

SECURE DEVICE INSTALL

draft-wkumari-opsawg-sdi-03

BACKGROUND / DISCLAIMER

- Idea percolating for a while
`draft-ietf-anima-autonomic-control-plane` final impetus
- Designed for **simplicity**
 - **implement and use**
- Examples use Cisco autoinstall, but works with anything with a config.
- Does not solve all use cases, solves common one

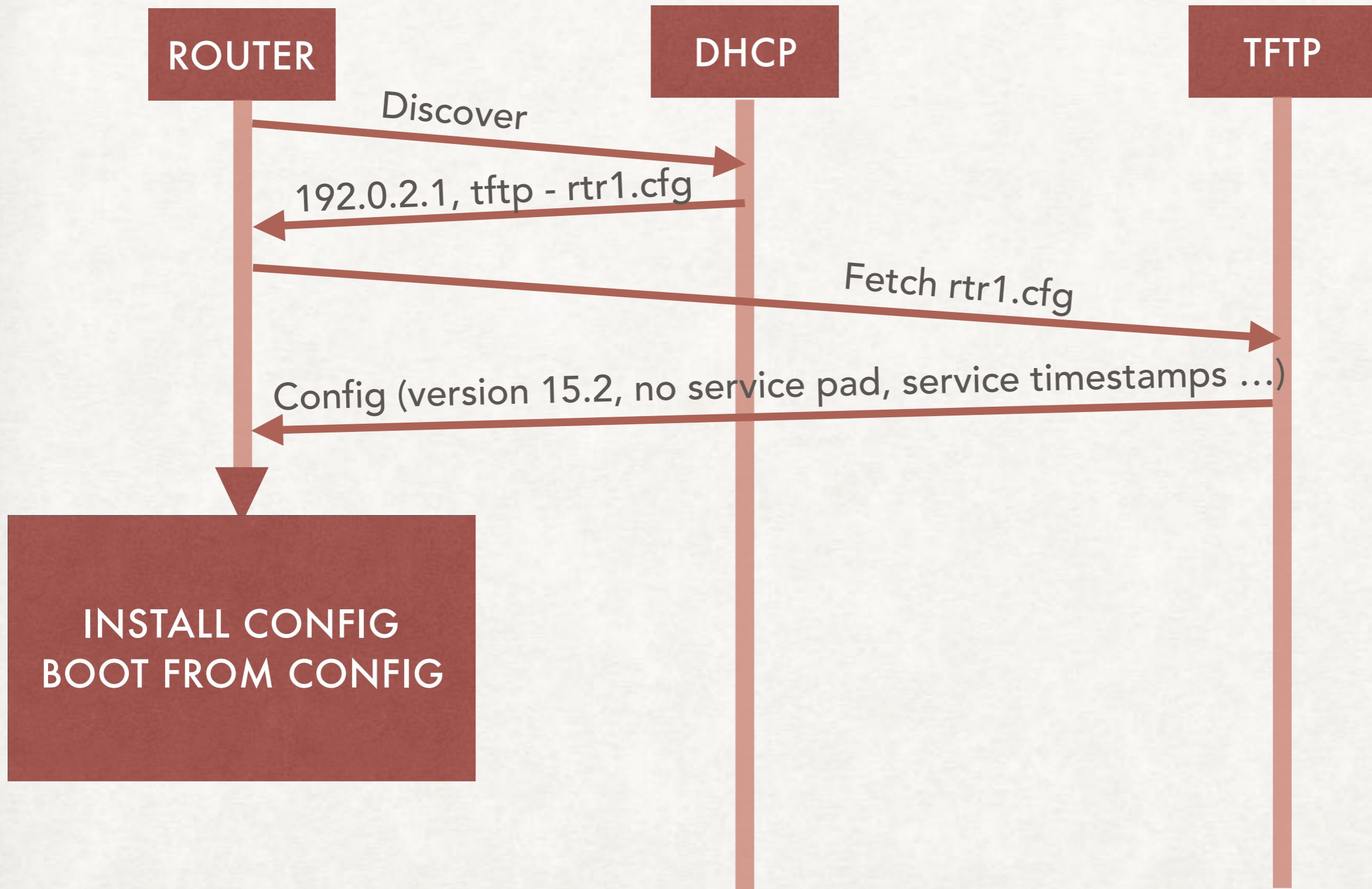
Use-case

SECURE DEVICE INSTALL

- Ship a device to an Internet Exchange
 - Already have a circuit
 - Have device able to be plugged in and Just Work
- Great! Use e.g autoinstall, problem solved!
 - Nope - Want **Secure Device Install**

Refresher on autoinstall

AUTOINSTALL



AUTOINSTALL



DHCP

TFTP

Discover

192.0.2.1, tftp - rtr1.cfg

Fetch rtr1.cfg

Config (version 15.2, no service pad, service timestamps ...)

