XEP-0487: Host Meta 2 - One Method To Rule Them All

Travis Burtrum
mailto:travis@burtrum.org
xmpp:travis@burtrum.org
https://moparisthebest.com/

2024-03-10
Version 0.1.0

Status	Type	Short Name
Experimental	Standards Track	connections-v2

This document defines an XMPP Extension Protocol for extending XEP-0156 by modifying the JSON Web Host Metadata Link format to support discovering all possible XMPP connection methods, for c2s and s2s

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction

Although XMPP Core [1] specifies the use of TCP as the method of connecting to an XMPP server, alternative connection methods exist, including the BOSH (XEP-0124) [2] method (for which XMPP Over BOSH (XEP-0206) [3] is the XMPP profile), the websocket subprotocol specified in RFC 7395 [4], SRV records for XMPP over TLS (XEP-0368) [5], XMPP over QUIC (XEP-0467) [6], and WebSocket S2S (XEP-0468) [7], and surely others that don't yet exist. For some of these methods, it is necessary to discover further parameters before connecting, such as the HTTPS URL of a BOSH or WebSocket request. Without ways to auto-discover these parameters, the relevant information would need to be provided manually by a human user (which is cumbersome and error-prone) or hard-coded into XMPP software applications (which is brittle and not interoperable)).
Additional things also require automatic discovery, like RFC 7711 [8] (replaced here by pinning public keys instead like RFC 7469 [9]), TLS Encrypted Client Hello [10], SNI names, and ALPN protocols.
This document defines a way to encapsulate information about all these connection methods and parameters for auto-discovery via Link entries in a server's "host-meta.json" file. It also provides a flag to signal to the client or server that all info is here and no other methods need be used.

2 HTTPS Lookup Method

2.1 Link Format

The HTTPS lookup method uses Web Host Metadata RFC 6415 [11] to categorize and list the URIs of alternative connection methods. It is intended to replace all current methods for looking up connection information by "native" clients and servers, as well as be used by web browsers.
Each alternative connection method is specified in the host-meta.json (JRD) file using a distinctive link relation RFC 5988 [12]. This specification defines several extension relation types, here links are provided to their respective transport definitions:

[1] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[2] XEP-0124: Bidirectional-streams Over Synchronous HTTP <https://xmpp.org/extensions/xep-0124.html>.
[3] XEP-0206: XMPP Over BOSH <https://xmpp.org/extensions/xep-0206.html>.
[4] RFC 7395: An Extensible Messaging and Presence Protocol (XMPP) Subprotocol for WebSocket <http://tools.ietf.org/html/rfc7395>.
[5] XEP-0368: SRV records for XMPP over TLS <https://xmpp.org/extensions/xep-0368.html>.
[6] XEP-0467: XMPP over QUIC <https://xmpp.org/extensions/xep-0467.html>.
[7] XEP-0468: WebSocket S2S <https://xmpp.org/extensions/xep-0468.html>.
[8] RFC 7711: PKIX over Secure HTTP (POSH)<http://tools.ietf.org/html/rfc7711>.
[9] RFC 7469: Public Key Pinning Extension for HTTP <http://tools.ietf.org/html/rfc7469>.
[10] TLS Encrypted Client Hello <http://tools.ietf.org/html/draft-ietf-tls-esni/>.
[11] RFC 6415: Web Host Metadata <http://tools.ietf.org/html/rfc6415>.
[12] RFC 5988: Web Linking <http://tools.ietf.org/html/rfc5988>.

- urn:xmpp:alt-connections:tls - c2s Direct TLS - SRV records for XMPP over TLS (XEP-0368) [13]
- urn:xmpp:alt-connections:quic - c2s Quic - XMPP over QUIC (XEP-0467) [14]
- urn:xmpp:alt-connections:s2s-websocket - s2s WebSocket - WebSocket S2S (XEP-0468) [15]
- urn:xmpp:alt-connections:s2s-tls - s2s Direct TLS - SRV records for XMPP over TLS (XEP-0368) [16]
- urn:xmpp:alt-connections:s2s-quic - s2s Quic - XMPP over QUIC (XEP-0467) [17]

And additionally re-uses some defined in Discovering Alternative XMPP Connection Methods (XEP-0156) [18]:

- urn:xmpp:alt-connections:websocket - c2s WebSocket - RFC 7395 [19]
- urn:xmpp:alt-connections:xbosh - c2s BOSH - XMPP Over BOSH (XEP-0206) [20]

Additionally a top level ”xmpp” object is defined, which currently has the following subfields defined:

