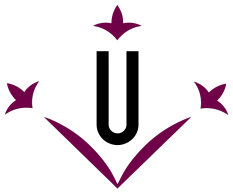

Universitat de Lleida

Sistemes

Seguretat d'Aplicacions i Comunicacions - Grau en Enginyeria Informàtica

Pablo Fraile Alonso

8 de novembre de 2022

Índex

Índex de figures

1 Sistema de fitxers

1.1 Particions

Mirem les particions del sistema:

```
[root@WEBSERVER usr]# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda               8:0    0   12G  0 disk
├─sda1            8:1    0  500M  0 part /boot
└─sda2            8:2    0 11,5G  0 part
  ├─centos-swap 253:0    0  1,2G  0 lvm  [SWAP]
  ├─centos-root 253:1    0  2,3G  0 lvm  /
  ├─centos-usr  253:2    0  6,9G  0 lvm  /usr
  └─centos-home 253:3    0  1,2G  0 lvm  /home
```

Veiem que empra diferents particions. On hi ha:

- Una partició per al boot
- Una partició de swap.
- Una partició per a root (/).
- Una partició per a /home (on hi han les configuracions dels usuaris, documents, imatges, etc).
- Una partició per a /usr (user-land data programs).

Tot i que ja hi han hagut varies particions que segueixen les bones pràctiques (com /home, /boot, etc), seria recomanable afegir noves particions, concretament:

- Una partició per a /tmp (ja que tots els usuaris poden escriure en aquesta partició).
- Una partició per a /var (ja veurem que aquesta màquina té instal·lada i configurada una base de dades, on les dades es guarden a /var[1]).

[1]Es podría fer backups d'aquesta partició en específic, sabent que conté dades crítiques/importants

- Una partició per a /var/log, per tal d'assegurar-nos que la màquina principal no es queda sense espai per culpa de guardar logs.
- Una partició per a /var/log/audit (aquesta màquina té audit habilitat i funcionant), i el motiu és el mateix que per crear una partició per a /var/log.

1.2 Boot Manager

En quant al boot manager, s'hauria d'activar Single Boot, o afegir una contrasenya a grub per tal de que no es puguin editar paràmetres d'arrancada desde el bootloader (figura 1).

