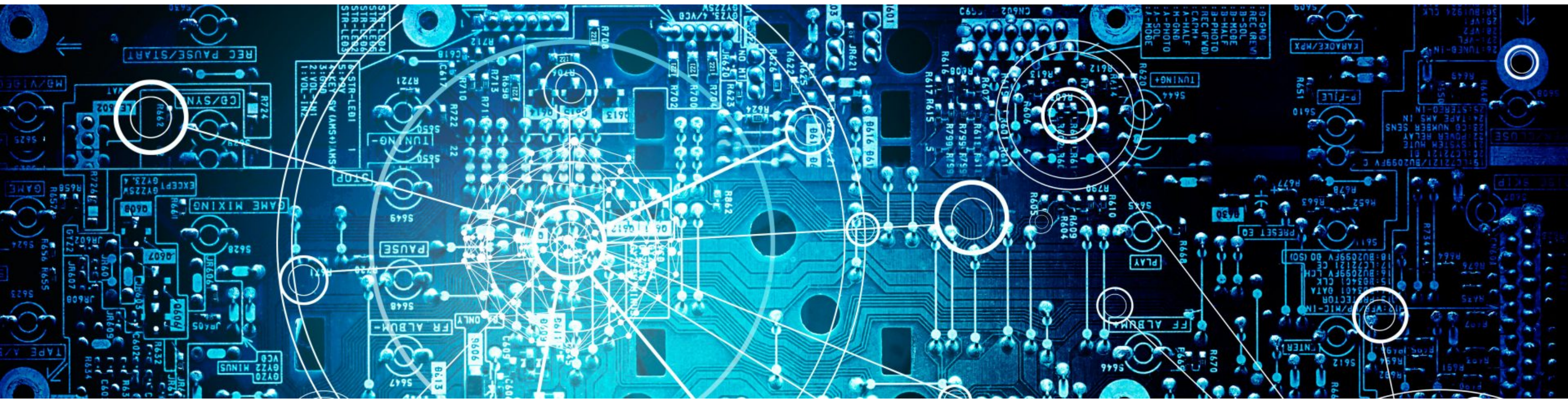
The Latest News from Research at Kudelski Security



SOFTRAID AND CRYPTO FOR OPENBSD 5.3/5.4

September 19, 2013 Romain Aviolat System administration 5 comments

DISCLAIMER: This how-to must be taken as is, it should not replace the official documentation and is not meant to do so. It may be useful as these features are quite new and not heavily documented on the net.

OpenBSD supports booting from a raid volume since version 5.3. Before that, the way to have redundancy for the root partition was to place /altroot on a second disk and to manually switch to it in the case of failure of the first disk.

It also supports booting from encrypted volumes. Sadly it doesn't supports booting from raid+encrypted volumes yet (they're working on it).

For my setup I want to have redundancy (meaning RAID1) and, ideally, crypto everywhere. As it's not yet possible I decided to create one softraid1 partition containing “/” and a second softraid1/encrypted partition containing some mount points: /tmp, /var, /usr, /usr/X11R6, /usr/local, /usr/src, /usr/obj

Below, I’ll describe the steps to obtain such setup.

Setup part:

1. boot an install media (I used PXE here)

2. drop to a (S)hell in the installation program

3. create the devices nodes:

```
cd /dev
sh MAKEDEV sd0    # <- our 1st HDD
sh MAKEDEV sd1    # <- our 2nd HDD
sh MAKEDEV sd2    # <- our 1st RAID1 volu
sh MAKEDEV sd3    # <- our 2nd RAID1 volu
sh MAKEDEV sd4    # <- our 2nd RAID1 volu
```

4. initialize the MBR (i = initialize, y = answer yes):

```
fdisk -iy sd0
fdisk -iy sd1
```

