Encrypted SNI

Metadata is powerful

- Censorship
- Surveillance
- "We kill people based on metadata." -- US General Michael Hayden

TLS Handshake is metadata

- And in the clear
- visible to passive and active attackers

Pervasive monitoring is an attack

TLS WG Charter explicitly states:

```
o Develop a mode that encrypts as much of the handshake as
  is possible to reduce the amount of observable data to
  both passive and active attackers.
```

TLS should protect this sensitive data.

Server Name Indication (SNI)

Extension in ClientHello that transmits the name of the desired host.

Hostnames are interesting:

- `alcoholicsanonymous.org`
- `cia.gov`
- `torproject.org`
- `falungong.org`
- `glaad.org`

Useful on shared IPs

- Major subdomain operations:
  - *example*.github.io
  - *example*.wordpress.com
  - ...
- Content distribution networks
- Shared hosting arrangements

Encrypted SNI necessary but insufficient

Hostnames still leak in DNS.

We need to fix what we can fix.

Work under way in `dns-privacy`.

Need pre-handshake key for 1RTT

Options:

- DNS
    - `draft-nygren-service-bindings`
    - DANISH
- previous history
- in-band, extra RTT?

Threat models

Defeats passive attacker

Active attacker connection fails (but SNI leaks)

With DNSSEC, defeats active attacker

Pre–handshake key ID

Services will eventually need to rotate their keys.

Enable this by providing Key Identifiers.

These should not be SNI–equivalent, but we can't prevent it.

How does it work?

If server has opted in, client's initial message is bare minimum ClientHello, but includes an extension with:

- Prehandshake Key Identifier (32 bits?)
- Client share
- Ciphered blob containing real ClientHello

What cipher?

- Defined by Prehandshake Key Identifier

Forward secrecy for handshake

We're moving toward all-forward-secret key exchange in TLS 1.3

Cacheable, redistributable pre-handshake keys will not have forward secrecy.

Data encrypted by these keys will lack forward secrecy

- without encrypted SNI, the data will lack any kind of secrecy
- this does not compromise the forawrd secrecy of the rest of the connection.

Failure modes

Client sends unknown key ID or undecryptable content

Server has two choices:

- respond to bare ClientHello as though non-SNI-capable client (e.g. TLS 1.2)
  - Client can continue connection or abort and retry
- respond with "use this other pre-handshake key/keyID"
  - Client sends initial flight again, using new pre-handshake key

Denver Interim

- much discussion
- no consensus for MTI

Pre-handshake key types

Costs depend on pre-handshake key types that we allow.

key types need to specify:

- public key
- encryption mechanism

Proposal:

- only two: NIST vs. non-NIST
    - ECDH(p256) + AES128-GCM
    - ECDH(curve25519) + AES128-GCM

Only protects the first flight, which is otherwise unprotected.

Questions?