- ”ttl” - integer - MANDATORY - seconds this document can be cached for
- ”public-key-pins-sha-256” - list of strings - OPTIONAL - base64 sha256 hashes of public keys to trust, use as defined in RFC 7469 [21]

The following are new fields defined in each link object:

- ”priority” and ”weight” - integers - MANDATORY - Use as defined in RFC 2782 [22]
- ”sni” - string - MANDATORY - name to send in the TLS/QUIC SNI extension
- ”ech” - string - OPTIONAL - Use as defined in TLS Encrypted Client Hello [23]

[13] XEP-0368: SRV records for XMPP over TLS <https://xmpp.org/extensions/xep-0368.html>.
[14] XEP-0467: XMPP over QUIC <https://xmpp.org/extensions/xep-0467.html>.
[15] XEP-0468: WebSocket S2S <https://xmpp.org/extensions/xep-0468.html>.
[16] XEP-0368: SRV records for XMPP over TLS <https://xmpp.org/extensions/xep-0368.html>.
[17] XEP-0467: XMPP over QUIC <https://xmpp.org/extensions/xep-0467.html>.
[18] XEP-0156: Discovering Alternative XMPP Connection Methods <https://xmpp.org/extensions/xep-0156.html>.
[19] RFC 7395: An Extensible Messaging and Presence Protocol (XMPP) Subprotocol for WebSocket <http://tools.ietf.org/html/rfc7395>.
[20] XEP-0206: XMPP Over BOSH <https://xmpp.org/extensions/xep-0206.html>.
[21] RFC 7469: Public Key Pinning Extension for HTTP <http://tools.ietf.org/html/rfc7469>.
[22] RFC 2782: A DNS RR for specifying the location of services (DNS SRV) <http://tools.ietf.org/html/rfc2782>.
[23] TLS Encrypted Client Hello <http://tools.ietf.org/html/draft-ietf-tls-esni/>.

- ”ips” - list of strings - at least one MANDATORY - IPv4 or IPv6 addresses only, connect to these

The ”href” field in websocket/bosh links remains unchanged from XEP-0156, but is replaced by ”port” (integer) in Direct TLS/Quic connections.

2.2 Business Rules

The following business rules apply:

1. host-meta files MUST be fetched only over HTTPS, and MUST only use secure connections (TLS or equivalent). This provides secure delegation, meaning you MUST send the provided SNI and validate that the certificate is valid for that host *or* the XMPP domain (or the public key hash is pinned).
2. host-meta responses with the top level ”xmpp” object mean this XEP is in use and legacy SRV/POSH/etc lookups SHOULD be skipped, alternatively if the top level ”xmpp” object does not exist, XEP-0156 rules apply instead.
3. Client/Server implementations SHOULD consider weight/priority as presumably the server admin has thought about which links can handle load best etc, but MAY prioritize certain protocols over others, for example a privacy client may want to use websocket to look most like HTTPS, or a mobile client might prefer Quic for connection roaming. Regardless server operators MUST NOT count on any ordering, a client can connect to any of these under even normal circumstances.

2.3 Examples

It is possible to use additionally a JSON-based format for host-meta information. The JSON representation of the host metadata is named JRD and specified in Appendix A of RFC 6415 [24]. The above XRD example would be presented in JRD as:

Listing 1: Result for /.well-known/host-meta.json

```
{
  "xmpp": {
    "ttl": 3000,
    "public-key-pins-sha-256": [
      "4/mggdlVx8A3pvHAWW5sD+qJyMtUHgiRuPjVC48N0XQ="
    ]
  },
  "links": [
    {
```

