HTTP Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

B. Schwartz
Google
M. Bishop
E. Nygren
Akamai Technologies

July 8, 2019

HTTPSSVC service location and parameter specification via the DNS (DNS HTTPSSVC)
draft-nygren-httpbis-httpssvc-03

Abstract

This document specifies an "HTTPSSVC" DNS resource record type to facilitate the lookup of information needed to make connections for HTTPS URIs. The HTTPSSVC DNS RR mechanism allows an HTTPS origin hostname to be served from multiple network services, each with associated parameters (such as transport protocol and keying material for encrypting TLS SNI). It also provides a solution for the inability of the DNS to allow a CNAME to be placed at the apex of a domain name. Finally, it provides a way to indicate that the origin supports HTTPS without having to resort to redirects, allowing clients to remove HTTP from the bootstrapping process.

By allowing this information to be bootstrapped in the DNS, it allows for clients to learn of alternative services before their first contact with the origin. This arrangement offers potential benefits to both performance and privacy.

TO BE REMOVED: This proposal is inspired by and based on recent DNS usage proposals such as ALTSVC, ANAME, and ESNIKEYS (as well as long standing desires to have SRV or a functional equivalent implemented for HTTP). These proposals each provide an important function but are potentially incompatible with each other, such as when an origin is load-balanced across multiple hosting providers (multi-CDN). Furthermore, these each add potential cases for adding additional record lookups in-addition to AAAA/A lookups. This design attempts to provide a unified framework that encompasses the key functionality of these proposals, as well as providing some extensibility for addressing similar future challenges.

This Internet-Draft will expire on January 9, 2020.

Table of Contents

1. Introduction

The HTTPSSVC RR is intended to address a number of challenges facing HTTPS clients and services, while also providing an extensible model to handle similar use-cases without forcing clients to perform additional DNS lookups and without forcing them to first make connections to a default service for the origin.

When clients need to make a connection to fetch resources associated with an HTTPS URI, they must first resolve A and/or AAAA address resource records for the origin hostname. This is adequate when clients default to TCP port 443, do not support Encrypted SNI [ESNI], and where the origin service operator does not have a desire to put an CNAME at a zone apex (such as for "https://example.com"). Handling situations beyond this within the DNS requires learning additional information, and it is highly desirable to minimize the

number of round-trip and lookups required to learn this additional information.

1.1. Introductory Example

As an introductory example, a set of example HTTPSSVC and associated A+AAAA records might be:

```
www.example.com.  2H  IN CNAME   svc.example.net.
example.com.      2H  IN HTTPSSVC 0 0 svc.example.net.
svc.example.net.  2H  IN HTTPSSVC 1 2 svc3.example.net. "h3=\":8003\"; \
                                  esnikeys=\"...\""
svc.example.net.  2H  IN HTTPSSVC 1 3 svc2.example.net. "h2=\":8002\"; \
                                  esnikeys=\"...\""
svc2.example.net. 300 IN A       192.0.2.2
svc2.example.net. 300 IN AAAA    2001:db8::2
svc3.example.net. 300 IN A       192.0.2.3
svc3.example.net. 300 IN AAAA    2001:db8::3
```

In the preceding example, both of the "example.com" and "www.example.com" origin names are aliased to use service endpoints offered as "svc.example.net" (with "www.example.com" continuing to use a CNAME alias). HTTP/2 is available on a cluster of machines located at svc2.example.net with TCP port 8002 and HTTP/3 is available on a cluster of machines located at svc3.example.net with UDP port 8003. An ESNI key is specified which allows the SNI values of "example.com" and "www.example.com" to be encrypted in the handshake with these service endpoints. When connecting, clients will continue to treat the authoritative origins as "https://example.com" and "https://www.example.com", respectively.

1.2. Goals of the HTTPSSVC RR

The goal of the HTTSSVC RR is to allow clients to resolve a single additional DNS RR in a way that:

- Provides service endpoints authoritative for an origin, along with parameters associated with each of these endpoints. In particular:
  - to support connecting directly to [HTTP3] (QUIC transport) service endpoints
  - to obtain the [ESNI] keys associated with a service endpoint
- Does not assume that all service endpoints have the same parameters (such as ESNI keys) or capabilities (such as [HTTP3]) or are even operated by the same entity. This is important as DNS

does not provide any way to tie together multiple RRs for the same name. For example, if www.example.com is a CNAME alias that switches between one of three CDNs or hosting enviroments, records (such as A and AAAA) for that name may have been sourced from different environments.

