The SVCB and HTTPS RRs

Service binding and parameter specification via the DNS

Presenter: Ben Schwartz <bemasc@google.com>
Coauthors:
Erik Nygren <erik+ietf@nygren.org>
Mike Bishop <mbishop@evequefou.be>

DNS-OARC February 2021
https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-svcb-https-02
rev. 1

Background: CNAME (RFC 1034, November 1987)

“A CNAME RR identifies its owner name as an alias, and specifies the corresponding canonical name in the RDATA section of the RR. If a CNAME RR is present at a node, **no other data should be present**; this ensures that the data for a canonical name and its aliases cannot be different.”

- But at a zone apex (e.g. example.com) **there must be an NS record**.
- Therefore **there cannot be a CNAME at the apex**.

Background: SRV (RFC 2782, February 2000)

```
_Service._Proto.Name TTL IN SRV Priority Weight Port Target
```

- Input: Service (e.g. “ldap”, “sip”), Protocol (“tcp” or “udp”)
- Output: Target name (alias), Port number, load balancing info
- **Allows indirection of the apex**
- Widely used for SIP, LDAP, DANE, and more
- **Not widely implemented by HTTP clients**
  - Much to the disappointment of some
- No change required to recursive resolvers, but
  - there is a latency penalty to chase the target if the recursive doesn’t help
  - POSIX getaddrinfo() and poorly designed middleboxes only support A/AAAA queries.

2000-2015: Rise of CDNs

- CDNs make HTTP scaling easy: just CNAME www.example.com to the CDN
- But what about users who type “example.com” into the URL bar?
  - Hardcode CDN IPs into the customer zone?
    - With some kind of anycast?
  - Let the CDN manage your zone?
    - Changes the business model, trust relationship, etc.
  - Make the authoritative resolve the alias itself
    - With an ALIAS record?
      - Not standardized, limited scope
    - With the (proposed) ANAME record?
      - Requires major changes to authoritative implementations
- Using a CDN for the apex domain is possible, but it’s **inconvenient**

2010-2020: Multi-CDN

- What if I want to host on *two* CDNs?
  - e.g. for reliability, **during a transition**, for business reasons, etc.
- For www.example.com, easy: just use two CNAMEs
  - Arguably a standards violation but as long as they're in separate RRSets it's fine.
- For example.com?
  - Mix the (hardcoded) IPs!
  - Works fine as long as the CDNs are fully interchangeable.

2018: Encrypted SNI (now “Encrypted ClientHello”)

- TLS server publishes a public key in the DNS
  - Initially prototyped using TXT records
- Client fetches the public key and uses it to encrypt the first message
- Multi-CDN Problem: **CDNs are no longer interchangeable**
  - Using CDN A’s public key with CDN B’s IP address is a **fatal error**
  - DNS doesn’t provide any way to bind together multiple RR types!
- Solution: Need a new record type that
  - **binds together** the service’s IP addresses, Encrypted ClientHello public key, and any other metadata needed for a successful connection, AND
  - allows **aliasing the apex** to a bound collection of information, if it exists:
- a **Service Binding Record**

SVCB Overview

```
_Port._Scheme.Name TTL IN SVCB SvcPriority TargetName [SvcParams...]
```

- Goal: bootstrap optimal connections from a single DNS query
- SvcPriority == 0: “AliasMode”
  - Enables apex aliasing (only for participating clients)
- SvcPriority != 0: “ServiceMode”
  - SvcParams: Arbitrary key-value data store
    - TLS ALPN hints
    - Port number
    - Encrypted ClientHello configuration
    - IP hints
- Like SRV, recursive resolver support makes lookups faster but is not required

HTTPS RR Overview

- “HTTPS” is a SVCB-compatible RR type specialized for HTTPS
  - **Avoids underscore prefixes**
    - Improves compatibility with wildcard domains
      - `*.example.com <TTL> IN HTTPS ...`
    - Compatible with existing CNAME delegations
  - Indicates that the origin defaults to HTTPS (similar to “HSTS”, prevents “SSL stripping”)
- DNS servers treat SVCB and HTTPS RRs identically

AliasMode (SvcPriority=0)

- Covers many “SRV” and “ANAME” use-cases

ServiceMode (SvcPriority>0)

- Covers Encrypted ClientHello use case and other protocol improvements

Lower SvcPriority means preferred

```
svc.example.net.  7200  IN HTTPS 2 svc3.example.net. alpn=h3 port=8003 \
                                                     echconfig=...
```

SvcParams encode protocol, port, ECH keys, and other params

```
svc.example.net.  7200  IN HTTPS 3 svc2.example.net. alpn=h2 port=8002 \
                                                     echconfig=...
```