[24] RFC 6415: Web Host Metadata <http://tools.ietf.org/html/rfc6415>.

```
    "rel": "urn:xmpp:alt-connections:websocket",
    "href": "wss://other.example.org/xmpp-websocket",
    "ips": [
      "1.2.3.4",
      "fd00:feed:dad:beef::1"
    ],
    "priority": 15,
    "weight": 50,
    "sni": "example.org",
    "ech": "eG1wcC1jbGllbnQ="
  },
  {
    "rel": "urn:xmpp:alt-connections:tls",
    "port": 443,
    "ips": [
      "1.2.3.4",
      "fd00:feed:dad:beef::1"
    ],
    "priority": 10,
    "weight": 50,
    "sni": "example.org",
    "ech": "eG1wcC1jbGllbnQ="
  },
  {
    "rel": "urn:xmpp:alt-connections:quic",
    "port": 443,
    "ips": [
      "1.2.3.4",
      "fd00:feed:dad:beef::1"
    ],
    "priority": 5,
    "weight": 50,
    "sni": "example.org",
    "ech": "eG1wcC1jbGllbnQ="
  },
  {
    "rel": "urn:xmpp:alt-connections:s2s-websocket",
    "href": "wss://other.example.org/s2s-xmpp-websocket",
    "ips": [
      "1.2.3.4",
      "fd00:feed:dad:beef::1"
    ],
    "priority": 15,
    "weight": 50,
    "sni": "example.org",
    "ech": "eG1wcC1jbGllbnQ="
  },
  {
    "rel": "urn:xmpp:alt-connections:s2s-tls",
```

```
      "port": 443,
      "ips": [
        "1.2.3.4",
        "fd00:feed:dad:beef::1"
      ],
      "priority": 10,
      "weight": 50,
      "sni": "example.org",
      "ech": "eG1wcC1jbGllbnQ="
    },
    {
      "rel": "urn:xmpp:alt-connections:s2s-quic",
      "port": 443,
      "ips": [
        "1.2.3.4",
        "fd00:feed:dad:beef::1"
      ],
      "priority": 5,
      "weight": 50,
      "sni": "example.org",
      "ech": "eG1wcC1jbGllbnQ="
    },
    {
      "rel": "urn:xmpp:alt-connections:xbosh",
      "href": "https://web.example.com:5280/bosh"
    }
  ]
}
```

3 Implementation Notes

3.1 For Server Administrators

For the forseeable future you will need to maintain legacy SRV records in addition to this file, and you should provide DANE TLSA records too if possible.
To make your server as accessible to other clients/servers no matter how bad the network they are on, it is advised to use port 443 when possible, as it looks the most like HTTPS.
Extra care must be taken in updating "public-key-pins-sha-256" similar to that which is required of HPKP and DANE, summarized here, you MUST add the new key to the file, continue using the old key until least 2 TTL periods have passed, and only then remove the old key from the file and start using the new key.
To make connection discovery work in web clients (including those hosted on a different domain) the host service SHOULD set appropriate CORS headers for Web Host Metadata files. The exact headers and values are out of scope of this document but may include: *Access-Control-Allow-Origin*, *Access-Control-Allow-Methods* and *Access-Control-Allow-Headers*.
Due care has to be exercised in limiting the scope of *Access-Control-Allow-Origin* response

header to Web Host Metadata files only.

```
Access-Control-Allow-Origin: *
```

Access-Control-Allow-Origin header with a value of * allows JavaScript code running on a different domain to read the content of Web Host Metadata files. Special value * ensures that the request will only succeed if it is invoked without user credentials (e.g. cookies, HTTP authentication).

3.2 For Programmers

As an author of a client or server, you presumably have users, and those users have a single desire, to communicate over XMPP. This means that they want to connect at any costs, they *do not* want to see the first error to appear and have all further attempts aborted until it's fixed. With this problem statement, here is a list of current best practices to make this happen:

- "ttl" is a guideline for when you SHOULD try to fetch a newer copy, if you can't, you MUST try to fetch connection info other ways, and also fall back to expired data if needed
- When trying a particular connection with a list of IPs, you can try them all in order, or pick a random one like DNS would do, but if you do, you MUST keep track of the ones you tried, and try all the rest later if your connection wasn't succesful
- An application MUST keep trying every possible connection until they have both an authenticated stream (the TLS certificate is valid) and have seen a valid XMPP stream element for the server you are trying to connect to. You may get an authenticated stream and an HTTP or IMAP or SMTP error, or even an XMPP stream header for another server, you MUST keep going until both of these are satisified.

3.3 For Future Spec Writers

Keep in mind this json file is defined in an RFC and we need to keep backwards compatibility with it, software only implementing XEP-0156 should be able to read and use this file as extended by this XEP only seeing the websocket/bosh connections.

4 Rationale

At the time of this writing, connecting to an XMPP server requires at least 5 seperate fetches of data, and doesn't support half the things this XEP does. I don't find adding more fetches a sustainable path forward, hence defininig one extensible method for all things going forward.
Here I will go through alternative solutions that were explored and explain their deficiencies

and why they were not chosen.

