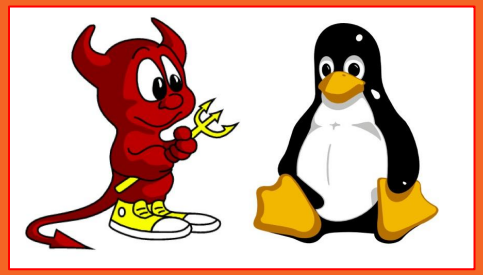


# Unix/Linux
## ISEN - AP4

# *6 - Hardening linux*

2019 - J. Hochart

# Infos

## Référence

Supports disponibles en ligne après le cours (j+1):

www.hochart.fr/LIN

## Contact

Pour toute question sur le cours ou les exos:

jul@hochart.fr

## Licence

# Références (2 - Hardening linux)

| | |
|---|---|
| Livres | • None |
| Web | • ANSSI - Note technique - Recommandations de configuration d'un système GNU/Linux |

# Intro

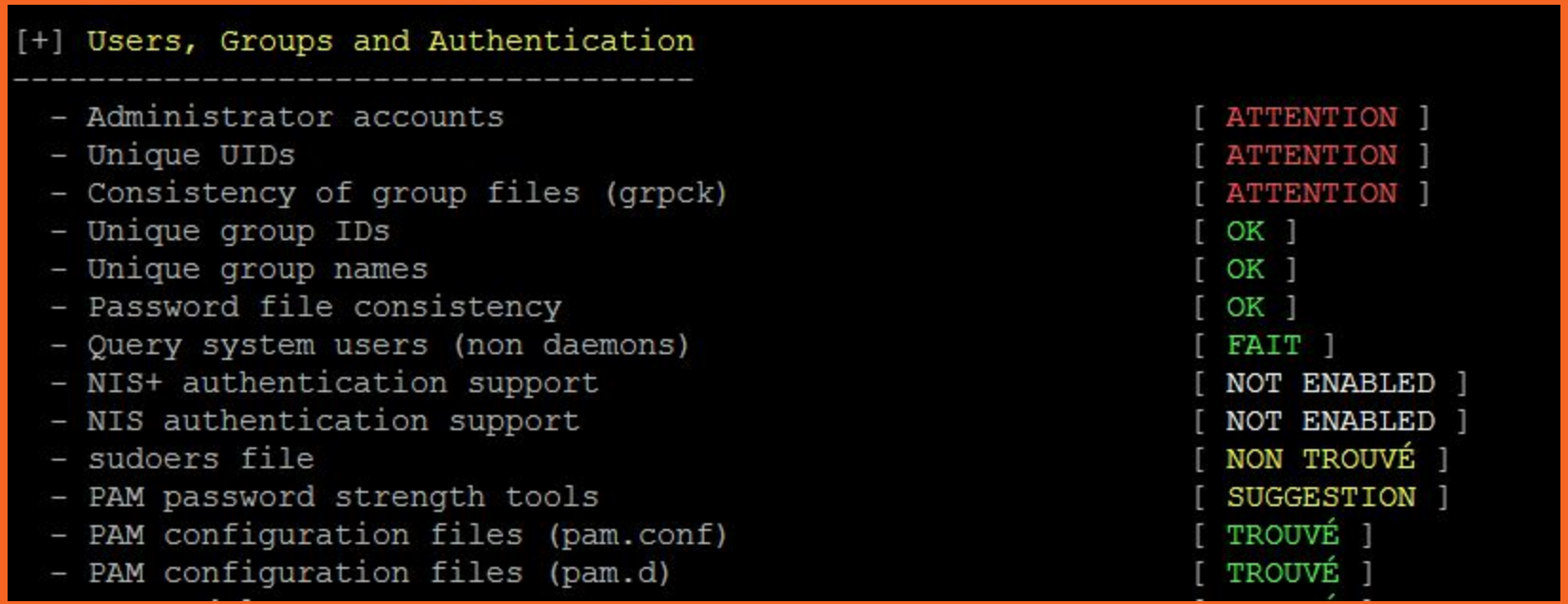
- Beaucoup de distributions
- Beaucoup d’usages, services

- Recommandations de hardening génériques
  - Basées sur la **compréhension** des mécanismes
  - Bon sens