- o Enables the functional equivalent of a CNAME at a zone apex (such as "example.com") for HTTPS traffic, and generally enables delegation of operational authority for an HTTPS origin within the DNS to an alternate name. This addresses a set of long-standing issues due to HTTP(S) clients not implementing support for SRV records, as well as due to a limitation that a DNS name can not have both a CNAME record as well as NS RRs (as is the case for zone apex names)

1.3. Overview of the HTTPSSVC RR

This subsection briefly describes the HTTPSSVC RR in a non-normative manner.

The HTTPSSVC RR has four primary fields:

1. SvcRecordType: A numeric flag indicating how to interpret the subsequent fields. When "0", it indicates that the RR contains an alias. When "1", it indicates that the RR contains an alternative service definition.

2. SvcFieldPriority: The priority of this record (relative to others, with lower values preferred). Applicable when SvcRecordType is "1", and otherwise has value "0". (Described in Section 3.2.)

3. SvcDomainName: The domain name of either the alias target (when SvcRecordType is "0") or the uri-host domain name of the alternative service endpoint (when SvcRecordType is "1").

4. SvcFieldValue: An Alternative Service field value describing the alternative service endpoint for the domain name specified in SvcDomainName (only when SvcRecordType is "1" and otherwise empty).

Cooperating DNS recursive resolvers will perform subsequent record resolution (for HTTPSSVC, A, and AAAA records) and return them in the Additional Section of the response. Clients must either use responses included in the additional section returned by the recursive resolver or perform necessary HTTPSSVC, A, and AAAA record resolutions. DNS authoritative servers may attach in-bailiwick

HTTPSSVC, A, AAAA, and CNAME records in the Additional Section to responses for an HTTPSSVC query.

When SvcRecordType is "1", the HTTPSSVC RR extends the concept introduced in the HTTP Alternative Services proposed standard [AltSvc]. Alt-Svc defines:

- an extensible data model for describing alternative network endpoints that are authoritative for an origin
- the "Alt-Svc Field Value", a text format for representing this information
- standards for sending information in this format from a server to a client over HTTP/1.1 and HTTP/2.

Together, these components provide a toolkit that has proven useful and effective for informing a client of alternative services for an origin. However, making use of an alternative service requires contacting the origin server first. This creates an obvious performance cost: users wait for a full HTTP connection initiation (multiple roundtrips) before learning of an alternative service that is preferred by the origin. The first connection also publicly reveals the user's intended destination to all entities along the network path.

The SvcFieldValue includes the Alt-Svc Field Value through the DNS. This is in its standard text format, with the uri-host portion of the alt-authority component moved into the SvcDomainName field of the HTTPSSVC RR. A client receiving this information during DNS resolution can skip the initial connection and proceed directly to an alternative service.

1.4. Additional Alt-Svc parameters

This document also defines one additional Alt-Svc parameter that can be used within SvcFieldValue:

- esnikeys (Section 7.1): The ESNIKeys structure from Section 4.1 of [ESNI] for use in encrypting the actual origin hostname in the TLS handshake.

1.5. Terminology

For consistency with [AltSvc], we adopt the following definitions:

- An "origin" is an information source as in [RFC6454].

- The "origin server" is the server that the client would reach when accessing the origin in the absence of Alt-Svc.

- An "alternative service" is a different server that can serve the origin.

Abstractly, the origin consists of a scheme (typically "https"), a host name, and a port (typically "443").

Additional DNS terminology intends to be consistent with [DNSTerm].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. The HTTPSSVC record type

The HTTPSSVC DNS resource record (RR) type (RRTYPE ???) is used to locate endpoints that can service an "https" origin. The presentation format of the record is:

RRName TTL Class HTTPSSVC SvcRecordType SvcFieldPriority \
SvcDomainName SvcFieldValue

where SvcRecordType is a numeric value of either "0" or "1", SvcFieldPriority is a number in the range 0-65535, SvcDomainName is a domain name, and SvcFieldValue is a string present when SvcRecordType is "1".

The algorithm for resolving HTTPSSVC records and associated address records is specified in Section 4.1.

2.1. HTTPSSVC RDATA Wire Format

The RDATA for the HTTPSSVC RR consists of:

- a 1 octet flag field for SvcRecordType, interpreted as an unsigned numeric value (0 to 255, with only values "0" and "1" defined here)

- a 2 octet field for SvcFieldPriority as an integer in network byte order. If SvcRecordType is "0", SvcFieldPriority MUST be 0.

- the uncompressed SvcDomainName, represented as a sequence of length-prefixed labels as in Section 3.1 of [RFC1035].

