IETF 112 DANCE

Architecture Document

Agenda

Problems

How DANE maps to the problem space

Use cases

Q&A

DISCLAIMER

This presentation and referenced document contains work beyond charter scope.

This is meant to provide context for alignment w/ future work

...but is not intended to obligate the WG to address all possible use cases

Problems

PKI-based client identity is:

- **Too restrictive, not interoperable:**
  - Every private PKI manages its own namespace
  - Multiple private PKI = risk of identifier collisions
  - Difficult to use across organizational boundaries
- **Operationally burdensome**:
  - Build or buy. Organization must manage its own private PKI
  - Cost-prohibitive for resource-constrained organizations

Problems: Too Restrictive

Every private PKI enforces its own namespace

No means for one private PKI to prevent another from creating an identifier collision

Private PKIs are typically scoped to organization/application

Establishing trust across organizational boundaries is complicated

No standardized method for client PKI discovery/lookup:

- Entity certs or public keys for object security
- Trust anchors for TLS client certificates

Problems: Operationally Burdensome

Cost-prohibitive for resource-constrained organizations

- Build or buy. Organization must manage its own private PKI
- This requires a skillset that understands how PKI works

Compare to DANE server identities:

- Put a public key in DNS, install the key pair in the TLS server.

Interlude: Best Practices

CSA, Identity and Access Management for the IoT Summary Guidance[1]:

Step 1a: Define a common namespace for IoT devices

What if we didn't create a new namespace…

And instead use the one we already have for servers: **DNS**

[1] https://downloads.cloudsecurityalliance.org/assets/research/internet-of-things/identity-and-access-management-for-the-iot.pdf

Life with DANE for Client Identity

PKI namespace is bound to DNS, recognized wherever DNS is used.
- Identifier namespace is no longer application- or organization-local

Organizations can obtain devices with pre-provisioned identities
- Leased device ID managed by leasing organization
- Hardware secure elements ship with universally-recognized IDs
- Organizations can **use** PKI-based identities **without managing** a PKI

Network access: Add the client's DNS name to an access list
Application access: Add the client's DNS name to the application client list
Object security: Use DNS for public key lookup, for signature verification

Example: Autonomous Cars

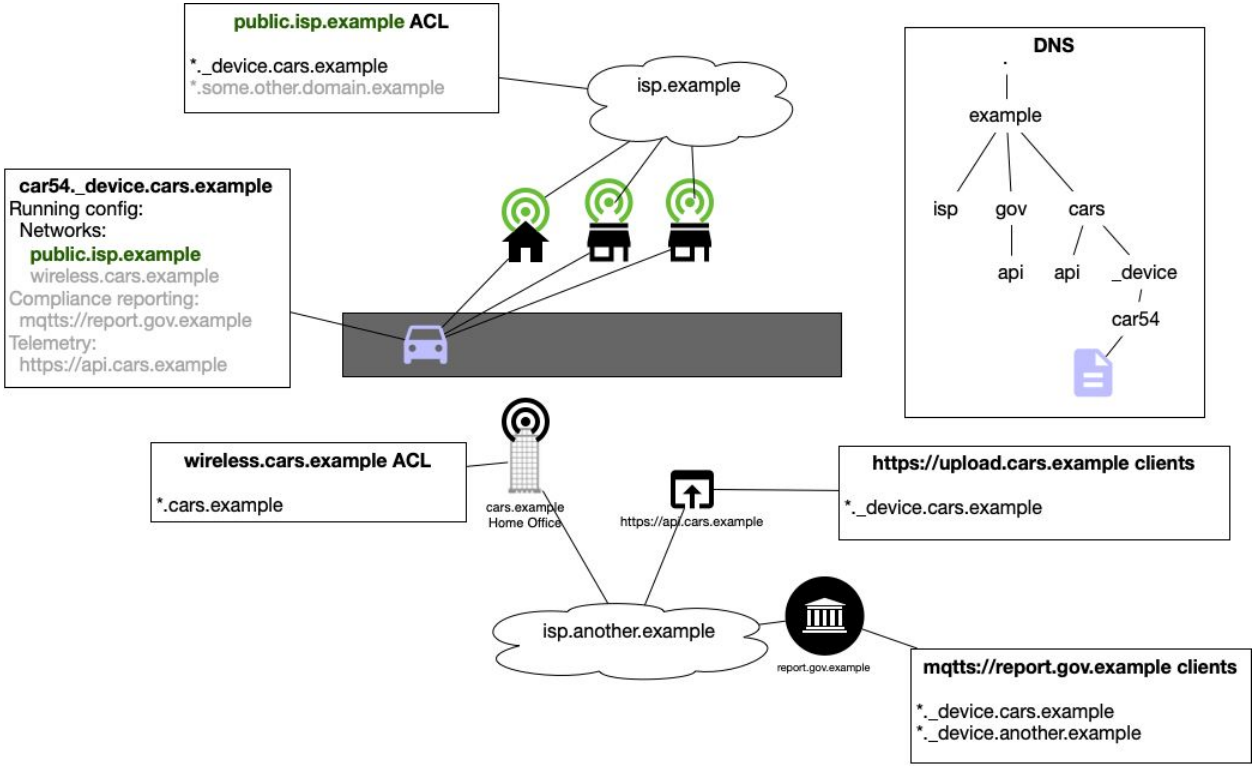
Premise:

An autonomous car company (**cars.example**) is allowed to operate on specific routes, and must regularly report paths taken to **gov.example** compliance dept. In signed JSON (JWS) format.

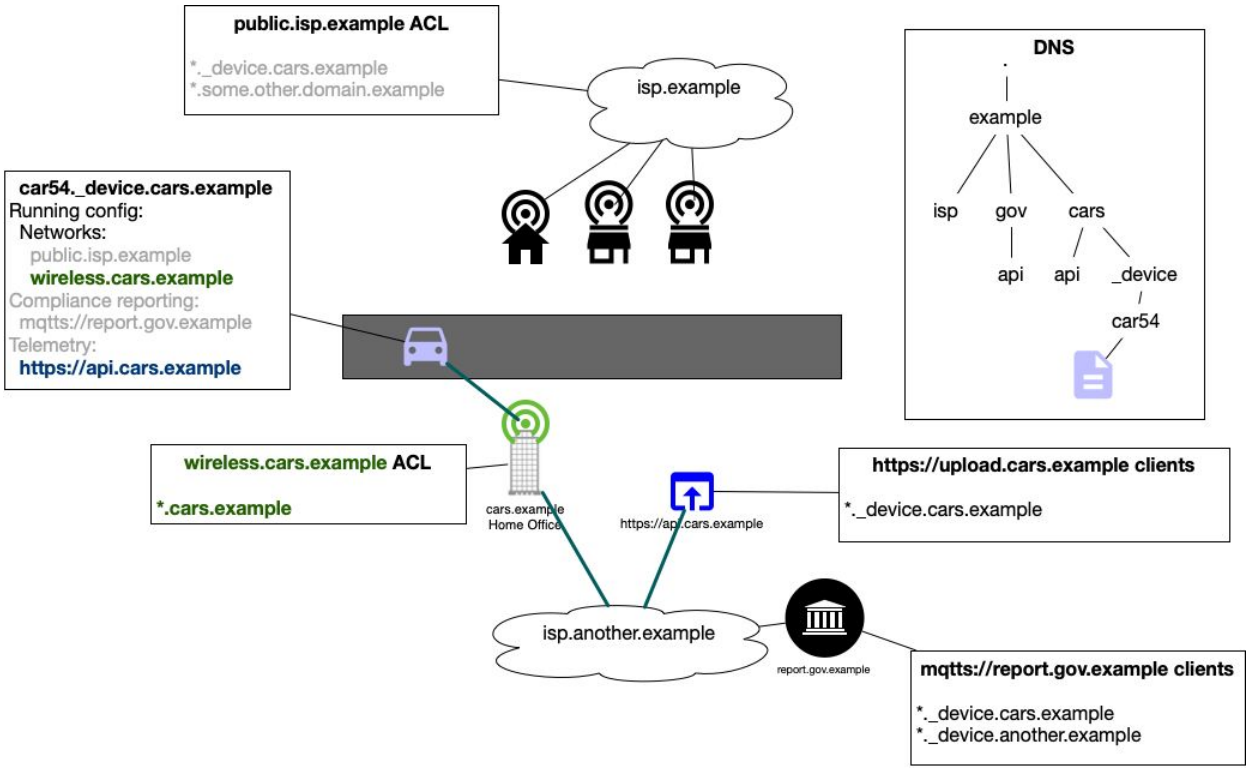
The autonomous cars generate LOTS of data while driving, and need to frequently upload sensor telemetry, videos, etc to cars.example data processing system.

An agreement exists between **cars.example** and **isp.example** to allow autonomous cars internet access via any isp.example access points.

