Network Working Group
Internet-Draft
Updates: 4556 (if approved)
Intended status: Standards Track
Expires: August 29, 2014

M. Wasserman
S. Hartman
Painless Security
M. Wasserman
JANET (UK)

February 25, 2014

Application Bridging for Federation Beyond the Web (ABFAB) Trust Router Protocol

draft-mrw-abfab-trust-router-02.txt

Abstract

A Trust Router is an infrastucture element used to construct multihop Application Bridging for Federated Authentication Beyond the Web (ABFAB) federations. This document defines both the Trust Router Protocol and the Temporary Identity Protocol, which can be used together to enable multihop ABFAB federations without requiring a centralized Public Key Infrastructure (PKI).

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 29, 2014.

Table of Contents

1. Introduction

A Trust Router is an infrastucture element used to construct multihop Application Bridging for Federated Authentication Beyond the Web (ABFAB) federations. This document defines the Temporary Identity Protocol and the Trust Router Protocol, which can be used together to enable multihop ABFAB federations without requiring a centralized Public Key Infrastructure (PKI).

This document defines a Temporary Identity Protocol that can be used by a AAA Client (such as a AAA Proxy near a Relying Party) to negotiate a shared key with the AAA Server(s) in a target IdP realm, so that the AAA Client can use the AAA Server(s) to authenticate users within the realm.

Temporary Identity requests are forwarded by Trust Routers across a chain of Trust Links, eventually reaching the AAA Servers within the target realm. Responses are returned along the same chain. Information about available Trust Links and the paths that can be used to reach AAA Servers within the IdP realms is propogated between Trust Routers using the Trust Router Protocol, which is also defined in this document.

2. Terminology

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 RFC 2119 [RFC2119].

This document introduces the following terms:

Trust Router: This is a logical ABFAB entity that exchanges information about Trust Paths that Relying Parties can use to create transtitive chains of trust across multihop ABFAB federations.

Trust Link: A Trust Link is an assertion that a given Trust Router is capable of providing a temporary identity to communicate with another ABFAB entity (either another Trust Router, or a AAA Server within an IdP).

Trust Path: A Trust Path is a concatenation of Trust Links that can be used by an RP to contruct a transitive trust chain across a federation to a target Identity Provider.

Temporary Identity Protocol: The Temporary Identity (TID) Protocol is used to negotiate a shared key between a AAA Client and a AAA Server that can be used for subsequent AAA authentication requests.

Trust Router Protocol: The Trust Router Protocol is the mechanism used by two Trust Routers to exchange information about Trust Links and Trust Paths.

Community of Interest A Community of Interest (COI) defines a group of Services and IdPs that have agreed to cooperate to provide access to a specific set of services only to those users within a particular community. Communities of Interest can be layered on top of the base Trust Router infrastructure to allow selected access to IdPs that have joined a specific group.

Authentication Policy Community An Authentication Policy Community (APC) is a type of community in which the members have agreed to specific policies regarding user authentication.

The terms Identity Provider (IdP), Relying Party (RP), Subject, and Federation are used as defined in [I-D.lear-abfab-arch].

3. Motivation

Figure 1 shows an example federation where the Relying Party Foo, has established relationships with various Identity Providers.

