HTTPS and SVCB Records

(New records for the Web)

Carsten Strotmann, Alan Clegg and the ISC Team

Welcome

Welcome to our Webinar on HTTPS and SVCB DNS records

In this Webinar

- Why new records?
- The issue with CNAME
- Use cases of HTTPS and SVCB records
- Implementation status
- Examples of HTTPS records

DNS Alias

The issue with CNAME

- DNS domain owner sometimes would like to alias names in DNS that is also the base (apex) name of a DNS zone
- Since the very beginning DNS has an alias function in form of the `CNAME` record
- However the `CNAME` record has limitations

The issue with CNAME

- The zone snippet below is not valid

```
$ORIGIN example.invalid.
$TTL 1h
@        IN SOA   ns1  .  1001 2h 1h 41d 1h
         IN NS    ns1
         IN NS    ns2
         IN CNAME www
www      IN AAAA  2001:db8::100
```

- A CNAME will alias **all** record types, including SOA, NS, A, AAAA etc
  - The presence of a CNAME for a domain name will trigger a new DNS name resolution with the new domain name
  - Having other records on the same domain name as the CNAME will create a consistency issue for DNS (see Appendix B of https://www.ietf.org/archive/id/draft-ietf-dnsop-aname-04.txt for details)
- Therefore a domain name that has a CNAME record cannot have any other record type (with the exception of RRSIG / NSEC / NSEC3 records)

The issue with CNAME

- For the same reason it is also not possible to alias a full domain at the zone apex

```
$ORIGIN example.invalid.
$TTL 1h
@        IN SOA   ns1  .  1001 2h 1h 41d 1h
         IN NS    ns1
         IN NS    ns2
         IN CNAME some-other-domain.invalid.
```

The issue with CNAME

- The DNAME resource record (RFC 6672) allows to alias full domains
  - But the DNAME must be placed into the parent (often the top level domain), that zone is unreachable for most domain owners

The issue with CNAME

- Because aliasing full domain names is often requested by domain owners, larger DNS companies (DNSimple, DNS Made Easy, EasyDNS, Cloudflare, Amazon, Dyn, and Akamai) started to create their own, non-standard solutions
- The IETF also tried to find solutions (`ANAME` record, `http` record …)

HTTPS and SVCB records

- The *HTTPS* and *SVCB* (Service Binding) records are currently standardized in the IETF
    - The document content is compete, the text is currenly in the IETF review process
    - There will be likely a new RFC published on these records in 2023
    - The records are already in use since 2020

The old way of DNS name resolution for services

- In the traditional way of name resolution for a service (for example a website), the client (browser) would resolve the domain name used in the URL …
  - … into an IPv6 address using an AAAA type query
  - … into an IPv4 address using an A type query
  - … into both an IPv6 and IPv4 addresses using asynchronous queries (Happy Eyeballs)

The problems with traditional service name resolution in the modern Internet

- There is a fixed connection between the *identity* (domain-name of the service) and the location (hostname of the server)
- It is not possible to alias a full domain (as `CNAME` records are not allowed on the zone apex)
- Upgrade to new protocol versions (such as HTTP/3 aka QUIC) require an additional round-trip
- There is no signaling of availability of secure transport (TLS via HTTPS)

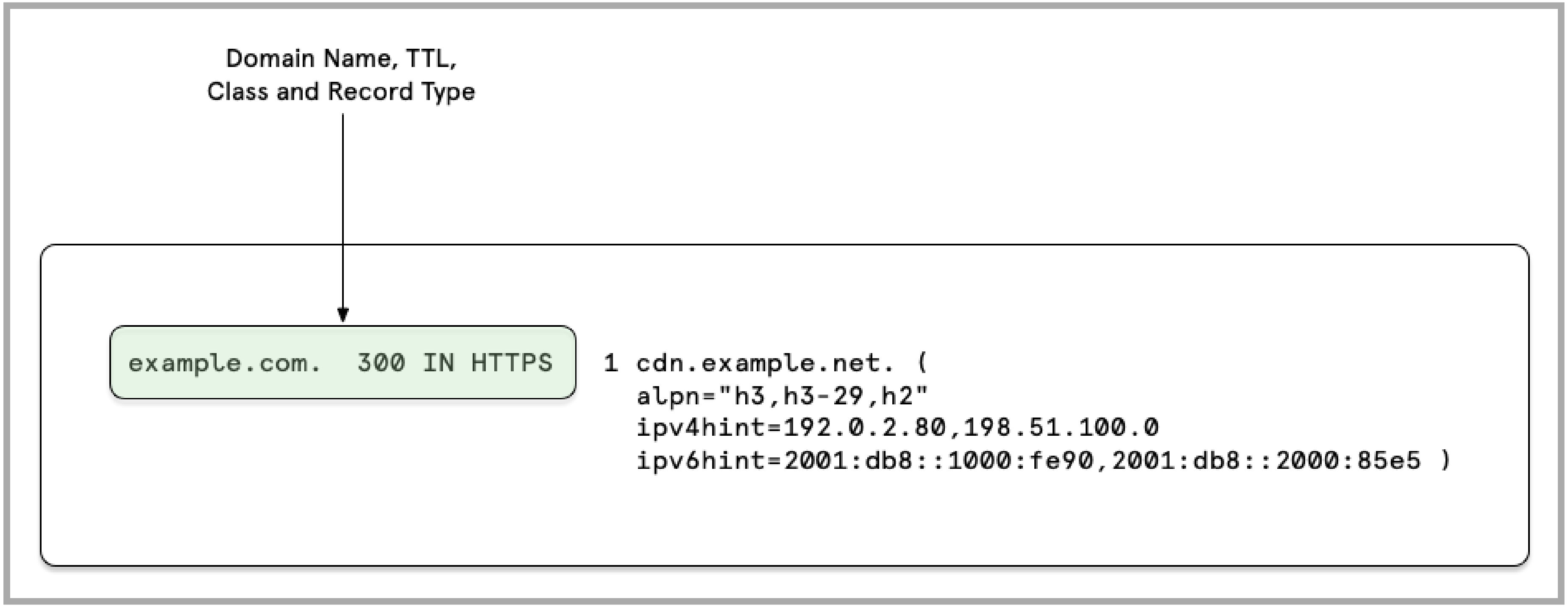
HTTPS Records

- The goals of the HTTPS (and the more general SVCB) record are
    - To provide all required information to make a connection to a service
    - Fewer DNS queries, so less round-trip time for application protocol connections
    - New DNS alias function (replacing proprietary records)
    - Separating identity (domain) from location (host)

HTTPS and SVCB records

- The HTTPS record is a special version of the SVCB record
- Almost all information given in this webinar is true for both the HTTPS and for the SVCB record
- As the HTTPS record is the simpler to understand, we will cover HTTPS first and SVCB later