# Intro

- Il existe des tools qui auditent un systeme en fonction d’une baseline. C’est juste des outils, qui ne remplacent pas le fait de comprendre pourquoi :)
- Lynis

# Intro




Lynis 2.4.0 / Debian

# Intro

```
Lynis security scan details:

Hardening index : 58 [###########          ]
Tests performed : 199
Plugins enabled : 1

Components:
- Firewall               [V]
- Malware scanner        [X]

Lynis Modules:
- Compliance Status      [?]
- Security Audit         [V]
- Vulnerability Scan     [V]

Files:
- Test and debug information      : /var/log/lynis.log
- Report data                     : /var/log/lynis-report.dat
```

Lynis 2.4.0 / Debian

# Besoin de sécurité

- Comme tout le temps en sécu, **les mesures à mettre en place dépendent du besoin de sécurité**
  - Serveur web de e-commerce: **Niveau élevé**
  - Serveur de dev sur un vlan isolé: **Niveau faible**

# Important

- En général un serveur fraîchement installé est solide
- Toutes ces recommandations sont à mettre dans le contexte d’un serveur sur lequel travaillent différentes personnes pendant plusieurs mois, faisant tourner plusieurs services complexes

# Niveaux de sécurité

**MINIMAL**
Authentification système
Comptes / passwords
Services inutiles
Patch management

**INTERMEDIAIRE**
Firewalling entrant
Hardening des services
Partitionnement
Setuid
Auth par annuaire
Export des logs
Sudo

**RENFORCE**
Firewalling sortant
Audit local
Conf système
Modules dynamiques
Binaires inutiles
Chroot des services

**ELEVE**
GRSecurity
RBAC SELinux
AppArmor

# Niveau: MINIMAL

# Minimiser les services réseau

- netstat -anpe | grep 0.0.0.0
- Désactiver tous les services qui ne sont pas utiles au fonctionnement principal du serveur
  - DHCP, Zeroconf
- Les modifier pour les binder sur 127.0.0.1 si possible
  - MySQL, Postgres, ... pour utilisation locale

# Défense en profondeur

- Franchissement de plusieurs barrières de technologies différentes
  - Authentification obligatoire
  - Journalisation centralisée
  - Mécanismes de prévention d’exploitation

# Patch management

- Process de mise à jour régulière de tous les logiciels du serveurs (pas seulement les composants de sécurité)
- Associé à un processus de veille croisé avec l'inventaire logiciel
- En général basé sur
  - Cron.d / outil de gestion des packages
  - Debian : unattended upgrades
  - Tools corporate custom basés sur puppet, ansible, ...

# Repos

- N’utiliser que les repos officiels de la distribution

# MDP Admin

- MDP root existant mais utilisé en **dernier recours** (après clefs, sudo, ldap …)
- MDP root unique sur chaque machine
- Si commun (souvent) à tous les serveurs, le pwd doit être non apprenable (long + complexe) et dans un coffre dans une enveloppe scellée

# Stockage des MDP

- Stockage des MDP hashés avec une fonction cryptographiquement sre (SHA256, SHA512, ...)
- Stockage des MDP salés
- Nombre de tours de hash assez grands

- **/etc/pam.d/common-password**
  - **password required pam_unix . so obscure sha512 rounds =65536**
- **/etc/login.defs**
  - **ENCRYPT_METHOD SHA512 SHA_CRYPT_MIN_ROUNDS 65536**

# Binaires avec bits de setuid et setgid

- N’utiliser le setuid et setgid qu’avec des programmes qui sont spécifiquement conçus pour cela
  - Elévation de privilèges suite à une vulnérabilité

# Services et daemons

- Désactiver les daemons qui ne sont pas utiles au service rendu
- Les désinstaller si possible
  - RPC: portmap, rpc.statd: utiles seulement pour un serveur NFS
  - Bureautique: dbus, hald, CUPS, …
  - Serveur X
  - Exim, postfix, ntpd, bind

# Services et daemons

- Lister les process et services en écoute
  - ps aux
  - netstat -aelonptu

# Sudo sans authentification

- Rendre obligatoire l’authentification lors de l’appel à sudo
  - Ne pas utiliser le keyword NOPASSWD
  - Audit: sudo -l et /etc/sudoers

