ANEXOS

Creación de una LiveCD de instalación personalizada basada en SecureBoot

Autor: Igor González Lerma

Tutor: Joaquín López Sánchez-Montañes

Índice

Tabla de contenido

Anexo 1 – preseed.cfg

```
# Configuración de localización para el idioma, país y «locale».

   d-i debian-installer/locale string es_ES

# Selección de teclado.

   d-i keyboard-configuration/xkb-keymap select es

# Excluir los paquetes del tpm del apt de la instalación.

   d-i pkgsel/exclude string tpm2 tpm2-tools libaspi0_2

# Instalación mínima

   ubiquity ubiquity/minimal_install boolean true
   d-i netcfg/choose_interface select auto
   d-i netcfg/get_hostname string mercury
   d-i netcfg/get_domain string mercury-domain

# Creación de la cuenta de usuario.

   d-i passwd/root-password password igor
   d-i passwd/root-password-again password igor
   d-i passwd/user-fullname string igor
   d-i passwd/username string igor
   d-i passwd/user-password password igor
   d-i passwd/user-password-again password igor


   d-i mirror/country string manual
   d-i mirror/http/hostname string http.us.debian.org
   d-i mirror/http/directory string /debian
   d-i mirror/http/proxy string
   d-i clock-setup/utc boolean true
   d-i time/zone string ES/Madrid
   d-i clock-setup/ntp boolean true
```

```
# Lanzar script justo antes de ejecutar partman

   d-i partman/early_command string sh randompass.sh


# Particionado guiado con LVM cifrado y password obtenido en el script
randompass.sh

   d-i partman-auto/method string crypto

   debconf-set partman-crypto/passphrase $PASS

   debconf-set partman-crypto/passphrase-again $PASS


# Particionado de forma automática sin confirmación.

   d-i partman-lvm/confirm boolean true

   d-i partman/choose_partition select finish

   d-i partman-auto-lvm/guided_size string max

   d-i partman-auto-lvm/new_vg_name string vg_root

   d-i partman-auto/choose_recipe select custom-lvm


# Esquema de particionado

   d-i partman-auto/expert_recipe string            \
      custom-lvm ::                      \
       538 538 1075 free                  \
       $iflabel{ gpt }                   \
       $reusemethod{ }                    \
       method{ efi }                     \
       format{ }                        \
       .                              \
       512 512 512 ext4 $primary{ } $BOOTable{ }       \
       mountpoint{ /BOOT }                  \
       method{ format }                    \
       format{ }                         \
       use_filesystem{ }                    \
       filesystem{ ext4 }                   \
       .                              \
```

```
2048 4095 4096 ext4 $lvmok{ }           \
mountpoint{ / }                    \
lv_name{ root }                    \
in_vg { vg_root }              \
method{ format }                   \
format{ }                      \
use_filesystem{ }                  \
filesystem{ ext4 }                 \
.                              \
2048 4094 4096 ext4 $lvmok{ }           \
mountpoint{ /update }                \
lv_name{ update }                  \
in_vg { vg_root }                  \
method{ format }                   \
format{ }                      \
use_filesystem{ }                  \
filesystem{ ext4 }                 \
.                              \
4096 5000 8192 ext4 $lvmok{ }           \
mountpoint{ /overlay }                   \
lv_name{ overlay }                 \
in_vg { vg_root }              \
method{ format }                   \
format{ }                      \
use_filesystem{ }                  \
filesystem{ ext4 }                 \
.                              \
4096 5500 -1 ext4 $lvmok{ }          \
mountpoint{ /log }                 \
lv_name{ log }                   \
in_vg { vg_root }              \
```

```
        method{ format }                    \
        format{ }                          \
        use_filesystem{ }                   \
        filesystem{ ext4 }                  \
        .                                 \
        512 4090 200% linux-swap $lvmok{ }          \
        lv_name{ swap }                    \
        in_vg { vg_root }                 \
        method{ swap }                     \
        format{ }                         \
        .
    d-i partman-lvm/device_remove_lvm boolean true
    d-i partman-md/device_remove_md boolean true
    d-i partman/confirm_write_new_label boolean true

# Confirmación para escribir las particiones lvm.

    d-i partman/confirm boolean true
    d-i partman-lvm/confirm_nooverwrite boolean true
    tasksel tasksel/first multiselect minimal

# Paquetes individuales a instalar efiBOOTmgr y libcurl3-gnutls.

    d-i pkgsel/include string efiBOOTmgr libcurl3-gnutls
    d-i pkgsel/upgrade select none
    popularity-contest popularity-contest/participate boolean false

# Configurar grub

    d-i grub-installer/with_other_os boolean true
    d-i cdrom-detect/eject boolean false
    d-i apt-setup/cdrom/set-first boolean false
    d-i apt-setup/cdrom/set-next boolean false
    d-i apt-setup/cdrom/set-failed boolean false
```

```
# Lanzar script justo al finalizar la instalación pero cuando todavía se puede
hacer uso de /target

   d-i preseed/late_command string sh efiBOOT.sh
```

