

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2013

P. Saint-Andre
M. Miller
Cisco Systems, Inc.
February 22, 2013

Domain Name Associations (DNA) in the Extensible Messaging and Presence Protocol (XMPP)

[draft-saintandre-xmpp-dna-01](#)

Abstract

This document improves the security of the Extensible Messaging and Presence Protocol (XMPP) in two ways. First, it specifies how "proofotypes" can establish a strong association between a domain name and an XML stream. Second, it describes how to securely delegate a source domain to a derived domain, which is especially important in virtual hosting environments.

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 August 26, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in [Section 4.e](#) of

Internet-Draft

XMPP DNA

February 2013

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology	3
3.	Flow Chart	3
4.	A Simple Scenario	6
5.	One-Way Authentication	7
6.	Piggybacking	7
6.1.	Assertion	7
6.2.	Supposition	9
7.	Alternate Prooftypes	10
8.	Virtual Hosting	11
9.	Prooftype Model	11
10.	Security Considerations	12
11.	IANA Considerations	12
12.	References	12
12.1.	Normative References	12
12.2.	Informative References	13
	Authors' Addresses	14

[1.](#) Introduction

The need to establish a strong association between a domain name and an XML stream arises in both client-to-server and server-to-server communication using the Extensible Messaging and Presence Protocol (XMPP), because XMPP servers are typically identified by DNS domain names. However, a client or peer server needs to verify the identity of a server to which it connects. To date, such verification has been established based on information obtained from the Domain Name System (DNS), the Public Key Infrastructure (PKI), or similar sources. This document (1) generalizes the model currently in use so that additional prooftypes can be defined, (2) provides a basis for modernizing some prooftypes to reflect progress in underlying technologies such as DNS Security [[RFC4033](#)], and (3) describes the flow of operations for establishing a domain name association.

Furthermore, the process for resolving the domain name of an XMPP service into the IP address at which an XML stream will be negotiated (defined in [[RFC6120](#)]) can involve delegation of a source domain (say, example.com) to a derived domain (say, hosting.example.net).

If such delegation is not done in a secure manner, then the domain name association cannot be authenticated. Therefore, this document provides guidelines for defining secure delegation methods.

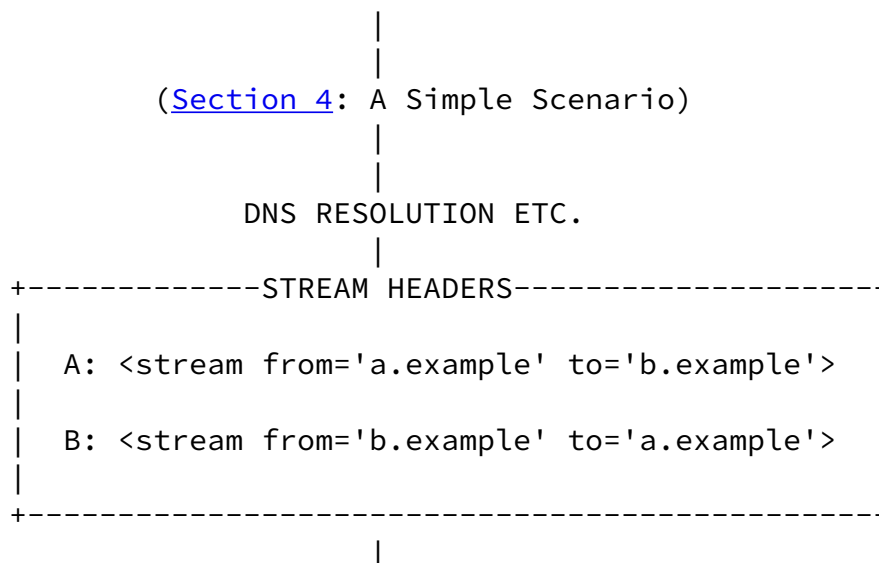
This document does not define any DNA prooftypes or secure delegation methods; such technologies are defined in companion documents.