- the SvcFieldValue byte string, consuming the remainder of the record (so smaller than 65535 octets and constrained by the RRDATA and DNS message sizes).

When SvcRecordType is "0", the SvcFieldValue SHOULD be empty ("") and clients MUST ignore the contents of non-empty SvcFieldValue fields.

2.2. RRNames

In the case of the HTTPSSVC RR, an origin is translated into the RRName in the following manner:

1. If the scheme is "https" and the port is 443, then the RRName is equal to the origin host name. Otherwise the RRName is represented by prefixing the port and scheme with "_", then concatenating them with the host name, resulting in a domain name like "_8443._https.www.example.com.".

2. When a prior CNAME or HTTPSSVC record has aliased to an HTTPSSVC record, RRName shall be the name of the alias target.

Note that none of these forms alter the HTTPS origin or authority. For example, clients MUST continue to validate TLS certificate hostnames based on the origin host.

As an example for schemes and ports other than "https" and port 443:

```
_8443._wss.api.example.com. 2H IN HTTPSSVC 0 0 svc4.example.net.
svc4.example.net.  2H  IN HTTPSSVC 1 3 svc4.example.net. "h2=\":8004\"; \
                                  esnikeys=\"...\""
```

would indicate that "wss://api.example.com:8443" is aliased to use HTTP/2 service endpoints offered as "svc4.example.net" on port 8004.

2.3. SvcRecordType

The SvcRecordType field is a numeric value defined to be either "0" or "1". Within an HTTPSSVC RRSet, all RRs must have the same value for SvcRecordType. Clients and recursive servers MUST ignore HTTPSSVC resource records with other SvcRecordType values. If an RRSet contains a record with type "0", the client MUST ignore any records in the set with type "1".

When SvcRecordType is "0", the HTTPSSVC is defined to be in "alias form".

When SvcRecordType is "1", the HTTPSSVC is defined to be in "alternative service form".

2.4. HTTPSSVC records: alias form

When SvcRecordType is "0", the HTTPSSVC record is to be treated similar to a CNAME alias pointing to the domain name specified in SvcDomainName. HTTPSSVC RRSets MUST only have a single resource record in this form. If multiple are present, clients or recursive resolvers SHOULD pick one non-determinstically.

The common use-case for this form of the HTTPSSVC record is as an alternative to CNAMEs at the zone apex where they are not allowed. For example, if an operator of https://example.com wanted to point HTTPS requests to a service operating at svc.example.net, they would publish a record such as:

example.com. 3600 IN HTTPSSVC 0 0 svc.example.net.

The SvcDomainName MUST point to a domain name that contains another HTTPSSVC record and/or address (AAAA and/or A) records.

Note that the RRName and the SvcDomainName MAY themselves be CNAMEs. Clients and recursive resolvers MUST follow CNAMEs as normal.

Due to the risk of loops, clients and recursive resolvers MUST implement loop detection. Chains of consecutive HTTPSSVC and CNAME records SHOULD be limited to (8?) prior to reaching terminal address records.

The SvcFieldValue in this form SHOULD be an empty string and clients MUST ignore its contents.

As legacy clients will not know to use this record, service operators will likely need to retain fallback AAAA and A records alongside this HTTPSSVC record, although in a common case the target of the HTTPSSVC record might have better performance, and therefore would be preferable for clients implementing this specification to use.

2.5. HTTPSSVC records: alternative service form

When SvcRecordType is "1", the combination of SvcDomainName and SvcFieldValue within each resource record associates an Alternative Service Field Value with an origin.

The SvcFieldValue of the HTTPSSVC resource record contains an Alt-Svc Field Value, exactly as defined in Section 4 of [AltSvc], but with the uri-host moved to the SvcDomainName field.

For example, if the operator of https://www.example.com intends to include an HTTP response header like

```
Alt-Svc: h3="svc.example.net:8003"; ma=3600, \
         h2="svc.example.net:8002"; ma=3600
```

they could also publish an HTTPSSVC DNS RRSet like

```
www.example.com. 3600 IN HTTPSSVC 1 2 svc.example.net. "h3=\":8003\""
                         HTTPSSVC 1 3 svc.example.net. "h2=\":8002\""
```

This data type can be represented as an Unknown RR as described in [RFC3597]:

```
www.example.com. 3600 IN TYPE??? \\# TBD:WRITEME
```

This construction is intended to be extensible in two ways. First, any extensions that are made to the Alt-Svc format for transmission over HTTPS are also applicable here, unless expressly mentioned otherwise.

