc/resolv.conf
# Generated by NetworkManager



# No nameservers found; try putting DNS servers into your
# ifcfg files in /etc/sysconfig/network-scripts like so:
#
```

```
# DNS1=xxx.xxx.xxx.xxx
# DNS2=xxx.xxx.xxx.xxx
# DOMAIN=lab.foo.com bar.foo.com
nameserver  192.168.1.1
[admin@machineTP ~]$
```

- Désactivation du démarrage interactif du processus init pour pas qu'un utilisateur lambda désactive un service.

```
[admin@machineTP ~]$  sudo  vi   /etc/sysconfig/init
[sudo] password for admin:
```

Mettre la valeur no a la variable PROMPT

```
[admin@machineTP ~]$  more   /etc/sysconfig/init  |  grep  PROMPT
PROMPT=no
[admin@machineTP ~]$
```

- Une interface graphique X windows permet l'exploit de failles potentielles et est un élément non obligatoire qui doit être désactivé. De plus , un bon nombre de packages préinstallés s'exécutent en mode graphique et seront donc inutilisables sans interface graphique

```
[admin@machineTP ~]$  sudo vi  /etc/inittab
```

Dans ce fichier on passe du runlevel 5 (avec interface graphique) à 3 (sans)

```
#
id:3:initdefault:
```

Si cela est possible, on va tenter de supprimer X Windows du système, ainsi que d'autres groupes de packages présents sur la machine dont les groupes sont donnés par la commande « yum grouplist »:

```
[admin@machineTP ~]$  sudo   yum groupremove  "X Window System"
[admin@machineTP ~]$  sudo   yum groupremove  "GNOME  Desktop Environment"
…
On supprime tous les groupes qui ne sont pas indispensables (trop long pour ce TP)
```

- Renforcement de la sécurité du programme init : Il faut éviter que lorsque le système démarre en mode « single user », un shell soit proposé à l'utilisateur sous l'identité root sans mot de passe. On va éditer /etc/inittab et rentrer avant la ligne du runlevel la commande suivante :

```
[admin@machineTP ~]$ cat  /etc/inittab  |  grep sulogin
~:S:wait:/sbin/sulogin
[admin@machineTP ~]$
```

- Protection de la pile TCP/IP : Il est nécessaire de protéger la pile contre des attaques potentielles exploitant des failles de sécurité ou la négligence de l'utilisateur au travers de clés rentrées dans le fichier /etc/sysctl.conf

```
[admin@machineTP ~]$ sudo sysctl -p
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.all.proxy_arp = 0
net.ipv4.icmp_ignore_bogus_error_responses = 1
error: "net.ipv4.icmp_echo_ignore_bogus_error_responses" is an unknown key
net.ipv4.icmp_echo_ignore_broadcasts = 1
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.tcp_syncookies = 1
[admin@machineTP ~]$
```

Le fichier /etc/sysctl.conf gère la manière dont doit se comporter la pile TCP/IP. Dans les options rentrées dans le fichier, entre autre, on désactive le routage sur la machine, on désactive les messages de diffusion ICMP , etc.. la commande sysctl -p active ces valeurs pour protéger la pile TCP/IP.
IL y a un message d'erreur que je ne comprend pas pour la valeur :
net.ipv4.icmp_echo_ignore_bogus_error_responses.
A mon avis, il se peut que cette clé soit une clé qui a été supprimée sur cette version de fedora et qui marchait sur d'anciennes versions .

- Mise en place de TCP_WRAPPER : ce démon installé par défaut sert de couche supplémentaire en plus du firewall iptables et permet de filtrer les services réseaux qui l'utilisent. Ce démon se base principalement sur 2 fichiers : /etc/hosts.allow d'abord pour vérifier qui a le droit d'accéder à quel service et /etc/hosts.deny ensuite, pour interdire les accès aux services. Le contenu de ces fichiers tel que je l'ai définit est le suivant :