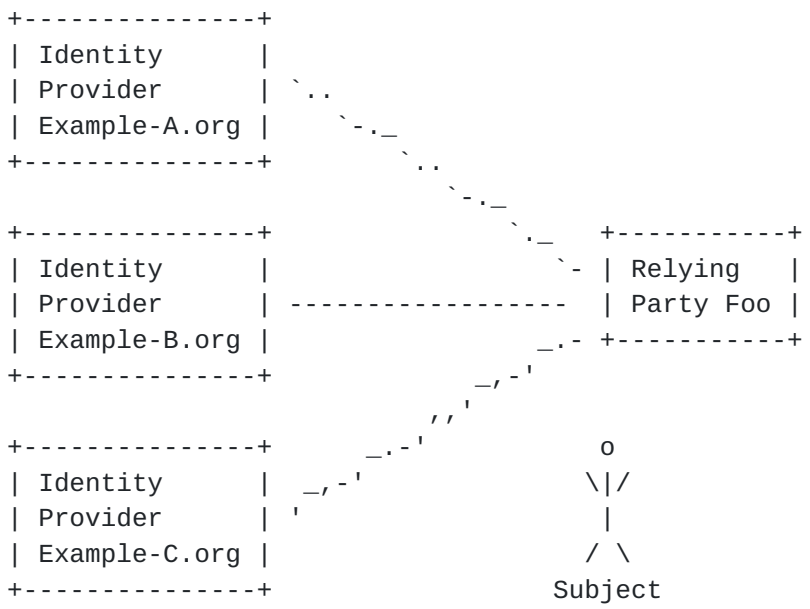

Figure 1: One-to-many Federation Example

When an RP receives a request to access a protected resource (or requires authentication for other purposes) the request includes a realm name that indicates the IdP the Subject has selected for this exchange. Offering the Subject the ability to choose among many different IdPs is necessary because a Subject may have, and want to maintain, uncorrelated identities in several different realms within a single federation (i.e. work, school, social networking, etc.). However, this also places a burden on the RPs to establish and maintain business agreements and exchange security credentials with a potentially large number of Identity Providers.

In order for a single-hop federation to function, each IdP needs to maintain business agreements and exchange credentials with every RP that its Subjects are authorized to access. Figure 2, shows the likely outcome, which is that a single-hop federation will come to resemble a dense mesh topology.

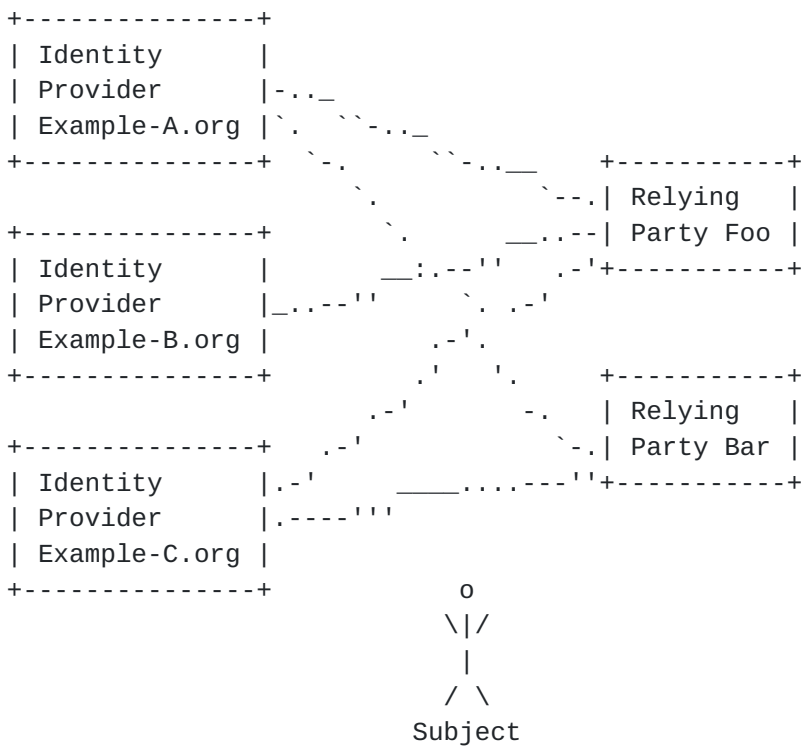

Figure 2: Mesh Federation Example

As discussed in section 2.1.1 of [I-D.lear-abfab-arch], as the number of organizations involved in a ABFAB federation increase, static configuration may not scale sufficiently. Also, using a Trust Broker to establish keys between entities near the RP and entities near the IDP with improve the security and privacy of an ABFAB federation. Figure 3 shows the structure of a federation where each IdP and RP has a single connection to the Trust Router infrastructure.

