Artemis

As all projects, this one needed to have a name as well. *Artemis* originates from the old Greek and means - amongst others - unharmful or safe. And exactly this Artemis should deliver to you: safety for your data and no harm to your communications.

Requirements

Hardware	Quantity
RaspberryPi + MicroUSB / Power Adapter	1
MicroSD Card	1
USB Thumbdrive	1
RJ45	1

Prerequisites

Creating temporary environment

Ensure that you download and install the ArchLinux ARM to a non-infected SD card on a secured computer. We recommand the following way of proceeding.

1. Download the latest ArchLinux ISO and its checksum.
2. Check against each other.
3. Copy the ISO to a clean USB thumbdrive.

<sxh bash;># dd bs=1m if=./archlinux-YYYY.MM.DD-dual.iso of=/dev/sdX</sxh>

1. Plug in a computer and boot it up.

ArchLinux ARM

1. Start fdisk to partition the SD card:

```
fdisk /dev/mmcplk0
```

1. At the fdisk prompt, delete old partitions and create a new one:
    1. Type o. This will clear out any partitions on the drive.
    2. Type p to list partitions. There should be no partitions left.
    3. Type n, then p for primary, 1 for the first partition on the drive, press ENTER to accept the default first sector, then type +100M for the last sector.
    4. Type t, then c to set the first partition to type W95 FAT32 (LBA).
    5. Type n, then p for primary, 2 for the second partition on the drive, and then press ENTER twice to accept the default first and last sector.
    6. Write the partition table and exit by typing w.
2. Create and mount the FAT filesystem:

<sxh>

```
    mkfs.vfat /dev/sdX1
    mkdir boot
    mount /dev/sdX1 boot</sxh>
- Create and mount the ext4 filesystem:
    <sxh>
    mkfs.ext4 /dev/sdX2
    mkdir root
    mount /dev/sdX2 root
    </sxh>
- Download and extract the root filesystem (as root, not via sudo):
  <sxh>
  wget http://archlinuxarm.org/os/ArchLinuxARM-rpi-latest.tar.gz
http://archlinuxarm.org/os/ArchLinuxARM-rpi-latest.tar.md5
  # GNU/Linux
  md5sum ArchLinuxARM-rpi-latest.tar.gz > md5sums.md5
  cat ArchLinuxARM-rpi-latest.tar.gz.md5 >> md5sums.md5
  md5sum -c md5sums.md5
  # MacOS X
  md5 ArchLinuxARM-rpi-latest.tar.gz > ArchLinuxARM-rpi-
latest.tar.gz.original.md5
  diff ArchLinuxARM-rpi-latest.tar.gz.original.md5 ArchLinuxARM-rpi-
latest.tar.gz.md5
  # If there is no return given / 0 as exit code, the match was successful.
  tar -xf ArchLinuxARM-rpi-latest.tar.gz -C root
  sync
  </sxh>
- Move boot files to the first partition:
  <code>
  mv root/boot/* boot
  </code>
- Unmount the two partitions:
  <code>
  umount boot root</code>
- Insert the SD card into the Raspberry Pi, connect ethernet, and apply 5V
power.
- Use the serial console or SSH to the IP address given to the board by your
router. The default root password is 'root'.
```

When finished, fire up the RaspberryPi with the new system connected to a display and keyboard, but **not** to a network. Log in with username **root** and password **root**.

SSHD

Configuration

Note: Ensure to be on a separate secured network or even better, plug yourself in a non-networked computer.

Firewall

In order to network-secure Artemis tight as possible, we are just going to drop every input and output traffic by default, and whitelist whatever services we need.

Purging iptables

Safe the following commands in /etc/iptables/purge-all-rules.sh

<sxh bash;Purging all iptable rules> iptables -F iptables -X iptables -t nat -F iptables -t nat -X iptables -t mangle -F iptables -t mangle -X iptables -t raw -F iptables -t raw -X iptables -t security -F iptables -t security -X iptables -P INPUT ACCEPT iptables -P FORWARD ACCEPT iptables -P OUTPUT ACCEPT </sxh>

<sxh bash;title:Dropping all packages;> # iptables -N TCP # iptables -N UDP </sxh>

References

[0] RaspberryPi With Root Partition Encryption, Unlocked Using Flash Drive
[1] Archlinux ARM encrypted root