Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2012

M. Wasserman
Painless Security

July 4, 2011

Application Bridging for Federation Beyond the Web (ABFAB) Multihop Federations

draft-mrw-abfab-multihop-fed-00.txt

Abstract

This document describes a mechanism for establishing trust across a multihop federation within the Application Bridging for Federation Beyond the Wed (ABFAB) framework.

This document introduces a new ABFAB entity, the Trust Router. Trust Routers exchange information about the availability of Trust Paths across a multiphop federation. They can be queried by a Relying Party to obtain the best Trust Path to reach a RADIUS or RADSEC server in a given realm. They also provide temporary identities that can be used by a Relying Party to traverse a Trust Path.

This document is currently limited to discussing a proposed mechanism to achieve a multihop federation in the ABFAB framework. Later versions of this document (or companion documents) will describe the protocols and algorithms in more detail.

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 January 6, 2012.

Copyright Notice

Table of Contents

1. Introduction

This document describes a mechanism for establishing trust across a multihop federation within the Application Bridging for Federation Beyond the Web (ABFAB) framework [I-D.lear-abfab-arch].

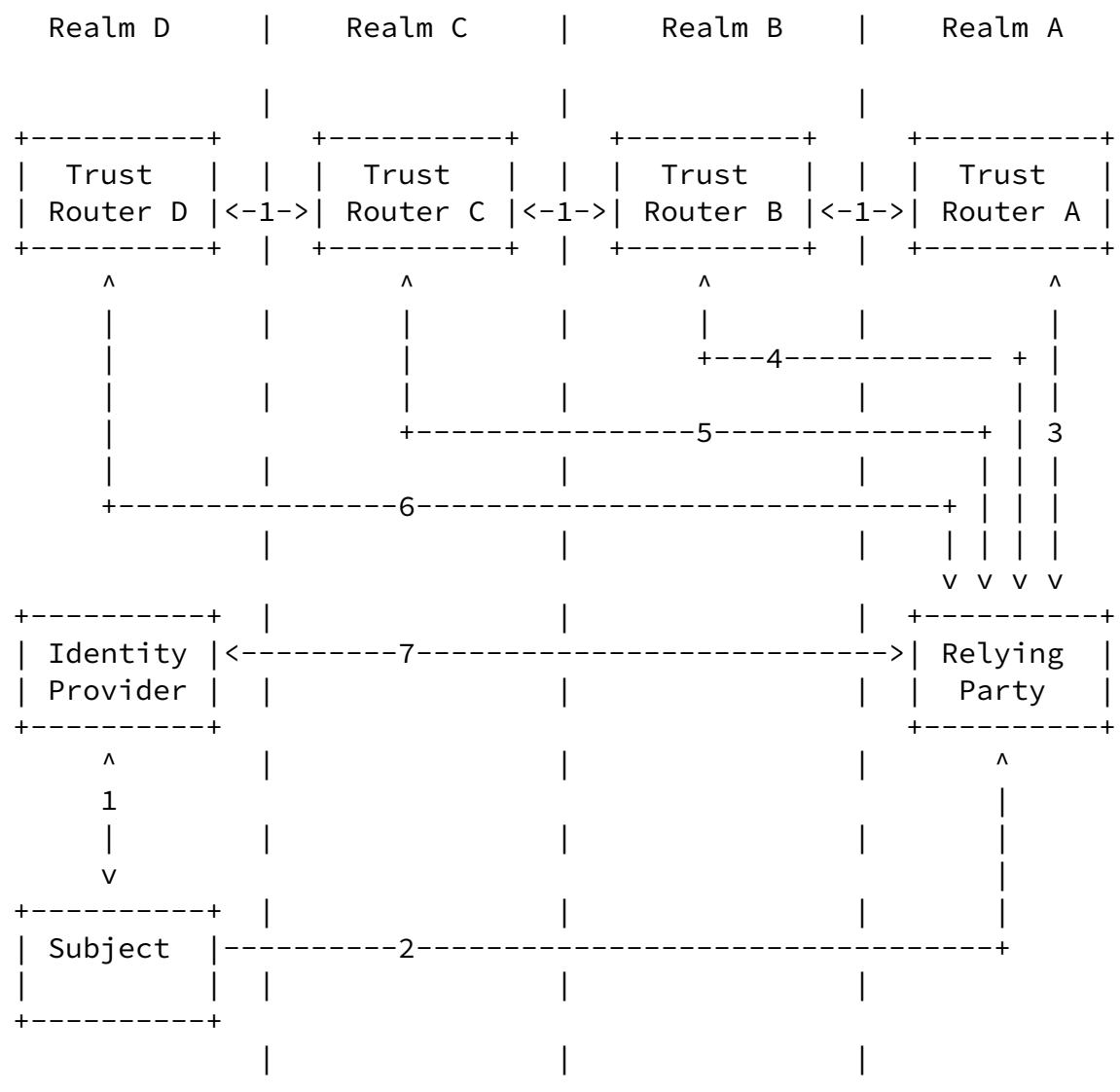
In a single hop federation, every Relying Party has a pre-existing relationship with every Identity Provider. In other words, each Relying Party is pre-configured with the required information and credentials to reach a RADIUS or RADSEC server in every Identity Provider withing the federation. In a multihop federation, this is not necessary, as a Relying Party can reach the RADIUS or RADSEC server within a previously unknown Identity Provider by traversing a transitive Trust Path across a federation.

