signed autoupdates for your zoo of embedded devices done right

Bastian Bittorf

February 22, 2017

signed autoupdates for embedded devices

- we discuss the concept, not the implementation
- we show possible workflows
- focus on OpenWrt/LEDE
  - can work for all your IoT/IoE devices
  - where updates are firmware-images
- secure, scalable, affordable, practicable

example: our test- and community-network

- 95 devices
  - e.g. wireless routers, NAS boxes, servers
  - everything runs OpenWrt or LEDE
- 35 different hardware models with 10 different architectures
  - e.g. TP-LINK TL-WDR4900 v1
  - e.g. Linksys WRT54G/GS/GL
  - e.g. Linksys WRT1900AC v1
- 20 different usecases
  - e.g. NAS, webcam, VPN, USBprinter,
  - e.g. DSLR-camera, WWAN, USB-audio...
- 3 different update-modi (stable, beta, testing)
  - sums up to: **171 different images**

problem: not much horsepower

- only a few admins
- who feels responsible for what target?
- who thrusts which admin?
- release early, release often?
- we need to protect ourselves!

typical release management

- stable
- beta
- testing (*aka* “avantgarde”)
- nightly

what is needed

- image + tarballs
- easy/simple check if there is a new version
- balance your individual threatlevel
- for powerusers
  - override-all aka “download and install NOW”
- for mama
  - configurable good-enough thrustlevel
  - how many signatures needed for ultimate thrust

needed infrastructure

- a build machine (or many)
  - crosscompiler/wrapper/
- a webserver for the images
- `usign` aka `signify` (OpenBSD)
- automated testing
- human testers

a manual walktrough

imagine, you are a router and you need fresh meat

- you know your... *network*
- you know your... *hardware-model*
- you know your... *update-mode*
- you know your... *usecase*

http://server/networks/berlin/firmware/TP-LINK TL-WR1043ND v2/testing/.usecase_hash/info.json

```
{
  "foo": "bar",
}
```

...not ready yet!