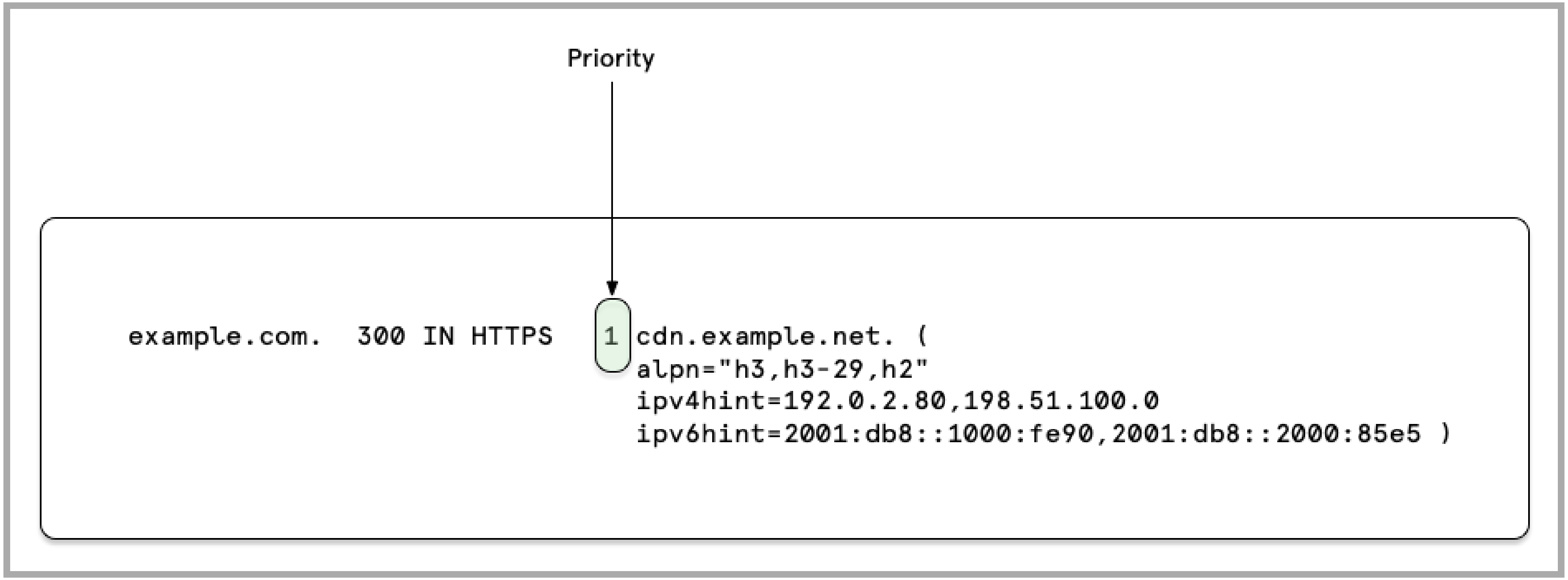
HTTPS Record - Example (1/6)



Domain Name, TTL, Class and Record Type

```
example.com.   300 IN HTTPS 1 cdn.example.net. (
                              alpn="h3,h3-29,h2"
                              ipv4hint=192.0.2.80,198.51.100.0
                              ipv6hint=2001:db8::1000:fe90,2001:db8::2000:85e5 )
```

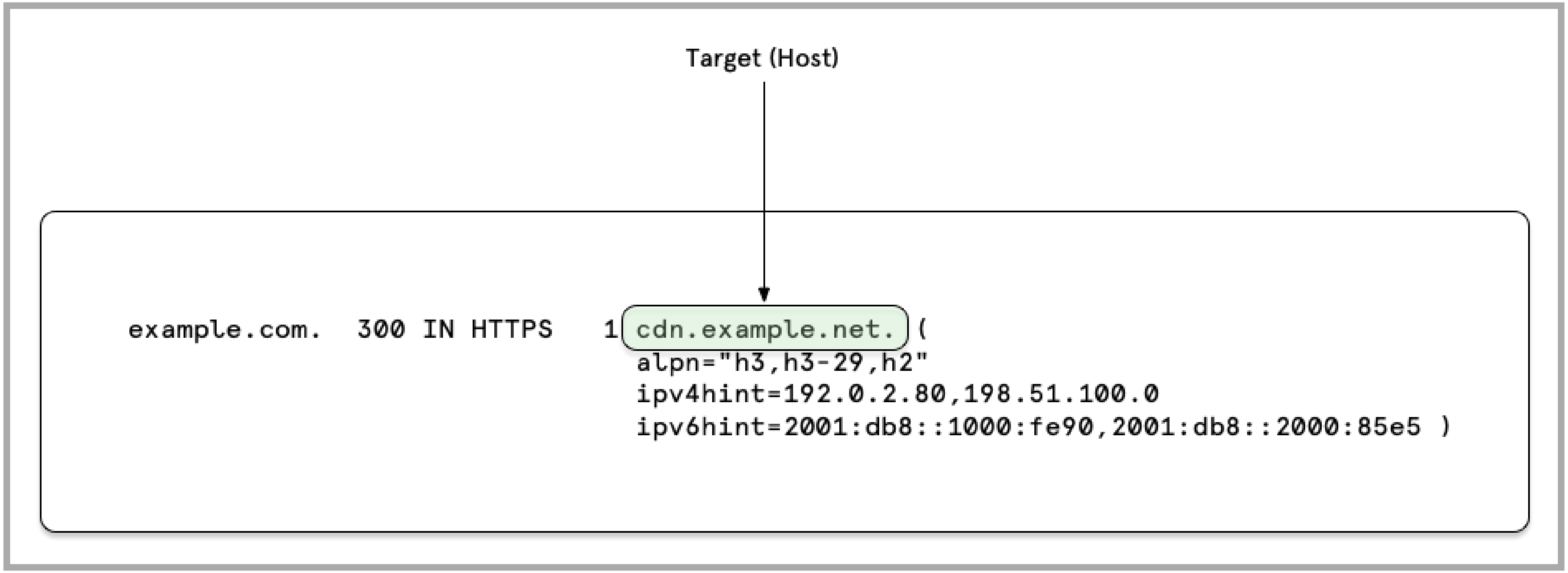
HTTPS Record - Example (2/6)



Priority

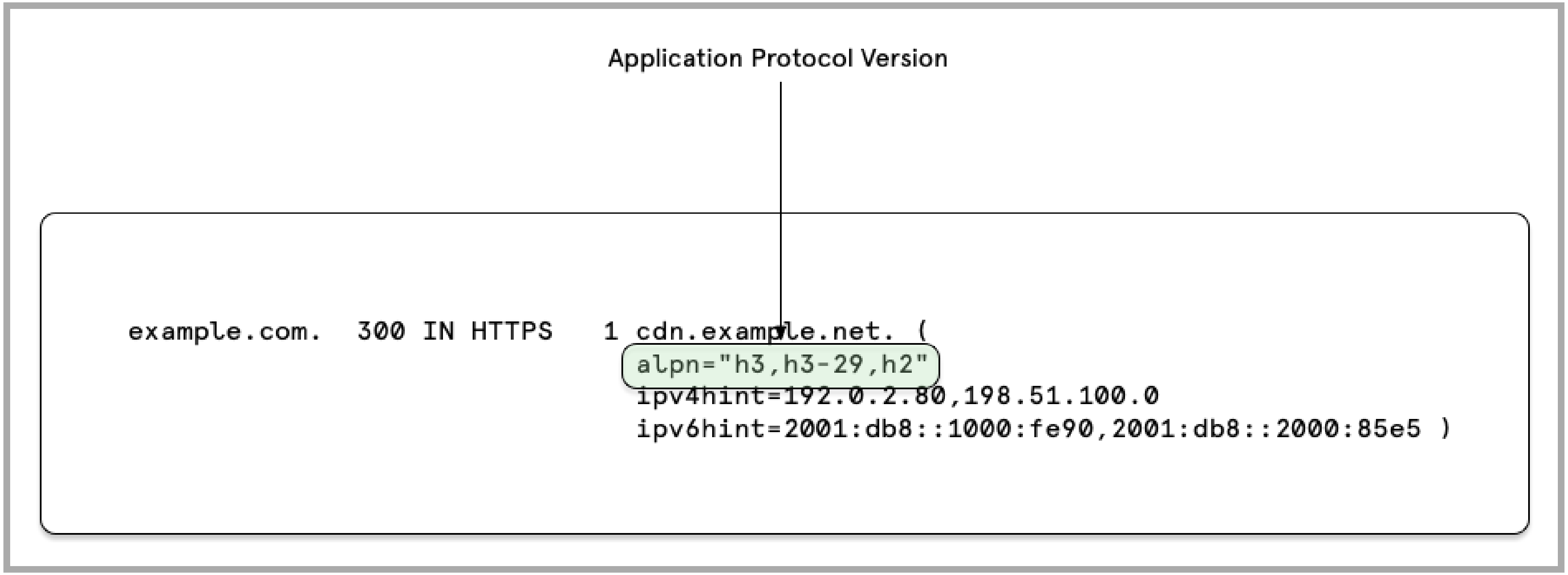
```
example.com.   300 IN HTTPS    1 cdn.example.net. (
                                 alpn="h3,h3-29,h2"
                                 ipv4hint=192.0.2.80,198.51.100.0
                                 ipv6hint=2001:db8::1000:fe90,2001:db8::2000:85e5 )
```

