Artificial truth

'Security/privacy made in Switzerland' has turned from a selling point into a punchline.

Hardening Proxmox against physical attacks

Sat 06 February 2016 — download

I'm using Proxmox (with a custom kernel) on my old (but still loved) T400 as an hypervisor, with its access only exposed in a dedicated VLAN.

While I'm pretty confident that it's non-privial to pwn it from the network, I was worried about physical attacks for a number of reasons. This article is about some basic things that I did to sleep better at night.

Encrypting proxmox

It seems that people who wants to encrypt their proxmox setup are first installing debian with FDE, then transforming it into proxmox, but since I'm too lazy to do this, I encrypted directly my proxmox instance. Of course, don't forget to shut down your vm before doing this.

```
apt-get install --no-install-recommends cryptsetup

# backup our vm.
cd /var/lib/vz
tar cvzf /root/vz.tar.gz ./*

# encrypt the partition
cd /root
umount /dev/pve/data
cryptsetup --verify-passphrase luksFormat /dev/pve/data
cryptsetup luksOpen /dev/pve/data data
mkfs.ext4 -v /dev/mapper/data
echo "data /dev/pve/data none luks" >> /etc/crypttab
sed -i 's#/dev/pve/data#/dev/mapper/data#g' /etc/fstab
```

```
# restore our vm
mount /var/lib/vz
cd /var/lib/vz
tar xvzf /root/vz.tar.gz
rm /root/vz.tar.gz

# encrypt the swap
swapoff -a
echo "swap    /dev/pve/swap    /dev/urandom swap" >> /etc/crypttab
# check your /etc/fstab for `/dev/mapper/swap    none    swap    default    0    0`

# apply the modifications
sync
reboot
```

Encrypting backups

Proxmox can backup your virtual machines directly on an NFS server, but it can't encrypt them before doing so. This is why I told it to put the backups in my local `/backups` folder, and put this in a crontab to take care of encrypting and moving the files:

```
#!/bin/bash

cd /backups/dumps

# encrypt the backups
gpg -r 0x04D041E8171901CC --batch --yes --encrypt-files *.vma.lzo

# move them over nfs to a remove machine
mount -t nfs 10.10.10.10:/tank/bak/dump /media/nfs -o rw,async,noatime,nolock
find /media/nfs -ctime 7 -type f -delete
mv *.vma.lzo.gpg /media/nfs
sync
umount /media/nfs
```

Panic shutdown

Do you know about inception framework? It's a fun thing that uses DMA to unlock your computer, dump memory, pop root shells or even spawn a metasploit reverse shell.

This is why you may want to blacklist some kernel modules, and/or if your motherboard has native DMA, start the following script at boot-time. It will trash your RAM and shutdown your computer when the kernel detects a new device.

```
#!/bin/bash

KEYWORDS='(Direct-Access|firewire|usb|pcmcia|scsi|…)'  # add your own keywords

fifo=$(tempfile).$$
mkfifo "${fifo}" || exit 1
nb_positives=$(dmesg | grep -E -i -c "${KEYWORDS}")
nb_positives=$(($nb_positives + 1))

dmesg -w >${fifo} &
grep -E -i -m "${nb_positives}" "${KEYWORDS}" "${fifo}" > /dev/null

#sync  # uncomment me if you want

# Trash RAM and shutdown everything
sdmem -fll
poweroff -f
```

Disable hibernation on lid closing

But default, Debian will put the computer into hibernation when the lid of the laptop is closed, allowing an attacker to get an unencrypted dump of your system. This is the magic one-liner to disable this behaviour:

```
sed -i 's/#HandleLidSwitch=hibernate/HandleLidSwitch=ignore/g' /etc/systemd/logind.conf
service systemd-logind restart
```

Disable the stupid annoying popup

Promox has by default an annoying popup with a stupid supported way to disable it. Here is an alternative:

```
sed -i "s/data.status !== 'Active'/false/g" /usr/share/pve-manager/ext4/pvemanagerlib.js
```