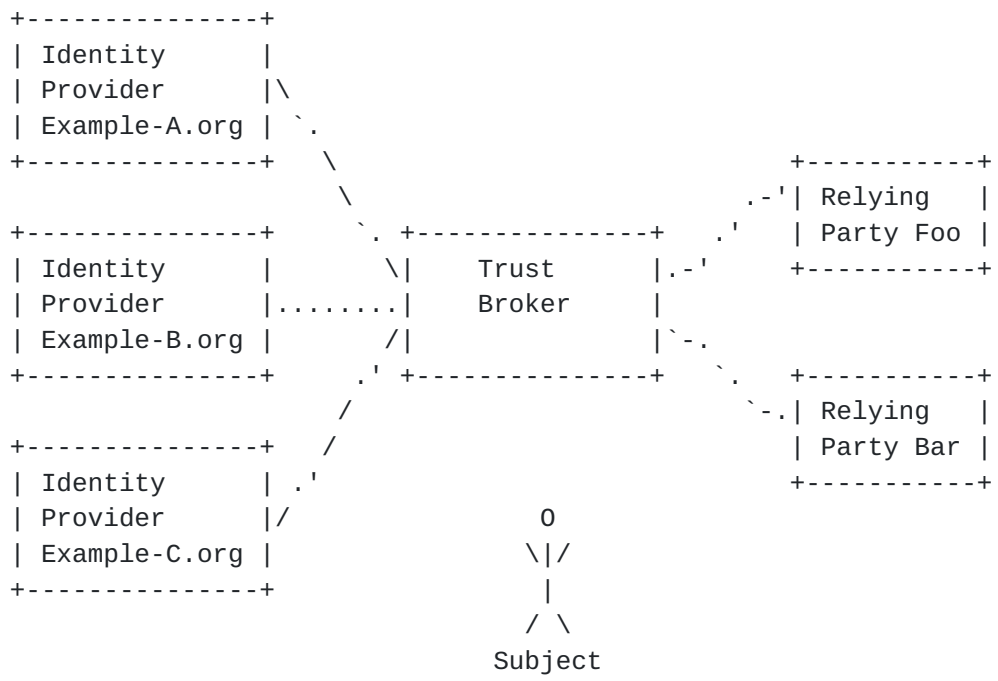

Figure 3: Federation Broker

To improve the operational scalability and security of large ABFAB federations, this document proposes a Trust Broker solution consisting of of a set of Trust Routers, as described in this document, running the Trust Router Protocol and forwarding Temporary Identity Requests between RPs and IdPs.

4. Multihop Federation Example

The diagram below shows an example of a successful exchange in a multihop federation using the Trust Routers to forward Temporary Identity Requests:

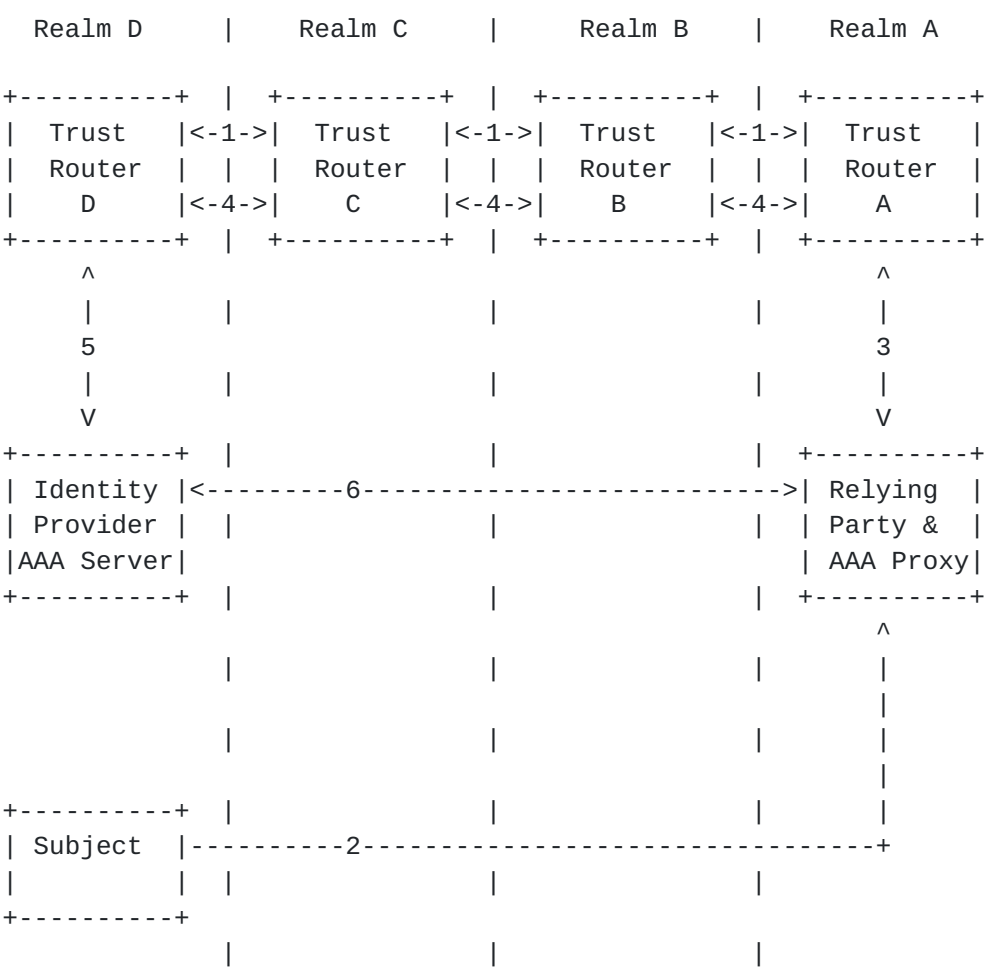

Figure 4: Example Message Exchange

A multihop federation exchange matching the above diagram can be summarized as follows:

1. We start with a single federation including four realms, each containing a single Trust Router. The Trust Routers are peered, such that their interconnections form a multihop federation.

2. A Subject (with an identity in Realm D) attempts to access a service provided by a Relying Party in Realm A.

3. The Relying Party does not have direct access to a AAA Server in Realm D that it can use to authenticate the Subject, so it asks its local Trust Router to forward a Temporary Identity Request to Realm D.

4. Trust Router A forwards the request along the Trust Path from A to B to C to D.

5. The AAA Server in Realm D receives the Temporary Identity request from the local Trust Router and configures a Temporary Identity for the AAA Proxy in Realm A.

6. The AAA Proxy in Realm A can now reach the AAA Server in Realm D to perform the authentication.

5. Temporary Identity Protocol

The Temporary Identity protocol is a simple request/response protocol that is used to established a shared secret between a AAA Client and a AAA Server that can be used for subsequent AAA exchanges. The shared secret is established via a Diffie-Helman (DH) exchange, and it therefore cannot be duplicated by the Trust Routers that forward the Temporary Identity Protocol messages.

5.1. Temporary Identity Request

A Temporary Identify request initially includes the RP Realm for which the identity is being requested, the Target Realm of the request, the Community in which the request is scoped, and a set of client DH parameters used to generate the shared secret.

As a Temporary Identity request is forwarded across the Trust Router chain, it may accumulate additional information, such at APC that corresponds to a COI in the original request and a set of Realm Constraints and Domain Constraints that will be stored by the AAA Server and used for Channel Binding to ensure that the later AAA Request comes from an appropriate AAA Client.

5.2. Temporary Identity Response

A Temporary Identity Response includes most of the fields in the original request. However, the client's DH information has been replaced by a list of AAA Servers for the target realm and a DH block corresponding to each server. The AAA Client can use that information to generate a shared key with each server.

5.3. Role of the Trust Router in Temporary Identity Requests

Trust Routers forward Temporary Identity Requests on behalf of their local AAA Proxies and their neighboring Trust Routers. As part of forwarding these requests, Trust Routers perform a COI to API conversion on the Community field, storing the original COI in an "orig_coi" field. They also add Realm and/or Domain Constraints that can later be used by the AAA Server to ensure that AAA Requests are coming from the correct AAA Client.

6. Trust Router Protocol

The Trust Router protocol is a TCP-based protocol that is used to exchange information between Trust Routers about available Trust Links within an ABFAB Federation.

As discussed in the multihop federation document, When a Trust Router advertises a Trust Link, such as A(T) -> B(T), it is making an assertion that Trust Router A is able, and willing, to provide temporary identities (via KNP) that can be used to reach Trust Router B.

Trust Routers use the information they receive about available Trust Links to construct Trust Paths that can be used to reach AAA Servers (i.e. RADIUS or DIAMETER servers) for a set of Identity Providers (IDPs) within a ABFAB federation. They then return the shortest path to a specific IDP in response to Trust Path Queries.