Second, by defining a way to map non-HTTPS schemes and non-default ports (Section 2.2), we provide a way for the HTTPSSVC to be used for them as needed. However, by using the origin name for the RRName for scheme https and port 443 we allow HTTPSSVC records to be included at the end of CNAME chains for existing site implementations without requiring changes in the zone containing the origin.

3. Differences from Alt-Svc as transmitted over HTTP

Publishing an alternative services form HTTPSSVC record in DNS is intended to be equivalent to transmitting this field value over HTTPS, and receiving an HTTPSSVC record is intended to be equivalent to receiving this field value over HTTPS. However, there are some small differences in the intended client and server behavior.

3.1. Omitting Max Age and Persist

When publishing an HTTPSSVC record in DNS, server operators MUST omit the "ma" parameter, which encodes the "max age" (i.e. expiration time) of an Alt-Svc Field Value. Instead, server operators SHOULD encode the expiration time in the DNS TTL, and MUST NOT set a TTL longer than the intended "max age".

When receiving an HTTPSSVC record, clients SHOULD synthesize a new "ma" parameter from the DNS TTL if the resulting alt-value is being passed to a subsystem that might employ caching.

When publishing an HTTPSSVC record, server operators MUST omit the "persist" parameter, which indicates whether the client should use this record on other network paths. When receiving an HTTPSSVC

record, clients MUST discard any records that contain a "persist" flag. Disabling persistence is important to prevent a local adversary in one network from implanting a forged DNS record that allows them to track users or hinder their connections after they leave that network.

3.2. Multiple records and preference ordering

Server operators MAY publish multiple SvcRecordType "1" HTTPSSVC records as an RRSET. When converting a collection of alt-values into an HTTPSSVC RRSET, the server operator MUST set the overall TTL to a value no larger than the minimum of the "max age" values (following Section 5.2 of [RFC2181]).

Each RR MUST contain exactly one alt-value, as described in Section 3 of [AltSvc].

As RRs within an RRSET are explicitly unordered collections, the SvcFieldPriority value is introduced to indicate priority. HTTPSSVC RRs with a smaller SvcFieldPriority value SHOULD be given preference over RRs with a larger SvcFieldPriority value.

Alt-values received via HTTPS are preferred over any Alt-value received via DNS.

When receiving an RRSET containing multiple HTTPSSVC records with the same SvcFieldPriority value, clients SHOULD apply a random shuffle within a priority level to the records before using them, to ensure randomized load-balancing.

3.3. Constructing Alt-Svc equivalent headers

For a client to construct the equivalent of an Alt-Svc HTTP response header:

1. For each RR, the SvcDomainName MUST be inserted as the uri-host. If SvcDomainName is has the value "." then the RRNAME for the final HTTPSSVC record MUST be inserted as the uri-host. (In the case of a CNAME or a HTTPSSVC SvcRecordType "0" record pointing to an HTTPSSVC record with SvcRecordType "1" and SvcDomainName "." then it is the RRNAME for the terminal HTTPSSVC record that must be inserted as the uri-host.)

2. The RRs SHOULD be ordered by increasing SvcFieldPriority, with shuffling for equal SvcFieldPriority values. Clients MAY choose to further prioritize alt-values where address records are immediately available for the alt-value's SvcDomainName.

3. The client SHOULD concatenate the thus-transformed-and-ordered SvcFieldValues in the RRSET, separated by commas. (This is semantically equivalent to receiving multiple Alt-Svc HTTP response headers, according to Section 3.2.2 of [HTTP]).

3.4. Granularity and lifetime control

Sending Alt-Svc over HTTP allows the server to tailor the Alt-Svc Field Value specifically to the client. When using an HTTPSSVC DNS record, groups of clients will necessarily receive the same Alt-Svc Field Value. Therefore, this standard is not suitable for uses that require single-client granularity in Alt-Svc.

Some DNS caching systems incorrectly extend the lifetime of DNS records beyond the stated TTL. Server operators MUST NOT rely on HTTPSSVC records expiring on time, and MAY shorten the TTL to compensate.

4. Client behaviors

4.1. Client resolution

When attempting to resolve a name HOST, clients should follow in-order:

1. Issue parallel AAAA/A and HTTPSSVC queries for the name HOST. The answers for these may or may not include CNAME pointers before reaching one or more of these records.

2. If an HTTPSSVC record of SvcRecordType "0" is returned for HOST, clients should loop back to step 1 replacing HOST with SvcDomainName, subject to loop detection heuristics.

