SHSP, or,
'what else could DNCP do?'

Markus Stenberg <markus.stenberg@iki.fi>

Motivation: Security is hard (5 minutes of Googling)

"Password Is "PASSWORD" - If You're Using One Of These 9 Baby Monitors, A Hacker Might Be Using It, Too" (Sep 2015)
https://www.fatherly.com/wi-fi-enabled-baby-monitors-vulnerable-to-hacks-1328437436.html

"Philips Light Bulb Vulnerability Could Leave Some In the Dark" (Aug 2013)
https://threatpost.com/philips-light-bulb-vulnerability-could-leave-some-in-the-dark/102007/

The uControl Smart Home Automation (aka de.ucontrol) application 1.2 for Android does not verify X.509 certificates from SSL servers, which allows man-in-the-middle attackers to spoof servers and obtain sensitive information via a crafted certificate. (Oct 2014)
https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-4892

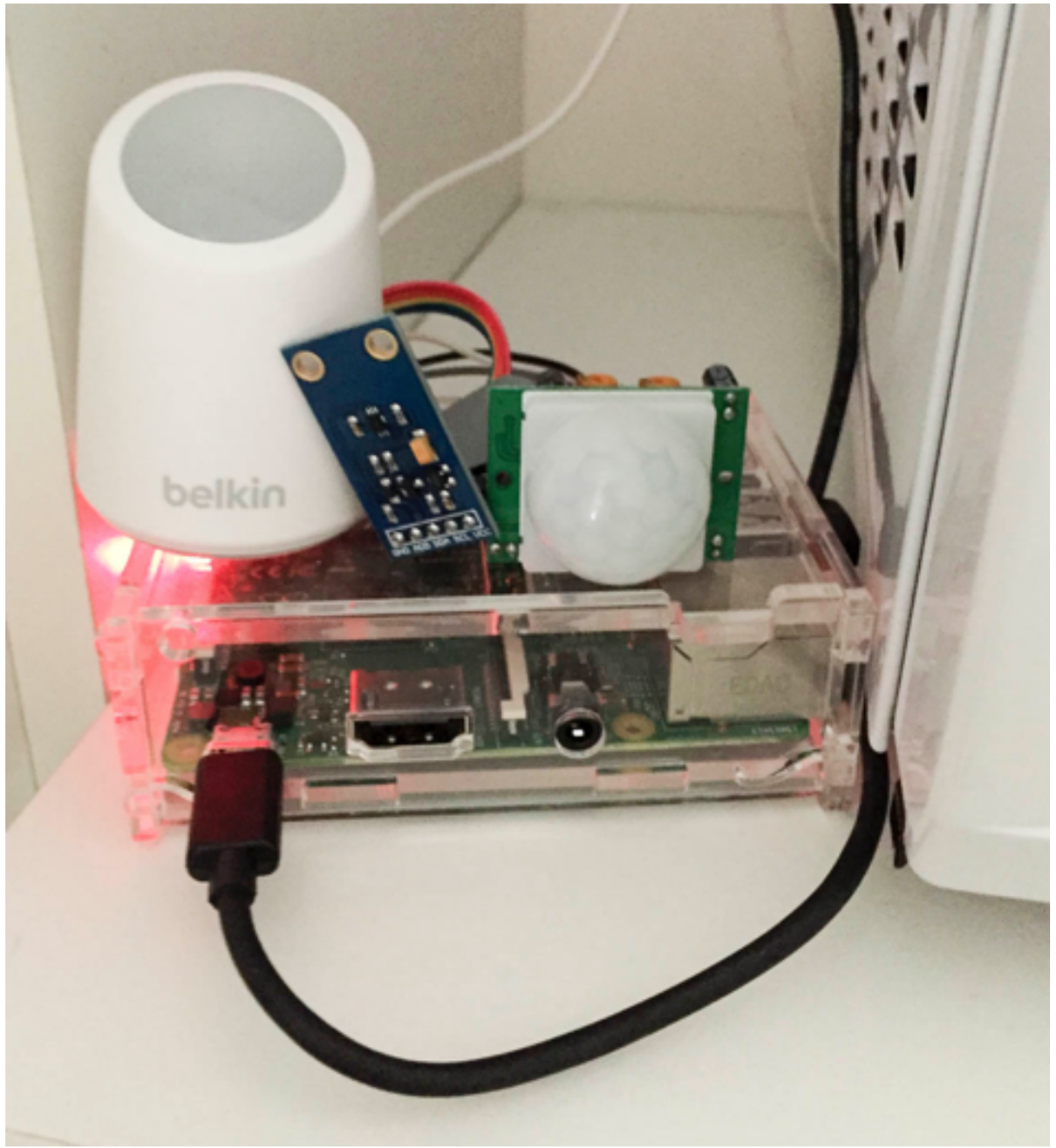
The Belkin WeMo Home Automation firmware before 3949 has a hardcoded GPG key, which makes it easier for remote attackers to spoof firmware updates and execute arbitrary code via crafted signed data. (2014)
https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-6952

Security 101

- If the risks outweigh the benefits, skip whatever you are planning to do
- (Directly) Internet connected devices that deal with e.g. mains power are not necessarily a good idea, given the implementations out there
- Anything with default password and Internet connectivity is a risk
- Limiting them in the network is also hard due to closed source and bad implementations - case Belkin WeMo:
  - NTP/call home attempt every few seconds until the end of time.
  - Many gigabytes of wifi traffic over few months hitting local firewall.
  - Also random stalls of software, possibly due to expecting call home to work or other software bugs.

