#SECURE HOME GATEWAY

Generating the Medkit Image

The medkit is the base system image of the Secure Home Gateway (SHG). It is required to turn a standard Turris Omnia router into a Secure Home Gateway (SHG) router.

The following procedure describes how to generate this image.

To generate the medkit image

A. Set up the Openwrt Build Environment

1. Clone and `cd` into the Openwrt official repository:

```
git clone https://git.openwrt.org/openwrt/openwrt.git
cd openwrt
```

2. From Turris' official repository (https://repo.turris.cz/hbs/omnia/packages/), open the git-hash file (https://repo/turris.cz/hbs/omnia/packages/git-hash).
This file contains the SHA-1 of the various repositories used to build the official medkit.

   a. Check out the specific `openwrt` version in your local copy.

   b. Open the `feeds.conf.default` file and set the various feeds to the SHA-1 found in the git-hash file. See the following syntax example:

```
src-git packages https://git.openwrt.org/feed/packages.
git^d3c324d8bc01638cf9bef4e34c8849fc16380b04
```

3. Add the SHG feed:

   a. Add the following line to the `feeds.conf.default` file:

```
src-git cira https://github.com/CIRALabs/SHG-feeds.git
```

   b. Update the feeds and activate the `shg` package:

```
./scripts/feeds update
./scripts/feeds install shg
```

4. Configure and compile the toolchain and kernel:

   a. Retrieve the `config-turris` file from: https://raw.githubusercontent.com/CIRALabs/shg_os4_build/shg/shg/config-turris

   b. Copy the file to the `openwrt` clone to `.config`.

   c. Run `make menuconfig`.

   d. Start compiling the toolchain:

```
make -j$(nproc) tools/download; make -j$(nproc) toolchain/download
ionice -c 3 nice -n19 make -j$(nproc) tools/install;
ionice -c 3 nice -n19 make -j$(nproc) toolchain/install
make target/linux/compile –j$(nproc)
```

5. Compile the SHG packages:

   a. Run `make menuconfig` and select the SHG package (Network → shg).

   b. Compile the package and its dependencies:

```
make package/feeds/cira/shg/{clean,compile}
```

The compiled packages (`ipk`) *will be located in the* `bin/packages/arm_cortex-a9_vfpv3/cira/` *directory.*

B. Generate the Medkit

1. Generate the keys:

   Note: As `usign` is required for this procedure, we recommend that you generate a first medkit without signing. See step 3 Build medkit below and remove the `--sign` and `--overlay` parameters.

   a. Locate the `usign` utility that was built during medkit generation: `medkit/turris-tools/usign/usign`

   b. Generate the keys: `usign –G –p repo.pub –s repo.key –c "CIRA Repository key"`

2. Create your own repository:

   a. Enter `mkdir repository`

   b. Move all IPK to `repository` folder

c. Enter `cd repository`

d. Enter `../ipkg-make-index.sh . > Packages.manifest`

e. Enter `grep –vE '^(Maintainer|LicenseFiles|Source|Require)' Packages.manifest > Packages`

f. Enter `gzip -9nc Packages > Packages.gz`

g. Enter `usign –S –m Packages –s path_to/repo.key`

3. Build the medkit:

a. Install fakeroot, libcurl devel (> 7.30), libevent devel, cmake, lua 5.1, asciidoc, uriparser, libb64, uthash

b. Copy public key to overlay: `mkdir –p overlay/etc/opkg/keys/; cp repo.pub overlay/etc/opkg/keys/<key id find in repo.pub>`

c. Enter `mkdir medkit; cd medkit`

d. Enter `../../generate_medkit –t omnia –b hbs –updater-script ../shg.lua --lists lxc -1 en --sign ../repo.key -- overlay ../overlay/`

Note: We have added the lxc list to include LXC utils that are used by mud supervisor. This also includes the "wget" tool that is required by opkg, but not installed by default.