Raspberry Pi - Feature #289

Grenade-style Encryption on Raspberry Pi

01/04/2014 05:40 PM - Daniel Curtis

Status:	Closed	**Start date:**	01/04/2014
Priority:	Normal	**Due date:**	
Assignee:	Daniel Curtis	**% Done:**	100%
Category:	Encryption	**Estimated time:**	4.00 hours
Target version:	Arch Linux	**Spent time:**	4.00 hours

Description

Prerequisites:

- 4GB or larger SD or MicroSD card
- RaspberryPi, I am using Model B / 512 MB RAM (predecessors should work as well)
- Monitor and keyboard
- Internet access

Notes

We will run no swap, like the default image, and leave the initial install on the 1.8 GB partition as a potential rescue disk. It is a good idea to be able to run ARM binaries if you ever need to recover.
Having a rescue partition that you can boot into will be invaluable and would have saved me from my last screw-up.

Download and install the Arch ARM image

```
cd ~/Downloads
wget http://archlinuxarm.org/os/ArchLinuxARM-rpi-latest.zip
unzip ArchLinuxARM-rpi-latest.zip
```

Run:

```
sudo dd if=archlinux-hf-2013-02-11.img | pv | dd bs=1M of=/dev/sdb
```

You know this bit, now put the card into your Pi.

Boot up and update the system

```
pacman -Syu
```

Based on the current image, you'll have to fix a few pacnew files.

Install rsync and vim

```
pacman -S vim rsync mkinitcpio
```

Reboot and make sure everything is working

```
reboot
```

Check the current kernel

```
uname -a
```

3.10.25-1-ARCH

Do any normal tasks now as you would otherwise when setting up a system.
All of this is optional:

1. Set your root password.
2. Set an IP, hostname, etc.
3. Add users, your favourite shell, configure any services, etc.
4. Setup whatever else you don't want to setup later or don't mind having in a rescue environment.

We will be rsyncing this setup to the encrypted partition.
Reboot again, make sure everything is good.

```
reboot
```

Create a new partition for encrypted space usage

Create a new partition of at least 2 GB, I normally just fill the rest of the SD card, it's up to you though.
Use fdisk on /dev/mmcblk0 and create a new primary partition.

```
fdisk /dev/mmcblk0
n
p
3
[Enter]
[Enter]
w
reboot
```

Login and check the new partition table

```
fdisk /dev/mmcblk0
p
q
```

There should now be an mmcblk0p1, mmcblk0p3, and mmcblk0p5

- mmcblk0p1 being the vfat boot partition, **do not mess with it**
- mmcblk0p3 being the new partition
- mmcblk0p5 being our current 1.8 GB root (recovery)

Securely erase new partition

dd /dev/zero over the new partition (p3), just to add a minimal amount of safety:

```
dd if=/dev/zero of=/dev/mmcblk0p3 bs=1M
```

This will take a long time. You'll get several kernel IO hung timeout messages while this runs, but it will finish. Be patient.

Create keyfile

```
dd if=/dev/urandom of=/path/to/keyfile.key bs=1024 count=8
```

And move it to a USB drive

```
mv keyfile.key /path/to/usb/keys
```

Identify the Universally Unique ID for the USB drive

```
blkid /dev/sdb1
```

Note: The actual device may vary, make sure to double check which device occupies which ID

Create the encrypted partition

When dd finishes, create a LUKS volume on /dev/mmcblk0p3

```
cryptsetup  -i 15000 -c aes-xts-plain:sha512 -y -s 512 luksFormat /dev/mmcblk0p3 /path/to/keyfile.key
```

Open the LUKS volume and put a filesystem on it:

```
cryptsetup -d /path/to/keyfile luksOpen /dev/mmcblk0p3 root
mkfs.ext4 /dev/mapper/root
```

Clone the running system into the encrypted partition

Mount the new filesystem:

```
mount /dev/mapper/root /mnt
```

Rsync the current system over:

```
rsync --progress -axv / /mnt/
```

Don't forget the trailing /*** on */mnt**/

This will take a long time.

Run the rsync again, just to make sure you have everything, this will be much quicker.

Update the initial ramdisk

We're not ready to do our "over SSH" unlock just yet, let's make sure we can at least boot the encrypted root.