HTTPS Record - Example (3/6)



```
example.com.   300 IN HTTPS    1 cdn.example.net. (
                                 alpn="h3,h3-29,h2"
                                 ipv4hint=192.0.2.80,198.51.100.0
                                 ipv6hint=2001:db8::1000:fe90,2001:db8::2000:85e5 )
```

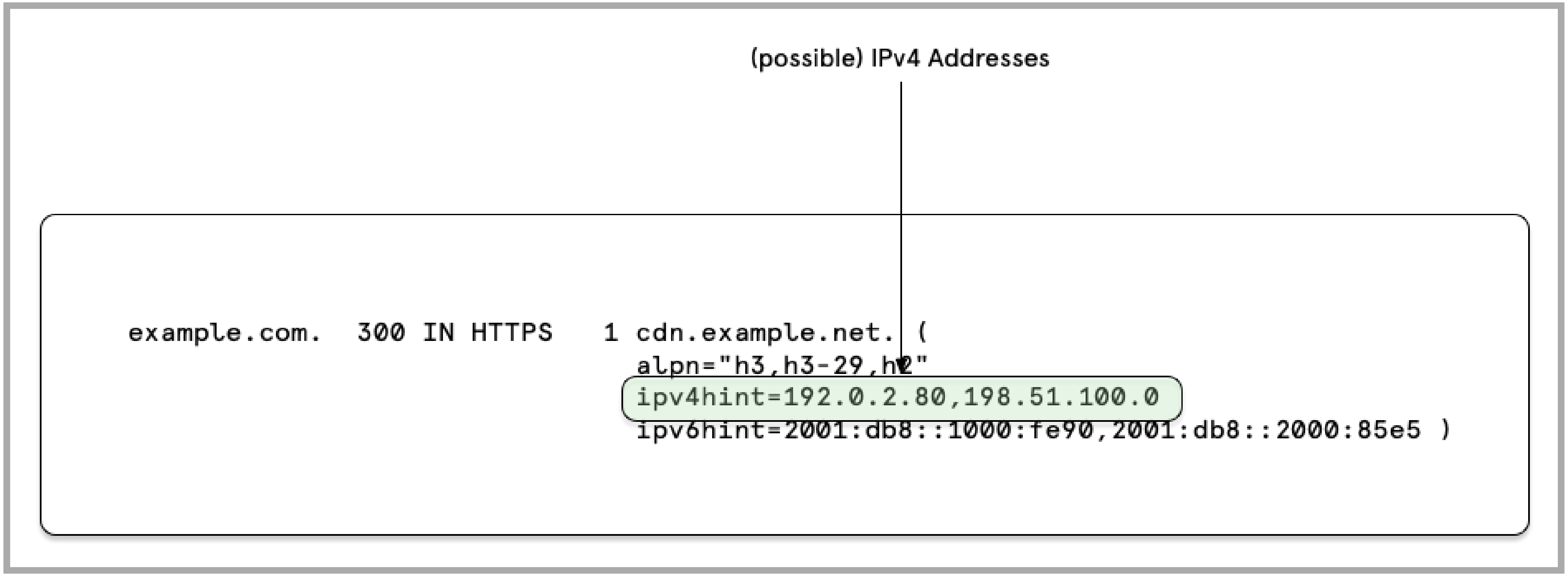
HTTPS Record - Example (4/6)



Application Protocol Version

```
example.com.   300 IN HTTPS    1 cdn.example.net. (
                                 alpn="h3,h3-29,h2"
                                 ipv4hint=192.0.2.80,198.51.100.0
                                 ipv6hint=2001:db8::1000:fe90,2001:db8::2000:85e5 )
```

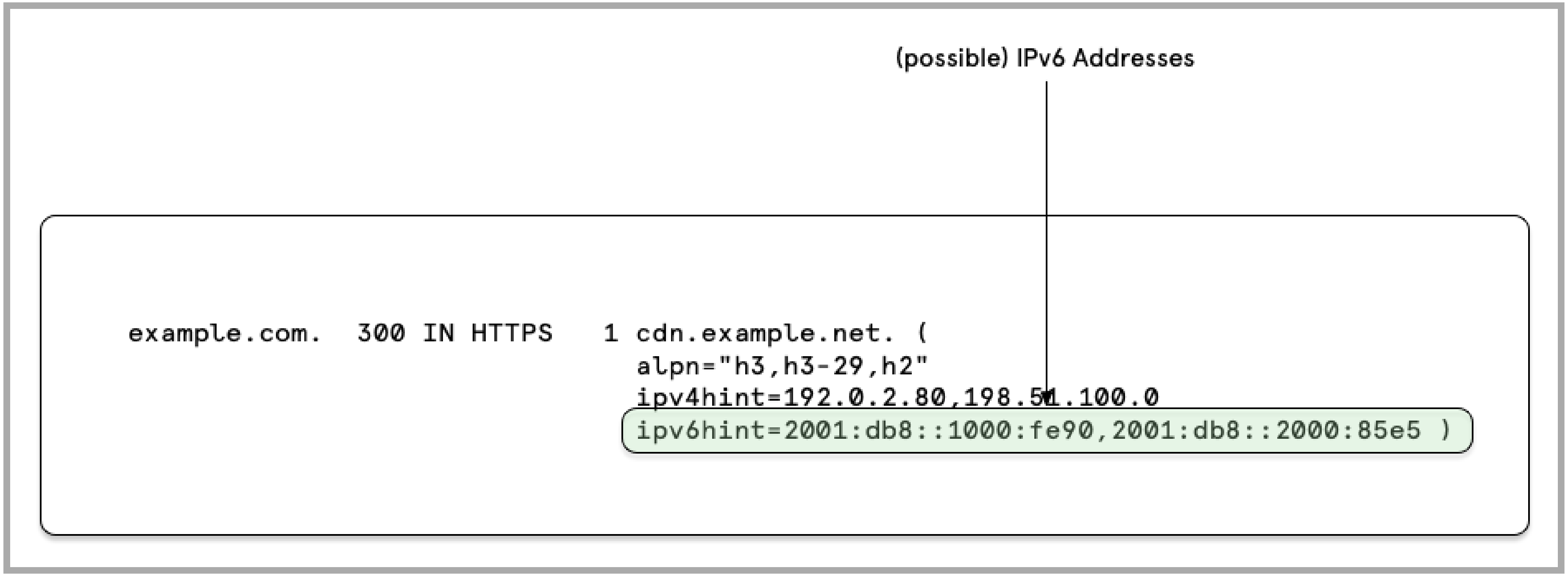
HTTPS Record - Example (5/6)



(possible) IPv4 Addresses

```
example.com.   300 IN HTTPS   1 cdn.example.net. (
                                alpn="h3,h3-29,h2"
                                ipv4hint=192.0.2.80,198.51.100.0
                                ipv6hint=2001:db8::1000:fe90,2001:db8::2000:85e5 )
```

HTTPS Record - Example (6/6)



(possible) IPv6 Addresses

```
example.com.   300 IN HTTPS   1 cdn.example.net. (
                                alpn="h3,h3-29,h2"
                                ipv4hint=192.0.2.80,198.51.100.0
                                ipv6hint=2001:db8::1000:fe90,2001:db8::2000:85e5 )
```

HTTPS/ SVCB Parameters

Select Application protocol (ALPN)

- The HTTPS record can signal the application protocol versions supported by the enpoint (server)
  - See RFC 7639 "The ALPN HTTP Header Field" and RFC 7301 "TLS Application-Layer Protocol Negotiation Extension"
- For HTTPS, this can be *HTTP/2 over TLS* (h2), *HTTP/2 over TCP* (h2c), *HTTP/3* (h3), *HTTP/1.1* (http/1.1)
- IANA Registry for ALPN https://www.iana.org/assignments/tls-extensiontype-values/tls-extensiontype-values.xhtml#alpn-protocol-ids

```
example.com.    IN HTTPS 1 . alpn="h3,h2,http/1.1"
```

