Secure Usable Intranet Browser/Browsing

IoT Security Foundation
ManySecured.net
SUIB WG
Michael Richardson (Sandelman)
Nick Allot (NQuiring Minds)
Jan Geertsma (Signify)
and help from Christian Amsüss

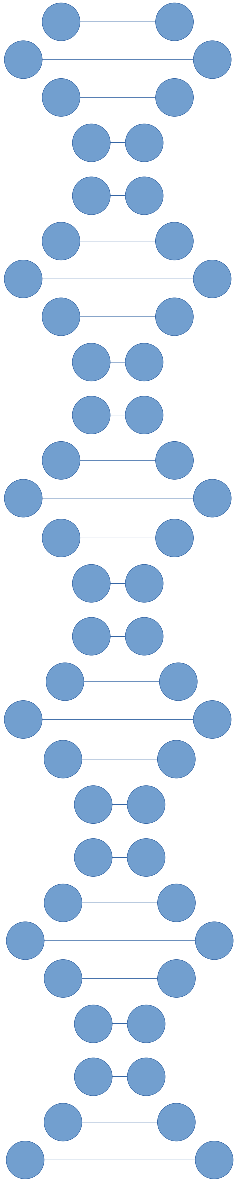
What we talked about last time

THE PROBLEM

- Browsers cannot connect securely (and useably) to local “intranet” resources
- https://datatracker.ietf.org/meeting/112/materials/slides-112-iotops-suib-browsing-local-web-resources-in-a-secure-usable-manner-iot-device-configuration-as-a-special-case-00
- https://specs.manysecured.net/suib/
- https://jabber.ietf.org/jabber/logs/iotops/2021-11-12.html many half-solutions proposed during the IETF112 talk.

TODAY

- some solutions that we think are detailed enough to throw darts at them.
- would like to have a full virtual interim on this topic
- IoTSF ManySecured meets every two weeks Wednesdays at 1500UTC.
- Note some connection to 6man work on IPv6-LL in URL work, https://datatracker.ietf.org/doc/draft-carpenter-6man-rfc6874bis/ If you could talk to it via IPv6-LL, you’d want to talk to it securely too, right?
  - PRINTERS



Try to Hack CA/Browser Forum

- “Change Browser Behaviour” https://specs.manysecured.net/suib/Solutions/browser-solution
- Try to make the warning look less scary when the connection is to a directly connected RFC1918, a directly connected IPv6 ULA, or an IPv6-Link-Local.
- Save Certificate to browser, it’s Trust on First Use.

Try to Hack DNS

- “DNS Based Device Name -
  Embedded Physical Address”
  https://specs.manysecured.net/suib/Solutions/dnsname-embedded-solution
- Device has factory provisioned certificate, with a name like:
  *.f0ocrypto1d.devices.vendor.example
- Device has http server that redirects to, for instance:
  https://c0a80001.f0ocrypto1d.devices.vendor.example
- query for c0a80001.f0ocrypto1d.devices.vendor.example returns 192.168.0.1
- Cleaver people will notice c0a80001 => 192.168.0.1. A records and AAAA are synthesized by devices.vendor.example server.
- This solution is actually already out there.

Try to Hack in an Onboarding Step

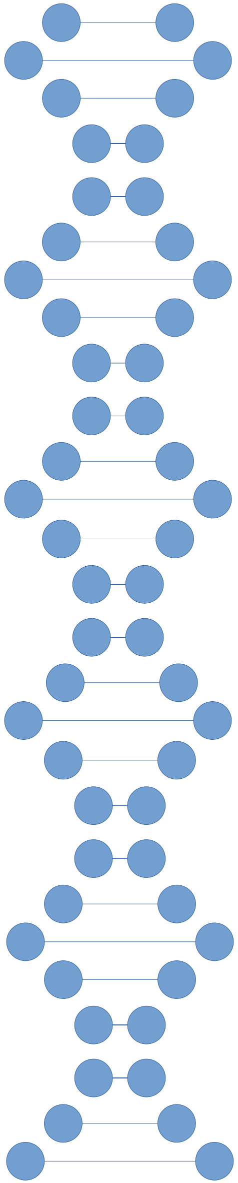
- “DNS Based Device Name - Dynamic DNS Resolution” https://specs.manysecured.net/suib/Solutions/dnsname-dynamic-solution
- Device has factory provisioned certificate, with a name like: f0ocrypto1d.devices.vendor.example
- When device boots, it calls home using provisioned certificate, and does Dynamic DNS update with correct A/AAAA record.
  - can use RFC3007, or wide variety of proprietary HTTP APIs: Dyn, no-ip, Cloudflare, whatever. It’s a vendor/device decision, not requiring standardization.
- Requires more state than previous solution, but a certificate needs to be provisioned at some point.
- Unclear if anyone does this for HTTPS, but certainly it’s popular for HTTP.

Try to Hack Local “Smart” mechanism

- “Application or Skill Issued Certificate”

  https://specs.manysecured.net/suib/Solutions/application-skill-solution

- A generic (not manufacturer) specific App or “Skill” interacts with device to collect a CSR, and then enrolls the device into a private PKI.
  - BRSKI RFC8995 does essentially this
  - CHIP/MATTER intends to do this
  - DPP could to this
- The private PKI is available “locally” only, on the smartphone, desktop or Intelligent Speaker that was in charge. It might propogate trust anchors to other parts of the home, and into `/etc/ssl/certs` or equivalent.
-



Try to Hack In IETF standard mechanisms

- “Gateway Issued Certificate”

  https://specs.manysecured.net/suib/Solutions/gateway-solution

- The home gateway runs the PKI and/or manages access to one, and does the enrollment. Same as before, but now with ISP delegated names.
  - BRSKI RFC8995 does essentially this, with brski-cloud, and draft-ietf-acme-integrations could provide ACME certificates

- Requires devices to do proper Onboarding

- Conclusions and Discussion

https://specs.manysecured.net/suib/

- “imagine you are in a field....”