Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2015

E. Nygren
Akamai Technologies

July 3, 2014

Service Binding DNS Records (DNS B)
draft-nygren-service-bindings-00

Abstract

This document describes a DNS "B" RR which binds together information needed to establish connection to a service across multiple protocol layers, including the location of the server, the application-level protocol, and security bootstrap information.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 4, 2015.

Table of Contents

1. Overview and rationale

As the protocols underlying services evolve to provide enhanced performance and security properties, clients have an increasing number of ways to contact any given service on a domain. Contacting any given implementation instance of a service on a domain may require parameters across multiple protocol layers. Different servers with different address records may even desire to implement the same service, especially in the case where older protocols lacked functionality required to make a smooth transition to newer protocols.

The DNS B RR allows administrators to bind together the parameters needed to contact an instance of a service on a domain into a single record, or "service binding". Multiple service bindings (multiple B records in an RRset) may provide a client with multiple ways to contact a given service, each with its own associated protocol(s) and related parameters.

Service bindings give clients a single DNS query [RFC1035] to resolve to obtain the B RRset for a service and domain. Clients are able to then filter B RRs based on which protocols they support. After selecting an B RR within this RRset, it should then be clear to the client whether other RRs need to be resolved prior to establishing communication with the service for a given protocol.

1.1. Relationship to SRV RR type

The B RR has close similarities of purpose to the SRV RR [RFC2782] but the B RR covers additional use-cases, including:

- Multiple application-level protocol implementations for a service
- Providing information for improving the security of a connection to a service (such as constraints on server certificates as well as handshake encryption keys)
- A single RRset name that can be queried that covers multiple lower-level protocol implementations for a service, such as when the same service is offered over multiple transport-layer protocols

Similar to SRV, the B RR provides:

- Multiple address records for a given domain
- Priorities and weights to guide client selection
- The information needed to contact a given server (such as target hostname and port number) is bound together in a single record

1.2. Introductory example

If a web browser supporting B records wishes to fetch the URL https://www.example.com/, it would use the URI scheme ("https") as the the service and perform a lookup of:

QNAME=_https._b.www.example.com, QCLASS=IN, QTYPE=B

which might return an RRset with multiple resource records:

```
1 0 server-experimental.example.com.
     { "a": "h2", "t": "fictionfast", "tp": 9443 }

3 0 server-primary.example.com.
     { "a": "h2", "t": "tcp", "tp": 443,
       "dane": "_443._tcp.server-primary-www-cert.example.com.",
       "tlshp": ["JgxJkCv0va0", "OBZVN8LPo+SB9eQ/"] }

5 0 server-legacy.example.com.
     { "a": "http/1.1", "t": "tcp", "tp": 443,
       "dane": "_443._tcp.server-legacy-www-cert.example.com" }
```

The first of these specifies using HTTP/2 over a fictional experimental secure transport protocol "fictionfast" over UDP port 9443 from the server(s) with address record server-experimental.example.com.

For clients not supporting this first item, HTTP/2 [I-D.ietf-httpbis-http2] is available over TLS via TCP port 443 with certificate information available from a DANE record at "_443._tcp.server-primary-www-cert.example.com." and with TLS 1.3 server handshake encryption parameters specified [I-D.rescorla-tls13-new-flows] from server(s) with the server-primary.example.com address record.

For clients not supporting HTTP/2, a separate set of legacy servers is available for HTTP/1.1 which happens to have different certificate information available from a separate DANE record.

NOTE: HTTPS is selected above for illustrative purposes only, and the HTTPS examples within this document should not be considered to be a definitive statement on the way for HTTPS to use B records.

2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Applicability Statement

In general, it is expected that service binding records will be used by clients for applications where the relevant protocol specification indicates that clients should use the B record. Such specification MUST define the symbolic name to be used in the Service field of the B record as described below. Such specification MUST also define the Parameters available as well as their interpretation. It also MUST

include security considerations. Service binding records SHOULD NOT be used in the absence of such specification.

The current version of this proposal includes specifics for how service bindings might be used in the context of HTTP/2 as well as TLS/1.3. It is expected that those would be split out to a separate draft.

4. The Service Binding Record

The service binding record is of the format:

_Service._b.Name TTL Class B Priority Weight Target Parameters

where:

Service The symbolic name of the desired service.
An underscore (_) is prepended to the service identifier to avoid collisions with DNS labels that occur in nature. In many specifications it is expected that the Service is likely to be the same as the URI Scheme [RFC3986] (such as "http" or "https").