2. Terminology

This document inherits XMPP terminology from [[RFC6120](#)] and [[XEP-0220](#)], DNS terminology from [[RFC1034](#)], [[RFC1035](#)], [[RFC2782](#)] and [[RFC4033](#)], and security terminology from [[RFC4949](#)] and [[RFC5280](#)]. The terms "source domain", "derived domain", "reference identity", and "presented identity" are used as defined in the "CertID" specification [[RFC6125](#)]. The terms "permissive federation", "verified federation", and "encrypted federation" are derived from [[XEP-0238](#)], although we substitute the term "authenticated federation" for the term "trusted federation" from that document.

3. Flow Chart

The following flow chart illustrates the protocol flow for establishing domain name associations between Server A and Server B, as described in the remaining sections of this document.



4. Server B sends a response stream header to Server A, asserting that it is b.example:

```
<stream:stream from='b.example' to='a.example'>
```
5. The servers attempt TLS negotiation, during which Server B (acting as a TLS server) presents a PKIX certificate proving that it is b.example and Server A (acting as a TLS client) presents a PKIX certificate proving that it is a.example.
6. Server A checks the PKIX certificate that Server B provided and Server B checks the PKIX certificate that Server A provided; if these proofs are consistent with the XMPP profile of the matching rules from [[RFC6125](#)], each server accepts that there is a strong domain name association between its stream to the other party and the DNS domain name of the other party.

Several simplifying assumptions underlie the happy scenario just outlined:

- o Server A presents a PKIX certificate during TLS negotiation, which enables the parties to complete mutual authentication.
- o There are no additional domains associated with Server A and Server B (say, a subdomain chatrooms.a.example on Server A or a second domain c.example on Server B).
- o The server administrators are able to obtain PKIX certificates in the first place.
- o The server administrators are running their own XMPP servers, rather than using hosting services.

Let's consider each of these "wrinkles" in turn.

5. One-Way Authentication

If Server A does not present its PKIX certificate during TLS negotiation (perhaps because it wishes to verify the identity of Server B before presenting its own credentials), Server B is unable to mutually authenticate Server A. Therefore, Server B needs to

negotiate and authenticate a stream to Server A, just as Server A has done:

1. Server B resolves the DNS domain name a.example.
2. Server B opens a TCP connection to the resolved IP address.
3. Server B sends an initial stream header to Server A, asserting that it is b.example:

```
<stream:stream from='b.example' to='a.example'>
```
4. Server A sends a response stream header to Server B, asserting that it is a.example:

```
<stream:stream from='a.example' to='b.example'>
```
5. The servers attempt TLS negotiation, during which Server A (acting as a TLS server) presents a PKIX certificate proving that it is a.example.
6. Server B checks the PKIX certificate that Server A provided; if it is consistent with the XMPP profile of the matching rules from [\[RFC6125\]](#), Server B accepts that there is a strong domain name association between its stream to Server A and the DNS domain name a.example.

Unfortunately, now the servers are using two TCP connections instead of one, which is somewhat wasteful. However, there are ways to tie the authentication achieved on the second TCP connection to the first TCP connection; see [\[XEP-0288\]](#) for further discussion.

[6.](#) Piggybacking

[6.1.](#) Assertion

Consider the common scenario in which Server B hosts not only b.example but also a second domain c.example. If a user of Server B associated with c.example wishes to communicate with a friend at a.example, Server B needs to send XMPP stanzas from the domain

c.example rather than b.example. Although Server B could open an new

TCP connection and negotiate new XML streams for the domain pair of c.example and a.example, that too is wasteful. Server B already has a connection to a.example, so how can it assert that it would like to add a new domain pair to the existing connection?

The traditional method for doing so is the Server Dialback protocol, first specified in [[RFC3920](#)] and since moved to [[XEP-0220](#)]. Here, Server B can send a <db:result/> element for the new domain pair over the existing stream.

```
<db:result from='c.example' to='a.example'>  
  some-dialback-key  
</db:result>
```

This element functions as Server B's assertion that it is (also) c.example, and thus is functionally equivalent to the 'from' address of an initial stream header as previously described.