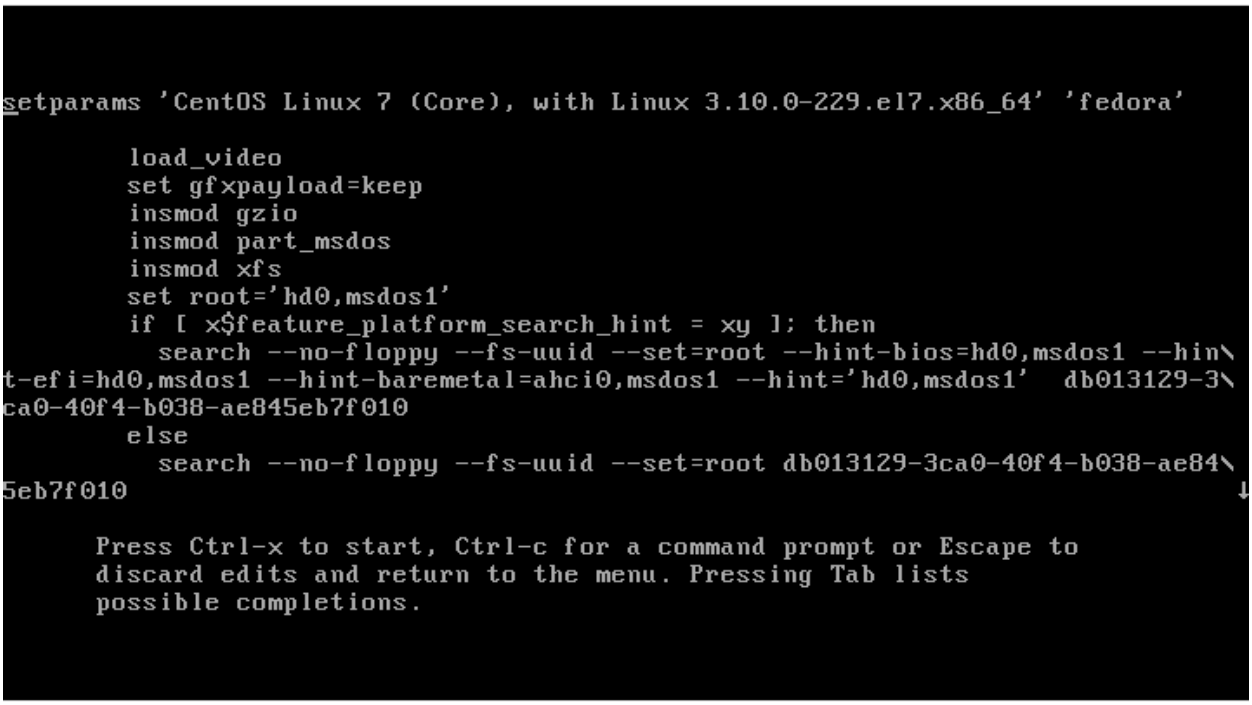

Figura 1: Es poden modificar paràmetres d'arrancada desde grub

1.3 Dispositius de xarxa

Sembla que l'ip s'ha configurat via DHCP, seria recomanable desactivar-lo per tal d'assegurar-nos tenir sempre la mateixa IP, a més d'evitar atacs com per exemple: DHCP Startvation i DHCP Spoofing.

Com la màquina empra NetworkManager, la comanda per a configurar manualment l'interfície serà:

```
nmcli con mod <interface> ipv4.addresses "<ADDRESS>/<MASK>"
nmcli con mod <interface> ipv4.gateway "<GATEWAY>"
nmcli con mod <interface> ipv4.dns "<DNS1> <DNS2>"
nmcli con mod <interface> ipv4.method manual
nmcli con mod <interface> connection.autoconnect yes
```

1.4 SUID

Es busquen tots els programes que tinguin el flag de SUID activat, amb la comanda:

```
[root@WEBSERVER sysconfig]# find / -perm /4000
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/newgrp
/usr/bin/su
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/umount
/usr/bin/ksu
/usr/bin/passwd
/usr/bin/staprun
/usr/bin/crontab
/usr/bin/pkexec
/usr/bin/at
/usr/bin/sudo
/usr/bin/vim
/usr/sbin/pam_timestamp_check
/usr/sbin/unix_chkpwd
```

```
/usr/sbin/userhelper
/usr/sbin/usernetctl
/usr/sbin/mount.nfs
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib64/dbus-1/dbus-daemon-launch-helper
/usr/libexec/sssd/krb5_child
/usr/libexec/sssd/ldap_child
/usr/libexec/sssd/selinux_child
/usr/libexec/abrt-action-install-debuginfo-to-abrt-cache
/usr/libexec/pulse/proximity-helper
```

Veiem alguns programes que **NO** haurien de tindre el SUID, aquests són:

1. VIM: El owner del binari és root i té el SUID activat, per tant, cada cop que obrim vim l'estarem executant com a root (i, per tant, els fitxers que guardem seràn propietat de root).

1.5 GUID

Es busquen tots els programes que tinguin el flag de GUID activat, amb la comanda:

```
[root@WEBSERVER /]# find / -perm /2000
/run/log/journal
/run/log/journal/d63bef2117f045eabe0735b25330096c
/usr/bin/wall
/usr/bin/write
/usr/bin/cgclassify
/usr/bin/ssh-agent
/usr/bin/cgexec
/usr/bin/locate
/usr/bin/vim
/usr/sbin/netreport
/usr/sbin/postdrop
/usr/sbin/postqueue
/usr/libexec/utempter/utempter
/usr/libexec/openssh/ssh-keysign
/usr/libexec/abrt-action-install-debuginfo-to-abrt-cache
```

Aquí s'observa que el grup de VIM és root i té el GUID habilitat, cosa que ens permetrà ser root durant l'execució d'aquest programa.