Edit /etc/mkinitcpio.conf and make sure this line has:
HOOKS="base udev autodetect modconf block usb keyboard encrypt filesystems fsck"
MODULES="vfat ext4"

Now generate an initrd:

```
mkinitcpio -k 3.10.25-1-ARCH -g /boot/initrd -c /etc/mkinitcpio.conf
```

Note: If you're reading this in the future, make sure -k has the correct kernel string **or else you're screwed**. The proper kernel string can be found using uname -a:
uname -a

To find the kernel version of a recent package upgrade, it can be found using:

```
pacman -Qs linux-raspberrypi
```

The correct kernel string for the -k flag is ***VERSION_NUMBER*-ARCH**

Edit /boot/config.txt and add to the end:

> initramfs initrd 0x00f00000

Configure the bootloader to use the encrpyted partition as root

Edit the kernel command line, leave whatever is there alone, add or modify the following (file is /boot/cmdline.txt):

> ipv6.disable=1 avoid_safe_mode=1 selinux=0 plymouth.enable=0 smsc95xx.turbo_mode=N dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mapper/root rootfstype=ext4

cryptdevice=/dev/mmcblk0p3:root cryptkey=/dev/disk/by-uuid/6F41-ED22:vfat:/keys/cimex.key initrd=0x00f00000 elevator=noop rootwait

The syntax for cryptdevice and cryptkey is:

- cryptdevice=*device*:*device-mapper-name*
- cryptkey=*device*:*fs-type*:*path*

Up to you if you want allow-discards or not, allowing discards has been said to have poor security implications

```
cryptdevice=/dev/mmcblk0p3:root:allow-discards root=/dev/mapper/root initrd=0x00f00000
```

Be sure to leave the "ro" option there.

Now add the following to fstab, edit /mnt/etc/fstab and ensure:

```
/dev/mmcblk0p1      /boot       vfat    defaults                    0       0
/dev/mapper/root    /           ext4    defaults,commit=120         0       1
```

Change options to what you want.
Reboot and hope it works!

```
reboot
```

Testing the grenade encryption

From now on, booting up the RPi with this disk will require the USB drive that the key was transferred to. Without it a password prompt will appear, and fail to boot.

Your root filesystem is now the LUKS encrypted mmcblk0p3 and not mmcblk0p5.

Make sure the **HOOKS** and **MODULES** line in /etc/mkinitcpio.conf on mmcblk0p3 matches what you edited before on mmcblk0p5.

Make sure /etc/fstab on this partition is correct (you did it right if it booted and you can do touch foo and write a file).

If you make any changes, reboot and ensure you can boot without any problems (if you are going to reboot, rebuild the initrd before you do -- just to be on the safe side).

You must rebuild the initrd after every kernel update.

Archlinux ARM kernel packages do not run mkinitcpio automatically after kernel update like regular x86_64 Arch does.
When rebuilding the initrd after an update, you must provide the correct string appropriate to the new kernel.

```
mkinitcpio -k CORRECT-KERNEL-VERSION -g /boot/initrd -c /etc/mkinitcpio.conf
```

The easiest way to recover, in my opinion, would be to:

- Leave the mmcblk0p5 partition intact
- When something goes wrong, put the SD card into another system and edit cmdline.txt to boot off **mmcblk0p5**:

  ipv6.disable=1 avoid_safe_mode=1 selinux=0 plymouth.enable=0 smsc95xx.turbo_mode=N dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p5 rootfstype=ext4 initrd=0x00f00000 elevator=noop rootwait

- Use the system on mmcblk0p2 to unlock the LUKS volume etc, and chroot into it, whatever you need to do to fix the system.

  ```
  cryptsetup -d /path/to/keyfile luksOpen /dev/mmcblk0p3 root
  mount /dev/mapper/root /mnt
  chroot /mnt
  ```

Resources

- https://gist.github.com//pezz/5310082
- https://bbs.archlinux.org/viewtopic.php?pid=1035124
- http://wiki.gentoo.org/wiki/DM-Crypt_LUKS
- https://wiki.archlinux.org/index.php/Mkinitcpio
- https://wiki.archlinux.org/index.php/Disk_Encryption
- http://www.freedesktop.org/software/systemd/man/crypttab.html

History

#1 - 01/04/2014 05:56 PM - Daniel Curtis

- Description updated

#2 - 01/05/2014 01:44 AM - Daniel Curtis

- Description updated

#3 - 01/13/2014 08:25 AM - Daniel Curtis

- Subject changed from Grendade-style Encryption on Raspberry Pi to Grenade-style Encryption on Raspberry Pi

- Status changed from Resolved to Closed

#4 - 03/07/2015 07:28 AM - Daniel Curtis

- Category set to Encryption

- Target version set to Arch Linux