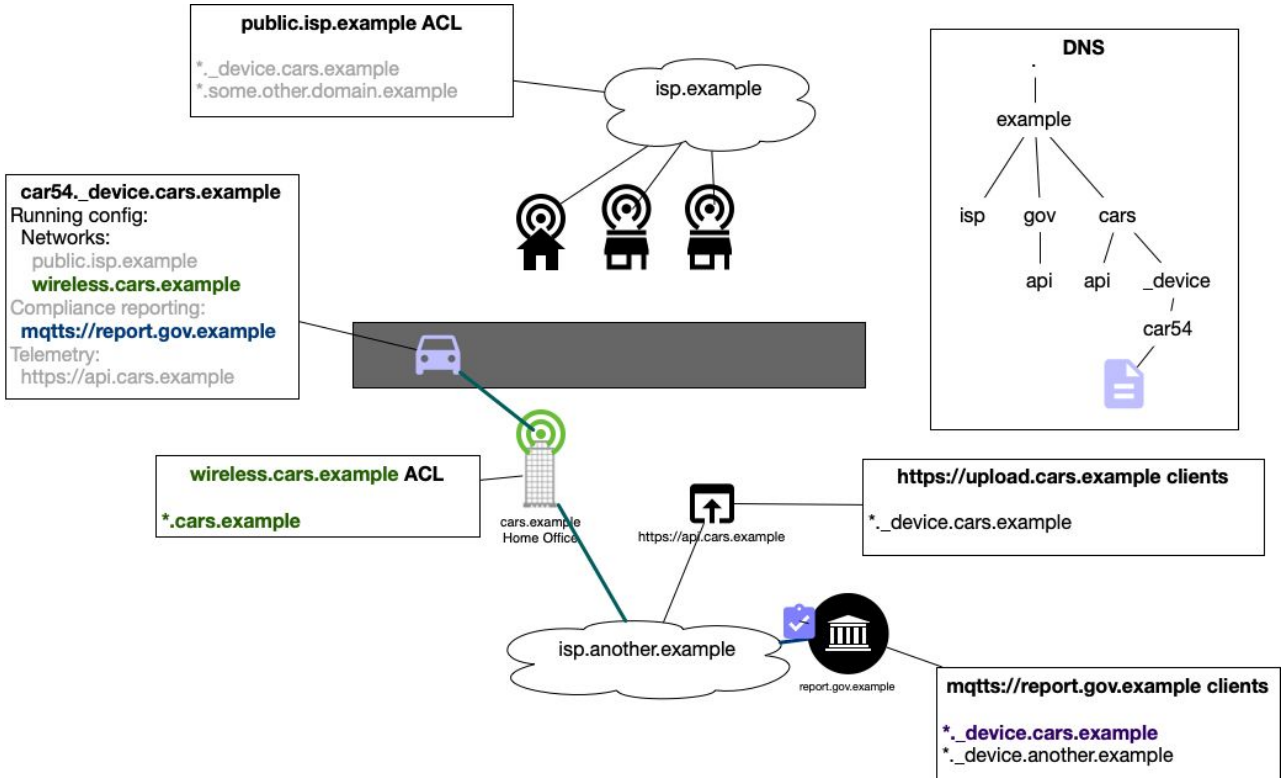
Universal client credential: EAP-TLS



Universal client credential: EAP-TLS + mTLS



Universal credential: EAP-TLS + mTLS + JWS



Summary

Car uses identity represented in DNS (car54._device.cars.example) to:

- Perform EAP-TLS client authentication with:
  - public.isp.example
  - wireless.cars.example
- Perform TLS client authentication with:
  - https://api.cars.example
  - mqtts://report.gov.example
- Sign messages for verification by:
  - gov.example

Possible protocol use cases

Mutual TLS/DTLS:
- STARTTLS
- EAP-TLS
- HTTPS
- MQTTS
- SIP/WebRTC
- LoRaWAN

Object security:
- JOSE/COSE
- XMPP E2E

Other:
- SSH

Anti-abuse protocols, security considerations

Anti-abuse:

- MUD Reporting
- XARF
- STIX/TAXII

Security:

- Confidentiality: Recommend DoT
- Integrity: Use DNSSEC validation in the stub resolver
- Availability: Only perform a DNS lookup for permitted client names (slow loris)

Links

https://datatracker.ietf.org/wg/dance/about/

https://datatracker.ietf.org/doc/draft-wilson-dance-architecture/

https://github.com/ashdwilson/draft-dance-architecture

Q&A