```
[admin@machineTP xinetd.d]$ cat /etc/hosts.allow | grep :
sshd : 192.168.1.142
ALL:LOCAL
[admin@machineTP xinetd.d]$
```

```
[admin@machineTP xinetd.d]$ cat /etc/hosts.deny | grep :
 ALL:ALL
[admin@machineTP xinetd.d]$
```

Le fichier /etc/hosts.allow est d'abord lu et autorise l'accès ssh pour la machine cliente de l'administrateur (et uniquement celle-ci). En local cependant tous les services sont accessibles, puisque l'OS est en principe sécurisé. Le fichier /etc/hosts.deny est ensuite lu interdisant tous les trafics qui ne sont pas mentionnés dans le fichier /etc/hosts.allow. La politique choisit est de tout interdire par défaut et d'autoriser que ce dont on a besoin.

Sécurité des systèmes de fichier

- Options des partitions : L'OS contient diverses partitions qui ont chacune différentes fonctions. Il est donc normal de ne pas avoir les mêmes droits sur ces partitions. Pour ce faire, nous allons éditer le fichier /etc/fstab et mettre les options comme indiqués sur les partitions :

```
[admin@machineTP ~]$ cat /etc/fstab


#
# /etc/fstab
# Created by anaconda on Fri May 29 19:57:09 2009
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or vol_id(8) for more info
#
UUID=ed3af808-d8da-490b-87b9-1fd515ca3370 /                       ext3    defaults        1 1
UUID=b16fcf8b-dc4a-4cf7-8c63-2eb9f7f7d2ca /var                    ext3    nosuid,noexec,nodev       1 2
UUID=1a8e642f-c349-4bdd-9487-7ab06a667bb1 /home                   ext3    nosuid,noexec,nodev       0 2
UUID=ca70f750-a6c1-4944-88f2-92e92a54aac8 /boot                   ext3    nosuid,noexec,nodev,ro       1 2
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
UUID=118ea8b3-606d-40c9-9003-a1927bfb780e swap                    swap    defaults        0 0
[admin@machineTP ~]$
```

/boot est employé par le serveur et il n'est donc pas utile d'écrire, d'exécuter des binaires ou d'activer le bit suid sur cette partition. /var est utilisé pour les logs et ne contient donc pas de fichiers spéciaux. Pour la même raison, il n'y a aucune raison de pouvoir exécuter des binaires sur cette partition et d'y écrire (modifier des fichiers de logs créé une faille de sécurité). /home est censé contenir les fichiers privés de l'utilisateur et par mesure de protection on désactive l'execution des binaires qui devraient se trouver dans /usr/bin. Le bit suid est également désactivé et les fichiers spéciaux n'ont a priori pas leurs place sur cette partition.

- Restreindre l'accès aux fichiers sensibles en vérifiant les permissions de ces fichiers et le propriétaire

```
[admin@machineTP etc]$ sudo chown root:root passwd shadow group gshadow
[admin@machineTP etc]$ sudo chmod 644 passwd group
[admin@machineTP etc]$ sudo chmod 400 shadow gshadow
[admin@machineTP etc]$ sudo chmod 750 /bin/mount
[admin@machineTP etc]$ sudo chmod 750 /bin/rpm
[admin@machineTP etc]$ sudo chmod 4750 /bin/su
…
```

Pour une liste des fichiers dont on a redéfinit les permissions, voir à la page 114 du livre eni.

Mécanisme de synchronisation de l'horloge avec NTP

- Il est important d'avoir un référentiel au niveau du temps pour corréler les traces recueillies suivant un même référentiel. Ainsi nous devons activer le service ntp et lui faire exécuter quotidiennement des synchronisations.

```
[admin@machineTP etc]$ sudo chkconfig ntpd on
```

On active le service pour que celui-ci puisse opérer au prochain démarrage.