# Niveau: INTERMEDIAIRE

# Configuration minimale des services

- Nettoyer les configurations par défaut qui ne sont pas assez restrictives
- Exemples
  - SSHD: port redirect
  - Apache: dirlisting

# Architecture d’install

- Installer la version 64 bits plutot que la version 32 bits
- Sauf si impossible pour des raisons matérielles
- Raisons
  - Présence des bits NX/XD/EVP
  - Espace d’adressage plus grand pour l’ASLR
  - Plus de mémoire pour les processes

# Partitionnement avec droits fins

| Mount point | Mount options |
| --- | --- |
| / | |
| /tmp , /home , /var , /var/log | Nosuid, nodev, noexec |
| /proc | hidepid=1 |
| /usr | nodev |
| /boot | Nosuid, nodev, noexec, **noauto (non standard)** |

# Packages installés

- Faire une install minimale de l’OS
- Installer packages utiles et leurs dépendances
- Ne pas utiliser les profils d’install

# Ne pas se logguer en root

- **Créer des comptes nominatifs (ou imputables) pour chaque administrateur**
  - Compte local ou ldap
- Ne pas leur donner l’id 0, mais utiliser sudo

# Configuration réseau (/etc/sysctl.conf)

```
# Pas de routage entre les interfaces
net.ipv4.ip_forward = 0
# Filtrage par chemin inverse
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
# Ne pas envoyer de redirections ICMP
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
# Refuser les paquets de source routing
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
# Ne pas accepter les ICMP de type redirect
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
# Loguer les paquets ayant des IPs anormales
net.ipv4.conf.all.log_martians = 1
```

# Configuration réseau (/etc/sysctl.conf)

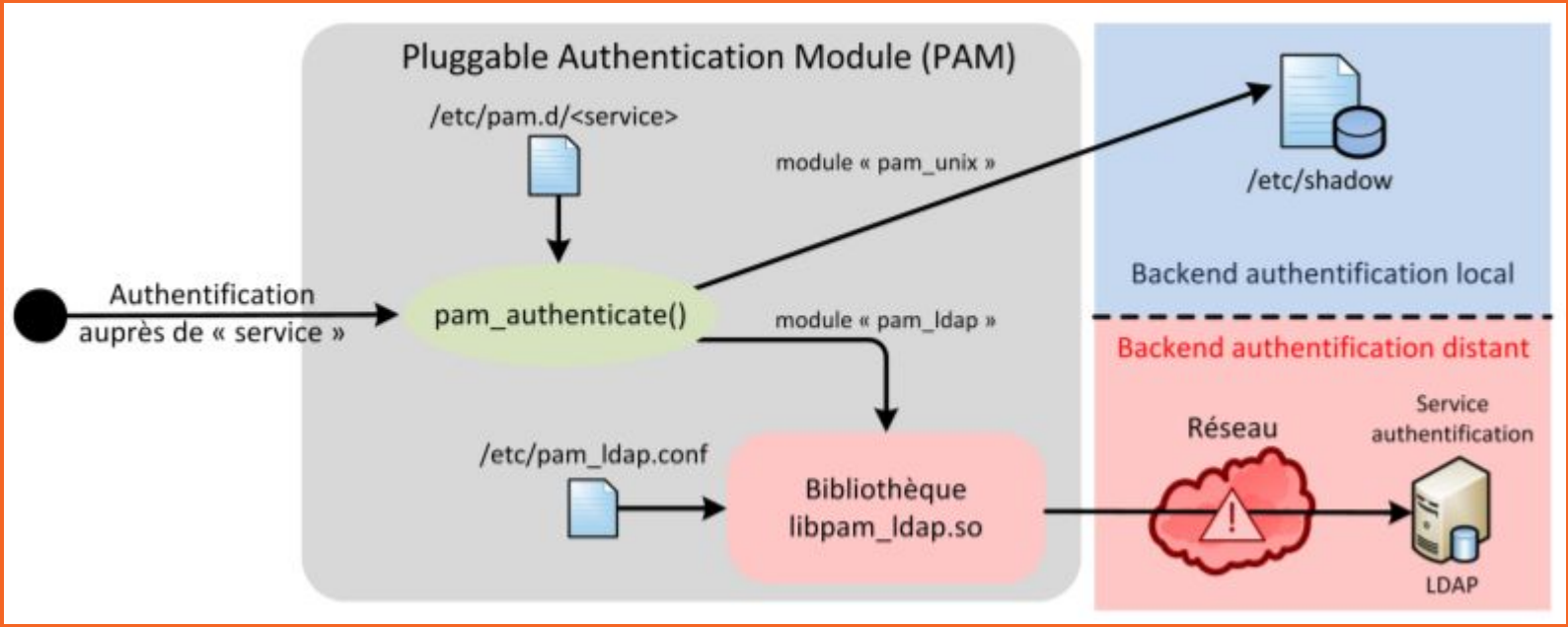
```
# Ignorer les réponses non conformes à la RFC 1122
net.ipv4.icmp_ignore_bogus_error_responses = 1
# Augmenter la plage pour les ports éphémères
net.ipv4.ip_local_port_range = 32768 65535
# Utiliser les SYN cookies
net.ipv4.tcp_syncookies = 1
# Désactiver le support des "router solicitations"
net.ipv6.conf.all.router_solicitations = 0
net.ipv6.conf.default.router_solicitations = 0
# Ne pas accepter les "router preferences" par "router advertisements"
net.ipv6.conf.all.accept_ra_rtr_pref = 0
net.ipv6.conf.default.accept_ra_rtr_pref = 0
# Pas de configuration auto des prefix par "router advertisements"
net.ipv6.conf.all.accept_ra_pinfo = 0
net.ipv6.conf.default.accept_ra_pinfo = 0
# Pas d'apprentissage du routeur par défaut par "router advertisements"
net.ipv6.conf.all.accept_ra_defrtr = 0
net.ipv6.conf.default.accept_ra_defrtr = 0
# Pas de configuration auto des adresses à partir des "router
    advertisements"
net.ipv6.conf.all.autoconf = 0
net.ipv6.conf.default.autoconf = 0
# Ne pas accepter les ICMP de type redirect
net.ipv6.conf.all.accept_redirects = 0
net.ipv6.conf.default.accept_redirects = 0
# Refuser les packets de source routing
net.ipv6.conf.all.accept_source_route = 0
net.ipv6.conf.default.accept_source_route = 0
# Nombre maximal d'adresses autoconfigurées par interface
net.ipv6.conf.all.max_addresses = 1
net.ipv6.conf.default.max_addresses = 1
```