SNMP COMMUNITY
TACACS KEY
FIREWALL CONFIG
USER ACCOUNTS

Attacker happy

Me sad

SECURE DEVICE INSTALL

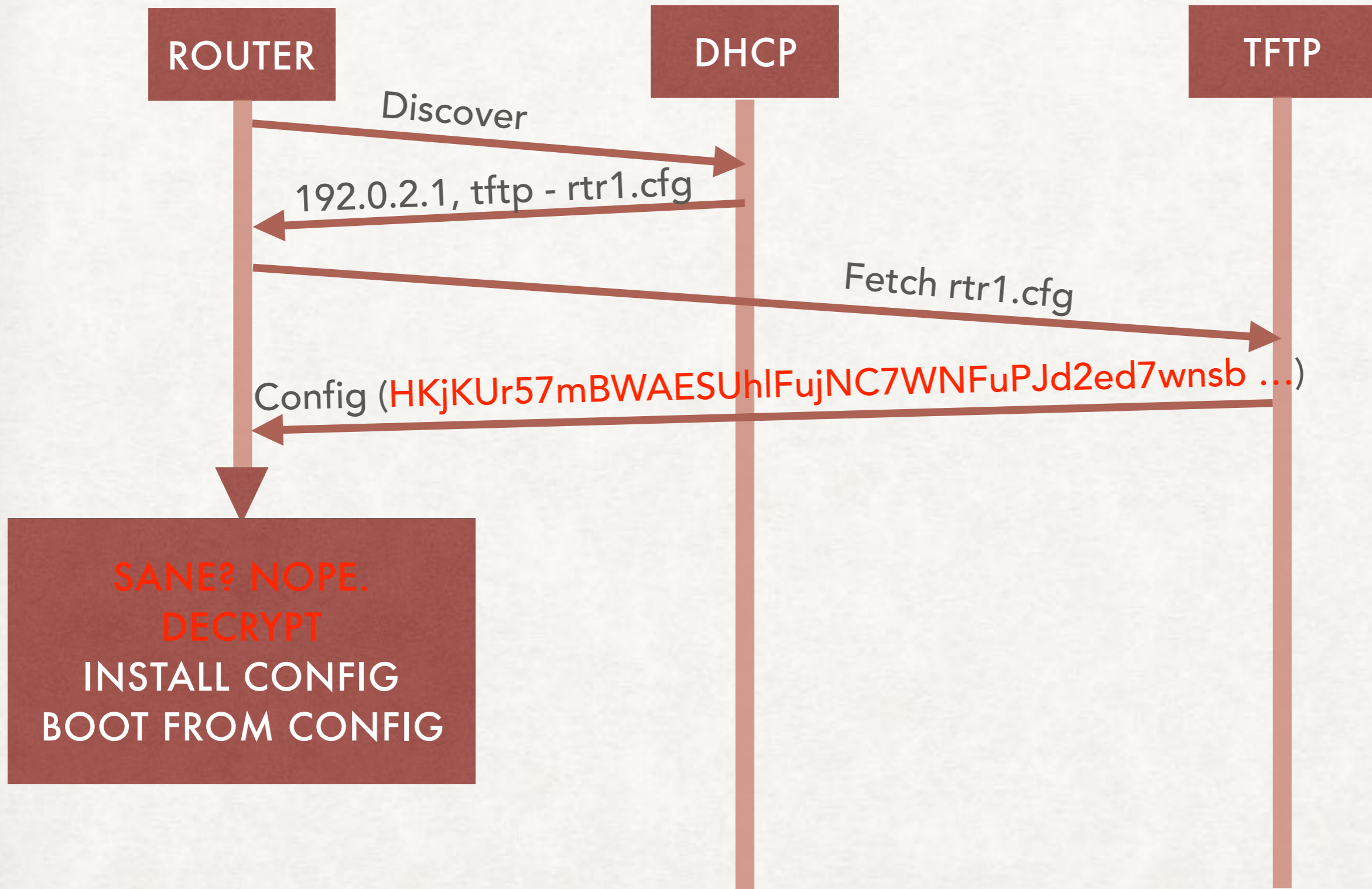
1. Vendors build device, make a keypair
2. They publish public key on <http://cert.vendor.example>

1. I order a device
2. Vendor says "Thanks, shipping you serial #4217"
3. I fetch <http://cert.vendor.example/sn-4217.crt>
4. Encrypt config file to key in sn-4217.crt