```
[admin@machineTP etc]$ sudo vi /etc/crontab
```

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
0 * * * * /usr/sbin/ntpdate -s 81.19.16.225 94.23.21.155
```

Grace à la crontab, le binaire ntpdate va synchroniser avec le serveur de temps fr.pool.ntp.org (sous forme d'IP dans le fichier) et ainsi être précis.

Sécurité des utilisateurs et des groupes

- Mettre un message d'avertissement lorsqu'un utilisateur se connecte pour le responsabiliser

```
[admin@machineTP etc]$ sudo vi /etc/issue
[admin@machineTP etc]$ sudo vi /etc/issue.net
```

- Changer le propriétaire des fichiers des comptes utilisateurs à supprimer :
  Je n'ai pas compris la commande du livre et je ne suis pas à même de la reproduire. Je ne vais donc pas supprimer les fichiers des utilisateurs à supprimer . je ne pense pas que cela soit grave, car le fichier a des accès restreints et que je peux toujours changer le propriétaire de ces fichiers en les octroyant à root.

- Supprimer/ Désactiver les comptes inutiles :

```
[admin@machineTP etc]$ sudo userdel -r adm
[sudo] password for admin:
userdel: error removing directory /var/adm
```

Apres la manipulation, il y a ce message d'erreur. J'imagine que ceci est la conséquence de l'étape précédente. Quoiqu'il en soit un « cat /etc/passwd | grep adm » nous indique que l'utilisateur adm est supprimé. Et de la même manière, le répertoire /var/adm a également disparu. Pour l'ensemble des suppression suivantes, on peut noter un problème similaire.

```
[admin@machineTP spool]$ sudo userdel -r lp
[admin@machineTP spool]$ sudo userdel -r sync
[admin@machineTP spool]$ sudo userdel -r shutdown
[admin@machineTP spool]$ sudo userdel -r halt
[admin@machineTP spool]$ sudo userdel -r news
[admin@machineTP spool]$ sudo userdel -r uucp
[admin@machineTP spool]$ sudo userdel -r operator
[admin@machineTP spool]$ sudo userdel -r games
[admin@machineTP spool]$ sudo userdel -r gopher
[admin@machineTP spool]$ sudo userdel -r ftp
[admin@machineTP spool]$ sudo userdel -r dip
[admin@machineTP spool]$ sudo userdel -r nscd
[admin@machineTP spool]$ sudo userdel -r rpc
[admin@machineTP spool]$ sudo userdel -r rpcuser
[admin@machineTP spool]$ sudo userdel -r mail
[admin@machineTP spool]$ sudo userdel -r mailnull
[admin@machineTP spool]$ sudo userdel -r rpcuser
[admin@machineTP spool]$ sudo userdel -r ident
[admin@machineTP spool]$ sudo userdel -r xfs
[admin@machineTP spool]$ sudo userdel -r gdm
```

En ce qui concerne d'autres compte utilisateurs dont je ne suis pas très sur, plutot que de les supprimer, je choisis de les désactiver.

```
[admin@machineTP ~]$ sudo /usr/sbin/usermod -L nobody
[admin@machineTP ~]$ sudo /usr/sbin/usermod -s /sbin/nologin nobody
```

- Supprimer /désactiver les groupes inutiles: A l'instar des users, je n'ai pas réussi à changer le propriétaire des fichiers appartenant au groupe.

```
[admin@machineTP ~]$  sudo  groupdel  ftp
[admin@machineTP ~]$ sudo  groupdel  news
[admin@machineTP ~]$ sudo  groupdel  uucp
[admin@machineTP ~]$ sudo  groupdel  gopher
[admin@machineTP ~]$ sudo groupdel  dip
[admin@machineTP ~]$ sudo groupdel  nscd
[admin@machineTP ~]$ sudo  groupdel  games
[admin@machineTP ~]$ sudo  groupdel  rpc
[admin@machineTP ~]$ sudo  groupdel  rpcuser
[admin@machineTP ~]$ sudo  groupdel  adm
[admin@machineTP ~]$ sudo groupdel  ppusers
[admin@machineTP ~]$ sudo groupdel  popusers
[admin@machineTP ~]$ sudo groupdel  slipusers
```