# Configuration système (/etc/sysctl.conf)

```
# Désactivation des SysReq
kernel.sysrq = 0
# Pas de core dump des exécutables setuid
fs.suid_dumpable = 0
# Interdiction de déréférencer des liens vers des fichiers dont
# l'utilisateur courant n'est pas le propriétaire
# Peut empêcher certains programmes de fonctionner correctement
fs.protected_symlinks = 1
fs.protected_hardlinks = 1
# Activation de l'ASLR
kernel.randomize_va_space = 2
# Interdiction de mapper de la mémoire dans les adresses basses (0)
vm.mmap_min_addr = 65536
# Espace de choix plus grand pour les valeurs de PID
kernel.pid_max = 65536
# Obfuscation des adresses mémoire kernel
kernel.kptr_restrict = 1
# Restriction d'accès au buffer dmesg
kernel.dmesg_restrict = 1
# Restreint l'utilisation du sous système perf
kernel.perf_event_paranoid = 2
kernel.perf_event_max_sample_rate = 1
kernel.perf_cpu_time_max_percent = 1
```

# Utilisation de PAM avec des protocoles secure

- Dans le cas d'une authentification réseau avec PAM, utiliser des protocoles sécurisés



## Utilisation de PAM avec des protocoles secure

- Pam_krb5 - Kerberos:  :)
- Pam_ldap, pam_mysql: Configuration explicite à faire

- Configurer PAM pour utiliser des modules de sécurité si applicables
  - Pam_time: restreindre l'accès à une plage horaire
  - Pam_cracklib: evalue la robustesse des mdp
  - Pam_passwdqc: application d'une pwd policy
  - Pam_tally: lock des comptes après n échecs