In response to this assertion, Server A needs to obtain some kind of proof that Server B really is also c.example. It can do the same thing that it did before:

1. Server A resolves the DNS domain name c.example.
2. Server A opens a TCP connection to the resolved IP address (which might be the same IP address as for b.example).
3. Server A sends an initial stream header to Server B, asserting that it is a.example:

```
<stream:stream from='a.example' to='c.example'>
```

4. Server B sends a response stream header to Server A, asserting that it is c.example:

```
<stream:stream from='c.example' to='a.example'>
```

5. The servers attempt TLS negotiation, during which Server B (acting as a TLS server) presents a PKIX certificate proving that it is c.example.
6. Server A checks the PKIX certificate that Server B provided; if it is consistent with the XMPP profile of the matching rules from [[RFC6125](#)], Server A accepts that there is a strong domain name association between its stream to Server B and the DNS domain name c.example.

Now that Server A accepts the domain name association, it informs Server B of that fact by sending verification of the Server Dialback key over the original connection:

```
<db:verify from='a.example' to='c.example' type='valid'/>
```

The parties can then terminate the second connection, since it was used only for Server A to associate a stream over the same IP:port combination with the domain name c.example (dialback key links the original stream to the new association).

[6.2.](#) Supposition

Piggybacking can also occur in the other direction. Consider the common scenario in which Server A provides XMPP services not only for a.example but also for a subdomain such as a groupchat service at chatrooms.a.example (see [\[XEP-0045\]](#)). If a user from c.example at Server B wishes to join a room on the groupchat service, Server B needs to send XMPP stanzas from the domain c.example to the domain chatrooms.a.example rather than a.example. Therefore, Server B needs to negotiate and authenticate a stream to chatrooms.a.example:

1. Server B resolves the DNS domain name chatrooms.a.example.
2. Server B opens a TCP connection to the resolved IP address.
3. Server B sends an initial stream header to Server A acting as chatrooms.a.example, asserting that it is b.example:

```
<stream:stream from='b.example' to='chatrooms.a.example'>
```

4. Server A sends a response stream header to Server B, asserting that it is chatrooms.a.example:

```
<stream:stream from='chatrooms.a.example' to='b.example'>
```

5. The servers attempt TLS negotiation, during which Server A (acting as a TLS server) presents a PKIX certificate proving that it is chatrooms.a.example.
6. Server B checks the PKIX certificate that Server A provided; if it is consistent with the XMPP profile of the matching rules from [\[RFC6125\]](#), Server B accepts that there is a strong domain name association between its stream to Server A and the DNS domain name chatrooms.a.example.

Internet-Draft

XMPP DNA

February 2013

As before, the parties now have two TCP connections open. So that they can close the now-redundant connection, Server B sends a dialback key to Server A over the new connection.

```
<db:result from='c.example' to='chatrooms.a.example'>
  some-dialback-key
</db:result>
```

Server A then informs Server B that it accepts the domain name association by sending verification of the dialback key over the original connection:

```
<db:verify from='chatrooms.a.example' to='c.example' type='valid'/>
```

Server B can now close the connection over which it tested the domain name association for chatrooms.a.example.

7. Alternate Proofatypes

The foregoing protocol flows assumed that domain name associations were proved using the standard PKI prooftype specified in [\[RFC6120\]](#): that is, the server's proof consists of a PKIX certificate that is checked according to a profile of the matching rules from [\[RFC6125\]](#), the client's verification material is obtained out of band in the form of a trusted root, and secure DNS is not necessary.

However, sometimes XMPP server administrators are unable or unwilling to obtain valid PKIX certificates for their servers (e.g., the administrator of im.cs.podunk.example can't receive certification authority verification messages sent to mailto:hostmaster@podunk.example, or hosting.example.net does not want to take on the liability of holding the certificate and private key for example.com). In these circumstances, prooftypes other than PKIX are desirable. Two companion documents, [\[XMPP-DANE\]](#) and [\[XMPP-POSH\]](#), define alternate prooftypes:

- o In the DANE prooftype, the server's proof consists of a PKIX

certificate that is compared as an exact match or a hash of either the SubjectPublicKeyInfo or the full certificate, and the client's verification material is obtained via secure DNS. See the accompanying [[XMPP-DANE](#)] spec for discussion and examples.

- o In the POSH (PKIX Over Secure HTTP) proofotype, the server's proof consists of a PKIX certificate that is checked according to the rules from [[RFC6120](#)] and [[RFC6125](#)], the client's verification material is obtained by retrieving the PKIX certificate over HTTPS

at a well-known URI [[RFC5785](#)], and secure DNS is not necessary since the HTTPS retrieval mechanism relies on the chain of trust from the public key infrastructure. See the accompanying [[XMPP-POSH](#)] spec for discussion and examples.