6.1. Trust Router Messages

6.1.1. Hello Message

Hello Messages are the first messages exchanged by Trust Routers when they bring up a new TCP connection, and they may be exchanged at other times to ensure that database information is synchronized, or to trigger a full Trust Link Database download. The first Hello messages exchanged over a new TCP connection are also used as the vehicle to establish an authenticated and encrypted GSS-API session.

6.1.2. Trust Link Database Message

A Trust Link Database Message contains a full (potentially filtered) set of Trust Links that can be reached through the sending Trust Router. This message may be quite large, and is only sent when solicited by the receiver.

6.1.3. Trust Link Update Message

Trust Routers send Trust Link Update messages to other Trust Routers to whom they are connected whenever their Trust Link Database is updated. Trust Link Update messages contain the portions of the Trust Link Database that have changed since the last update. They also contain a serial number that can be used by the receiving Trust Router to determine if any updates have been missed, in which case a full Trust Router Database download is needed.

6.2. Trust Router Operation

This section describes how Trust Routers work, in general. Detailed message formats are described in later sections of the document.

6.2.1. Hello Message Exchange

6.2.2. Exchanging Trust Link Databases

6.2.3. Trust Link Updates

6.2.4. Serial Numbers

6.2.5. TCP Connection Handling

Trust Routers communicate by exchanging full JSON-encoded messages over a TCP connection. If incomplete messages are received, or if the TCP connection is interrupted before a complete message is received, the incomplete messages will be discarded, and no protocol actions will be taken based on the contents of the incomplete message.

In the Trust Router Protocol, no information about the availability of Trust Links is inferred from a TCP reset, or a retransmission timeout on the TCP connection to another Trust Router. A Trust Router is only considered unreachable after an attempt to reestablish a TCP connection to that Trust Router is reset or times out.

When a Trust Router is found to be unreachable, the Trust Links supplied by that Trust Router are not removed from the local Trust Link Database. They will however, be marked as deprecated until a connection can be reestablished with the Trust Router that sent them, and it can be verified that the sequence number of that Trust Router's Database still matches the sequence number of the most recent Trust Link information received.

When Trust Links are marked as deprecated, they will not be used if another, non-deprecated path exists to reach the target Identity Provider. If there are no paths to the target Identity Provider that traverse only non-deprecated Trust Links, a path containing a deprecated Trust Link will be used.

6.2.6. Conceptual Data Structures

6.2.7. Peer Table

6.2.8. Trust Link Database

7. Message Representation

This section provides details about the contents and encoding of both Trust Router Protocol messages and Trust Path Query messages.

7.1. Message Encoding

The Trust Router Protocol and Trust Path Query messages are encoded in JavaScript Object Notation (JSON) [RFC4627].

7.2. Temporary Identity Protocol Representation

7.2.1. Temporary Identity Request

7.2.2. Temporary Identity Response

7.3. Trust Router Protocol Message Representation

7.3.1. Hello Message Representation

Name or Realm (??) Auth-Token (??) Database-Serial-Number Database-Request

Database-Serial-Number field contains the current serial number of the sending Trust Router's Trust Link Database. This information may be used by a receiving Trust Router to determine whether it should request a full Trust Link Database download.

The Database-Request field indicates whether the receiving Trust Router should respond to this message with a Trust Link Database message, to share its full Trust Link Database with the sending Trust Router. If this field has a value of "true", a download is requested. If it is "false", a download is not requested.

7.3.2. Trust Link Database/Update Representation

In the Trust Router Protocol, each Trust Router will send a (potentially filtered) set of Trust Links to its neighboring Trust Routers. The representation of these Trust Links is designed for efficient encoding, and to allow easy population of a conceptual Trust Link Table on the receiving Trust Router. Each Trust Router will only distribute a set of Trust Links that form a connected tree rooted at the sending Trust Router.

Conceptually, a Trust Link consists:

- A Trust Router that is willing to provide a temporary identity.
- The Trust Router or AAA Server which the identity can be provided.
- The Communities-of-Interest to whom the link is available.
- A lifetime for this link, in seconds.

However, the actual Trust Links passed in the Trust Router protocol rely on inference and ordering to eliminate the need to include the first Trust Router identity in each distributed link. Instead, we use an Index variable, which indicates each Trust Link's level in a conceptual tree, and we order the Trust Links, so that a Trust Link with an Index of N is subordinate to the closest previous Trust Link with an index of N-1 that applies to the same Community-of-Interest. Each conceptual tree is rooted at the sending Trust Router, which is represented by an an entry with an Index value of 0.