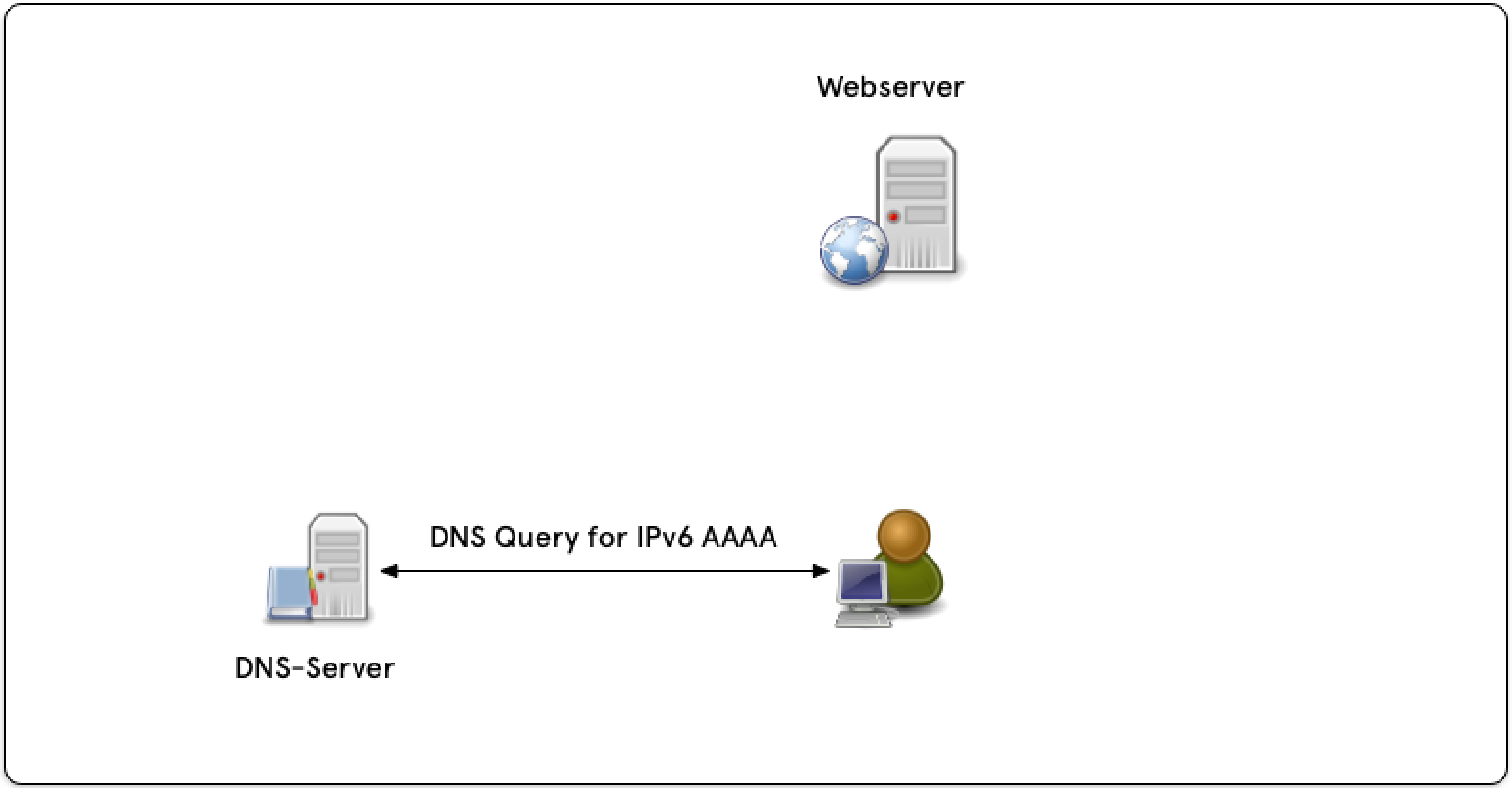
No support for the default protocol

- Most protocols have a *default* protocol version
- The default protocol version will be used as a *last resort* if all other ALPN protocol versions fail
- The HTTPS key `no-default-alpn` can be used to disable the default protocol version

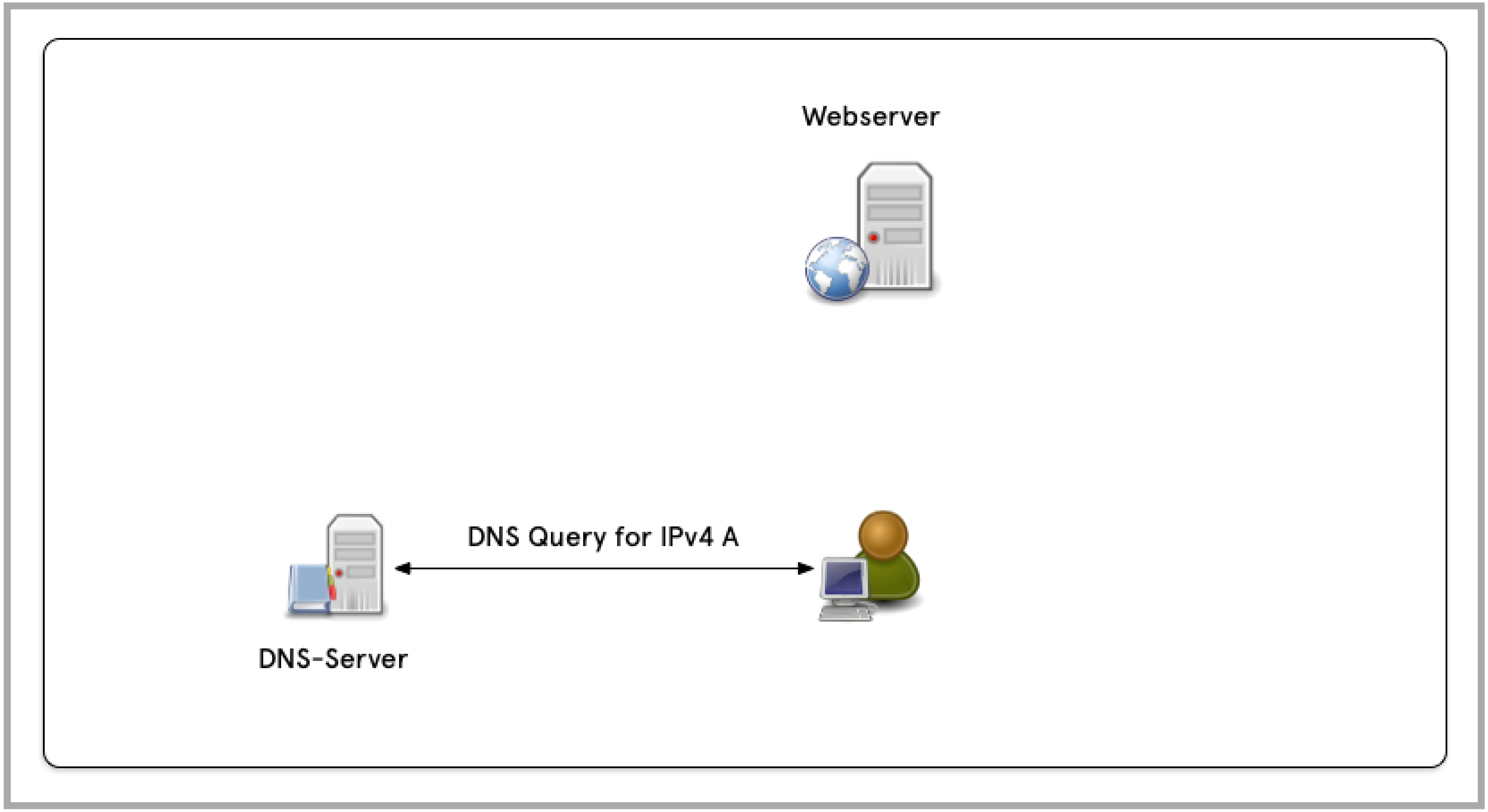
```
example.com.    IN HTTPS 1 . alpn="h3" no-default-alpn
```

- The `no-default-alpn` key does not have a value, so the equal sign = can be omitted

