Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 12, 2012

M. Wasserman
Painless Security
H. Tschofenig
Nokia Siemens Networks
S. Hartman
Painless Security

July 11, 2011

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

draft-mrw-abfab-multihop-fed-01.txt

Abstract

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

This document introduces a new 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 an Identity Provider. They also provide temporary identities that can be used by a Relying Party to traverse a Trust Path.

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 12, 2012.

Table of Contents

1. Introduction 3
2. Terminology 4
3. Motivation 5
4. Multihop Federation Example 7
5. Trust Router Protocol 9
6. Trust Path Query 10
7. Temporary Identity Request 10
8. Security Considerations 11
8.1. Threat Model 12
8.2. Security Requirements 13
8.3. Data Origin validation and signatures 13
9. IANA Considerations 13
10. Acknowledgements 13
11. References 13
11.1. Normative References 13
11.2. Informative References 14
Authors' Addresses 14

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].

This document introduces a new ABFAB entity, the Trust Router. Trust Routers exchange information about the availability of Trust Paths across a multihop federation. ABFAB entity, the Trust Router. These paths are used by RPs to contruct transitive trust chains across a federation to a Radius or Diameter server within a target IdP.

A Trust Path consists of one or more Trust Links. A Trust Link is an assertion that a specific Trust Router is capable of providing temporary identies that can be used to access another entity in the ABFAB system. At this point, we anticipate that there will be two types of Trust Links in ABFAB: a Trust Link that indicates that one Trust Router can be used to reach another Trust Router, and a Trust Link that indicates that a Trust Router can be used to reach a Radius or Diameter Server. The first type (Trust Router Links) are shown as A->B(T), which indicates that the Trust Router in realm A can create identities to reach the trust router in Realm B. The second type (Radius/Diameter Links) are shown as A->B(R), to indicate that a trust router in Realm A can be used to reach a Radius, RadSec or Diameter server in Realm B.

Trust Routers exchange information about available Trust Links within a federation, and each Trust Router maintains a tree of available paths to reach all of the IdPs within the federation that can be reached from the local realm of the Trust Router.

When an RP receives a request from a party within a realm that not known directly to the RP, the RP will query its local Trust Router to obtain the best Trust Path to reach that IdP. Note that we use the term 'best' here to highlight that there may well be multiple paths to reach an IdP from a given RP, and the selection of the 'best' path

may involve several factors in addition to the length of the path, such as security and privacy practices, or monetary costs.

The RP will travers the Trust Path obtained from it's local Trust Router. At each step, the RP will request a temporary identity to access the next step in the Trust Path, contructing a transitive chain of trust to a Radius or Diameter server within the target IdP.

To summarize, the Trust Router performs three functions:

- Trust Routers peer with other Trust Routers to exchange information about available Trust Links, and Trust Paths. This

  information is exchanged between Trust Routers using the Trust Router Protocol. The Trust Router Protocol is described in more detail in Section 5.

- Trust Routers respond to queries from Relying Parties to make information about Trust Paths available. This exchange is referred to as a Trust Path Query Protocol, which is described in Section 6.

- To follow the Trust Path across a federation, the RP will use KNP to 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, which is described in Section 7.

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 Radius/Diameter 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 IdP.

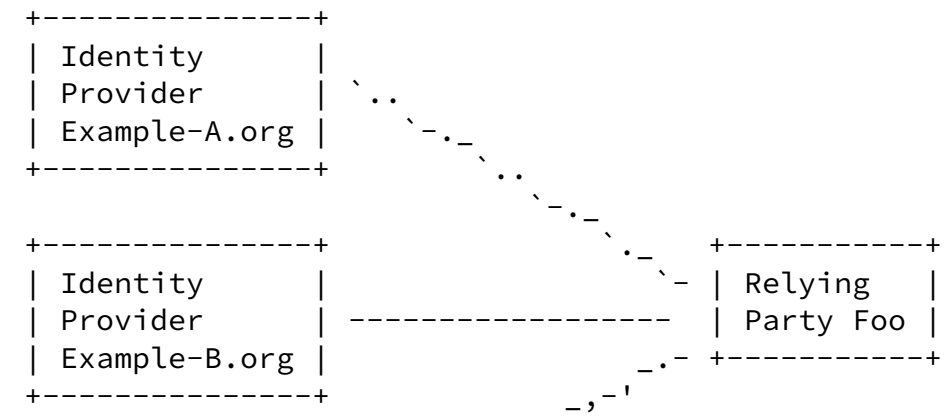
Trust Router Protocol:

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

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.