7.3.3. Trust Link Ordering

7.3.4. Entity Identity

When we send Trust Router or AAA Server identities in the Trust Router Protocol, that information will be sent in an Entity Identity structure containing the following fields:

- Name
- Type
- Realm

The Name field will typically contain a fully-qualified domain name (FQDN) that can be used to reach the indicated entity (e.g. "tr-A.example.net").

The Type field indicates that the entity is a Trust Router (Type = "T") or a AAA Server (Type = "R", "D", or "S" for a RADIUS Server, DIAMETER Server or RADSEC Server, respectively).

The Realm field contains the security realm associated with the entity (e.g. "example.net").

7.3.5. Trust Link Entry

As transmitted in the Trust Router Protocol, a Trust Link entry will have the following fields:

- Index
- Target-Entity
- Communities-of-Interest
- Lifetime

The Index field contains a non-zero integer value, indicating the depth of this Trust Link in a conceptual tree of links rooted at the sending Trust Router. The maximum value of this field is 255.

The Target-Entity field contains a the Trust Router or AAA Server for which temporary identities can be generated. This also represents the Trust Router that can generate identities for any directly subordinate nodes in the conceptual tree.

The Communities-of-Interest field contains an array of strings, each containing a Community-of-Interest for which this link is available.

The Lifetime field contains an integer that indicates the lifetime of this Trust Link in seconds. Links are removed from the the conceptual Trust Link Table if their lifetime expires.

7.3.6. Trust Link Database Message

A Trust Link Database will consist two fields:

- Serial-Number
- Trust-Links

The Serial-Number field contains an integer indicating the version of the information contained in this database. The maximum value for this field is (2^32 - 1).

The Trust-Links field contains an array of Trust Link Entries.

7.3.7. Trust Link Update Message

8. Message Examples

8.1. Temporary Identity Request Example

```
{"msg_type": "TIDRequest",
 "msg_body":
    {"rp_realm": "foo.example.com",
     "target_realm": "bar.example.net",
     "community": "trust-router-hackers.baz.example.edu",
     "dh_info":
        {"dh_p": "FFFFFFFF...",
         "dh_g": "02",
         "dh_pub_key": "FBF98ABB..."
        }
    }
}
```

8.2. Temporary Identity Response Example

```
{"msg_type": "TIDResponse",
 "msg_body":
    {"rp_realm": "foo.example.com",
     "target_realm": "bar.example.net",
     "community": "apc.baz.example.edu">
     "orig_coi": "trust-router-hackers.baz.example.edu",
     "aaa_servers":
        [{"server_name":"aaa-server.bar.example.net",
          "dh_info:
             {"dh_p": "FFFFFFFF...",
              "dh_g": "02",
              "dh_pub_key": "FBF98ABB..."
             }
          }]
     "realm_constraints":"*.foo.example.com"
    }
}
```

9. Security Considerations

[TBD]

10. IANA Considerations

IANA has allocated the following TCP port numbers for use by protocols described in this document:

[TBD]

11. Acknowledgements

This document was written using the xml2rfc tool described in RFC 2629 [RFC2629].

The following people provided useful comments or feedback on this document: Daniel Kouril, Linus Nordberg, Jim Schaad, Rhys Smith, Kevin Wasserman.

12. Change Log

12.1. Changes between -01 and -02

- Changed Trust Path Query protocol to Temporary Identity Request Protocol
- Added TID details based on implemented code.
- Restructured document to remove need for separate multihop federations document.

12.2. Changes between -00 and -01

- Minor revisions, added authors.

13. References

13.1. Normative References

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

13.2. Informative References

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

Authors' Addresses

Margaret Wasserman
Painless Security
14 Summer Street, Suite 202
Malden, MA 02148
USA

Phone: +1 781 405-7464
Email: mrw@painless-security.com
URI: http://www.painless-security.com

Sam Hartman
Painless Security
14 Summer Street, Suite 202
Malden, MA 02148
USA

Email: hartmans@painless-security.com
URI: http://www.painless-security.com

Margaret Wasserman
JANET (UK)

Email: josh.howlett@ja.net