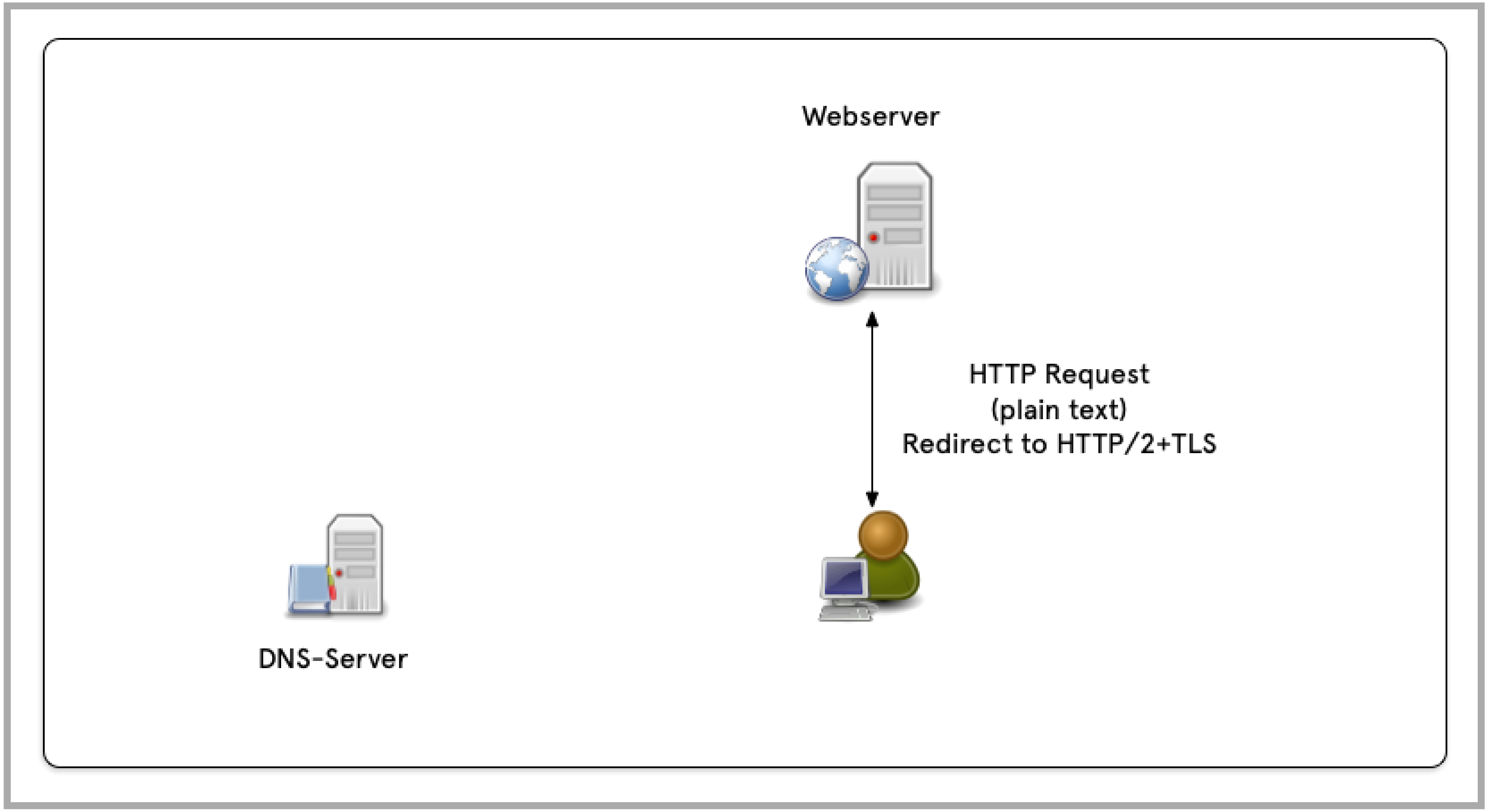
Session establishment with Alt-Svc HTTP Header (1/5)



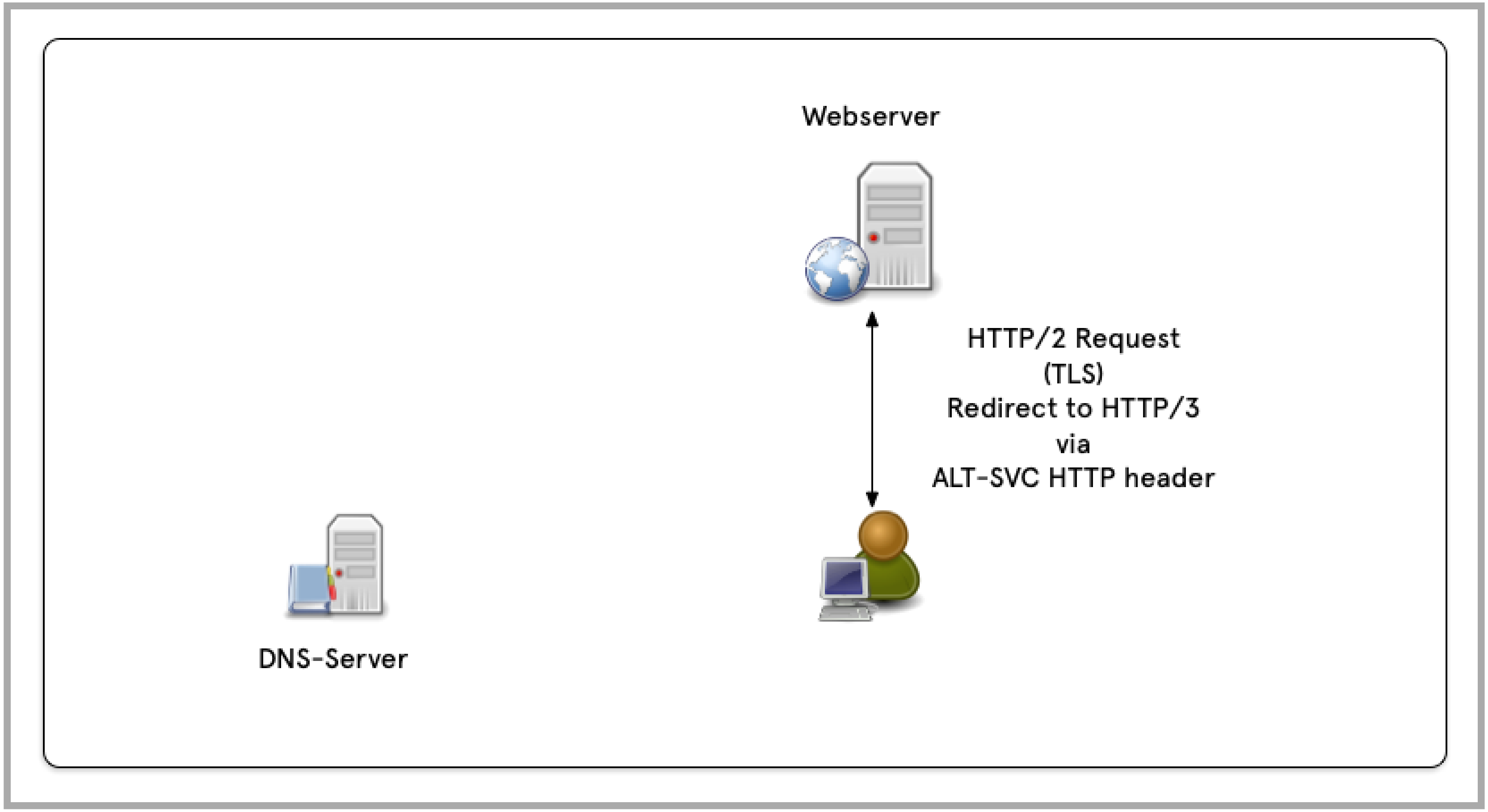
Session establishment with Alt-Svc HTTP Header (2/5)