Rather non-novel idea

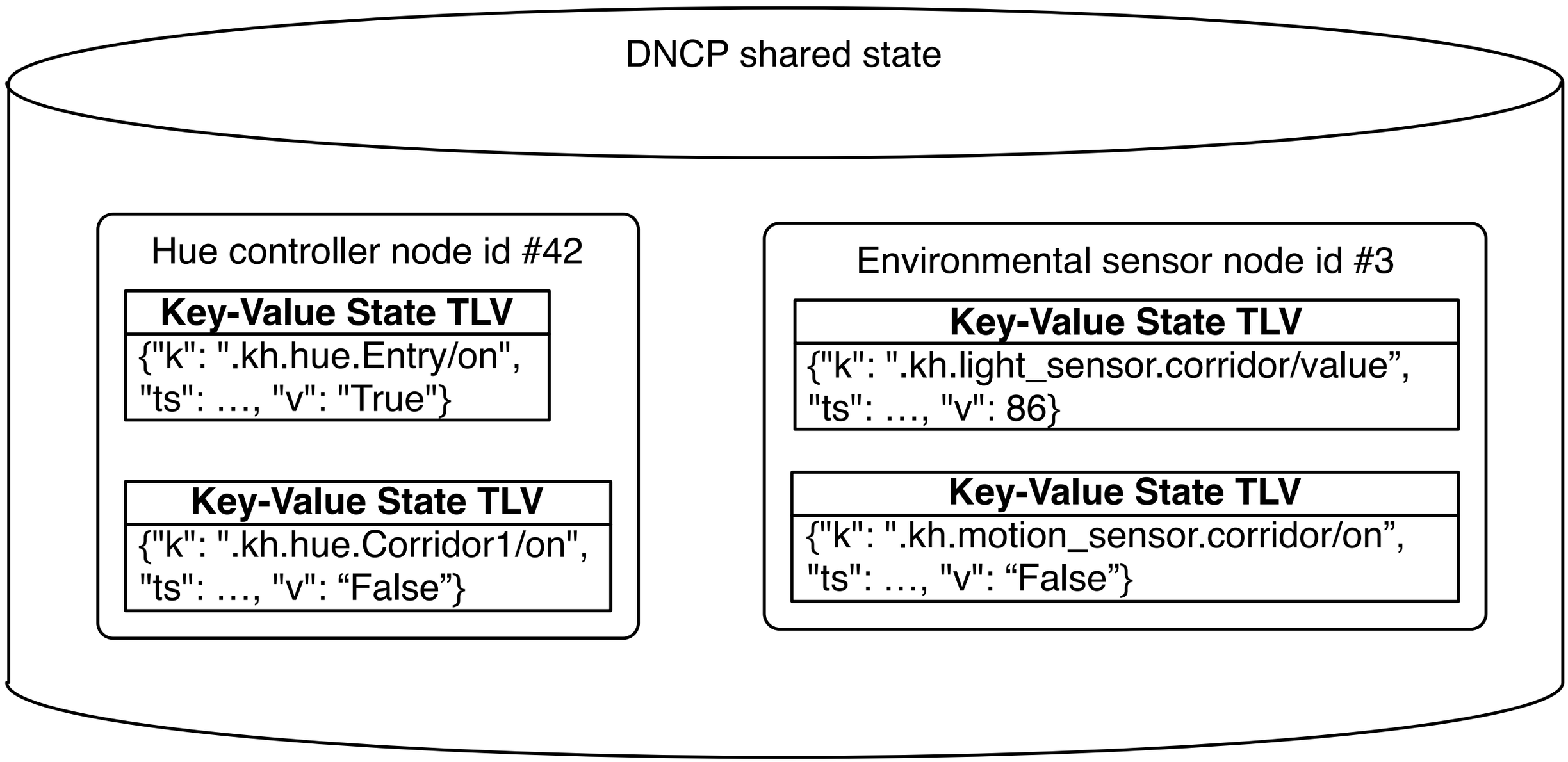
- No Internet access for most of home automation
- Let (somewhat more secured and maintained) devices deal with the Internet
- Implies need to share state with minimal configuration on sensors/devices
  - Hello, DNCP!



SHSP in nutshell

- DNCP "network" - no need to configure addresses, hop-by-hop, IPv6 linklocal-only = win
- HNCP transport (implementation), or some better option for the protocol (TCP in the draft)
- Each node publishes a set of TLVs that contain
    - key name (string); (node id, key name) unique within the network
    - value (arbitrary JSON string)
    - last changed timestamp (UTC seconds)
- Outcome:
    - No need to configure addresses, all devices have equal access to the data
- For 'write' access, assume key names are themselves are actually network-wide unique, and just provide updated values (last changed higher) for the keys owned by other nodes

Example SHSP state



Interesting DNCP-ish problem: Local operations on DNCP state

- Given the SHSP (main) process(es) are long-lived, an interface is needed to probe the state, and to share the port on the node

- In hnetd (implementation), we implemented IPC mechanism to get the DNCP state; only one long-lived process

- Laziness reigned in my home networking effort:

  - Used the SHSP transport itself on 'client' sockets bound to ::1 (and some other port) with unicast-only transport with the 'server' socket that actually communicates with the outside world

  - Pro: Little code needed.

  - Con: Duplicates whole DNCP state in N processes.

Summary

- Amusing research-y hacking project which is partially open source:
    - Python SHSP/HNCP implementation (https://github.com/fingon/pysyma) and
    - Home automation logic (https://github.com/fingon/kodin-henki) are,
        - but the database they use (prdb) is not currently.
- It already works better than the stuff it replaces (+- some complaints about ugly hardware constructed from Raspberry Pis by other people)
- Unfortunately the industry trend seems to be the reverse direction
    - Everyone providing their own Internet service for devices to connect with proprietary APIs, and either bad behavior or unusable devices if they cannot connect it

Questions?