# Configuration PAM

Exemple avec /etc/pam.d/su et /etc/pam.d/sudo :

```
# Bloque l'accès à root aux membres du groupe 'wheel'
auth            required        pam_wheel.so
```

Exemple avec /etc/pam.d/passwd :

```
# Au moins 12 caractères, pas de répétition ni de séquence monotone,
# 3 classes différentes (parmi majuscules, minuscules, chiffres, autres)
password        required        pam_cracklib.so minlen=12 minclass=3  \
                                dcredit=0 ucredit=0 lcredit=0 \
                                ocredit=0 maxrepeat=1         \
                                maxsequence=1 gecoscheck      \
                                reject_username enforce_for_root
```

Exemple avec /etc/pam.d/login et /etc/pam.d/sshd :

```
# Blocage du compte pendant 5 min après 3 échecs
auth            required        pam_tally.so deny=3 lock_time=300
```

# Permissions sur les fichiers sensibles

```
-rw-r----- root root  /etc/gshadow
-rw-r----- root root  /etc/shadow
-rw------- foo  users /home/foo/.ssh/id_rsa
```

## Sticky bit et permission W

- **Les repertoires ecrivables par tous doivent avoir le sticky bit armé**
- **Le propriétaire doit etre root**
- **Sinon les users peuvent ecraser les fichiers des autres**

```
find / -type d \( -perm -0002 -a \! -perm -1000 \) -print 2>/dev/null
```

```
find / -type d -perm -0002 -a \! -uid 0 -print 2>/dev/null
```

# Sécurité des sockets et pipes nommés

- Ne pas créer de sockets locales à la racine d’un /tmp accessible à tout le monde en écriture
- ss -xp

# Journalisation

- **Chaque service doit logguer dans un log dédié**
- **Ce fichier ne doit être accessible que par syslog**
- **Ce fichier ne doit pas être R, W, D par le service qui l'à produit**

→ Utilisation de l'API syslog(),

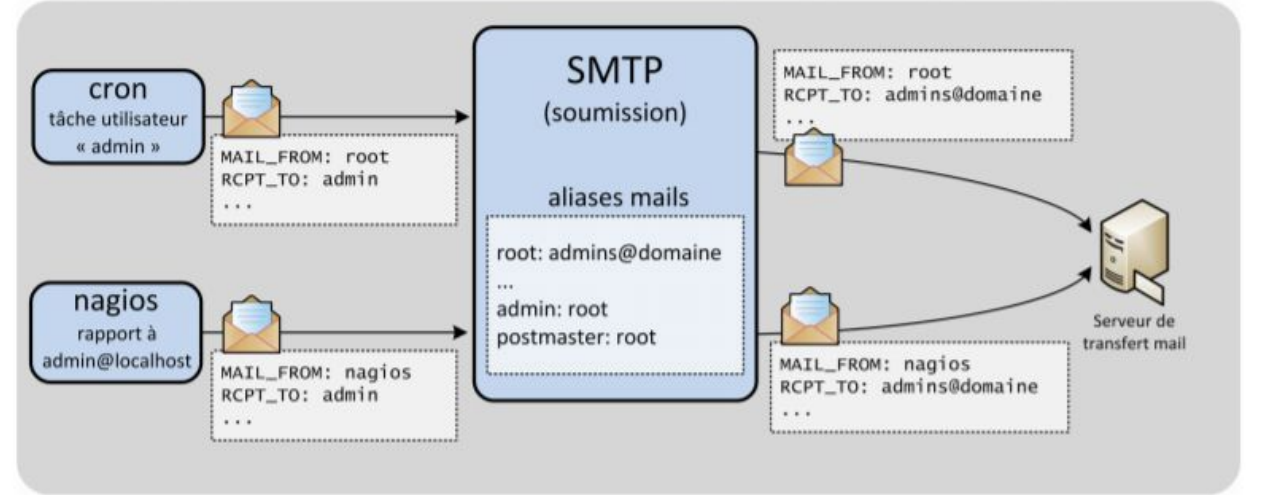
→ 'logger' en CLI à minima

# MTA (Mail transport Agent)

- Sauf cas particulier, le MTA doit être configuré pour
  - Rejeter les mails qui ne sont pas à destination d'un user local du serveur
  - Ne pas écouter sur une interface réseau autre que la loopback (127.0.0.1)

