Firmware Rollout

Wie kommen kompilierte Firmware-Images auf Server und Geräte?

Checkliste

- Firmware als stable bauen
- Manifest signieren lassen (Wahl der Schlüssel so, dass für alle aktiven Versionen gültige Signaturen vorhanden sind)
- Firmware und signiertes Manifest ablegen (gw1, /var/www/firmware/stable/0.xx)
- Symlink latest auf entsprechende Firmware setzen
- Info über Mailingliste
- Blogeintrag schreiben
- Ankündigung auf Twitter/ Facebook
- Anpassung der Download-Seite (https://md.freifunk.net/mitmachen/firmware/), Links und Changelog

Firmware bauen

Wir nutzen Gluon als Firmware für unsere Knoten. Das erfordert einige angepasste Dateien im Unterordner `site` des geklonten Gluon-Repositories, für Freifunk Magdeburg liegen diese bei GitHub derzeit. In der entsprechenden `README.md` ist eine Anleitung zum Bauen der Firmware für FFMD.

- Build mit Docker

Per default wird für *experimental* gebaut. Für die anderen Branches muss das Argument *beta* oder *stable* angegeben werden:

```
site/build.sh beta
```

Firmware signieren

In der `site.conf` sind kryptografische Schlüssel der Entwickler hinterlegt. Für ein *experimental* build ist die Signatur eines Entwicklers, für *beta* und *stable* jeweils zwei Signaturen nötig, damit Autoupdate auf den Knoten funktioniert.

Gluon enthält ein Skript zum Signieren, Voraussetzung sind die *ecdsautils*:

```
contrib/sign.sh $SECRETKEY output/images/sysupgrade/experimental.manifest
```

Wie die Entwickler die Manifest-Dateien zum Hinzufügen mehrerer Signaturen austauschen, bleibt ihnen überlassen. Eine Möglichkeit ist bspw.: der erste Entwickler lädt alle Images plus Manifest auf

den Webserver. Weitere Entwickler laden sich das Manifest von dort, signieren, laden die neue Version wieder auf den Webserver.

Schlüsselerzeugung mit ecdsautils

Installation kann aus dem Debian-Paket, dem Paket der gewünschten Distribution oder aus dem Quellcode erfolgen.

bis 0.3.2

```
% ecdsakeygen -h
ecdsautils 0.3.2

Usage: ecdsakeygen [-s] [-p] [-h]
        -s      generate a new secret on stdout
        -p      output public key of secret read from stdin
        -h      display this short help and exit
```

```
% ecdsakeygen -s > secret
% cat secret
68b12c0eaf88bf17fbcdf560780136b9cc4be352fb8aa7148215fbd65887db7b
```

```
% ecdsakeygen -p < secret
1f63ef7450760af9062ff697995eb536eef25a555822087fa4cfd9a82d9faa79
```

ab 0.4.0

```
% ecdsautil generate-key > secret
% cat secret
68b12c0eaf88bf17fbcdf560780136b9cc4be352fb8aa7148215fbd65887db7b
```

```
% ecdsautil show-key < secret
1f63ef7450760af9062ff697995eb536eef25a555822087fa4cfd9a82d9faa79
```

Schlüssel sicher speichern

Der erzeugte Schlüssel ist ebenso sorgfältig abzulegen wie private Schlüssel von PGP oder SSH. Leserechte sollte nur der Nutzer haben:

```
% chmod go-rwx secret
% ls -l secret
-rw------- 1 adahl adahl 65 Jan 31 15:07 secret
```

Firmware hochladen

Die Firmware-Images plus Manifest mit Signatur(en) wird vom Build-Rechner auf den Webserver kopiert, dazu kann bspw. *rsync* genutzt werden.

```
rsync -rlthcvz output/images/
gw1.md.freifunk.net:/var/www/firmware/stable/0.37
```

Anschließend auf dem Webserver kontrollieren, dass die Rechte korrekt gesetzt sind und ggf. symbolische Links anpassen.

Der Webserver hat die Images in einer Ordnerstruktur liegen (nicht doppelt), liefert aber je nach VHOST ggf. unterschiedliche Seiten/Pfade aus. Über http://firmware.md.freifunk.net/ bspw. Images zum Download durch Nutzer über das Internet und den Webbrowser und über http://experimental.updates.firmware.ffmd/ bspw. experimental Updates für die Knoten. Im Freifunk-Netz muss dementsprechend die Zone `.ffmd` korrekt auf den Webserver aufgelöst werden und der Webserver muss intern im Freifunk-Netz erreichbar sein (nicht (nur) aus dem Freifunk-Netz über das Internet).

Symlinks:

- LATEST im Firmware-Oberverzeichnis (z.B. stable/LATEST)
- manifest auf z.B. manifest.stable

From:
https://wiki.netz39.de/ - **Netz39**

Permanent link:
https://wiki.netz39.de/freifunk:anleitungen:rollout

Last update: **2018-07-24 21:31**