- Vérification de l'activation du mécanisme de « shadowing »

```
[admin@machineTP ~]$  sudo  pwconv
[admin@machineTP ~]$  sudo  pwck
```

- Mettre un masque par défaut pour restreindre les droits des fichiers d'un utilisateur

```
[admin@machineTP ~]$ sudo   vi  /etc/profile
[admin@machineTP ~]$ sudo  vi  /etc/bashrc
[admin@machineTP ~]$ sudo  vi  /etc/csh.cshrc
```

Entrer dans chaque fichier la valeur umask 077 pour restreindre les droits. Quelque soit le shell installé sur le système, au démarrage d'une session utilisateur, ces fichiers vont être lus et le masque sera appliqué.

- Création d'un utilisateur standard « romain » et sécuriser l'environnement :

```
[admin@machineTP ~]$ sudo  useradd  romain
[sudo] password for admin:
[admin@machineTP ~]$ sudo  passwd  romain
Changing password for user romain.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[admin@machineTP ~]$ cat  /etc/passwd   |  grep  romain
romain:x:501:501::/home/romain:/bin/bash
[admin@machineTP ~]$
```

On va créer un groupe « utilisateurs » qui va contenir tous les comptes utilisateurs humains et qui va servir a uniformiser certaines restrictions sur le système pour tous les membres du groupe. Je considère que

l'utilisateur 'admin' fait parti de ce groupe pour une question de sécurité et que si besoin est, il peut s'accorder plus de droits sur les ressources du système

```
[admin@machineTP ~]$ sudo groupadd utilisateurs
[admin@machineTP ~]$ cat /etc/group | grep utilisateurs
utilisateurs:x:502:
```

On va rajouter les utilisateurs humains dans ce groupe en éditant le fichier /etc/group avec « sudo vi /etc/group » .

```
[admin@machineTP ~]$ cat /etc/group | grep utilisateurs ; id
utilisateurs:x:502:romain,admin
uid=500(admin) gid=500(admin) groups=10(wheel),500(admin),502(utilisateurs)
context=unconfined_u:unconfined_r:unconfined_t:s0
[admin@machineTP ~]$
```

On édite ensuite le fichier /etc/security/limits.conf pour restreindre au groupe « utilisateurs » des accès aux ressources. Les membres du groupe ne peuvent créer plus de 20 processus, un utilisateur ne peut employer plus de 10Mo de mémoire résidente, la génération de fichiers core est désactivée.

```
[admin@machineTP ~]$ cat /etc/security/limits.conf | grep utilisateurs
@utilisateurs hard core 0
@utilisateurs hard rss 10000
@utilisateurs hard nproc 20
[admin@machineTP ~]$
```

- Protéger les comptes avec les PAM : On va redéfinir les paramètres présents dans les fichiers de configuration /etc/pam.d/system-auth et /etc/login.defs. La stratégie de sécurisation à mettre en œuvre est la suivante :
    - Historique des derniers mots de passe non réutilisables : 5
    - Durée de vie maximale du mot de passe : 5 semaines
    - Longueur minimale du mot de passe : 8 caractères

```
[admin@machineTP ~]$ sudo vi /etc/pam.d/system-auth
[sudo] password for admin:
[admin@machineTP ~]$ cat /etc/pam.d/system-auth | grep password
password    required     pam_cracklib.so try_first_pass retry=3
password    sufficient   pam_unix.so sha512 shadow nullok try_first_pass use_authtok remember=5
password    required     pam_deny.so
[admin@machineTP ~]$
```

```
[admin@machineTP ~]$ sudo vi /etc/login.defs
[sudo] password for admin:
[admin@machineTP ~]$ cat /etc/log
login.defs     logrotate.conf  logrotate.d/    logwatch/
[admin@machineTP ~]$ cat /etc/login.defs | grep PASS_MAX_DAYS
#       PASS_MAX_DAYS   Maximum number of days a password may be used.
PASS_MAX_DAYS   35
[admin@machineTP ~]$
```