# Alias mail

- Pour chaque service de la machine, et pour root, créer un alias mail vers une adresse valide pour qu'il puisse recevoir d'éventuelles alertes



# Chroot

- **Utiliser chroot pour chaque service qui le supporte**
- Le chroot doit être **compris** pour éviter les risques qu'il engendre
  - L'utilisateur qui exécute le service ne doit pas être root
  - L'utilisateur doit être un utilisateur dédié à cela, non utilisé ailleurs
  - **L'utilisateur ne doit pas avoir accès en W à la / chrootée** (sinon il peut par exemple se créer un /etc/shadow)

# Groupe pour sudo

- Créer un groupe dédié à l'usage de sudo
- Seuls les users membre de ce groupes pourront utiliser sudo
- Surveiller la persistence de ces droits par script (peut être reset par un update du package)

```
# ls -al /usr/bin/sudo
-rwsr-x---. 2 root sudogrp [...] /usr/bin/sudo
```

# Directives de configuration sudo

- Configurer sudo par défaut avec
  - **noexec:** appliquer le tag NOEXEC par défaut sur les commandes *(ne pas pouvoir à leur tour exec d'autres commandes)*
  - **requiretty:** imposer à l'utilisateur d'avoir un tty de login
  - **umask=0027:** forcer umask à un masque plus restrictif (override)
  - **ignore_dot:** ignorer le '.' dans $PATH
  - **env_reset:** réinitialiser les variables d'environnement
  - **passwd_timeout=1:** allouer 1 minute pour entrer son mot de passe.

```
Defaults noexec,requiretty,use_pty,umask=0027
Defaults ignore_dot,env_reset,passwd_timeout=1
```

# Utilisateur à privilège sudo

- Eviter d’utiliser root en cible mais un utilisateur spécifique

# Négation sudo

- Ne pas utiliser la syntaxe négative pour sudo
- Trivialement contournable, exemple:
  - **user ALL = ALL,!/bin/sh**
  - **cp /bin/sh ./yop**
  - **sudo ./yop  #w00t**

# Sudo et arguments explicites

- Similaire au globbing du shell
- Inclut les arguments
- Toutes les specs dans le fichier sudoers doivent spécifier les arguments
- Eviter *, et préciser l’absence d’args par “”

# Niveau: RENFORCE

# Cloisonnement des services réseau

- Ne pas faire tourner des services complémentaires (par exemple HTTPD + DB sur le même host
- 1 service par host
- Cloisonnement
  - Deny all par défaut entre les hosts
  - Allow des flux TCP nécessaires à la communication entre les composants

# Journalisation de l’activité des services

- Journalisation de l’OS + des services
- Export des logs sur un concentrateur, par exemple via syslog

# Accès à la System.map

- /boot/System.map
- Utilisé en lecture par des programmes malveillants pour attaquer le kernel
- Si pas possible de mounter /boot en noauto
  - Lecture uniquement pour root

## Bootloader + password

- Grub
- Configurer Grub pour qu'il demande un MDP si on tente de modifier les options de boot
- Dans tous les cas, restreindre **l'accès physique (toucher le serveur en baie) aux sysadmins**

# Installation des secrets et certificats

- Le faire **rapidement** après l'install du serveur
  - Génération des clefs privées **(sur place!)**
  - Importation des certificats, des CA

# Interdire le chargement des modules kernel

- Vérifier que cela n’entraine pas d’effet de bord sur le fonctionnement normal du serveur
- /etc/sysctl.conf
  - kernel . modules_disabled = 1
- Configurer Yama
  - security=yama en paramètre de boot du kernel
  - Systcl kernel.yama.ptrace_scope >= 1

# Désactiver les comptes inutiles

- Désactivation du compte (suppression du champ password dans le shadow)
  - Usermod -L user
- Désactivation du shell de l’user
  - Usermod -s /bin/false user

# Comptes systeme de service

- Utiliser un compte de service unique par service

# Timeout de session interactive

- Configurer le shell pour que la session timeout après un temps raisonnable d'inactivité (par exemple 10 min)
  - Bash: export TMOUT=600
  - Industrialiser le positionnement de cette variable d'environnement par défaut pour chaque compte ayant un shell

