INCOMPATIBLE PROTOCOL DOWNGRADE ATTACK PROTECTIONS

When ALPN isn't enough

draft-thomson-tls-snip, Martin Thomson, TLS, IETF 108

Incompatible Protocols

ALPN enables negotiation of **compatible** protocols

A connection attempt can offer both HTTP/1.1 and HTTP/2
…the result is a connection with either

The same connection attempt can’t result in HTTP/3

HTTP/3 is **incompatible** with HTTP/1.1 and HTTP/2

Incompatible Protocols

TLS and DTLS are incompatible

QUIC is also incompatible with either
…but potentially has multiple versions
…QUIC versions can be incompatible with each other

Incompatible protocols mean clients choose

ALPN results in selecting the **server's preference**
…from a set of **compatible** options offered by the client

Servers cannot decide between incompatible protocols

For incompatible protocols, clients choose what to attempt

Clients might choose not to act

- e.g., Clients likely have to tolerate HTTP/3 being unavailable

Don't Negotiate, Authenticate

Choice between incompatible protocols can't be negotiated

Authenticate instead

As the client decides, **provide client with all available options**
…the client can then attempt their most preferred

In other words, if the client chooses, the client also has to act

BAD IDEA: DNSSEC

SVCB lists what protocols are supported

DNSSEC can authenticate SVCB

Problem solved, right?

DNSSEC Problems

DNSSEC uses a different authority to the connection
…weird incongruities with ALPN result

Clients don't get all DNS records
…choosing which records to serve is a critical resolver feature

Deployments aren't uniform
…during deployment of new versions
…due to being served by multiple providers

Deployments aren’t uniform

Server doesn’t mean single computer
…HTTP is deployed on multiple computers
…and operated by multiple providers

Different providers need to support different protocol versions

Parts of deployments too
…during upgrades
…as a result of deployment constraints

PROPOSAL: PROTOCOL AUTHENTICATION SCOPES

Divide deployments into compartments
… called **protocol authentication scopes** in the draft

Allow an attacker to pick which compartment is used
…more importantly allow DNS resolvers to choose

Homogenous deployments will get full downgrade protection

Heterogeneous deployments are exposed to downgrade
…to the scope with the least-preferred set of protocols

Proposal:
Protocol Authentication Scopes

Only protect against downgrade in a narrow scope

By default scope is a single endpoint
…in effect, no downgrade protection

For SVCB, define a scope based on the ServiceForm name

Allows for full downgrade protection
… if you only use SVCB and have uniform protocol support

Proposal: Protocol Authentication Scopes

A client receives information about scopes from two places

- Any discovery/lookup process (DNS, SVCB, DDDS)
- The TLS handshake

Only the TLS handshake is authoritative

Proposal:
TLS Extension

List all incompatible protocols that are available
…in the scopes that the server supports(*)

Tied to the authority of the connection

ISSUE:
AGREEING ON APPLICABLE SCOPES

The -00 draft assumes that the server can just pick a scope

This leaves the client open to downgrade attack

- Attacker suppresses SVCB records (which have wider scope)
- Client only gets A/AAAA records (with zero scope)
- Client cannot act if the server advertises a more-preferred protocol

Proposal:

- Server lists all scopes that apply
- Client can use any or all scopes that it understands

ISSUE:
ALPN IDENTIFIERS

This proposes using ALPN identifiers

Future ALPN identifiers will need to be limited to one protocol
…just TLS, DTLS, or (a set of compatible) QUIC versions

Most already do this, with one exception
…STUN/TURN (RFC 7443)

- STUN/TURN uses the same identifier for both TLS and DTLS
- So they are both compatible AND incompatible at the same time

QUESTIONS

Was this sufficiently clear?

Is the draft sufficiently clear?

Problem worth solving?

Solution generally reasonable?

Why no code, syntax, nor diagrams in this presentation?