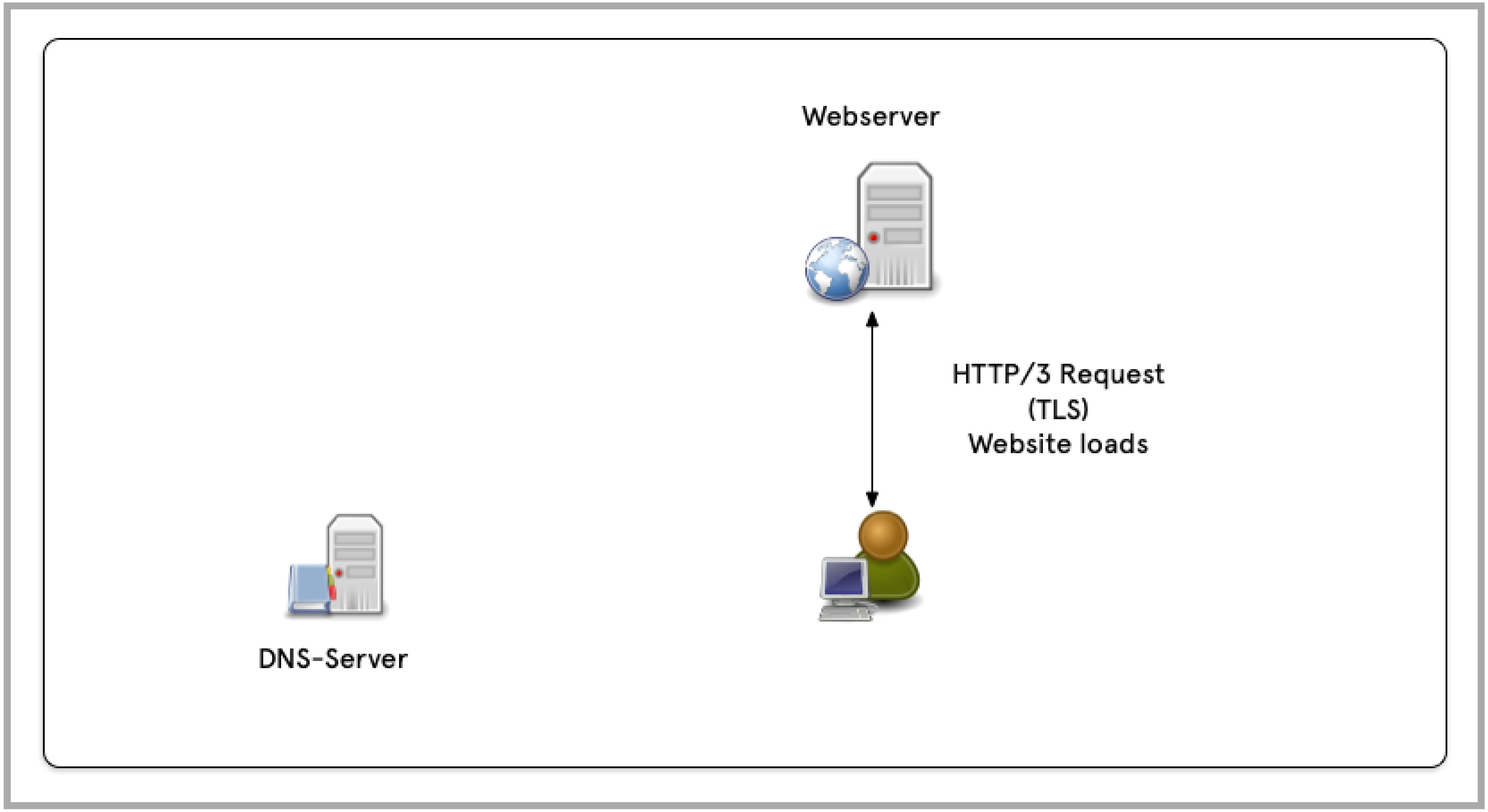
Session establishment with Alt-Svc HTTP Header (3/5)



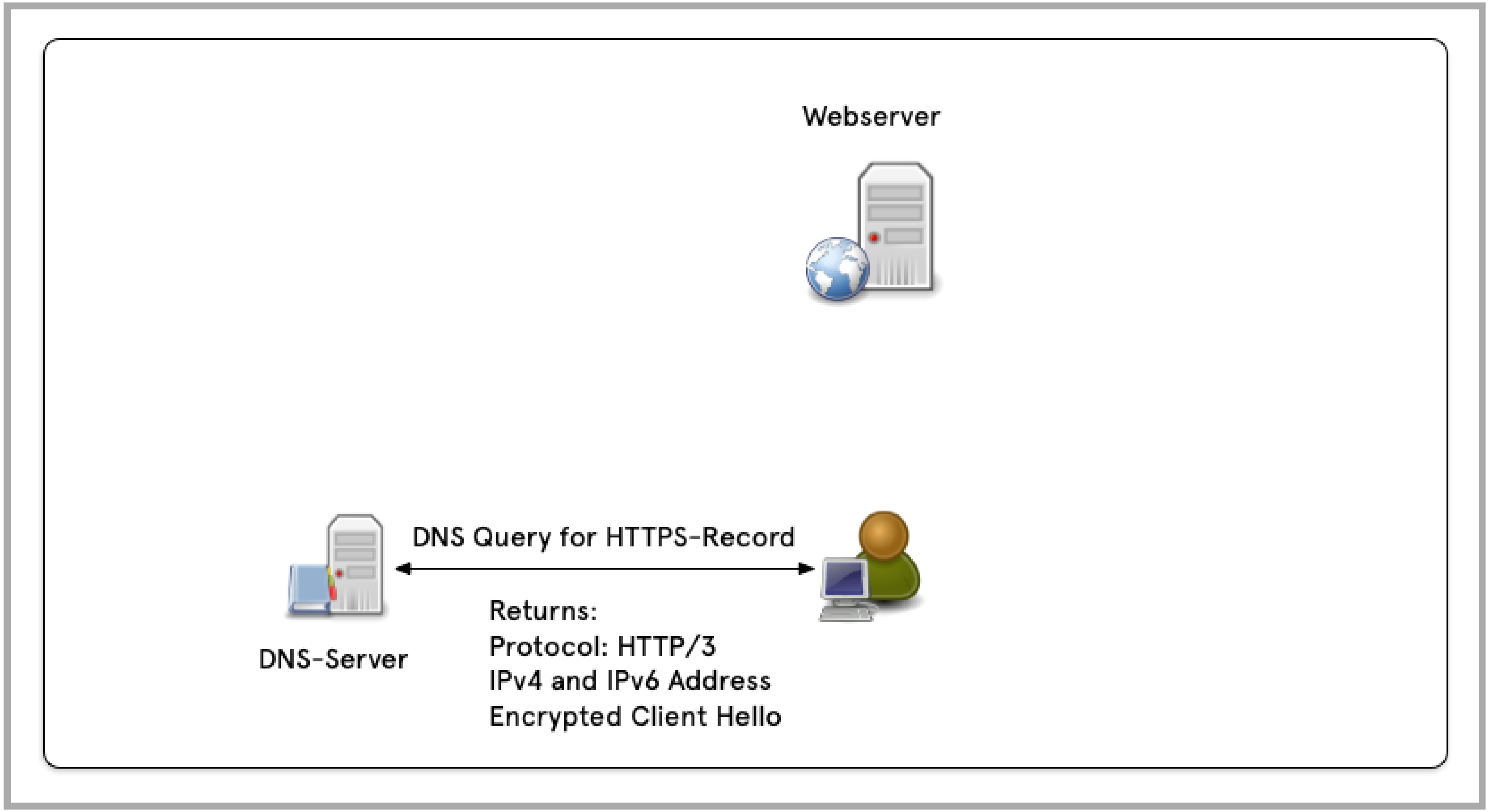
Session establishment with Alt-Svc HTTP Header (4/5)



Session establishment with Alt-Svc HTTP Header (5/5)



Session establishment with HTTPS Record (1/2)