- Délai d'inactivité maximal: La session shell d'un utilsateur non utilisé se terminera au bout de 5 min en cas d'inactivité en éditant le fichier /etc/profile en entrant les valeurs suivantes :
  Readonly TMOUT=300
  Export TMOUT

Sécurité des services

- Désactivation des services inutiles : Moins il y a de service, moins il y a d'attaques potentielles et de ports ouverts. Nous allons donc supprimer les services suivants qui ne présentent pas un intérêt majeurs et doivent donc être désactivés

```
[admin@machineTP init.d]$  sudo  chkconfig  anacron  off
[admin@machineTP init.d]$  sudo  chkconfig   avahi-daemon off
[admin@machineTP init.d]$ sudo  chkconfig  cups off
[admin@machineTP init.d]$ sudo  chkconfig  gpm off
[admin@machineTP init.d]$ sudo  chkconfig  haldaemon  off
[admin@machineTP init.d]$  sudo  chkconfig  ip6tables  off
[admin@machineTP init.d]$  sudo  chkconfig mdmonitor off
[admin@machineTP init.d]$  sudo  chkconfig  messagebus  off
[admin@machineTP init.d]$  sudo  chkconfig  microcode_ctl  off
[admin@machineTP init.d]$  sudo  chkconfig  netfs  off
[admin@machineTP init.d]$  sudo  chkconfig  nfslock  off
[admin@machineTP init.d]$  sudo  chkconfig  rpcgssd  off
[admin@machineTP init.d]$  sudo  chkconfig  rpcidmapd  off
[admin@machineTP init.d]$  sudo chkconfig  sendmail  off
[admin@machineTP init.d]$  sudo  chkconfig setroubleshoot  off
[admin@machineTP init.d]$  sudo  chkconfig livesys  off
[admin@machineTP init.d]$  sudo chkconfig livesys-late  off
[admin@machineTP init.d]$  sudo chkconfig portreserve  off
[admin@machineTP init.d]$  sudo chkconfig  rpcbind  off
[admin@machineTP init.d ]$  sudo chkconfig irqbalance off
```

Les services de type « livesys » ont été désactivés car ils font référence au système X window que nous avons supprimés. Le service « portreserve » a été désactivé car il a pour but d'aider a la réservation de ports TCP et fonctionne avec portmap. C'est donc un élément non obligatoire. Le service rpcbind contient quant à lui une faille de sécurité exploitable en spoofant une adresse IP source et il vaut mieux le désactiver. En revanche le service udev_post n'a pas été désactivé, car il semblerait que son éxécution sécurise le système opérant sur les droits des fichiers. Les autres services sont des services usuels que l'on peut sans problème désactiver. En règle générale, il est d'usage de désactiver les services r* car ils déportent l'authentification, mais j'ai choisis de conserver le service rsyslog car son rôle de centralisation de logs est un élément important en terme de sécurité.

Nous allons vérifier les ports TCP/UDP actuellement ouverts (le service web n'étant pas encore fonctionnel) pour voir quels services sont accessibles depuis le réseau. Pour ce faire nous allons installer nmap avec la commande « sudo yum install nmap »

```
[admin@machineTP ~]$ sudo nmap -sSU 127.0.0.1 | grep /
Starting Nmap 4.68 ( http://nmap.org ) at 2009-06-12 19:31 CEST
Warning: RateMeter::update: negative time delta; now=1244827904.985429;
last_update_tv=1244827904.985724
22/tcp open ssh
123/udp open|filtered ntp
[admin@machineTP ~]$
```

On voit bien qu'un nombre restreint de services sont ouverts. Seuls ceux qui sont vraiment utiles.