_b The literal label "_b", intended to prevent collisions with DNS labels that occur in nature.

Name The domain this RR refers to. Similar to SRV RR [RFC2782], the name one searches for is not this name.

TTL Standard DNS meaning [RFC1035].

Class Standard DNS meaning [RFC1035]. B records occur in the IN Class.

Priority The priority of this service binding. After filtering service bindings for protocols that it supports, the client MUST attempt to contact the target host with the lowest-numbered priority it can reach; target hosts with the same priority SHOULD be tried in an order defined by the weight field. The range is 0-65535. This is a 16 bit unsigned integer in network byte order.

Weight A server selection mechanism. The weight field specifies a relative weight for entries with the same priority. Larger weights SHOULD be given a proportionately higher probability of being selected. The range of this number is 0-65535. This is a 16 bit unsigned integer in network byte order. Domain administrators SHOULD use Weight 0 when there isn't any server selection to do, to make the RR easier to read for humans (less noisy). In the presence of records containing weights greater

than 0, records with weight 0 should have a very small chance of being selected. Clients SHOULD use the same weight selection algorithm as described in [RFC2782].

Target The domain name of the target host. There MUST be one or more address records for this name (A RR's and/or AAAA RR's, or their most modern equivalent). The name MAY be a CNAME alias (in the sense of [RFC1034]). Implementors are encouraged, but not required, to return the address record(s) in the Additional Data section where possible and appropriate. Unless and until permitted by future standards action, name compression is not to be used for this field.

The special target value of "." indicates that this service is not available on this domain. When used, the RRset MUST contain only this single record. However, the record MAY have additional parameters (not yet defined), such as to specify that an alternate service should be used instead.

Parameters A collection of tag:value parameters specifying additional attributes for this service binding. These are encoded in canonicalized CBOR [RFC7049] using the map data type. Their textual representation is JSON, with binary byte arrays (such as literal keys) being base64 [RFC4648] encoded. While some tags are defined here, additional tags may be defined in the specifications for the particular Service, as well as in specifications for additional protocols for implementing the Service.

4.1. B record RDATA encoding

A more formal definition of the RDATA encoding and parameter textual representation syntax will be included in a future version of this draft once the core concepts have stabilized.

4.2. Service Binding Parameters

The parameters of the B record provide extensibility, allowing additional parameters to be specified depending on the particular service and protocol.

Clients MUST use parameters to filter out service bindings specifying protocols or other parameters that they do not support, as specified by those parameters.

Any parameters not specified in the initial specification of Service Binding usage for a given Service and protocol combination must be capable of being safely ignored by a client.

Some parameters MUST NOT be used by a client unless all relevant DNS records can be securely validated, such as with DNSSEC [RFC2535]. This means that both the B record, any aliases leading to it, and any aliases and records leading from names specified in that parameter must be validated. This is discussed in more length in Security Considerations [1].

The specification for any given parameter SHOULD include:

- The tag used to identify the parameter
- The valid type and allowed value ranges for the parameter
- Whether the parameter is optional, and its default value if not
- What is the behavior if the DNS records can not be securely validated
- A description of how the client should use the parameter value

4.3. Standard Service Binding Parameters

Application Layer Protocol (tag "a") A string representing the application layer protocol to be used when contacting the service. The valid values are service-dependent and should be the same values as used in ALPN [I-D.ietf-tls-applayerprotoneg]. For example, "h2" represents HTTP/2.

If not specified, the default application-layer protocol for the given service is assumed.

Clients MUST filter out service bindings utilizing application layer protocols that they do not support or recognize, as well as to use the proper application protocol when contacting servers using this binding.

Transport Layer Protocol (tag "t") A string representing the transport layer protocol to be used when contacting the service. The valid values are service dependent. For example, "tcp" is likely to be used for many services.

If not specified, the default transport-layer protocol for the given service is assumed.

Clients MUST filter out service bindings utilizing transport layer protocols that they do not support or recognize, as well as to use the proper transport protocol when contacting servers using this binding.

Note for discussion: it may make sense to omit this as a separate parameter and to have the Application Layer Protocol tag describe the entire stack, as in [I-D.ietf-httpbis-http2].

Transport Layer Port (tag "tp") An integer specifying the transport layer protocol port to be used when contacting the service. The valid values are transport layer protocol dependent.