1.6 nodev en particions no root

S'ha d'evitar que particions que no siguin root tinguin accés a muntar dispositius, en aquest cas mirem les particions:

```
[root@WEBSERVER /]# findmnt
TARGET                          SOURCE                  FSTYPE     OPTIONS
......
├─/run                          tmpfs                   tmpfs      rw,nosuid,nodev,mode=755
├─/usr                          /dev/mapper/centos-usr  xfs        rw,relatime,attr2,
                                                                   inode64,noquota
├─/boot                         /dev/sda1               xfs        rw,relatime,attr2,
                                                                   inode64,noquota
└─/home                         /dev/mapper/centos-home xfs        rw,relatime,attr2,
                                                                   inode64,noquota
```

I veiem que /home no té activada l'opció nodev (tot i que seria recomanable tenir-la activada).

1.7 noexec nosuid en dispositius externs

Els dispositius haurien de tenir aquestes opcions activades:

- nosuid: no permetre que el *set-user-identifier (SUID)* o el *set-group-identifier (GUID)* prenguin efecte.
- noexec: no permetre l'execució dels binaris montats en aquell sistema de fitxers.
- nodev: no intrepretar dispositius de caràcters (teclats, etc) o de blocs (sistemes de fitxers) al sistema.

En el nostre cas veiem el següent:

```
TARGET                          SOURCE                  FSTYPE     OPTIONS
/                               /dev/mapper/centos-root xfs        rw,relatime,
                                                                   attr2,inode64,noquota
├─/dev                          devtmpfs   devtmpfs   rw,nosuid,size=499140k,
                                                      nr_inodes=124785,mode=755
│ ├─/dev/shm                    tmpfs      tmpfs      rw,nosuid,nodev
│ ├─/dev/pts                    devpts     devpts     rw,nosuid,noexec,relatime,
```

```
                                                    gid=5,mode=620,ptmxmode=000
│ ├─/dev/hugepages              hugetlbfs  hugetlbfs  rw,relatime
│ └─/dev/mqueue                 mqueue     mqueue     rw,relatime
```

Com veiem l'opció **noexec** no es troba activada a tots els dispositius, a més, seria recomanable afegir l'opció **noauto** per tal de que els dispositius no es muntessin automàticament.

1.8 Permisos dels fitxers principals

S'han revisat els fitxers /etc/passwd i /etc/shadow i s'ha vist que tenen els permisos adequats, és a dir:

1. El fitxer /etc/shadow té els permisos:

   ```
   ---------- 1 root root 1119 24 oct  2016 /etc/shadow
   ```

2. El fitxer /etc/passwd té els permisos:

   ```
   -rw-r--r-- 1 root root 1964 24 oct  2016 /etc/passwd
   ```

1.9 Fitxers executables

Busquem si es troba activat el *exec shield*, que afegeix mesures de seguretat per a prevenir exploits que aprofiten un desbordament en la pila:

```
[root@WEBSERVER dev]# sysctl kernel.exec-shield
sysctl: cannot stat /proc/sys/kernel/exec-shield: El fitxer o directori no existeix
```

Veiem que no existeix, però si busquem informació sobre el kernel actual (3.10.0), diu que exec-shield bé habilitat per defecte per motius de seguretat, i no es pot deshabilitar.

Mirem si l'aleatoritat d'adreces dels programes en temps d'execució es troba activada:

```
[root@WEBSERVER dev]# sysctl kernel.randomize_va_space
kernel.randomize_va_space = 2
```

Veiem que té el valor = 2, això implica que fa aleatòria tant les adreces de la pila, VDSO[2]i memòria compartida com les adreces del bloc de dades (per tant, ens proporciona més seguretat).