- Sécurisation des comptes de service : Les démons n'ont pas besoin de shells interactifs, on va donc les désactiver en ajoutant le shell « /bin/false » aux utilisateurs démons.

```
[admin@machineTP ~]$ cat /etc/passwd | grep /bin/false
ntp:x:38:38::/etc/ntp:/bin/false
tcpdump:x:72:72::/:/bin/false
avahi:x:497:494:avahi-daemon:/var/run/avahi-daemon:/bin/false
openvpn:x:495:492:OpenVPN:/etc/openvpn:/bin/false
apache:x:48:48:Apache:/var/www:/bin/false
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/bin/false
haldaemon:x:68:68:HAL daemon:/:/bin/false
[admin@machineTP ~]$
```

- Sécurisation de ssh : Ce service fait parti des quelques services qui tournent sur le système et qui a un port ouvert sur la machine. On pourrait entre autre changer le port d'SSH , mais objectivement, cela ne représente que peu d'intérêt compte tenu du fait qu'un « nmap » liste les ports ouverts et qu'il est aisé de tester une connexion sur les ports affichés. Le plus important est de sécuriser correctement le service en éditant le fichier /etc/ssh/sshd_config de la manière suivante : (On ne parlera que des modifications effectués à partir du fichier de base)

Option	Valeur	Commentaire
AllowUsers	admin	On autorise uniquement cet utilisateur
PermitRootLogin	no	Root ne doit pas se connecter, on utilise sudo
IgnoreRhosts	yes	On ne reprend pas les faiblesses des services R*
HostbasedAuthentication	no	Même raison
PermitEmptyPasswords	no	trivial
Banner	/etc/issue.net	On affiche un message d'avertissement
GatewayPorts	no	Pas de redirection de port

Mise en place du firewall Iptables

- Commençons déjà par s'assurer que le firewall sera bien lancé lors du boot :

```
[admin@machineTP ~]$ sudo chkconfig iptables on
```

- Entrons ensuite les nouvelles règles firewall. Elles consistent à autoriser en entrée uniquement les flux dont on a besoin et refuser tout le reste.
  La commande «sudo iptables -L » nous donne les règles actuellement chargées. On va supprimer les règles qui ne correspondent pas à la politique précitée en tapant « sudo iptables –D INPUT *numéro_règle* .

  On rentre ensuite les règles suivantes :

```
[admin@machineTP ~]$ sudo iptables -A INPUT -s 192.168.1.142 -m state --state
NEW,ESTABLISHED,RELATED -p tcp --dport 22 - j ACCEPT

[admin@machineTP ~]$ sudo iptables -A INPUT -m state --state NEW,ESTABLISHED,RELATED -p tcp --
dport 80 -j ACCEPT

[admin@machineTP ~]$ sudo iptables -A INPUT -i eth0 -m state --state ! INVALID -p tcp --dport
123 -j ACCEPT


[admin@machineTP ~]$ sudo iptables -A INPUT -p tcp --dport 139 -j ACCEPT

[admin@machineTP ~]$ sudo iptables -A INPUT -i eth0 -m state --state ! INVALID -p udp --dport
123 -j ACCEPT


[admin@machineTP ~]$ sudo iptables -A INPUT -j REJECT --reject-with icmp-host-prohibited
```

  On accepte en entrée la connexion SSH a partir d'une seule adresse IP correspondant au poste de l'administrateur. Puis le serveur web, le protocole NTP (on laisse passé les traffics TCP et UDP), puis le partage samba et finalement , tous les autres types de traffics sont refusés par le firewall (les règles sont prises dans l'ordre).
  Si l'on veut conserver cette politique de sécurité au prochain démarrage, il est necessaire de sauvegarder les règles firewall en tapant la commande :

```
[admin@machineTP ~]$ sudo service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
[admin@machineTP ~]$
```