3. If one or more HTTPSSVC record of SvcRecordType "1" is returned for HOST, clients should synthesize equivalent Alt-Svc Field Values based on the SvcDomainName and SvcFieldValue. If one of these alt-values is selected to be used in a connection, the client will need to resolve AAAA and/or A records for SvcDomainName.

4. If only AAAA and/or A records are present for HOST (and no HTTPSSVC), clients should make a connection to one of the IP addresses contained in these records and proceed normally.

When selecting between AAAA and A records to use, clients may use an approach such as [HappyEyeballsV2]

Some possible optimizations are discussed in Section 6 to reduce latency impact in comparison to ordinary AAAA/A lookups.

4.2. HTTP Strict Transport Security

By publishing an HTTPSSVC record, the server operator indicates that all useful HTTP resources on that origin are reachable over HTTPS, similar to HTTP Strict Transport Security [HSTS]. When an HTTPSSVC record is present for an origin, all "http" scheme requests for that origin SHOULD logically be redirected to "https".

Prior to making an "http" scheme request, the client SHOULD perform a lookup to determine if an HTTPSSVC record is available for that origin. To do so, the client SHOULD construct a corresponding "https" URL as follows:

1. Replace the "http" scheme with "https".

2. If the "http" URL explicitly specifies port 80, specify port 443.

3. Do not alter any other aspect of the URL.

This construction is equivalent to Section 8.3 of [HSTS] , point 5.

If an HTTPSSVC record is present for this "https" URL, the client should treat this as the equivalent of receiving an HTTP "307 Temporary Redirect" redirect to the "https" URL. Because HTTPSSVC is received over an often insecure channel (DNS), clients MUST NOT place any more trust in this signal than if they had received a 307 redirect over cleartext HTTP.

If the HTTPSSVC query results in a SERVFAIL error, and the connection between the client and the recursive resolver is cryptographically protected (e.g. using TLS [RFC7858] or HTTPS [RFC8484]), the client SHOULD abandon the connection attempt and display an error message. A SERVFAIL error can occur if the domain is DNSSEC-signed, the recursive resolver is DNSSEC-validating, and an active attacker between the recursive resolver and the authoritative DNS server is attempting to prevent the upgrade to HTTPS.

Similarly, if the client enforces DNSSEC validation on A/AAAA RRs, it SHOULD abandon the connection attempt if the HTTPSSVC RR fails to validate.

4.3. Cache interaction

If the client has an Alt-Svc cache, and a usable Alt-Svc value is present in that cache, then the client SHOULD NOT issue an HTTPSSVC DNS query. Instead, the client SHOULD proceed with alternative service connection as usual.

If the client has a cached Alt-Svc entry that is expiring, the client MAY perform an HTTPSSVC query to refresh the entry.

5. DNS Server Behaviors

Recursive DNS servers SHOULD resolve SvcDomainName records and include them in the Additional Section (along with any relevant CNAME records). For SvcRecordType=0, recursive DNS servers SHOULD attempt to resolve and include A, AAAA, and HTTPSSVC records. For SvcRecordType=1, recursive DNS servers SHOULD attempt to resolve and include A and AAAA records.

Authoritative DNS servers SHOULD return A, AAAA, and HTTPSSVC records (as well as any relevant CNAME records) in the Additional Section for any in-bailiwick SvcDomainNames.

6. Performance optimizations

For optimal performance (i.e. minimum connection setup time), clients SHOULD issue address (AAAA and/or A) and HTTPSSVC queries simultaneously, and SHOULD implement a client-side DNS cache. With these optimizations in place, and conforming DNS servers, using HTTPSSVC does not add network latency to connection setup.

A nonconforming recursive resolver might return an HTTPSSVC response with a nonempty SvcDomainName, without the corresponding address records. If all the HTTPSSVC RRs in the response have nonempty SvcDomainName values, and the client does not have address records for any of these values in its DNS cache, the client SHOULD perform an additional address query for the selected SvcDomainName.

The additional DNS query in this case introduces a delay. To avoid causing a delay for clients using a nonconforming recursive resolver, domain owners SHOULD choose the SvcDomainName to be a name in the origin hostname's CNAME chain if possible. This will ensure that the required address records are already present in the client's DNS cache as part of the responses to the address queries that were issued in parallel.