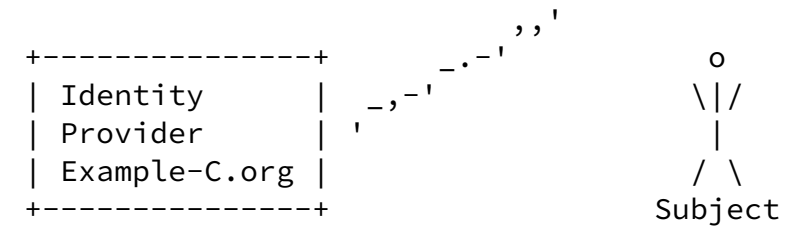
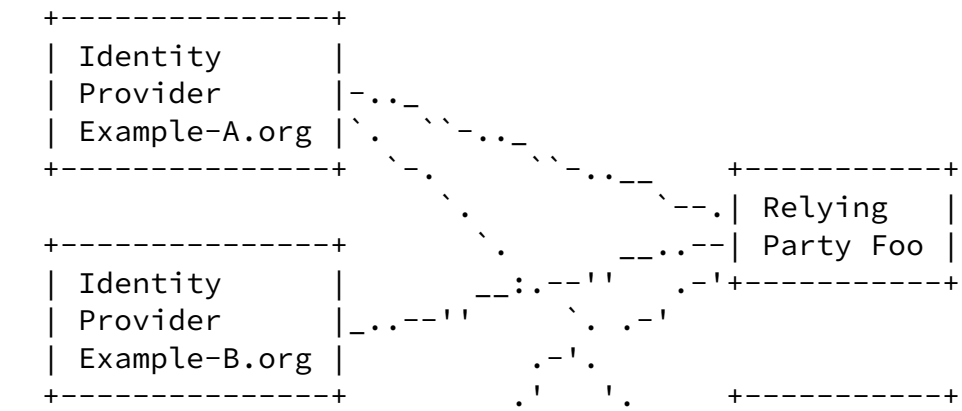


Figure 1: One-to-many Federation Example

When an RP receives a request to access a protected resource (or requires authentication and authorization 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.



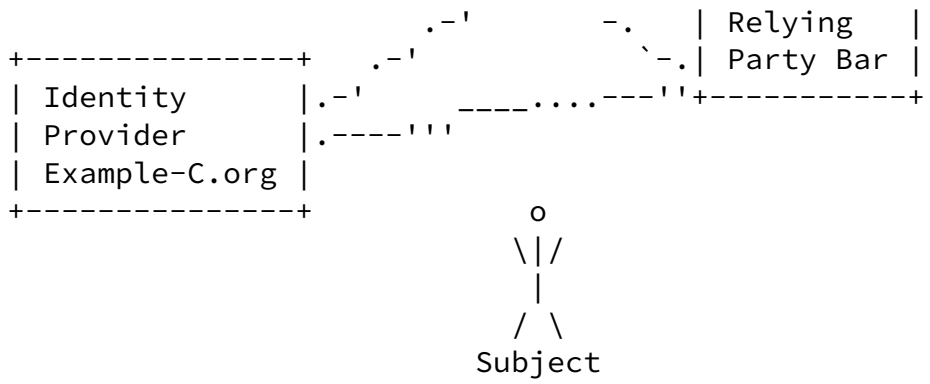
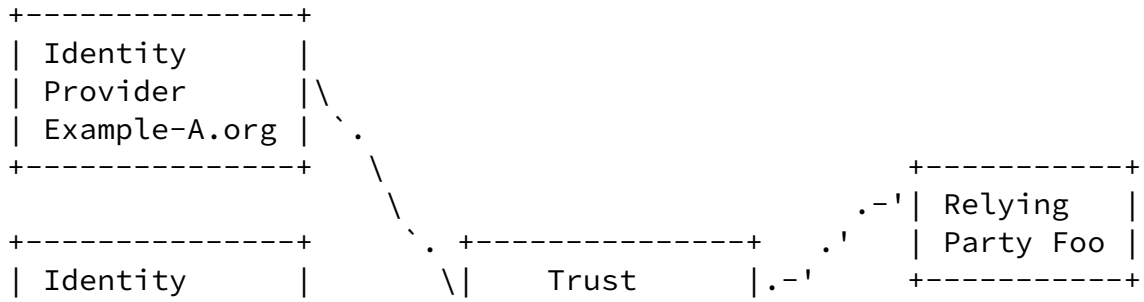


Figure 2: Mesh Federation Example

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.