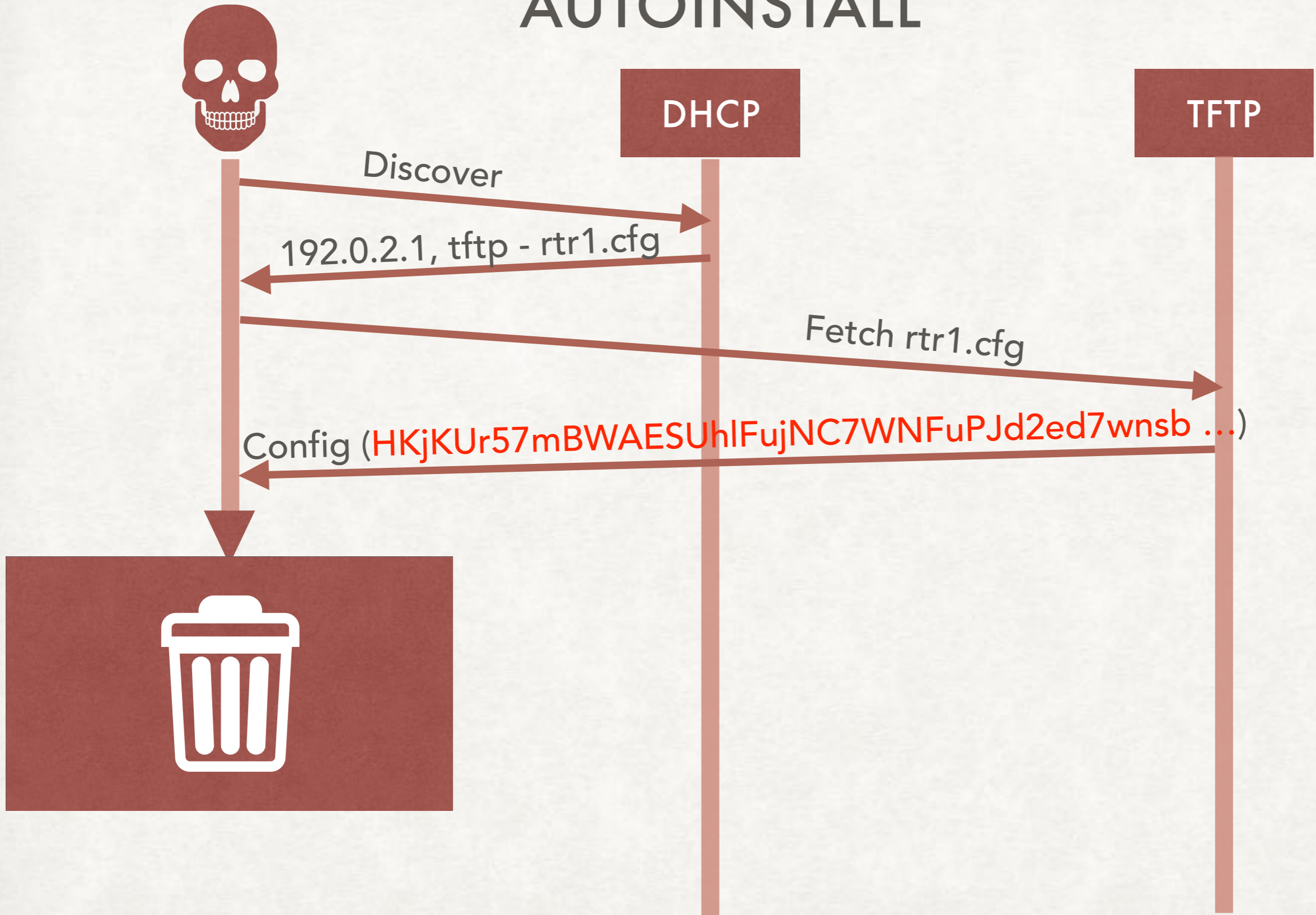
When device boots, it fetches rtr1.cfg:

1. If rtr1.cfg looks sane, install and boot
2. If not, decrypt with private key, install and boot
3. Done.

SECURE AUTOINSTALL



AUTOINSTALL



Me happy

Attacker Sad

FAQ

- Why not `draft-ietf-netconf-zero-touch?!`
 - that's better, but more complex
- Why not ANIMA ACP and BRSKI?!
 - that's much more complex
- Why not `$Vendor $Cool_Product?`
 - Simplicity, standardized
- Why not use the 802.1AR IDevID certificate?
 - If you have it and can use it, go right ahead
- Can I delete / replace the cert?
 - Probably? Your choice
- I deploy a staging config, then SSH in and upgrade...
 - Yay! Keep doing that, use this for the staging config.

SIMPLE? REALLY!?

Step 1: Fetch the certificate.

```
$ wget http://keyserv.example/certificates/SN19842256.crt
```

Step 2: Encrypt the config file.

```
$ openssl smime -encrypt -aes-256-cbc -in
SN19842256.cfg\
  -out SN19842256.enc -outform PEM SN19842256.crt
-----BEGIN PKCS7-----
MIICigYJKoZIhvcNAQcDoIICezCCAncCAxggE+MIIBOgIBADAI MBUxEzARB
gNVBAMMC1NOMTk4NDIyNTYCCQDJVuBOb1DANBgkqhkiG9w0BAQEFAASCAQB
ABvM3...
LZoq08jq1WhZZWhTKs4XPGHUdmnZRYIP8KXyEtHt
-----END PKCS7-----
```

Step 3: Profit!

QUESTIONS?