Anexo 2 – randompass.sh

```
#! /bin/sh
# Script debconf-compatible
. /usr/share/debconf/confmodule
# Obtención de la password aleatoria
PASS=$(cat /dev/urandom | tr -dc A-Za-z0-9 | head -c 13)
# Obtención del disco
FIRST_DISK=`list-devices disk | head -n1`
# Almacenar variables en el fichero variable.txt para su posterior uso
echo "PASS=$PASS" > /tmp/variable.txt
echo "DISK=$FIRST_DISK" >> /tmp/variable.txt
# Creación del la plantilla
cat > /tmp/randompass.template <<'!EOF!'
Template: random-pass/title
Type: text
Description: LVM PASSWORD
Template: random-pass/pass
Type: note
Description: Please Keep it:${passw}
!EOF!
# Carga de la plantilla
debconf-loadtemplate random-pass /tmp/randompass.template
# Poner el titulo a la pantallla de dialogo
db_settitle random-pass/title
db_input critical random-pass/pass
# Presentación de password en pantalla!
db_subst random-pass/pass passw $PASS
db_go
#  Seteo de las variables preseed desde el script.
debconf-set partman-auto/disk "$FIRST_DISK"
db_set partman-crypto/passphrase $PASS
```

```
db_set partman-crypto/passphrase-again $PASS
```

Anexo 3 – efiBOOT.sh

```
#/bin/bash
# Creación de la carpeta donde almacenar el BOOT
mkdir -p /target/BOOT/efi/EFI/BOOT
# Copia del BOOT firmado en la máquina
cp /cdrom/igor/BOOTX64.EFI /target/BOOT/efi/EFI/BOOT/BOOTX64.EFI
# Copia del contenido de la carpeta igor(scripts, .deb, etc) en la máquina
cp -r /cdrom/igor /target/tmp
# Fichero que contiene las variables password y tipo de disco
cp /tmp/variable.txt /target/tmp/variable.txt
# Copia del script
cp script_ch.sh /target/tmp
cp pcrs.bin /target/pcrs.bin
# Montaje del entorno chroot
for i in /sys /proc /dev; do mount --bind $i /target$i; done
cat <<EOF | chroot /target
# Ejecución del script dentro del entorno chroot
sh /tmp/script_ch.sh
# Salida y desmontaje del entorno chroot
exit;
EOF
for i in /sys /proc /dev; do umount /target$i; done
umount -lf /target
umount /dev/.static/dev
```

```
# Copia de la imagen base mediante el comando dd
dd if=/cdrom/igor/systemimage.sqsh of=/dev/mapper/vg_root-root bs=128K
```