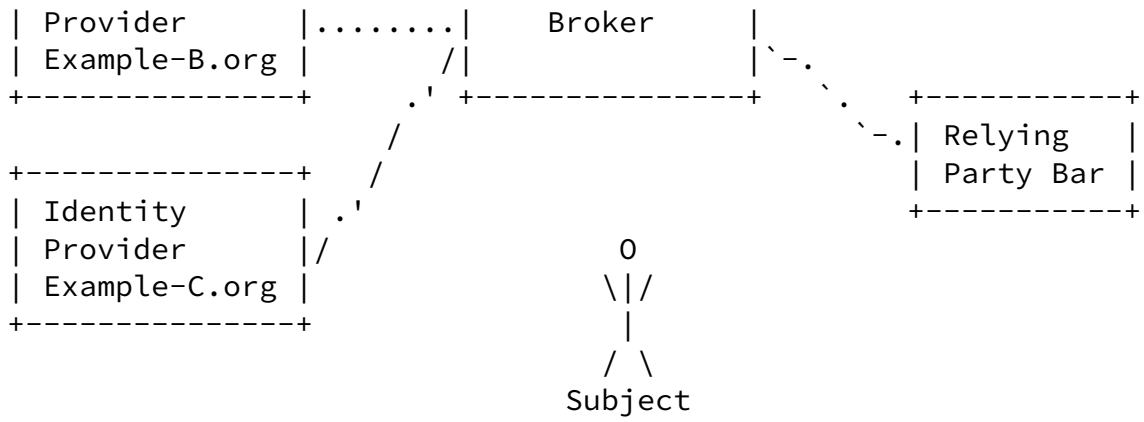


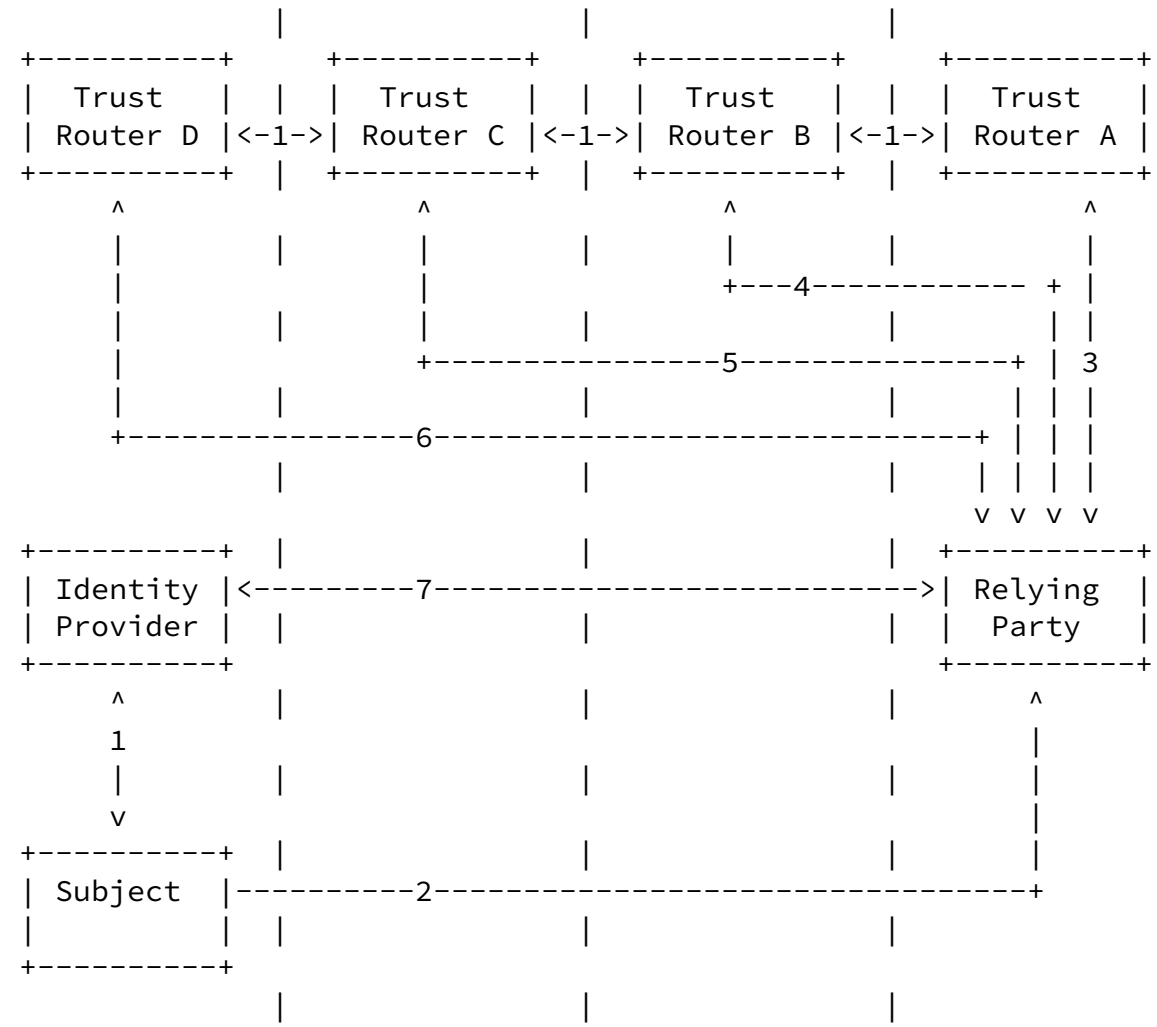
Figure 3: Federation Broker

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, and the Key Negotiation Protocol (KNP), as described in [I-D.howlett-radsec-knp].

4. Multihop Federation Example

The diagram below shows an example of a successful exchange in a multihop federation using the Trust Router Protocol and KNP:

Realm D | Realm C | Realm B | Realm A



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 Radius or Diameter server in Realm D that it can use to authenticate the Subject, 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 Subject.

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 Diameter 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].

5. Trust Router Protocol

Trust Routers use the Trust Router Protocol to exchange information about available Trust Links, and Trust Paths across a federation.

The Trust Router Protocol differs from an Internet Routing Protocol in a couple of important ways:

- Trust Links are unidirectional. It can not be assumed that the fact that a Trust Router in Realm A is authorized to create temporary identities to access a Trust Router in realm B, that the opposite is also true (A -> B(T) does not imply B->A(T)).

- Realm names are not necessarily hierarchical. Although aggregation might be possible as a later optimization, the ability to aggregate realm names based on shared roots is not currently assumed.

In addition to the existence of the links themselves, Trust Links have a set of associated attributes that can be used for filtering and tree computation, including:

- The cost of the link.

- Any security and privacy characteristics associated with the link.
- 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.

6. Trust Path Query

A Trust Path Query is generated by a RP 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 zero or more Trust Router steps and ends with a Radius or Diameter server within the IdP for the indicated realm.

The Trust Path Query is initiated by the RP, 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 RP, and check local policy information to determine whether or not to reply.

Assuming that the RP 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 Policy 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 or Radius/Diameter), 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.

7. 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 or Diameter infrastructure that can be used by the Relying Party to communicate with the Trust Router or Radus/Diameter server that represents the

next step in the Trust Path. Current thinking is that KNP will be used as the protocol mechanism for these requests.

These Temporary Identities will have a finite lifetime and, when authenticated, will include a Radius Attribute/Diameter AVP 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.