1.10 Accés root

Si som puristes, s'hauria de restringir l'accés root a només el device consola. En aquest dispositiu, aquesta mesura no està activada. Si es volgués activar, s'hauria d'editar el fitxer /etc/securetty i deixar únicament el paràmetre:

```
console
```

A més, veiem que el usuari *sjefe* pertany al grup wheel. Això indica que pot executar comandes com a root. S'hauria de valorar si aquest usuari realment necessita aquests permisos i, en cas de que necessiti root per qualsevol programa, crear un grup amb els permisos únicament per aquest context.

1.11 Restriccions d'accés

S'hauria de restringir l'accés a quins usuris poden accedir al sistema. El recomanable seria que root mai podés accedir remotament i haver de gestionar tot el manteniment del sistema a partir de l'accés físic a la màquina [3].

La configuració de la restricció d'accés es fa a l'arxiu /etc/security/acces.conf, seguint la sintaxi: *permís : usuari : origen*.

S'observa que no hi ha cap regla configurada, ja que:

```
cat /etc/security/access.conf | grep -v "^#"
```

no mostra cap contingut.

A més, tampoc hi ha configurat un timeout de fallada per si un atacant intenta un atac de força bruta, això es podria fer mitjançant la comanda

[2]Virtual Dynamic Shared Object

[3]Tot i que això aporta bastants inconvenients a l'hora de configurar màquines, però a nivell de seguretat és l'idoni

authconfig.

A més, també és important que, si hi han usuaris que no necessiten shell, aquesta s'hauria de redirigir a /sbin/nologin, amb la comanda:

```
$ usermod -s /sbin/nologin <user>
```

1.12 root com a propietari

Hi han carpetes on el propietari no hauria d'acostumar a ser root. Una d'aquestes pot ser la carpeta /var, on teòricament hi van dades variables, però a la pràctica és on van les dades de la BD, les pàgines webs, etc.

Busquem carpetes on el owner sigui root a la carpeta var amb la comanda:

```
$ find /var -type d -user root -perm -u+w
/var/lib/tomcat
/var/lib/tomcat/webapps
/var/lib/tomcat/webapps/host-manager
/var/lib/tomcat/webapps/host-manager/META-INF
/var/lib/tomcat/webapps/host-manager/WEB-INF
/var/lib/tomcat/webapps/host-manager/WEB-INF/jsp
/var/lib/tomcat/webapps/host-manager/images
/var/lib/tomcat/webapps/manager
/var/lib/tomcat/webapps/manager/META-INF
/var/lib/tomcat/webapps/manager/WEB-INF
/var/lib/tomcat/webapps/manager/WEB-INF/jsp
/var/lib/tomcat/webapps/manager/images
/var/lib/tomcats
....
/var/www
/var/www/cgi-bin
/var/www/html/blog/WordPress-4.6
/var/www/html/blog/WordPress-4.6/wp-admin/*
/var/www/html/blog/WordPress-4.6/wp-content/*
/var/www/html/blog/WordPress-4.6/wp-includes/*
```

Veiem que tant wordpress com tomcat són carpetes on l'usuari propietari és root. Seria recomanable crear un usuari (que no fos administrador) per a

que sigui el propietari d'aquestes carpetes[4]. Així, en cas de que s'accedeixi a partir d'una fallada de wordpress (que és bastant habitual) o de tomcat (servidor web de java, tot i que no es troba habilitat) es trobarà que el owner és un usuari sense gairebé privilegis.

2 Nombre de serveis (i quins) actius en arrancar

Observem els serveis que s'activen amb la comanda:

```
[root@WEBSERVER /]# systemctl list-unit-files --type=service --state=enabled | grep enabled
abrt-ccpp.service                           enabled
abrt-oops.service                           enabled
abrt-vmcore.service                         enabled
abrt-xorg.service                           enabled
abrtd.service                               enabled
accounts-daemon.service                     enabled
atd.service                                 enabled
auditd.service                              enabled
bluetooth.service                           enabled
chronyd.service                             enabled
crond.service                               enabled
dbus-org.bluez.service                      enabled
dbus-org.fedoraproject.FirewallD1.service   enabled
dbus-org.freedesktop.NetworkManager.service enabled
dbus-org.freedesktop.nm-dispatcher.service  enabled
display-manager.service                     enabled
dmraid-activation.service                   enabled
firewalld.service                           enabled
gdm.service                                 enabled
getty@.service                              enabled
httpd.service                               enabled
irqbalance.service                          enabled
iscsi.service                               enabled
kdump.service                               enabled
```