This document introduces a new ABFAB entity, the Trust Router. Trust Routers exchange information about the availability of Trust Paths across a multihop federation. They can be queried by a Relying Party to obtain the best Trust Path to reach a RADIUS or RADSEC server in a given realm. They also provide temporary identities that can be used by a Relying Party to traverse a Trust Path.

This document is currently limited to discussing a proposed mechanism to achieve a multihop federation in the ABFAB framework. Later versions of this document (or companion documents) will describe the protocols and algorithms in more detail.

1.1. Multihop Federation Example

The diagram below shows an example of a successful authentication exchange in a multihop federation:


Realm D
Realm C
Realm B
Realm A
Trust Router D
Trust Router C
Trust Router B
Trust Router A
1
1
1
4
5
6
3
Identity Provider
7
Relying Party
1
Subject
2

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

1. We start with a single federation including 4 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 RADIUS or RADSEC server in Realm D that it can use to authenticate the user, so it asks its local Trust Router for a Trust Path to reach Realm D. The Trust Router in Realm A returns the path A->B(T)->C(T)->D(T)->D(R), which indicates that the Relying Party should use the Trust Routers in Realms B, C and D to reach a

RADIUS or RADSEC server in Realm D, which could then be used to authenticate the user.

4. The Relying Party contacts a trust router in Realm B (using its permanent identity in Realm A), and requests the creation of a temporary identity that can be used to communicate with the Trust Router in Realm C.

5. The Relying Party then contacts the Trust Router in Realm C (using the temporary identity returned in the previous step), and asks for a temporary identity that can be used to communicate with the Trust Router in realm D.

6. The Relying Party then contacts the Trust Router in Realm D (using the temporary identity returned in the previous step), and asks the Trust Router to provision an identity that it can use to speak to the RADIUS or RADSEC server in Realm D (which is part of Realm D's Identity Provider).

7. At this point, the Relying Party can reach the Subject's Identity provider, and the rest of the ABFAB exchange can continue, as described in [I-D.lear-abfab-arch].

1.2. Trust Router Overview

As shown in the example above, the Trust Router performs three functions:

- Trust Routers peer with one another to exchange information about available Trust Paths. This information is exchanged between Trust Routers using the Trust Router Protocol. The Trust Router Protocol is described in more detail in a later section.

- The Relying Party queries a local Trust Router to determine the best path to use to reach the destination realm. This exchange is referred to as a Trust Path Query, and is described in more detail in a later section.

- The Relying Party will ask each Trust Router along the path to provision a temporary identity that can be used to gain access to the next step in the path. This mechanism is called a Temporary Identity Request, and is described in more detail in a later section.

1.3. Multiple Federations

The example above shows a number of Trust Routers running within a single federation. In real deployments, it is expected that some Trust Routers will serve multiple federations. Also, it is possible that services will be available across multiple federations, or that Subjects with have identities within multiple federations. In order to support these cases, a Policy Regime (essentially a federation name) is passed as a parameter or attribute in many of the exchanges shown above, typically paired with the name of a realm. Trust Routers will, conceptually, calculate a separate tree for each Policy Regime, and the Trust Path provided to the Relying Party will consist of Trust Links within a single Policy Regime (or federation).