# Umask

- Setter le default umask SYSTEME à 027
  - Debian: /etc/init.d/rc.d
  - Tout nouveau fichier n'est lisible que par l'owner et le group
  - Tout nouveau fichier n'est writable que par l'owner
- Setter le default umask USER à 077
  - Debian: /etc/login.defs
  - Tout nouveau fichier n'est lisible que par l'owner
  - Tout nouveau fichier n'est writable que par l'owner

# Executables en setuid root

- **Limiter au maximum les binaires en setuid**
- Si ils sont destinés aux sysadmins, utiliser plutôt su/sudo (auditabilité)
- Contrôler cela après chaque upgrade du système
  - find / -type f \( -perm -2000 -o -perm -4000 \) -print 2>/dev/null
  - chmod u-s / g-s

| Exécutable | Commentaire |
|---|---|
| /bin/mount | À désactiver, sauf si absolument nécessaire pour les utilisateurs. |
| /bin/netreport | À désactiver. |
| /bin/ping6 | (IPv6) Idem ping. |
| /bin/ping | (IPv4) Retirer droit setuid, sauf si un programme le requiert pour du monitoring. |
| /bin/su | Changement d'utilisateur. Ne pas désactiver. |
| /bin/umount | À désactiver, sauf si absolument nécessaire pour les utilisateurs. |
| /sbin/mount.nfs4 | À désactiver si NFSv4 est inutilisé. |
| /sbin/mount.nfs | À désactiver si NFSv2/3 est inutilisé. |
| /sbin/umount.nfs4 | À désactiver si NFSv4 est inutilisé. |
| /sbin/umount.nfs | À désactiver si NFSv2/3 est inutilisé. |
| /sbin/unix_chkpwd | Permet de vérifier le mot de passe utilisateur pour des programmes non root. À désactiver si inutilisé. |
| /usr/bin/at | À désactiver si *atd* n'est pas utilisé. |
| /usr/bin/chage | À désactiver. |
| /usr/bin/chfn | À désactiver. |
| /usr/bin/chsh | À désactiver. |
| /usr/bin/crontab | À désactiver si *cron* n'est pas requis. |
| /usr/bin/fusermount | À désactiver sauf si des utilisateurs doivent monter des partitions FUSE. |
| /usr/bin/gpasswd | À désactiver si pas d'authentification de groupe. |
| /usr/bin/locate | À désactiver. Remplacer par `mlocate` et `slocate`. |
| /usr/bin/mail | À désactiver. Utiliser un mailer local comme `ssmtp`. |
| /usr/bin/newgrp | À désactiver si pas d'authentification de groupe. |

| Exécutable | Commentaire |
|---|---|
| /usr/bin/passwd | À désactiver, sauf si des utilisateurs non root doivent pouvoir changer leur mot de passe. |
| /usr/bin/pkexec | À désactiver si PolicyKit n'est pas utilisé. |
| /usr/bin/procmail | À désactiver sauf si *procmail* est requis. |
| /usr/bin/rcp | Obsolète. À désactiver. |
| /usr/bin/rlogin | Obsolète. À désactiver. |
| /usr/bin/rsh | Obsolète. À désactiver. |
| /usr/bin/screen | À désactiver. |
| /usr/bin/sudo | Changement d'utilisateur. Ne pas désactiver. |
| /usr/bin/sudoedit | Idem `sudo`. |
| /usr/bin/wall | À désactiver. |
| /usr/bin/X | À désactiver sauf si le serveur X est requis. |
| /usr/lib/dbus-1.0/dbus-daemon-launch-helper | À désactiver quand D-BUS n'est pas utilisé. |
| /usr/lib/openssh/ssh-keysign | À désactiver. |
| /usr/lib/pt_chown | À désactiver (permet de changer le propriétaire des PTY avant l'existence de devfs). |
| /usr/libexec/utempter/utempter | À désactiver si le profil utempter SELinux n'est pas utilisé. |
| /usr/sbin/exim4 | À désactiver si Exim n'est pas utilisé. |
| /usr/sbin/suexec | À désactiver si le `suexec` Apache n'est pas utilisé. |
| /usr/sbin/traceroute | (IPv4) Idem `ping`. |
| /usr/sbin/traceroute6 | (IPv6) Idem `ping`. |