[4]De fet, nginx ja crea un usuari anomenat www-data, per als seus fitxers del servidor web

```
libstoragemgmt.service                      enabled
lvm2-monitor.service                        enabled
mariadb.service                             enabled
mdmonitor.service                           enabled
microcode.service                           enabled
multipathd.service                          enabled
NetworkManager-dispatcher.service           enabled
NetworkManager.service                      enabled
postfix.service                             enabled
rngd.service                                enabled
rsyslog.service                             enabled
rtkit-daemon.service                        enabled
smartd.service                              enabled
sshd.service                                enabled
sysstat.service                             enabled
systemd-readahead-collect.service           enabled
systemd-readahead-drop.service              enabled
systemd-readahead-replay.service            enabled
tuned.service                               enabled
```

Expliquem breument cadascun dels serveis:

- abrt*[5]: Serveis per a gestionar els errors del sistema.
- accounts-daemon: Permet als programes gestionar informació d'usuari a partir de missatges D-BUS de forma estandaritzada.
- atd: Permet planificar tasques d'un sol cop en un moment determinat amb la comanda *at*.
- auditd[6]: Permet gestionar els registres d'auditoria del sistema.
- bluetooth: Permet gestionar dispositius bluetooth.
- chronyd: Permet sincronitzar l'hora del sistema amb un servidor NTP.
- crond: Permet planificar tasques de forma periòdica.

[5] Automatic Bug Reporting Tool
[6] Security Auditing Service

- dbus*[7]: Serveis que permeten gestionar els missatges del sistema a través de D-BUS.
- display-manager: Permet gestionar els gestors de pantalla (relacionat amb gdm).
- dmraid-activation: Permet gestionar els dispositius RAID.
- firewalld: Permet gestionar la configuració de tallafocs del sistema.
- gdm[8]: Gestor de pantalla gràfic
- getty@: Permet gestionar les tty's del sistema.
- httpd: Servidor http.
- irqbalance: Permet gestionar les interrupcions del sistema.
- iscsi: Permet gestionar els dispositius iSCSI.
- kdump[9]: Permet guardar core-dumps en cas de que hi hagi un panic al kernel.
- libstoragemgmt: Permet gestionar els dispositius de magatzem.
- lvm2-monitor: Permet gestionar els volums lògics.
- mariadb: Servidor de bases de dades opensource que sorgeix d'un fork de MySQL.
- mdmonitor: Permet gestionar els dispositius RAID.
- microcode: Permet gestionar el microcodi de la cpu.
- multipathd: Permet gestionar els dispositius multipath.
- NetworkManager: Permet configurar i detectar xarxes i connectar-s'hi automàticament.

[7] D-Bus System Message Bus
[8] Gnome Display Manager
[9] Crash recovery kernel arming

- Postfix: Agent de transport de correu (MTA) que encamina i lliura correus electrònics en sistemes Linux.
- rngd [10]: Permet generar entropia per algoritmes pseudoaleatoris.
- rtkit-daemon: Un *daemon* de "gestió". En lloc d'aplicacions que demanen recursos directament al kernel (i necessiten els permisos adequats per a això, normalment root), li demanen a aquest daemon.
- smartd[11]: Permet mirar l'estat dels discs.
- sysstat: Reseteja els logs de les activitats de sistema.
- systemd-readahead-*: Optimitza i carrega a la caché els blocs més emprats del sistema operatiu.
- rsyslog: System Logging Service.
- sshd: Daemon per permetre ssh.
- tuned: Permet adaptar dinàmicament configuracions del sistema depenent de l'ús de recursos del mateix.