Highly performance-sensitive clients MAY implement the following special- case shortcut to avoid increased connection time: if (1) one

of the HTTPSSVC records returned has SvcRecordType=0, (2) its SvcDomainName is not in the DNS cache, and (3) the address queries for the origin domain return usable IP addresses, then the client MAY ignore the HTTPSSVC records and connect directly to the origin domain. When the SvcDomainNames and any needed HTTPSSVC records are available, the client SHOULD make subsequent requests over connections specified by the HTTPSSVC records.

Server operators can therefore expect that publishing HTTPSSVC records with SvcRecordType=0 should not cause an additional DNS query for performance-sensitive clients. Server operators who wish to prevent this optimization should use SvcRecordType=1.

7. Extensions to enhance privacy

7.1. Alt-Svc parameter for ESNI keys

An Alt-Svc "esnikeys" parameter is defined for specifying ESNI keys corresponding to an alternative service. The value of the parameter is an ESNIKeys structure [ESNI] encoded in [base64], or the empty string. ESNI-aware clients SHOULD prefer alt-values with nonempty esnikeys.

This parameter MAY also be sent in Alt-Svc HTTP response headers and HTTP/2 ALTSVC frames.

The Alt-Svc specification states that "the client MAY fall back to using the origin" in case of connection failure [AltSvc]. This behavior is not suitable for ESNI, because fallback would negate the privacy benefits of ESNI.

Accordingly, any connection attempt that uses ESNI MUST fall back only to another alt-value that also has the esnikeys parameter. If the parameter's value is the empty string, the client SHOULD connect as it would in the absence of any ESNIKeys information.

For example, suppose a server operator has two alternatives. Alternative A is reliably accessible but does not support ESNI. Alternative B supports ESNI but is not reliably accessible. The server operator could include a full esnikeys value in Alternative B, and mark Alternative A with esnikeys="" to indicate that fallback from B to A is allowed.

7.2. Interaction with other standards

The purpose of this standard is to reduce connection latency and improve user privacy. Server operators implementing this standard SHOULD also implement TLS 1.3 [RFC8446] and OCSP Stapling [RFC6066],

both of which confer substantial performance and privacy benefits when used in combination with HTTPSSVC records.

To realize the greatest privacy benefits, this proposal is intended for use with a privacy-preserving DNS transport (like DNS over TLS [RFC7858] or DNS over HTTPS [RFC8484]). However, performance improvements, and some modest privacy improvements, are possible without the use of those standards.

This RRType could be extended to support schemes other than "https". Any such scheme MUST have an entry under the HTTPSSVC RRType in the IANA DNS Underscore Global Scoped Entry Registry [Attrleaf] The scheme SHOULD have an entry in the IANA URI Schemes Registry [RFC7595]. The scheme SHOULD be one for which Alt-Svc is defined.

8. Security Considerations

Alt-Svc Field Values are intended for distribution over untrusted channels, and clients are REQUIRED to verify that the alternative service is authoritative for the origin (Section 2.1 of [AltSvc]). Therefore, DNSSEC signing and validation are OPTIONAL for publishing and using HTTPSSVC records.

TBD: expand this section in more detail. In particular: * Just as with [AltSvc], clients must validate the TLS server certificate against hostname associated with the origin. Clients MUST NOT use the SvcDomainName as any part of the server TLS certificate validation. * ...

9. IANA Considerations

Per [RFC6895], please add the following entry to the data type range of the Resource Record (RR) TYPEs registry:

TYPE	Meaning	Reference
HTTPSSVC	HTTPS Service Location	(This document)

Per [Attrleaf], please add the following entries to the DNS Underscore Global Scoped Entry Registry:

RR TYPE	_NODE NAME	Meaning	Reference
HTTPSSVC	_https	Alt-Svc for HTTPS	(This document)

Per [AltSvc], please add the following entries to the HTTP Alt-Svc Parameter Registry:

Alt-Svc Parameter	Meaning	Reference
esnikeys	Encrypted SNI keys	(This document)

10. Acknowledgements and Related Proposals

There have been a wide range of proposed solutions over the years to the "CNAME at the Zone Apex" challenge proposed. These include [I-D.draft-bellis-dnsop-http-record-00], [I-D.draft-ietf-dnsop-aname-03], and others.

Thank you to Ian Swett, Ralf Weber, Jon Reed, Martin Thompson, Lucas Pardue, Ilari Liusvaara, and others for their feedback and suggestions on this draft.

11. References

11.1. Normative References

[AltSvc] Nottingham, M., McManus, P., and J. Reschke, "HTTP Alternative Services", RFC 7838, DOI 10.17487/RFC7838, April 2016, <https://www.rfc-editor.org/info/rfc7838>.