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. Terminology

- Trust Router
- Trust Path
- Trust Link
- Trust Router Protocol
- Policy Regime

4. Trust Router Protocol

The Trust Router Protocol has some similarities to an Internet Routing Protocol, with some exceptions: Links are unidirectional, and realm names are not hierarchical, so no aggregation is possible. Also, it is necessary to associate a set of information with each link that can be used for filtering and tree computation, including: the cost of the link, the Policy Regime associated with the link, and information indicating how/if the link should be propagated across the federation.

Current thinking is that we will use a BGP-based algorithm for computation of the local tree at each Trust Router, and that we will

communicate a similar set of information between Trust Routers as would be communicated between Internet Routers running BGP. However, BGP does not have the security properties that would be desirable in a Trust Router Protocol, so we will not use the standard BGP encapsulation.

5. Trust Path Query

A Trust Path Query is generated by a Relying Party to request a Trust Path to reach a specific realm within a given Policy Regime. If possible, the Trust Router will reply with a Trust Path that consists of one or more Trust Router steps and ends with a RADIUS or RADSEC server within the Identify Provider for the indicated realm. The returned Trust Path represents the best path for the Relying Party to use to reach an Identity Provider in the destination realm.

The Trust Path Query is initiated by the Relying Party, and the initial query message will contain the destination realm and Policy Regime.

When a Trust Path Query is received by a Trust Router, the router will first authenticate the Relying Party, and check local policy information to determine whether or not to reply.

Assuming that the Relying Party is successfully authenticated and the request passes local policy checks, the Trust Router will search it's tree of Trust Path information to determine whether a Trust Path exists that will reach the destination Realm within the indicated regime. If so, the shortest/best Trust Path will be returned to the Relying Party.

A Trust Path will consist of a list of steps, each of which will contain: The type of the step (Trust Router, RADIUS or RADSEC), the Policy Regime associated with each step, information needed to reach the indicated Trust Router or server (domain name or IP address), and any special attributes associated with that step.

6. Temporary Identity Request

A Temporary Identity Request is issued by a Relying Party in order to obtain an identity that can be used to traverse each step in the Trust Path. When a Temporary Identity is requested, a Trust Router will provision a new identity in its local RADIUS infrastructure that can be used by the Relying Party to communicate with the Trust Router or RADIUS/RADSEC server that represents the next step in the Trust Path.

These Temporary Identities will have a finite lifetime and, when

authenticated, will include a RADIUS attribute indicating that they were generated based on a Temporary Identity Request. This attribute will inlcude the chain of identities that preceeded the current identity in the traversal of the Trust Path.

The details of how these messages will be encoded has not yet been determined. However, it is expected that, for each Trust Router step in the Trust Path, the following actions will take place:

1. The Relying Party will send a Temporary Identity Request message to the Trust Router, containing the identity of the next step in the Trust Path, the destination realm that it is trying to reach, and the Policy Regime in use. This request will be sent using the identity that the Trust Router obtained from the previous step in the Trust Path (or the Trust Router's permanent identity in it's home realm, if this is the first step).

2. The Trust Router will authenticate the Relying Party.

3. If the authentication is successful, the Trust Router will check local policy to determine whether it should provision an identity for the Relying Party for the indicated purpose (details of this check may be implementation dependent).

4. If the request passes any policy requirements, the Trust Router will provision a temporary identity for the Relying Party within the Trust Router's local realm that can be used to access the next-hop Trust Router or RADIUS/RADSEC server in the Trust Path.

7. Security Considerations

TBD.

8. IANA Considerations

There are no IANA actions required for this document at this time.

9. Acknowledgements

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

10. References

10.1. Normative References

[I-D.lear-abfab-arch]
Howlett, J., Hartman, S., Tschofenig, H., and E. Lear, "Application Bridging for Federated Access Beyond Web (ABFAB) Architecture", draft-lear-abfab-arch-02 (work in progress), March 2011.

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

10.2. Informative References

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

Author's Address

Margaret 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