If not specified, the default port for the given service and transport protocol is assumed.

4.4. Optional Security Parameters

The following parameters should likely move to separate specifications:

DANE Certificate Controls (tag "dane") The value of this optional parameter is a DNS domain name pointing to a DANE TLSA [RFC6698] record associated with this service binding.

This is particularly useful to bind TLSA records with specific servers, especially in the case where different servers have different certificates and perhaps even different operators.

This is also useful to indicate that TLSA records are available (to reduce the need for speculative DNS lookups)

If relevant DNS records can not be securely validated, clients MUST NOT loosen their security policies based on the TLSA record. However, clients MAY use the record to apply more restrictive policies even if the TLSA record could not be securely validated.

In particular, if this parameter is present, a client SHOULD resolve the named TLSA record and use it to constrain the TLS server certificates that it will accept. If the the relevant DNS records can not be securely validated, the TLSA record must only be used in addition to policies already enforced by the client. For example, if a "CA Constraint" certificate usage is specified, then this SHOULD limit the CA allowed to those specified in the TLSA record(s), but only if the CA is also in a trust chain that the client would already accept.

TLS Handshake Parameters (tag "tlshp") This optional parameter specifies ServerParameters for bootstrapping the TLS 1.3 handshake, such as for encrypting the SNI and other parts of the ClientHello to make them less vulnerable to passive eavesdropping [I-D.rescorla-tls13-new-flows].

The value of this parameter is a list containing the binary values of the ServerParameter label followed by the ServerDHParams or ServerECDHParams.

Clients MAY use these values even if the B record is not securely validated, as the intent of these parameters is to protect against passive eavesdropping. Note that there may be limited value in handshake encryption unless DNS lookups are also encrypted.

4.5. Examples for other possible future Parameters

Some of these may also make sense to include in an future version of this draft:

- A parameter indicating that this DNS record must have been securely validated (such as with DNSSEC) or must otherwise be skipped. This could allow for incremental deployment of DANE-managed certificates on some alternate set of servers even without universal DNSSEC adoption.

- Minimum and/or maximum version of a protocol supported. (For example, this could be helpful to mitigate downgrade attacks.)

- Which extensions to a protocol are supported, allowing a client to opportunistically send them in its handshake.

- Hints as to whether a target address record has A and/or AAAA records. (Careful consideration would need to be given to how this played with DNS64 as well as other transition technologies.) A reasonable compromise would be to have a hint indicating that a AAAA record was available and encouraged (such that clients might wait longer to get a response from nameservers) as well as a separate hint indicating that no A record is available.

- Selected Service Binding Indicator - a label that can be passed from client to server indicating which service binding it selected. This may be helpful for both load reporting/feedback, and diagnostics.

- HSTS-style [RFC6797] indicator parameter. This would be returned along with a target of "." on a less secure service to indicate that a more secure scheme/service should be used (such as "https" instead of "http"). This might be a boolean "sts=1" parameter, such as with a record like:

```
_http._b.www.example.com 2D IN B 0 0 . { "sts": 1 }
```

5. Selecting a Service Binding to Use

When a client supporting Service Bindings wishes to make a connection to a service, it SHOULD query the B records for the service. It MAY choose to opportunistically also issue DNS queries for the address records (eg, A and AAAA) to reduce the latency for the case where no B record is available.

5.1. Handling B record responses

In the case where an RRset for B records is returned, the client should:

1. Filter out any B RR's from the RRset which it does not support. In particular, it MUST ignore any B records containing an Application Layer Protocol that is unknown, unsupported, or explicitly disallowed for the particular service.

2. The remaining B RR's should be sorted by priority. An entry should be selected from the top priority (lowest numeric priority value). It multiple records have the same priority, the weighting algorithm specified in [RFC2782] SHOULD be used to order them.

3. The client should attempt to contact the service using the parameters specified in the first selected record from the sorted list.

4. If a given attempt fails, the client MAY attempt to contact the service using the next record in the sorted list. After some number of tries, the client MAY attempt to contact the service directly using the address records for the domain.

5.2. Handling a lack of Service Binding records

Not all clients will immediately implement Service Binding records for any given Service, and administrators may not always put Service Binding records in-place. Just as importantly, experiments have shown that new DNS RR types take awhile before they are accessible to most clients. (_Reference needed_)

As such, client SHOULD directly lookup and use the address records for the domain name when no valid B record is unavailable. In this case, the default application and transport layer protocols SHOULD be used. Clients and Servers MAY then use inline upgrade or signaling mechanisms such as AltSvc [I-D.ietf-httpbis-alt-svc] or ALPN [I-D.ietf-tls-applayerprotoneg] for upgrading to newer protocols instantiating the Service.