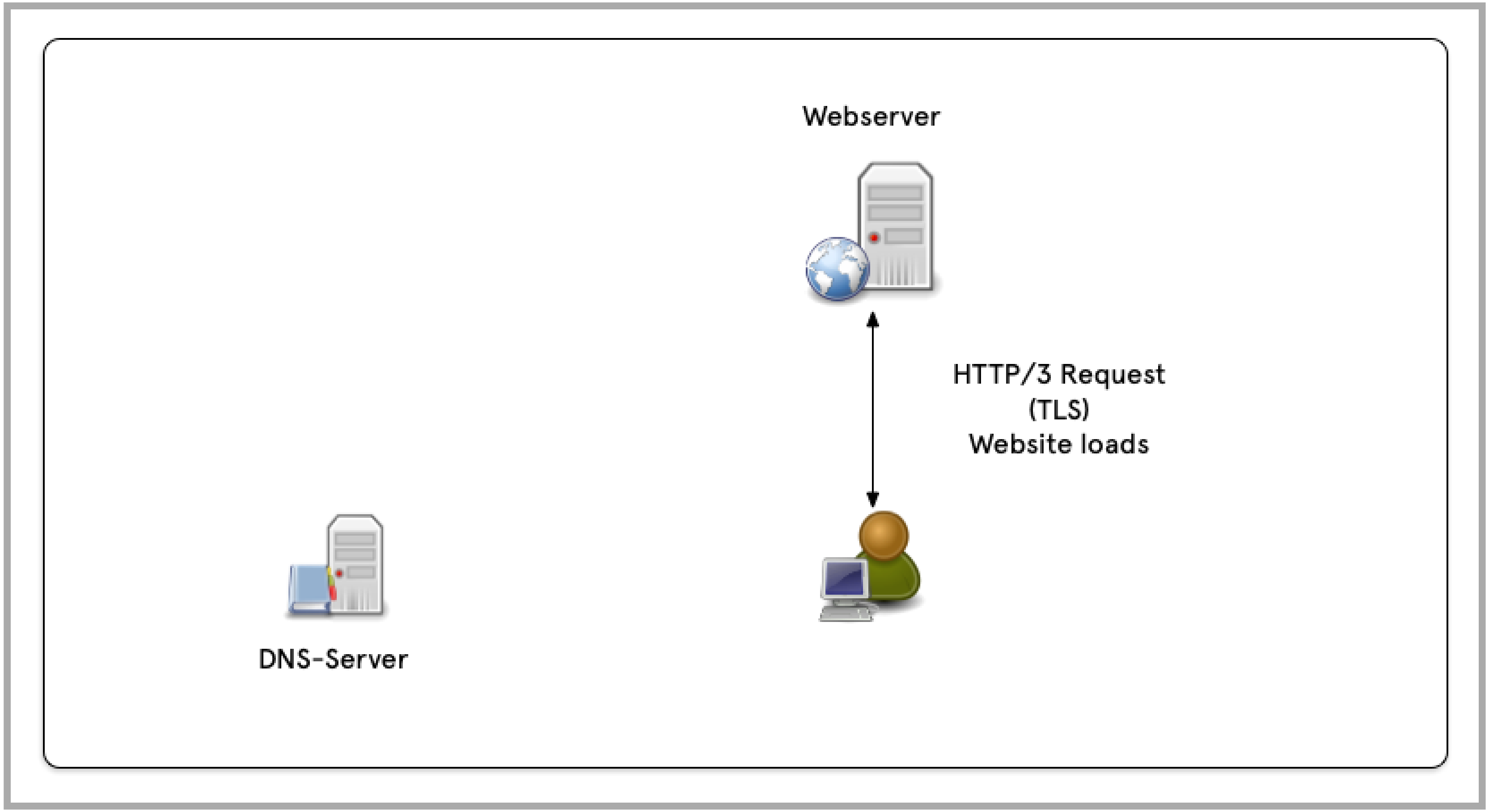
Webserver

DNS Query for HTTPS-Record

DNS-Server

Returns:
Protocol: HTTP/3
IPv4 and IPv6 Address
Encrypted Client Hello

Session establishment with HTTPS Record (2/2)



Encrypted client hello

- The `HTTPS` record can contain an `ech` parameter to signal *Encrypted Client Hello* support for the endpoint
- With ECH, the client can start sending encrypted data from the start without an initial clear text communication

Port number for endpoint

- With the `port` parameter, the HTTPS record can signal that the service is running on a non-standard port

IPv4/IPv6 address hints

- The HTTPS record can carry IPv6 and IPv4 address hints for the endpoint
- These addresses *can* be used by a client in case the addresses of an endpoint are not already known
  - If the addresses are already known, the hints from the HTTPS record should not be used
- The hints can speed up the initial connection to a service endpoint

What about SRV records?

- The `SRV` Record (RFC 2782 "A DNS RR for specifying the location of services (DNS SRV)") shares some functions with the HTTPS/SVCB records
- Notable differences
  - `SRV` records are not extensible, `HTTPS` / `SVCB` records are
  - `SRV` records always require an extra address resolution step
  - `SRV` records have a `weight` field to influence the load distribution to multiple endpoints

SVCB records

The differences between SVCB and HTTP

- The `SVCB` record (type64) is the generalized version of the `HTTPS` record (type65)
- `SVCB` encodes a service name (as an *Attrleaf* naming pattern, RFC 8552 and RFC 8553)
  - `_service.domain.tld.` or `_ssh.example.com.`
  - The service label can be pretended with a *Attrleaf* label for a non-standard port: `_4422._ssh.example.net.` (Service ssh on port 4422)

Name resolution with HTTPS/SVCB

Client resolution steps

- The client application does the bulk of the work when working with `HTTPS` and `SVCB` records
- Clients should try higher-priority service endpoints first, with fallback to lower-priority alternatives.

Clients, HTTPS and DNSSEC

- If the client is DNSSEC aware, and the zone containing the HTTPS record is DNSSEC signed, a failure in DNS resolution is fatal
  - Clients must not fall back to insecure address resolution using just A/AAAA records
  - This is to prevent downgrade attacks

DNS resolver and HTTPS/SVCB records

- HTTPS/SVCB does not require extra support from a DNS resolver (other than responding transparent for HTTPS/SVCB records)
- A DNS resolver *can* support HTTPS/SVCB by resolving the target domain addresses and delivering them in the response (additional section)

Authoritative DNS server and HTTPS/SVCB records

- HTTPS/SVCB does not require extra support from an authoritative DNS server (other than support for the encoding of HTTPS/SVCB records or RFC3597 style unknown resource records)
- A HTTPS/SVCB aware DNS server should return the *in zone* address records of domain names in the target field of an HTTPS/SVCB answer

Use cases of HTTPS/SVCB records

Zone Alias

- One main function of HTTPS records is to alias whole domains without the need to configure a HTTP redirect on the web-server

```
$TTL 3600
example.de.    IN SOA  ns1.example.com.   .   1001 2h 1h 41d 1h
example.de.    IN NS   ns1.example.com.
example.de.    IN NS   ns2.example.com.
example.de.    IN HTTPS 0 example.com.  # Alias example.de -> example.com
```

Load-Distribution

- Using the same value in the *priority* field of the HTTPS/SVCB records will trigger a simple load-distribution scheme inside the client
    - Clients will randomly connect to one if the endpoints
    - This is similar to good old address record *round robin*, but it is implemented on the client, not in the DNS resolver

```
auth.example.com. 400  IN HTTPS 1 server1.datacenter.example.
auth.example.com. 400  IN HTTPS 1 server2.datacenter.example.
```

- Don't use priority value `0` for load-distribution, as a `0` will trigger the alias mode, not the service mode

Backup-Service

- Using different values in the *priority* field of the HTTPS/SVCB record will create a fallback configuration for the service

```
dav.example.com. 400  IN HTTPS 100 server1.datacenter.example.
dav.example.com. 400  IN HTTPS 200 server2.datacenter.example.
```

- A client will first try `server1.datacenter.example.`, and if that endpoint cannot be reached, it will try `server2.datacenter.example.`

Different Port

- The `HTTPS` record can indicate different TCP/UDP ports for the server implementing the service

```
cloud.example.com. 86400  IN HTTPS 100 . port=8443
cloud.example.com. 86400  IN HTTPS 200 backup.example.net. port=8800
```

- The web-service `cloud.example.com` has two endpoints
  - The server `cloud.example.com` running on port 8443
  - The server `backup.example.net` running on port 8800