Apache Chrooting

gunzip httpd-2.2.11.tar.gz
tar xvf httpd-2.2.11.tar
cd httpd-2.2.11

```
./configure --disable-actions --disable-alias --disable-asis --disable-autoindex --disable-cgi --disable-cgid --
disable-charset-lite --disable-env --disable-imagemap --disable-include --disable-negociation --disable-setenvif --
disable-userdir
make
umask 022
make install
chmod -R root:sys /usr/local/apache2

cd /usr/local/apache2
bin/apachectl start
ps -ef | grep http

http://192.168.10.10/
cat logs/access_log
cat logs/error_log


mkdir -p /chroot/httpd/dev
mkdir -p /chroot/httpd/etc
mkdir -p /chroot/httpd/lib
mkdir -p /chroot/httpd/var/run
mkdir -p /chroot/httpd/usr/lib
mkdir -p /chroot/httpd/usr/libexec
mkdir -p /chroot/httpd/usr/local/apache2/bin
mkdir -p /chroot/httpd/usr/local/apache2/lib
mkdir -p /chroot/httpd/usr/local/apache2/logs
mkdir -p /chroot/httpd/usr/local/apache2/conf
mkdir -p /chroot/httpd/usr/local/apache2/htdocs

chown -R root:root /chroot
chmod -R 0755 /chroot

ls -als /dev/null
mknod /chroot/httpd/dev/null c 1 3
chown root:root /chroot/httpd/dev/null
chmod 666 /chroot/httpd/dev/null

ldd /usr/local/apache2/bin/httpd
strings /usr/local/apache2/bin/httpd | grep lib
strace /usr/local/apache2/bin/httpd 2>&1 | grep open

cp -p /lib/libm.so.6 /chroot/httpd/lib/.
cp -p /usr/local/apache2/lib/libaprutil-1.so.0 /chroot/httpd/usr/local/apache2/lib/.
cp -p /usr/local/apache2/lib/libexpat.so.0 /chroot/httpd/usr/local/apache2/lib/.
cp -p /usr/local/apache2/lib/libapr-1.so.0 /chroot/httpd/usr/local/apache2/lib/.
cp -p /lib/librt.so.1 /chroot/httpd/lib/.
cp -p /lib/libcrypt.so.1 /chroot/httpd/lib/.
cp -p /lib/libpthread.so.0 /chroot/httpd/lib/.
cp -p /lib/libdl.so.2 /chroot/httpd/lib/.
cp -p /lib/libc.so.6 /chroot/httpd/lib/.
```

cp -p /lib/ld-linux.so.2 /chroot/httpd/lib/.

cp -p /lib/libnss_files.so.2 /chroot/httpd/lib/.

cp -p /usr/local/apache2/bin/httpd /chroot/httpd/usr/local/apache2/bin/.
cp -p /usr/local/apache2/conf/httpd.conf /chroot/httpd/usr/local/apache2/conf/.
cp -p /usr/local/apache2/conf/mime.types /chroot/httpd/usr/local/apache2/conf/mime.types
cp -p /etc/hosts /chroot/httpd/etc/.
cp -p /etc/resolv.conf /chroot/httpd/etc/.
cp -p /etc/host.conf /chroot/httpd/etc/.
cp -p /etc/group /chroot/httpd/etc/.
cp -p /etc/passwd /chroot/httpd/etc/.
cp -p /etc/shadow /chroot/httpd/etc/.

cp -p /usr/local/apache2/htdocs/* /chroot/httpd/usr/local/apache2/htdocs/.

NB: Le but étant de sécuriser l'OS et d'installer Apache, j'ai utilisé le fichier commande.txt vu en TP pour chrooter le service httpd.

Références

- www.nsa.gov/ia/_files/os/redhat/rhel5-guide-i731.pdf

- **GNU/Linux Fedora special sécurité (editions eni)**