6. Operational Examples

A few illustrative examples follow to help to demonstrate how Service Binding records might be used. More examples may be added in a future version of this draft. (_In particular, an example should be added on how a client might resolve and use some specific B records.)

6.1. Example: Moving a service and domain between operators

A common deployment model is for different administrators (and sometimes entirely separate organizations) to operate the DNS for a domain than those that operate a service on the domain. Either or both might even be out-sourced to separate entities such as a hosting/infrastructure provider or Content Delivery Network (CDN). It is critical that it be possible to move domains and services between operators and administrators without any disruption in service and with minimal coordination between the different organizations. Service Binding records simplify this by allowing a delegation of name via a single DNS alias per service/domain. All other properties (such as TLSA records) then follow directly along from this.

For example with:

```
_https._b.www.example.com.
         6H IN CNAME _https._b.www.example.com.example-1.net.

_https._b.www.example.com.example-1.net.
         6H IN B 0 0 server-primary.example-1.net.
         { "a": "h2",
         "dane": "_443._tcp.server-primary-www-cert.example-1.net.",
         "tlshp": ["JgxJkCv0va0", "OBZVN8LPo+SB9eQ/"] }
```

the administrator of the example.com DNS zone delegated control over www.example.com to an operator example-1.net.

In the example above, it would be recommended for the example-1.net authorities to also return ADDITIONAL records for the "server-primary.example-1.net." address records and for the "_443._tcp.server-primary-www-cert.example-1.net." TLSA record along with the B record response.

The DANE TLSA records and TLS 1.3 handshake parameters and supported Application Layer Protocols are bound in this case to the "server-primary.example-1.net." target as they should be. If the administrator of example.com were to move the "www.example.com." CNAME to point to a different operator (eg, "example-2.net"), that

second operator would provide their own service bindings which clients would use as a unit.

The value of this Service Binding approach can be seen here when contrasting against having multiple independent records, each with their own TTL. For example, if "www.example.com" is a CNAME to address records on example-1.net and "_443._tcp.www.example.com" is a CNAME to TLSA records on example-1.net, there is no way to change either of these CNAMEs to point to example-2.net without a situation where some clients are getting address records for example-1 and TLSA records for example-2, resulting in a possible denial-of-service.

7. Open Areas for Discussion

As this is a early version of this draft, a number of design choices are still open for active discussion:

- What encoding should be used for the Parameters? Some options include:
  - CBOR [RFC7049] - has the advantage of allowing new parameters to be added with self-describing types in a way that unknown new parameters can be ignored.
  - Something ad-hoc, such as the tag-value system of DKIM [RFC6376] section 3.2. This might be more compact, but also ends up being more ad-hoc.
  - What textual representation should be used? JSON, or something more in-line with other DNS record types? The illustrative examples here use JSON for the time-being.
- Should there be a more formal general definition of the transport vs protocol layering and filtering? It may be preferable to have a single identifier (ALPN) specifying the full stack.
  - Which layers should be included? More or less?
  - How does TLS fit in?
  - What more examples should we give?
- How much do we need to worry about packet size limitations, and what can we do about them?
  - Should we do any form of name compression?
  - Should we allow names (such as the DANE names) to be relative?

- Should the TLS 1.3 handshake parameters move to their own record that is just named from the B record?

- What should be the name of the Resource Record type: is "B" a good name, or should something longer such as "SB" or "SBND" be used?

- What should be a standard part of the record and what should be parameters? Everything could be a parameter, including priority and target and weight. In the other direction, some existing parameters could be standard. There may be a reasonable balance in the current proposal.

- There is an operational risk that the B record and normal address records (A and AAAA) could diverge over time due to administrative management skew. For a CDN or hosting provider, taking advantage of B records also means that the content provider would need to CNAME over an additional records per-service.

- We may want to give examples of additional use-cases where this is helpful:

  1. Apex zone domain names - can be used instead of a CNAME to delegate over to a CDN or hosting provider

  2. Dealing with the SNI challenge for TLS: specify that clients using a Service Binding MUST send TLS SNI. This would allow the use of a larger set of SNI-only servers for the B record targets than for the normal address records on the domain,

- Should there be a parameter to indication which service binding was used when making a connection? In particular, a parameter which is a label or string that gets passed from the client to the server (such as a response header or as "Indicated Service")?