Different Application Protocol / Protocol Upgrade

- The `HTTPS` record signals to the client the protocol versions to supported
  - And a recommended order

```
example.com. 86400  IN HTTPS 100 . alpn="h3,h2"
example.com. 86400  IN HTTPS 200 backup.example.net. alpn="h2"
```

- Here, the server at `example.com` does support HTTP/3 (QUIC) and HTTP/2
  - The backup server at `backup.example.net` only supports HTTP/2

Encrypted Client Hello

- The "ech" parameter transports the ECH configuration for the service running on the target host
- The value of the parameter is an *ECHConfigList* (see https://datatracker.ietf.org/doc/html/draft-ietf-tls-esni) containing the *HPKE* public key of the service enpoint
    - It's base64 encoded binary data

```
example.com.  IN HTTPS  1 . ech="aHR0cHM6Ly9k[...]scy1lc25pCg=="
```

Using HTTPS records for onion services

- Web-Browser that are able to use *The Onion Router* overlay network can use the HTTPS record to provide Onion-Services under a well known (and human readable) domain name

```
example.com. 86400  IN HTTPS 1 ydahwfjqeqghj47srnj5zvn5jdm4o5zcgrqqhbs25sd75dhmz6sfbvqd.onion alpn="h2"
example.com. 86400  IN HTTPS 2 . alpn="h3,h2"
```

- In the example above, the browser will first try to reach the service over the alternative *onion* address (staying within the TOR network), before trying to fall back to the connection through the TOR network reaching the service on the Internet via an exit node
- Issue tracker for this feature request https://gitlab.torproject.org/tpo/applications/tor-browser/-/issues/41325

Implementations

General implementation status

- The HTTPS/SVCB records are supported by the current versions of popular open source DNS software (such as BIND 9)
- Client implementation (operating systems and web-browser) varies
- No client currently (Nov 2022) implements *all* features of the HTTPS/SVCB records

Apple iOS 14 / macOS 12

- Apple started with support for the HTTPS resource record in 2020
- Shortly after the release of iOS 14 in Fall 2020, the HTTPS (type65) RR was seen as the third most requested record on large ISPs DNS resolver

HTTPS is Firefox

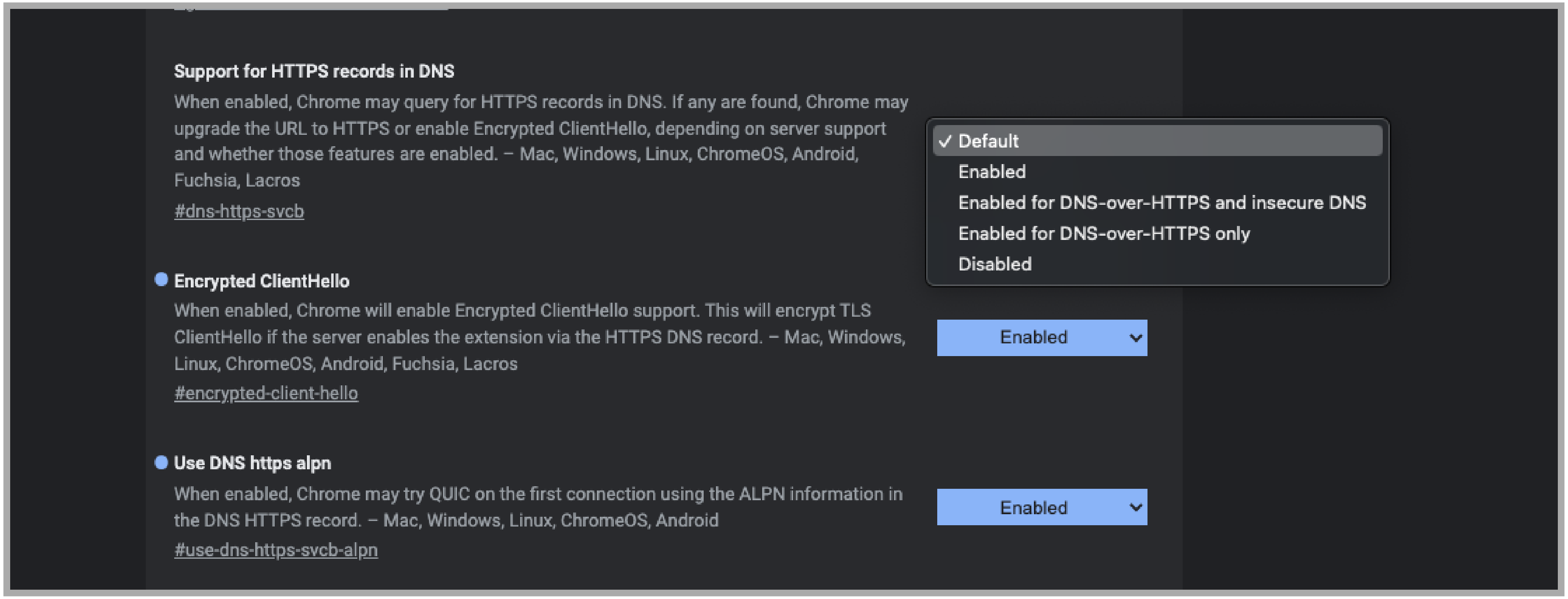
- Setting `network.dns.use_https_rr_as_altsvc` in Firefox 85+
- Mozilla Bug-Tracker on `HTTPS` record support https://bugzilla.mozilla.org/show_bug.cgi?id=1634793

HTTPS Records in Google Chrome

- Google Chrome design document on the support for the HTTPS record https://docs.google.com/document/d/1k461sRbddjDGj7ZKHZvmB-ENUWSdX_3Fpp2dmXQ/edit

Google Chrome

- Controlling the use of `HTTPS` records in Google Chrome



Support for HTTPS records in DNS

When enabled, Chrome may query for HTTPS records in DNS. If any are found, Chrome may upgrade the URL to HTTPS or enable Encrypted ClientHello, depending on server support and whether those features are enabled. – Mac, Windows, Linux, ChromeOS, Android, Fuchsia, Lacros

#dns-https-svcb

✓ Default
Enabled
Enabled for DNS-over-HTTPS and insecure DNS
Enabled for DNS-over-HTTPS only
Disabled

● **Encrypted ClientHello**

When enabled, Chrome will enable Encrypted ClientHello support. This will encrypt TLS ClientHello if the server enables the extension via the HTTPS DNS record. – Mac, Windows, Linux, ChromeOS, Android, Fuchsia, Lacros

#encrypted-client-hello

Enabled

● **Use DNS https alpn**

When enabled, Chrome may try QUIC on the first connection using the ALPN information in the DNS HTTPS record. – Mac, Windows, Linux, ChromeOS, Android

#use-dns-https-svcb-alpn

Enabled

DNS Server

- BIND 9 since 9.18.0 / 9.16.21
- Knot DNS since 3.10 https://www.knot-dns.cz/2021-08-02-version-310.html
- NSD since 4.3.7 https://www.nlnetlabs.nl/news/2021/Jul/22/nsd-4.3.7-released/
- PowerDNS since 4.4.0 - https://doc.powerdns.com/authoritative/guides/svcb.htm

Literature and Links

- The Use Cases and Benefits of SVCB and HTTPS DNS Record Types https://www.domaintools.com/resources/blog/the-use-cases-and-benefits-of-svcb-and-https-dns-record-types
- Speeding up HTTPS and HTTP/3 negotiation with… DNS https://blog.cloudflare.com/speeding-up-https-and-http-negotiation-with-dns/
- Apple WWDC 2020 "Boost performance and security wit modern networking" https://developer.apple.com/videos/play/wwdc2020/101

Literature and Links

- Encrypted Client Hello: the future of ESNI in Firefox https://blog.mozilla.org/security/2021/01/07/encrypted-client-hello-the-future-of-esni-in-firefox/
- Internet Draft "TLS Encrypted Client Hello" https://datatracker.ietf.org/doc/html/draft-ietf-tls-esni
- Internet Draft "Service binding and parameter specification via the DNS (DNS SVCB and HTTPS RRs)" https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-svcb-https

Upcoming ISC Webinars

- 15th Dec 2022 - Memory management in BIND 9.16/9.18

Questions and Answers