3 Configuracions de logging

L'estàndard de logging a Unix és syslog. Tal i com s'ha vist a la secció 1.12, el sistema empra rsyslogd per tal de manegar els logs del sistema. A més, sempre podem emprar *journalctl* per a veure el *journal* de systemd, a més de *dmesg* per a veure els missatges del kernel.

Mirem el fitxer de configuració de rsyslog (/etc/rsyslog.conf) i veiem que no es fa log de les següents facilities:

- kernel
- auth
- user

[10] Hardware RNG Entropy Gatherer Daemon
[11] Self Monitoring and Reporting Technology (SMART) Daemon

- daemon
- syslog

Per tant, s'hauria d'afegir les següents línies a l'arxiu de configuració:

```
kern.*     /var/log/kernel.log
auth,user.*     /var/log/messages.log
daemon.*        /var/log/daemon.log
syslog.*        /var/log/syslog.log
```

A més, mirant els permisos de les diferents carpetes de logging, s'observa que:

- La carpeta de logs té els permisos drwxr-xr-x, cosa que permet que un usuari no root llegeixi els logs del sistema. Si fos un atacant, podria llegir els logs i donar-se compte de les versions dels demés programes, fallades, etc.
- El fitxer de configuració de rsyslog, també es pot llegir per *others*, per tant, també seria recomanable modificar els permisos per tal de que un atacant si no és ni l'usuari root ni del grup root no pugui veure que estem loggejant.

3.1 Rotacions

La configuració de la rotació dels logs es prou correcta, i s'ha observat que aquests roten setmanalment. Tot i això, es podrien fer els següents mètodes per tal de millorar la seguretat:

1. Afegir el paràmetre size al fitxer /etc/logrotate.conf de manera que s'estableix una mida màxima de fitxer de log, que provocaria una rotació si és superada. Aquesta configuració permet dificultar a l'atacant esgotar l'espai del disc dur d'una màquina plenant els fitxers de log.
2. Enviar els fitxers de log a un servidor apart. Així a un atacant li seria molt més difícil esborrar els logs, a més de que facilitaria la feina a un forense que vol saber com ha passat un atac, ja que els logs estarien centralitzats (tenint en compte que es té sincronitzada correctament l'hora del sistema amb un servidor NTP centralitzat).

4 Configuracions de seguretat

4.1 SeLinux

Veiem si selinux es troba configurat amb la comanda:

```
[root@WEBSERVER ~]# sestatus
SELinux status:                 disabled
```

Tot i que, si mirem els logs amb journalctl, veiem el següent:

```
[root@WEBSERVER ~]# journalctl | grep SELinux
nov 08 15:41:45 WEBSERVER.acme.com kernel: SELinux:  Initializing.
nov 08 15:41:45 WEBSERVER.acme.com kernel: SELinux:  Starting in permissive mode
nov 08 15:41:45 WEBSERVER.acme.com kernel: SELinux:  Registering netfilter hooks
[root@WEBSERVER ~]#
```

I a la configuració del selinux veiem que s'intentava habilitar-lo amb una política permissiva i amb un mode de seguretat mínim:

```
[root@WEBSERVER ~]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=permisive
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=minimum
```

La conclusió que s'ha arribat es que Selinux s'ha intentat habilitar, però no s'han carregat els mòduls de LSM corresponents, i per tant no s'ha pogut activar.