[8.](#) Virtual Hosting

One common method for deploying XMPP services is virtual hosting: e.g., the XMPP service for example.com is actually hosted at hosting.example.net. Such an arrangement is relatively convenient in XMPP given the use of DNS SRV records [[RFC2782](#)], such as the following pointer from example.com to hosting.example.net:

```
_xmpp-server._tcp.example.com. 0 IN SRV 0 0 5269 hosting.example.net
```

To improve security and limit liability, in typical deployments the administrators of hosting.example.net do not wish to hold the certificate and private key for example.com and the owners of example.com do not wish to share their certificate and private key with the administrators of hosting.example.net. In practice this means that server-to-server communications to example.com go unencrypted or the communications are TLS-encrypted but the certificates are not checked (which is functionally equivalent to an unencrypted connection). This is also true of client-to-server communications, forcing end users to override certificate warnings or configure their clients to accept certificates for hosting.example.net instead of example.com. The fundamental problem here is that if DNSSEC is not used the act of delegation is inherently insecure.

This document does not describe how to achieve secure delegation.

However, [[XMPP-DANE](#)] explains how to use DNSSEC for secure delegation in the PKI and DANE prooftypes and [[XMPP-POSH](#)] explains how to use HTTPS redirects for secure delegation in the POSH proftype.

9. Proftype Model

In general, a DNA proftype conforms to the following definition:

proftype: A mechanism for proving an association between a domain name and an XML stream, where the mechanism defines (1) the nature of the server's proof, (2) the matching rules for comparing the client's verification material against the server's proof, (3) how the client obtains its verification material, and (4) whether the mechanism depends on secure DNS.

The PKI, DANE, and POSH prooftypes adhere to this model. In addition, other prooftypes are possible (examples might include PGP keys rather than PKIX certificates, or a token mechanism such as Kerberos or OAuth).

Some prooftypes depend on (or are enhanced by) secure DNS and therefore also need to describe how secure delegation occurs for that proftype.

10. Security Considerations

This document supplements but does not supersede the security considerations provided in [[RFC6120](#)] and [[RFC6125](#)].

11. IANA Considerations

This document has no actions for the IANA.

12. References

12.1. Normative References

- [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.
- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", [RFC 2782](#), February 2000.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", [RFC 4033](#), May 2005.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", [RFC 4949](#), August 2007.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", [RFC 5280](#), May 2008.
- [RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", [RFC 5785](#), April 2010.

- [RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", [RFC 6120](#), March 2011.
- [RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", [RFC 6125](#), March 2011.
- [XEP-0220] Miller, J., Saint-Andre, P., and P. Hancke, "Server Dialback", XSF XEP 0220, August 2011.
- [XMPP-DANE] Miller, M. and P. Saint-Andre, "Using DNS Security Extensions (DNSSEC) and DNS-based Authentication of Named Entities (DANE) as a Proof-type for XMPP Domain Name Associations", [draft-miller-xmpp-dnssec-proof-type-04](#) (work

in progress), February 2013.

[XMPP-POSH]

Miller, M. and P. Saint-Andre, "Using PKIX over Secure HTTP (POSH) as a Proofotype for XMPP Domain Name Associations", [draft-miller-xmpp-posh-proofotype-03](#) (work in progress), February 2013.

12.2. Informative References

[RFC3920] Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol (XMPP): Core", [RFC 3920](#), October 2004.

[XEP-0045]

Saint-Andre, P., "Multi-User Chat", XSF XEP 0045, February 2012.

[XEP-0238]

Saint-Andre, P., "XMPP Protocol Flows for Inter-Domain Federation", XSF XEP 0238, March 2008.

[XEP-0288]

Hancke, P. and D. Cridland, "Bidirectional Server-to-Server Connections", XSF XEP 0288, August 2012.

Authors' Addresses

Peter Saint-Andre
Cisco Systems, Inc.
1899 Wynkoop Street, Suite 600
Denver, CO 80202
USA

Email: psaintan@cisco.com

Matthew Miller
Cisco Systems, Inc.
1899 Wynkoop Street, Suite 600
Denver, CO 80202
USA

Email: mamille2@cisco.com