## Tracker et attribuer les fichiers sans owner

- Si ils ont un owner ID qui n’est pas dans /etc/passwd, il est possible qu’ils se retrouvent à appartenir à un nouvel user qui serait créé dans le futur et qui aurait cet ID

- find / -type f \( -nouser -o -nogroup \) -print 2>/dev/null

# Auditd

- Journaliser l'activité système avec auditd
- Exemple /etc/audit/auditd.rules

```
# Exécution de insmod, rmmod et modprobe
-w /sbin/insmod    -p x
-w /sbin/modprobe  -p x
-w /sbin/rmmod     -p x
# Journaliser les modifications dans /etc/
-w /etc/ -p wa
# Surveillance de montage/démontage
-a exit,always -S mount -S umount2
# Appels de syscalls x86 suspects
-a exit,always -S ioperm -S modify_ldt
# Appels de syscalls qui doivent être rares et surveillés de près
-a exit,always -S get_kernel_syms -S ptrace
-a exit,always -S prctl

# Rajout du monitoring pour la création ou suppression de fichiers
# Ces règles peuvent avoir des conséquences importantes sur les
# performances du système
-a exit,always -F arch=b64 -S unlink -S rmdir -S rename
-a exit,always -F arch=b64 -S creat -S open -S openat -F exit=-EACCESS
-a exit,always -F arch=b64 -S truncate -S ftruncate   -F exit=-EACCESS

# Verrouillage de la configuration de auditd
-e 2
```

# Restreindre l’environnement des processes

- Restreindre au strict nécessaire l’environnement des services
  - **Filesystem: Chroot**
  - **Process: Modele unix**
  - **Réseau: Modele unix**

# Restreindre l'environnement des processes

- Cloisonner par containers (docker, …)
- Cloisonner par emulation (QEmu, Virtualbox, …)
- Cloisonner par hyperviseur léger (Xen, HyperV)
- Cloisonner par hyperviseur kernel (KVM)

- **Toujours garder à l'esprit que l'ajout d'une brique de virtualisation apporte ses risques**
  - Nouvelles interfaces critiques à protéger (Rancher, vsphere, …)
  - Si l'host / kernel est compromis, tous les containers aussi
  - **Appliquer les recommandations de durcissement applicables à la techno de virtualisation employée**

## Limiter l’utilisation du tag exec dans les regles de sudo

- Des programmes complexes offrent la possibilité d’exécuter un sous process
- Permet des escalades de privilèges, par exemple !sh dans less
- Sudo permet un certain contrôle sur les sous-process mais qu’on pourra presque toujours subvertir, car un programme peut virtuellement appeler n’importe quel syscall (execve, …)

# Niveau: EXPERT

# SELINUX

- Permet d’implémenter des politiques de contrôle d’accès
- Ensemble de patches au kernel, qui permettent de contrôler strictement l’accès de process à
  - Fichiers
  - Processes
  - Sockets
  - Ports
  - ...

# APPARMOR

- Fonctionnalités similaires
- Learning mode
- Configuration plus simple que SELinux

```
# Last Modified: Wed Jul 27 23:19:01 2016
#include <tunables/global>

/home/user/bin/example.sh {
  #include <abstractions/base>
  #include <abstractions/bash>

  /bin/bash ix,
  /dev/tty rw,
  /etc/ld.so.preload r,
  /home/user/bin/data/sample.txt rw,
  /home/user/bin/example.sh r,
  /home/user/bin/sample.txt w,
  /usr/bin/rm rix,
  /usr/bin/touch rix,
```

# TRIPWIRE, SAMHAIN, ...

- FIM = File integrity monitoring
- On spécifie une liste de fichiers à contrôler
- Ils sont hashés, et l'output est signé
- De manière récurrente, on répète le process (schédulé) et on compare les écarts à la baseline

# GRSECURITY

- Patches de hardening du kernel
  - PAX: Marque les zones DATA (heap, stack, …) comme NX
    - Buffer overflows
  - ASLR: Randomization des addresses mémoires
  - RBAC
  - Patches sur chroot: cloisonnement des signaux, de la mémoire partagée, du mount, …
- Grsecurity n’est pas mergé dans le kernel, et les patches ne sont plus publiques