4.2 AppArmor

Emprar apparmor i selinux a la vegada és impossible per les limitacions establertes al *Linux Security Module* del kernel. Igualment, comprovem que AppArmor no està instal·lat amb la comanda:

```
[root@WEBSERVER ~]# aa-enabled
-bash: aa-enabled: no s'ha trobat l'ordre
```

5 Nivell d'actualització del sistema

Veiem el nivell d'actualització del sistema amb les següents comandes:

```
[root@WEBSERVER /]# uname -r
3.10.0-229.el7.x86_64
[root@WEBSERVER /]# cat /etc/redhat-release
CentOS Linux release 7.1.1503 (Core)
[root@WEBSERVER /]# cat /etc/os-release
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
HOME_URL="https://www.centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"

CENTOS_MANTISBT_PROJECT="CentOS-7"
CENTOS_MANTISBT_PROJECT_VERSION="7"
REDHAT_SUPPORT_PRODUCT="centos"
REDHAT_SUPPORT_PRODUCT_VERSION="7"
```

A dia d'avui (2 de novembre de 2022), l'última versió de CentOS-7 és 7.9.2009. Per tant, el sistema no es troba actualitzat.

No s'ha tingut en compte que CentOS ha estat descontinuada, i que a partir d'ara s'ha d'emprar CentOS stream o bé Rocky Linux (que també té

les sources directament de RHEL). Si s'hagués d'instal·lar una nova versió, es recomanaria instal·lar una d'aquestes distribucions, ja que tindran suport.

Tot i això, la versió 7 de CentOS, continuarà rebent actualitzacions de seguretat fins al 2024.

6 Altres programaris instal·lats i destacables fora dels repositoris oficials

Alguns paquets instal·lats i les seves versions són:

- PHP (versió 5.4.16): Línia de comandes per al llenguatge de programació php.
- MariaDB (versió 5.5.50): Base de Dades.
- iptables (versió 1.4.21): Tallafocs
- ModSSl (versió 2.4.6): Mòdul per al servidor web Apache que permet el suport per SSL/TLS.
- httpd (versió 2.4.6): Servidor Apache http.
- MariaDB (versió 5.5.50)

A més, es veu que s'ha configurat wordpress a la màquina (ja que existeixen directoris a /var/www/html/blog/wp-*). Concretament, la versió 4.6:

```
[root@WEBSERVER blog]# pwd
/var/www/html/blog
[root@WEBSERVER blog]# ls -la | grep Word
drwxr-xr-x  5 root   root    4096 16 ago  2016 WordPress-4.6
[root@WEBSERVER blog]#
```

També s'ha detectat magento (/var/www/html/magento/*), que és una plataforma de comerç electrònic escrita en php.

6.1 Vulnerabilitats

Al ser programes molt antics, hi han múltiples exploits per aquestes versions de programari. Sol per mostrar-ne algunes:

1. PHP 5.4.16: Totes les vulnerabilitats es poden trobar aquí. Hi han bastantes de RCE, DoS, etc.
2. MariaDB: Hi han també un munt de CVE's, que s'arreglen al fer upgrade a la versió 10 (que tot i això no és l'última).
3. Servidor Apache: Atacs de DoS, bypass, etc.

7 Contrasenyes

Tal i com s'ha comentat a la secció 1.9, l'usuari sjefe pertany al group wheel, que li permet executar com a root amb la comanda sudo.
El problema, és que la seva contrasenya és: *123456*, que és una contrasenya coneguda i que apareix en tots els diccionaris de contrasenyes. Per tal de crackejar-la, s'ha fet cat del fitxer /etc/shadow i s'ha obtingut el hash de l'usuari:

```
sjefe:$6$8tNfuznnrJUqrmwz$NXsbEY.SQoJzVOQnGJUpMWfd2jr59KphPph6zfB1XPv6fmrAsgPf
V1E2brxeD2MyL15ez1vzdfAZ1JOf7/spu/:......
```

A partir d'aquest hash, s'ha crackejat la contrasenya emprant hashcat amb la comanda:

```
$ hashcat -m 1800 -a 0 hash.txt /usr/share/wordlists/rockyou.txt
```

I s'ha vist que era 123456.

Les contrasenyes haurien de ser:

- Mínim 12 caràcters.
- Que contingui majúscules, caràcters especials, números i minúscules.
- No emprar paraules que puguin estar en un diccionari, emprar mnemotècnica.

El mateix és aplicable per a la contrasenya root, i s'ha d'intentar que la contrasenya dels usuaris i la de root siguin diferents.