Anexo 4 – script_ch.sh

```
#!/bin/bash

# Obtener las variables, password y el disco, del fichero temporal

PASS=`grep -e PASS /tmp/variable.txt| cut -d'=' -f 2`

DISK=`grep -e DISK /tmp/variable.txt| cut -d'=' -f 2`

# Carga de los módulos del tpm

modprobe tpm

modprobe tpm_tis

modprobe tpm_tis_spi

# Instalación de los paquetes necesarios para interactuar con el TPM

dpkg -i /tmp/igor/libsapi0_2.0-1_amd64.deb

dpkg -i /tmp/igor/tpm2_1.0-1_amd64.deb

dpkg -i /tmp/igor/tpm2-tools_3.0.2-1_amd64.deb

# Interactuación con el TPM y almacenado de la password. En caso que se
encuentre almacenado ya el handle de la memoria persistente se borra.

tpm2_evictcontrol -c 0x81000010

install -d pcrs_instalation

cp pcrs.bin pcrs_instalation

cd pcrs_instalation

echo -n $PASS > secret.bin

tpm2_pcrlist -L sha1:0,2,3,7 > pcrs_instalation.txt

tpm2_createpolicy \--policy-pcr -L sha1:0,2,3,7 -F pcrs.bin -o policy.digest

tpm2_createprimary -a e -g sha1 -G rsa -o primary.context

tpm2_flushcontext -t

tpm2_create -g sha256 -u obj.pub -r obj.priv -C primary.context -L
policy.digest -b "noda|adminwithpolicy|fixedparent|fixedtpm" -i secret.bin

tpm2_flushcontext -t

tpm2_load -C primary.context -u obj.pub -r obj.priv -o load.context

tpm2_flushcontext -t

tpm2_evictcontrol -c load.context -p 0x81000010

# Obtener el numero dentro del gestor de arranque en el que se encuentre
TFG
```

```
EFINUMBER=`efiBOOTmgr | grep TFG | cut -d "*" -f1 | cut -d "t" -f2`

# Eliminar la entrada del arranque

efiBOOTmgr -b $EFINUMBER -B

# Creación de la entrada del arranque apuntando al BOOT firmado

efiBOOTmgr -c -d $DISK -p 1 -w -L TFG -l '\EFI\BOOT\BOOTX64.EFI'

# Borrado de todos los ficheros

rm secret.bin pcrs.bin policy.digest primary.context obj.pub obj.priv
load.context
```

Anexo 5 – signed_BOOT.sh

```
#/bin/bash

# Montaje de efi.img

install -d /tmp/BOOTdebian

mount netinst/BOOT/grub/efi.img /tmp/BOOTdebian

# Firmado de los arranques tanto la Live-CD como el suministrado

sbsign --key certificates/db.key --cert certificates/db.crt --output
/tmp/BOOTdebian/efi/BOOT/BOOTx64.efi /tmp/BOOTdebian/efi/BOOT/BOOTx64.efi

sbsign --key certificates/db.key --cert certificates/db.crt --output
/tmp/BOOTdebian/efi/BOOT/grubx64.efi /tmp/BOOTdebian/efi/BOOT/grubx64.efi

sbsign --key certificates/db.key --cert certificates/db.crt --output
netinst/igor/BOOTX64.EFI netinst/igor/BOOTX64.EFI

# Verificación del firmado

sbverify --cert certificates/db.crt /tmp/BOOTdebian/efi/BOOT/grubx64.efi

sbverify --cert certificates/db.crt /tmp/BOOTdebian/efi/BOOT/BOOTx64.efi

sbverify --cert certificates/db.crt netinst/igor/BOOTX64.EFI
```

Anexo 7 – createiso_with_preseed.sh

```
#!/bin/bash

# Permisos de lectura sobre el directorio.

chmod +w -R netinst/install/

# Creación de carpeta.

mkdir -p tmp/initrd

cd tmp/initrd

# Extracción del initrd.gz.

zcat ../../netinst/install.amd/initrd.gz | cpio -iv

# Copia de ficheros y scripts.

cp ../../preseed.cfg .

cp ../../randompass.sh .

cp ../../efiBOOT.sh .

cp ../../script_ch.sh .

cp ../../pcrs.bin .

mkdir -p lib/firmware

cp -a ../../rtl_nic lib/firmware

# Creación del nuevo initrd.gz.

find . -print0 | cpio -0 -H newc -ov | gzip -c > ../../netinst/install.amd/initrd.gz

cd ../../

rm -fr tmp/initrd

# Eliminación de permisos.

chmod -w -R netinst/install

# Permisos de lectura md5sum.txt.

chmod +w netinst/md5sum.txt

# Se crea el md5sum con los cambios.

find netinst -follow -type f ! -name md5sum.txt -print0 | xargs -0 md5sum > netinst/md5sum.txt

# Eliminación de permisos.

chmod -w netinst/md5sum.txt

cd netinst
```

```
# Creación de la imagen ISO.

xorriso -as mkisofs -o netinst_preseed.iso -isohybrid-mbr /usr/lib/ISOLINUX/
isohdpfx.bin -c isolinux/BOOT.cat -b isolinux/isolinux.bin -no-emul-BOOT -
BOOT-load-size 4 -BOOT-info-table -eltorito-alt-BOOT -e BOOT/grub/efi.img -
isohybrid-gpt-basdat -no-emul-BOOT ../netinst

cd ..

mv netinst/netinst_preseed.iso .
```