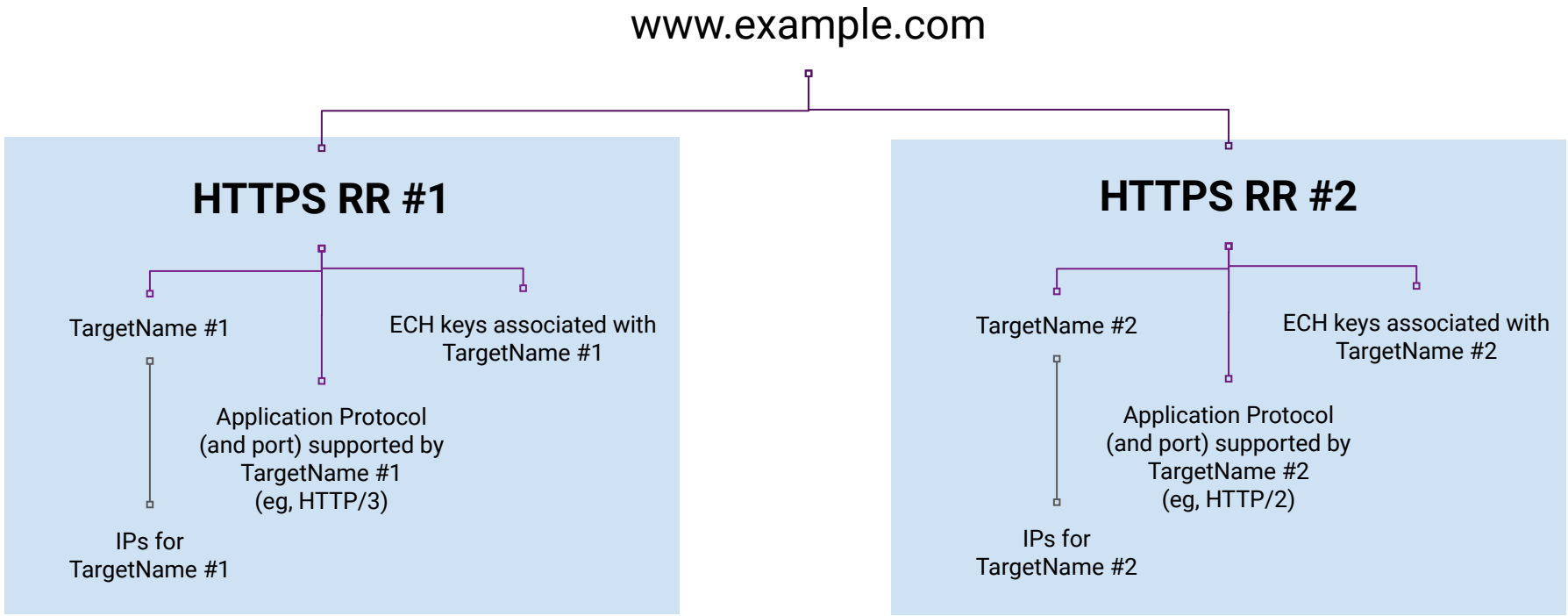
“Please use QUIC to UDP svc3.example.net:8003 with this ECH configuration, or use HTTP/2 to TCP svc2.example.net:8002 with this other ECH configuration.” (Also, HTTP/1.1 is always supported unless explicitly disclaimed.)

Query flow

- (Dual-stack) client issues, A, AAAA, and HTTPS queries simultaneously
  - **50% increase in query volume**
- Client must follow AliasMode records (A, AAAA, and HTTPS again)
- Client must resolve ServiceMode TargetName (A and AAAA)
- Participating recursives can provide these records in the Additional Section
- Extra-special requirements for Encrypted ClientHello
  - Client **must wait** for the HTTPS response before starting TLS, in case it enables ECH
  - Client **cannot fall back** if the HTTPS query fails with a timeout or SERVFAIL
    - This would open a downgrade attack.
  - **Potential operational challenge!**

Example: the HTTPS RR and Multi-CDN hosting

Clients may end up on one or more service endpoints (i.e. sets of servers) which may have different capabilities and keys, such as on different CDNs. The HTTPS RR binds each endpoint together.



Quick note: IP hints

```
@ 7200 IN HTTPS 1 svc.cdn.example ipv4hint=192.0.2.1,192.0.2.2
```

- **OPTIONAL**: Clients are free to ignore IP hints, zones are free to omit them
- Only used if the recursive resolver didn’t chase the A record for svc.cdn.ex
- Avoids delay while the client chases the A record
- Not rewritten by DNS64

Current Status

- IANA has assigned RR type codepoints (SVCB=64, HTTPS=65)
- draft-ietf-dnsop-svcb-https is nearly final
- Deployed
  - iOS 14, macOS 11
  - Cloudflare CDN
  - Akamai Cacheserve Resolver (See Ralf Weber's session for results!)
- In Beta
  - Chrome (new!)
  - Firefox
- In development
  - BIND
  - PowerDNS
  - Unbound
  - many others