Network Working Group	M. Wasserman
Internet-Draft	Painless Security
Intended status: Standards Track	October 24, 2011
Expires: April 26, 2012	

Application Bridging for Federation Beyond the Web (ABFAB) Trust Router Protocol
draft-mrw-abfab-trust-router-00.txt

Abstract

A Trust Router is an infrastucture element used to construct multihop Application Bridging for Federated Authentication Beyond the Web (ABFAB) federations, as discussed in draft-mrw-abfab-multihop-fed-01.txt. This document defines both the Trust Router Protocol and the Trust Path Query, as discussed in the multihop federation document.

Status of this Memo

Copyright Notice

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, as discussed in draft-mrw-abfab-multihop-fed-01.txt. This document defines both the Trust Router Protocol and the Trust Path Query, as discussed in the multihop federation document. This document defines the protocol used between Trust Routers to exchange information about Trust Paths available within an ABFAB federation. It also defines the messages that a federated service will use to obtain Trust Path information from its local Trust Router, so that it can use the ABFAB Key Negotiation Protocol (KNP) to forge a Chain of Trust across a federation. The Chain of Trust will lead to an Authentication, Authorization and Accounting (AAA) Server for a user's

Identity Provider, which will then be used to authenticate and authorize the user.

2. Requirements 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].

3. 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.

4. Trust Router Messages

4.1. 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.
TBD: We need to discuss how GSS-API will be used with this protocol. Maybe we need separate authentication messages before the Hello messages are exchanged?

4.2. 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.

4.3. 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.

5. Trust Router Operation

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

5.1. Hello Message Exchange

5.2. Exchanging Initial Trust Link Databases

5.3. Trust Link Updates

5.4. Serial Numbers

5.5. 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.

5.6. Conceptual Data Structures

5.6.1. Peer Table

5.6.2. Trust Link Database

6. Trust Path Query

6.1. Trust Path Query Messages

6.1.1. Trust Path Query Request

6.1.2. Trust Path Query Response

6.2. Trust Path Query Operation

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. Hello Message Representation

Name or Realm (??) Auth-Token (??) Database-Serial-Number Database-Request
TBD: It is unclear what sort of authentication information needs to be in this message for GSS-API authentication.
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. 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.

7.3.1. Trust Link Ordering

7.3.2. 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").
The Realm field contains the security realm associated with the entity (e.g. "example.net").

7.3.3. 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.

7.3.4. Trust Link Database Message

A Trust Link Databases 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.

7.3.5. Trust Link Update Message

7.4. Trust Path Query Representation

7.4.1. Trust Link Query Request

TBD: Pending resolution of open architectural questions regarding what will be queried/returned in these messages.

7.4.2. Trust Link Query Response

TBD: Pending resolution of open architectural questions regarding what will be queried/returned in these messages.

7.5. Message Examples

This section contains example of Trust Router Protocol and Trust Query messages encoded in JSON, as they will be sent over the nework.

7.5.1. Hello Message Example

7.5.2. Trust Link Database Example

7.5.3. Trust Link Update Example

7.5.4. Trust Path Query Request Example

TBD: Pending resolution of open architectural questions regarding what will be queried/returned in these messages.

7.5.5. Trust Path Query Response Example

TBD: Pending resolution of open architectural questions regarding what will be queried/returned in these messages.

8. Security Considerations

[TBD]

9. IANA Considerations

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

10. 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: Sam Hartman, Josh Howlett.

11. References

11.1. Normative References

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

11.2. Informative References

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

Author's Address

Margaret Wasserman Wasserman Painless Security 356 Abbott Street North Andover, MA 01845 USA Phone: +1 781 405 7464 EMail: mrw@painless-security.com URI: http://www.painless-security.com