[AltSvcSNI] Bishop, M., "The "SNI" Alt-Svc Parameter", draft-bishop-httpbis-sni-altsvc-02 (work in progress), May 2018.

[Attrleaf] Crocker, D., "DNS Scoped Data Through "Underscore" Naming of Attribute Leaves", draft-ietf-dnsop-attrleaf-16 (work in progress), November 2018.

[base64] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[ESNI] Rescorla, E., Oku, K., Sullivan, N., and C. Wood, "Encrypted Server Name Indication for TLS 1.3", draft-ietf-tls-esni-03 (work in progress), March 2019.

[HappyEyeballsV2]
Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <https://www.rfc-editor.org/info/rfc8305>.

[HSTS] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, DOI 10.17487/RFC6797, November 2012, <https://www.rfc-editor.org/info/rfc6797>.

[HTTP3] Bishop, M., "Hypertext Transfer Protocol Version 3 (HTTP/3)", draft-ietf-quic-http-20 (work in progress), April 2019.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997, <https://www.rfc-editor.org/info/rfc2181>.

[RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource Record (RR) Types", RFC 3597, DOI 10.17487/RFC3597, September 2003, <https://www.rfc-editor.org/info/rfc3597>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <https://www.rfc-editor.org/info/rfc6066>.

[RFC6454] Barth, A., "The Web Origin Concept", RFC 6454, DOI 10.17487/RFC6454, December 2011, <https://www.rfc-editor.org/info/rfc6454>.

[RFC7595] Thaler, D., Ed., Hansen, T., and T. Hardie, "Guidelines and Registration Procedures for URI Schemes", BCP 35, RFC 7595, DOI 10.17487/RFC7595, June 2015, <https://www.rfc-editor.org/info/rfc7595>.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://www.rfc-editor.org/info/rfc7858>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

11.2. Informative References

[DNSTerm] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/info/rfc8499>.

[HTTP] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[I-D.draft-bellis-dnsop-http-record-00] Bellis, R., "A DNS Resource Record for HTTP", draft-bellis-dnsop-http-record-00 (work in progress), November 2018.

[I-D.draft-ietf-dnsop-aname-03] Finch, T., Hunt, E., Dijk, P., Eden, A., and W. Mekking, "Address-specific DNS aliases (ANAME)", draft-ietf-dnsop-aname-03 (work in progress), April 2019.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <https://www.rfc-editor.org/info/rfc2782>.

[RFC6895] Eastlake 3rd, D., "Domain Name System (DNS) IANA Considerations", BCP 42, RFC 6895, DOI 10.17487/RFC6895, April 2013, <https://www.rfc-editor.org/info/rfc6895>.

Appendix A. Additional examples

A.1. Equivalence to Alt-Svc records

The following:

```
www.example.com.  2H  IN CNAME   svc.example.net.
example.com.      2H  IN HTTPSSVC 0 0 svc.example.net.
svc.example.net.  2H  IN HTTPSSVC 1 2 svc3.example.net. "h3=\":8003\"; \
                                  esnikeys=\"ABC...\""
svc.example.net.  2H  IN HTTPSSVC 1 3 . "h2=\":8002\"; \
                                  esnikeys=\"123...\""
```

is equivalent to the Alt-Svc record:

```
Alt-Svc: h3="svc3.example.net:8003"; esnikeys="ABC..."; ma=7200, \
         h2="svc.example.net:8002"; esnikeys="123..."; ma=7200
```

for the origins of both "https://www.example.com" and "https://example.com".

Appendix B. Comparison with alternatives

The HTTPSSVC record type closely resembles some existing record types and proposals. A complaint with all of the alternatives is that web clients have seemed unenthusiastic about implementing them. The hope here is that by providing an extensible solution that solves multiple problems we will overcome the inertia and have a path to achieve client implementation.

B.1. Differences from the SRV RRTYPE

An SRV record [RFC2782] can perform a similar function to the HTTPSSVC record, informing a client to look in a different location for a service. However, there are several differences:

- SRV records are typically mandatory, whereas clients will always continue to function correctly without making use of Alt-Svc or HTTPSSVC.
- SRV records cannot instruct the client to switch or upgrade protocols, whereas Alt-Svc can signal such an upgrade (e.g. to HTTP/2).
- SRV records are not extensible, whereas Alt-Svc and thus HTTPSSVC can be extended with new parameters. For example, this is what allows the incorporation of ESNI keys in HTTPSSVC.

- Using SRV records would not allow a client to skip processing of the Alt-Svc information in a subsequent connection, so it does not confer a performance advantage.