4.1 Why not DNS?

- SRV records cannot hold the additional parameters needed and are not extensible.
- The web solves these problems in HTTP3 by introducing another set of DNS records (RFC 9460 [25]) but that poses a problem for XMPP because we don't have the clout to introduce our own DNS records and hope for any sort of adoption with the myriad of DNS setup panels most people use.
- We could register our own SvcParamKeys and use SVCB records, but that suffers the same problem as above.
- Perhaps most importantly, while we all hope and pray for DNSSEC (and DANE), many TLDs don't yet support it, and we can't trust this info (pinned keys or secure delegation) without cryptographic authenticity.
- Some of these parameters not even the web trusts over plaintext, DNS-over-TLS/HTTPS is a requirement for ECH.
- TXT records can hold arbitrary data, but the authenticity/privacy problems above persist, practical size limits make this a non-starter, and writing a custom parser isn't great from a security perspective.

4.2 Why not host-meta.xml?

- XMPP uses XML right? So we already have a XML parser to use! Actually no, XMPP uses a (very) strict subset of XML, and an XML parser properly configured to parse XMPP *will not* parse host-meta.xml. Some more security conscious XML parsers are explicitly built for XMPP and cannot be configured to parse host-meta.xml. It is *dangerous* from a security perspective to have an XML parser capable of parsing host-meta.xml anywhere in your client/server, see CVE-2022-0217.
- Extending host-meta.xml the ideal way where things that can only have 1 value per link and should be an attribute (like port) would require namespaced attributes which are not widely supported or currently used at all in XMPP (though legal).
- It would also require dummy values in Link like href='DUMMY' or so when a port is set. Or introducing another tag... Regardless it won't be very clean, and I consider the first issue a complete deal breaker.

[25] RFC 9460: Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records) <http://tools.ietf.org/html/rfc9460>.

4.3 Why host-meta.json instead of something else?

- A major design goal here is enabling 1 single fetch to get everything needed to connect to a server. This file already exists and is already being grabbed by clients, since we can extend it without breaking compatibility with those clients, this doesn't introduce another network fetch.
- RFC 7711 [26] already exists for clients and servers and uses JSON, same parser can be used.
- As stated in the section above, I find a JSON parser much less of a potential security vulnerability to have available than an XML parser capable of parsing host-meta.xml

4.4 Why do we even need these additional parameters?

- SNI fixes a fundamental problem we have with current methods of secure delegation (XEP-0156 and DNSSEC signed SRV records), in that these enable certificate validation to allow a certificate if it contains either one of two names, the service name *or* the target name. But you can only send one name with SNI, and nothing indicates which to send! If you send the wrong one the connection will fail, this necessitates sending one and, if it fails, sending the other, which is of course less than ideal.
- ECH allows encrypting SNI+ALPN on the wire and so is invaluable for privacy.
- Pinned public keys are a better replacement for POSH (they don't need changed on certificate renewal which commonly happens every 60 days now), and an alternative to DANE which can't be used where DNSSEC is not available.

4.5 Misc

- host-meta.json has an 'expires' key defined already, why 'ttl' instead? I was almost convinced to use this until I realized this means that the file needs served by a program that can update 'expires' dynamically on fetch, or at least on a schedule, and that is far more complicated and therefore less preferable than a simple ttl that never needs updated allowing clients to keep track of their own expiry.
- What would be the next best option? Well I think a new file over https/.well-known that is in an XMPP-subset-of-XML format and so can safely share the same parser. It would be similar to HACX which was rejected with the direction to look into extending Discovering Alternative XMPP Connection Methods (XEP-0156) [27], which this is doing.

[26] RFC 7711: PKIX over Secure HTTP (POSH)<http://tools.ietf.org/html/rfc7711>.
[27] XEP-0156: Discovering Alternative XMPP Connection Methods <https://xmpp.org/extensions/xep-0156.html>.

5 Security Considerations

It should be noted this allows your web host to hijack your XMPP connection, but that's actually been true for quite some time, they could already bypass the need for a certificate with POSH, or get one from LetsEncrypt if you didn't have the proper CAA records, or hijack it for websocket/bosh supporting clients, so this doesn't really open up new avenues of attack.
Please refer to the security considerations and warnings of RFC 7469 [28] with regards to having a backup public key and being careful to not break your domain for the whole TTL. For this reason and others it is advised to put a max limit on TTL of 1 week (604800).
Validating certs is full of edge cases and must be done with the utmost of care and precision.

6 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [29].

7 XMPP Registrar Considerations

This document requires no interaction with the XMPP Registrar [30].

[28] RFC 7469: Public Key Pinning Extension for HTTP <http://tools.ietf.org/html/rfc7469>.

[29] The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.

[30] The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <https://xmpp.org/registrar/>.