XEP-0418: DNS Queries over XMPP (DoX)

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

2019-03-29
Version 0.1.0

Status	Type	Short Name
Deferred	Standards Track	dox

This specification defines an XMPP protocol extension for sending DNS queries and getting DNS responses over XML streams. Each DNS query-response pair is mapped into an IQ exchange.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2024 by the XMPP Standards Foundation (XSF).

Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the ”Specification”), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

Warranty

NOTE WELL: This Specification is provided on an ”AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF’s Intellectual Property Rights Policy (a copy of which can be found at <https://xmpp.org/about/xsf/ipr-policy> or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

Contents

1 Introduction 1

2 Requirements 1

3 Protocol 1

4 Use Cases 1

5 Determining Support 2

6 Implementation Notes 3

7 Security Considerations 3

8 IANA Considerations 4

9 XMPP Registrar Considerations 5

9.1 Protocol Namespaces 5

10 XML Schema 5

1 Introduction

This document defines a specific protocol, DNS over XMPP (DoX), for sending DNS RFC 1035 [1] queries and getting DNS responses over XMPP Core [2] (and therefore TLS RFC 8446 [3] security for integrity and confidentiality.
The integration with XMPP provides a transport suitable for both existing DNS clients and native XMPP applications seeking access to the DNS.
This protocol is almost identical in scope to DNS Queries over HTTPS (DoH) RFC 8484 [4]

2 Requirements

This specification addresses the following requirements:

1. Sending a DNS query.
2. Responding with a DNS response.

3 Protocol

The DoX protocol is extremely simple:

1. The requesting entity (requestor) sends an IQ-get containing a <dns/> element qualified by the 'urn:xmpp:dox:0' namespace, which contains the DNS query.
2. The resolving entity (resolver) returns either an IQ-result containing a <dns/> element qualified by the 'urn:xmpp:dox:0' namespace, which contains the DNS response (if it supports the namespace) or an IQ-error (if it does not).
3. In both the query and response, the content of the <dns/> element is the DNS on-the-wire format as defined in RFC 1035 [5]. The body MUST be encoded with base64 RFC 4648 [6]. Padding characters for base64 MUST NOT be included.

4 Use Cases

Sending a DNS query is done by sending an <iq/> get over the stream from the requestor to the resolver.

[1] RFC 1035: Domain Names - Implementation and Specification <http://tools.ietf.org/html/rfc1035>.
[2] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[3] RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3 <http://tools.ietf.org/html/rfc8446>.
[4] RFC 8484: DNS Queries over HTTPS (DoH) <http://tools.ietf.org/html/rfc8484>.
[5] RFC 1035: Domain Names - Implementation and Specification <http://tools.ietf.org/html/rfc1035>.
[6] RFC 4648: The Base16, Base32, and Base64 Data Encodings <http://tools.ietf.org/html/rfc4648>.

Listing 1: Query

```
<iq from='romeo@montague.lit/home' to='juliet@capulet.lit/chamber'
 id='s2c1' type='get'>
  <dns xmlns='urn:xmpp:dox:0'>
      vOIBIAABAAAAAAABB2V4YW1wbGUDb3JnAAABAAEAACkQAAAAAAADAAKAAj5HO5JuEe
      +mA</dns>
</iq>
```

If the resolver supports the dns namespace, it MUST return an IQ-result, which contains the DNS response:

Listing 2: Response

```
<iq from='juliet@capulet.lit/chamber' to='romeo@montague.lit/home'
 id='s2c1' type='result'>
  <dns xmlns='urn:xmpp:dox:0'>
      vOKBoAABAAEAAAABB2V4YW1wbGUDb3JnAAABAAHADAABAAEAAAhjAARduNgiAAApEAAAAAAAAAA
      </dns>
</iq>
```

If the resolver does not support the dns namespace, it MUST return a <service-unavailable/> error:

Listing 3: DNS Not Supported

```
<iq from='juliet@capulet.lit/chamber' to='romeo@montague.lit/home' id=
   's2c1' type='error'>
  <dns xmlns='urn:xmpp:dox:0'>
      vOIBIAABAAAAAAABB2V4YW1wbGUDb3JnAAABAAEAACkQAAAAAAADAAKAAj5HO5JuEe
      +mA</dns>
  <error type='cancel'>
    <service-unavailable xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

The other error conditions defined in RFC 6120 [7] could also be returned if appropriate.

5 Determining Support

If an entity supports the DoX protocol, it MUST report that fact by including a service discovery feature of "urn:xmpp:dox:0" in response to a Service Discovery (XEP-0030) [8] information request:

[7] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[8] XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.

Listing 4: Service Discovery information request

```
<iq type='get'
    from='juliet@capulet.lit/balcony'
    to='capulet.lit'
    id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>
```

Listing 5: Service Discovery information response

```
<iq type='result'
    from='capulet.lit'
    to='juliet@capulet.lit/balcony'
    id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='urn:xmpp:dox:0'/>
    ...
  </query>
</iq>
```

In order for an application to determine whether an entity supports this protocol, where possible it SHOULD use the dynamic, presence-based profile of service discovery defined in Entity Capabilities (XEP-0115) [9]. However, if an application has not received entity capabilities information from an entity, it SHOULD use explicit service discovery instead.
Support could also be pre-arranged between parties by putting a resolver at a known JID, in which case the requestor can just start sending queries to the resolver

6 Implementation Notes

Some XMPP clients do not respond to IQ stanzas containing unsupported payloads. Although this is in violation of XMPP Core [10], this behavior can result in disconnection of clients that are in fact actively connected to the server.

7 Security Considerations

Running DNS over XMPP relies on the security of the underlying XMPP transport, therefore all queries and responses MUST use TLS or equivalent connection security. This mitigates classic amplification attacks for UDP- based DNS.
Session-level encryption has well-known weaknesses with respect to traffic analysis, which might be particularly acute when dealing with DNS queries. DoX resolvers can also add DNS

[9] XEP-0115: Entity Capabilities <https://xmpp.org/extensions/xep-0115.html>.
[10] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.

padding RFC 7830 [11] if the DoX requestor requests it in the DNS query. An experimental effort to offer guidance on choosing the padding length can be found in RFC 8467 [12].
The TLS connection provides transport security for the interaction between the DoX resolver and requestor, but it does not provide the response integrity of DNS data provided by DNSSEC. DNSSEC and DoX are independent and fully compatible protocols, each solving different problems. The use of one does not diminish the need nor the usefulness of the other. It is the choice of a requestor to either perform full DNSSEC validation of answers or to trust the DoX resolver to do DNSSEC validation and inspect the AD (Authentic Data) bit in the returned message to determine whether an answer was authentic or not.
In the absence of DNSSEC information, a DoX resolver can give a requestor invalid data in response to a DNS query. A DoX capable requestor MUST discard any responses not specifically requested, this prohibition does not guarantee protection against invalid data, but it does reduce the risk.
If a server receives a dns request directed to a full JID <localpart@domain.tld/resource> associated with a registered account but there is no connected resource matching the 'to' address, RFC 6120 [13] requires it to reply with a <service-unavailable/> error and set the 'from' address of the IQ-error to the full JID provided in the 'to' address of the dns request. If a connected resource receives a dns request but it does not want to reveal its network availability to the sender for any reason (e.g., because the sender is not authorized to know the connected resource's availability), then it too MUST reply with a <service-unavailable/> error. This consistency between the server response and the resolver response helps to prevent presence leaks.

8 IANA Considerations

No interaction with the Internet Assigned Numbers Authority (IANA) [14] is necessary as a result of this document.

[11] RFC 7830: The EDNS(0) Padding Option <http://tools.ietf.org/html/rfc7830>.
[12] RFC 8467: Padding Policies for Extension Mechanisms for DNS (EDNS(0)) <http://tools.ietf.org/html/rfc8467>.
[13] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[14] 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/>.

9 XMPP Registrar Considerations

9.1 Protocol Namespaces

The XMPP Registrar [15] includes ”urn:xmpp:dox:0” in its registry of protocol namespaces (see <https://xmpp.org/registrar/namespaces.html>).

10 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='urn:xmpp:dox:0'
    xmlns='urn:xmpp:dox:0'
    elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-XXXX: https://xmpp.org/extensions/inbox/dox.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name='dns' type='base64Binary'/>

</xs:schema>
```

[15] 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/>.