- Why a separate domain name (eg, "_http._b.www.example.com") rather than just a record on "www.example.com" (as in the DANISH proposal)?

  * The reason is that you _do_ want a separate name-per-service, as otherwise the RRset will get too big if you have multiple names.

  * Is the "_b" actually needed? Or could this just be "_http.www.example.com" ?

- Which of the additional parameters above should we include in subsequent versions of this draft?

8. IANA Considerations

The IANA will need to assign an RR type value to the B RR.

The IANA will also need to maintain a registry of Parameters allowed B RRs. (Details on this will need to be expanded in a future version of this draft.)

9. Security Considerations

Service Bindings have many of the same security issues as SRV records or most other DNS record types (such as A and AAAA address records). In particular, if the client is not securely validating DNS resolutions then a man-in-the-middle attacker could trick a client into contacting an attacker selected server and port and protocol. Similar attacks could also force a client to downgrade to a less secure protocol or to fall back to using address records.

Clients MAY chose to bound the TTLs of B RRsets to a reasonable value and/or forget B RRsets on network changes to limit the damage that can be done by such a man-in-the-middle.

For each service binding parameter type, specific attention must be paid to the security considerations relevant to it. In particular:

Application Layer Protocol Care must be taken to disallow and filter out any invalid ALPN and Service combinations. For example, clients MUST NOT allow ALPN "h2c" in combination with Service "https".

DANE Certificate Controls DANE TLSA records can be used to either restrict the set of allowed server certificates to a subset of those that a client would normally allow. They can also be used to specify alternate certificates or trust roots to use for validation, relying on DNSSEC instead of the CA hierarchy.

If a man-in-the-middle attacker can inject DNS responses (and the client is not securely validating the responses), then in the former case the attacker can only force a denial-of-service by causing the client to fail its validation. Allowing this may be an acceptable trade-off to allow administrators to limit the scope of allowed certificates even for clients not performing validation. As such, clients MAY use B and TLSA records to constrain allowed certificates to a strict subset of those that would otherwise be allowed.

However, in the former case (where alternate certificates that are not ones a client would already trust can be specified), an

attacker or unscrupulous DNS resolver operator could utilize this to force a client to trust an adversary-controlled server. As such, clients MUST NOT use DANE TLSA records unless the B RRset, the TLSA RRset, all aliases (eg, CNAMEs) pointing to them, and all authorities of these can be securely validated.

(_Additional security considerations will need to be added to a future version of this draft._)

10. Acknowledgments

This draws heavily on many previous DNS RFCs such as [RFC2782].

Discussions with Eric Rescorla, Daniel Kahn Gilmore, Mark Nottingham, and others on the TLS and HTTPBIS working groups have heavily influenced this proposal. Suggestions from Richard Barnes and others have also been incorporated into this draft.

11. References

11.1. Normative References

[I-D.ietf-tls-applayerprotoneg]
Friedl, S., Popov, A., Langley, A., and S. Emile, "Transport Layer Security (TLS) Application Layer Protocol Negotiation Extension", draft-ietf-tls-applayerprotoneg-05 (work in progress), March 2014.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2535] Eastlake, D., "Domain Name System Security Extensions", RFC 2535, March 1999.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

[RFC6797] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, November 2012.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, October 2013.

11.2. Informative References

[I-D.ietf-httpbis-alt-svc]
Nottingham, M., McManus, P., and J. Reschke, "HTTP Alternative Services", draft-ietf-httpbis-alt-svc-01 (work in progress), April 2014.

[I-D.ietf-httpbis-http2]
Belshe, M., Peon, R., and M. Thomson, "Hypertext Transfer Protocol version 2", draft-ietf-httpbis-http2-13 (work in progress), June 2014.

[I-D.rescorla-tls13-new-flows]
Rescorla, E., "New Handshake Flows for TLS 1.3", draft-rescorla-tls13-new-flows-01 (work in progress), February 2014.

[RFC6376] Crocker, D., Hansen, T., and M. Kucherawy, "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, September 2011.

11.3. URIs

[1] #security

Author's Address

Erik Nygren
Akamai Technologies

Email: erik+ietf@nygren.org
URI: http://erik.nygren.org/