5. partition the disks. We will create one “a” partition for the raid1 root fs, “b” for the encrypted one and “d” for the raid1 + crypto

```
disklabel -E sd0
a a   #1G, FS type: RAID root partition
a b   #2G, swap partition (OpenBSD automa
a d   #[all available space], FS type: R
w     #write
q     #quit
```

6. do the same for sd1:

```
disklabel -E sd1
```

7. create two RAID1 devices, one for the / and one for the encrypted partition:

```
bioctl -c 1 -l /dev/sd0a,/dev/sd1a softra
bioctl -c 1 -l /dev/sd0d,/dev/sd1d softra
```

8. two devices should be created: sd2 and sd3, we will empty their first few sectors:

```
dd if=/dev/zero of=/dev/rsd2c bs=1m coun
dd if=/dev/zero of=/dev/rsd3c bs=1m coun
```

9. create a partition on the new devices:

```
disklabel -E sd2
a a    #whole disk, FS type BSD
w      #write
q      #quit
```

10. and for the crypto partition:

```
disklabel -E sd3
a a    #whole disk, FS type RAID
w      #write
q      #quit
```

11. create the crypto partition:

```
bioctl -c C -r 8192 -l /dev/sd3a softrai
```

12. start the installer:

```
install
```

13. choose to use (W)hole disk sd2 and partition it like that:

```
a a    #whole disk, FS BSD mount /
w      #write
q      #quit
```

14. Which one do you want to initialize? sd4

```
a d   #size 4G, /tmp
a e   #size 7G, /var
a f   #size 2G, /usr
a g   #size 1G, /usr/X11R6
a h   #size 7G, /usr/local
a i   #size 2G, /usr/src
a j   #size 2G, /usr/obj
a k   #size [left space ~24G], /home
w     #write
q     #quit
```

15. choose your mirror and the stuffs you want to install

16. finish the install and reboot

OS part

the boot should yell that some partitions can't be mounted and drop you to a shell. It's normal, crypted partitions aren't supported out of the box by the boot process. You will have to decrypt the partition by hand (sd3a), then you will be asked for your passphrase:

```
bioctl -c C -l /dev/sd3a softraid0 && ex
```

next step we will do a modification of the boot order asking for the passphrase of the encrypted device instead of "crashing" with error messages. We will add the following script to the end of the file **/etc/rc.conf.local**:

```
bioctl sd4 > /dev/null 2>&1

if [ $? -ne 0 ]; then
    echo unlocking encrypted device
bioctl -c C -l /dev/sd3a softraid0
  fi
```

This script will hang the boot process until the passphrase for the encrypted device is entered, then it will be able to mount the system partitions that are on the crypted sd4 device (/tmp, /usr, …).

The location of my script is a bit tricky. In theory I should have put it in /etc/rc.securelevel, the problem is that rc.securelevel is called too late in the boot process, after the mount of the partitions.

17. reboot and check if the passphrase is asked and working, if not redo the step “17”.

Next step is only required for OpenBSD 5.3, it has been corrected since

The last “problem” we have is when we shutdown your system, OpenBSD will remove the devices approximately in the same order they are created. “sd3” gets shutdown before sd4, hence sd4 will be unable to write metadata to the underlying sd3 device. The next reboot will say that the sd4 device was not correctly unmounted and may request a fsck. To avoid this we will create a script that unmount the partitions located on the crypted device and then destroy the crypted disk before the sd3 devices gets removed.

18. add the following to the **/etc/rc.shutdown** file:

```
for device in $mounted; do
bioctl $device | grep -q CRYPTO
iscrypto="$?"

if [ "$iscrypto" -eq 0 ]; then
echo $device is a crypto device, umount
# umount partitions related to crypto de
tounmount=`mount | grep "^/dev/$device"

for cryptdev in $tounmount; do
umount -f $cryptdev
    done

bioctl -d $device

```

```
    fi
done
```

Last but not least, if you have some sensitive files containing passwords like IPsec secrets or pf configurations it may be good to put them into an encrypted partition, like /home, and symlink them to their official location.

Thanks for reading!

Romain