B.2. Differences from the proposed HTTP record

Unlike [I-D.draft-bellis-dnsop-http-record-00], this approach is extensible to cover Alt-Svc and ESNIKeys use-cases. Like that proposal, this addresses the zone apex CNAME challenge.

Like that proposal it remains necessary to continue to include address records at the zone apex for legacy clients.

B.3. Differences from the proposed ANAME record

Unlike [I-D.draft-ietf-dnsop-aname-03], this approach is extensible to cover Alt-Svc and ESNIKeys use-cases. This approach also does not require any changes or special handling on either authoritative or master servers, beyond optionally returning in-bailiwick additional records.

Like that proposal, this addresses the zone apex CNAME challenge for clients that implement this.

However with this HTTPSSVC proposal it remains necessary to continue to include address records at the zone apex for legacy clients. If deployment of this standard is successful, the number of legacy clients will fall over time. As the number of legacy clients declines, the operational effort required to serve these users without the benefit of HTTPSSVC indirection should fall. Server operators can easily observe how much traffic reaches this legacy endpoint, and may remove the apex's address records if the observed legacy traffic has fallen to negligible levels.

B.4. Differences from the proposed ESNI record

Unlike [ESNI], this approach is extensible and covers the Alt-Svc case as well as addresses the zone apex CNAME challenge.

By using the Alt-Svc model we also provide a way to solve the ESNI multi-CDN challenges in a general case.

Unlike ESNI, this is focused on the specific case of HTTPS, although this approach could be extended for other protocols. It also allows specifying ESNI keys for a specific port, not an entire host.

B.5. SNI Alt-Svc parameter

Defining an Alt-Svc sni= parameter (such as from [AltSvcSNI]) would have provided some benefits to clients and servers not implementing ESNI, such as for specifying that "_wildcard.example.com" could be sent as an SNI value rather than the full name. There is nothing precluding HTTPSSVC from being used with an sni= parameter if one were to be defined, but it is not included here to reduce scope, complexity, and additional potential security and tracking risks.

Appendix C. Design Considerations and Open Issues

This draft is intended to be a work-in-progress for discussion. Many details are expected to change with subsequent refinement. Some known issues or topics for discussion are listed below.

C.1. Record Name

Naming is hard. The "HTTPSSVC" is proposed as a placeholder. Other names for this record might include ALTSVC, HTTPS, HTTPSSRV, B, or something else.

C.2. Applicability to other schemes

The focus of this record is on optimizing the common case of the "https" scheme. It is worth discussing whether this is a valid assumption or if a more general solution is applicable. Past efforts to over-generalize have not met with broad success.

C.3. Wire Format

Advice from experts in DNS wire format best practices would be greatly appreciated to refine the proposed details, overall.

C.4. Extensibility of SvcRecordType

Only values of "0" and "1" are allowed for SvcRecordType. Should we give more thought to potential future values? The current version tries to leave this open by indicating that resource records with unknown SvcRecordType values should be ignored (and perhaps should be switched to MUST be ignored)?

C.5. Where to include Priority

The SvcFieldPriority could alternately be included as a pri= Alt-Svc attribute. It wouldn't be applicable for Alt-Svc returned via HTTP, but it is also not necessarily needed by DNS servers. It is also not used when SvcRecordType=0. A related question is whether to omit it

from the textual representation when SvcRecordType=0. Regardless, having a series of sequential numeric values in the textual representation has risk of user error, especially as MX, SRV, and others all have their own variations here.

C.6. Whether to include Weight

Some other similar mechanisms such as SRV have a weight in-addition to priority. That is excluded here for simplicity. It could always be added as an optional Alt-Svc attribute.

Appendix D. Change history

- draft-nygren-httpbis-httpssvc-03
  - Change redirect type for HSTS-style behavior from 302 to 307 to reduce ambiguities.
- draft-nygren-httpbis-httpssvc-02
  - Remove the redundant length fields from the wire format.
  - Define a SvcDomainName of "." for SvcRecordType=1 as being the HTTPSSVC RRNAME.
  - Replace "hq" with "h3".
- draft-nygren-httpbis-httpssvc-01
  - Fixes of record name. Replace references to "HTTPSVC" with "HTTPSSVC".
- draft-nygren-httpbis-httpssvc-00
  - Initial version

Authors' Addresses

Ben Schwartz
Google

Email: bemasc@google.com

Mike Bishop
Akamai Technologies

Email: mbishop@evequefou.be

Erik Nygren
Akamai Technologies

Email: erik+ietf@nygren.org