8. Security Considerations

As discussed in [I-D.lear-abfab-arch], the trust broker architecture is a mechanism for establishing technical trust in an ABFAB federation. Technical trust mechanisms have three primary responsibilities in ABFAB. They are responsible for integrity protection of AAA traffic. They are responsible for constraining the naming of ABFAB entities: for example the technical trust mechanism assures that the entity claiming to be the IDP is authenticated and authorized to act as the IDP for the realm containing the subject. The technical trust mechanism also determines where AAA messages are routed.

The trust broker architecture described in this document is designed

to meet the security and operational requirements of federations and groups of federations with large numbers of organizations. In these environments depending on any common credentials or trust mechanism does not make sense. While federations are expected to interconnect, they are not expected to have a common set of trust anchors for a public-key infrastructure. Each realm needs to be able to choose the appropriate credentials and security policies to use when establishing a relationship with another realm.

by design, this approach provides flexibility. Parts of the interconnected set of realms can use high-assurance processes and mechanisms including strong authentication mechanisms and rigorous credentialing and enrollment processes. Other realms can use lower-assurance mechanisms and processes, balancing cost and speed against security. However this flexibility complicates the security policy. Just because the local realm has a high-assurance trust link does not mean that the path is high-assurance. Operational mechanisms are required in order for RPs to express their security requirements and for the trust routers to make sure that resulting trust paths meet these requirements. Similarly, trust routers need to make sure that paths to a given IDP are not announced unless that IDP's security requirements will be met.

8.1. Threat Model

Like all Internet protocols, the trust router protocols and KNP need to have strong protection against parties who are not authorized to be part of an exchange. Such attackers do not start out knowing credentials necessary to participate in the system. However these attackers can be assumed to observe trust router, KNP, AAA and ABFAB exchanges. The system needs to maintain integrity of all data, confidentiality of keys and in some cases confidentiality of other data even when these attackers can insert, suppress, modify or replay packets. Reasonable defenses against attacks on the availability of the system are required, although obviously there are limits to these defenses. An attacker who can disrupt connectivity with a realm can impact availability.

The interesting threat model surrounds malicious participants authorized to participate in the system. The threat model is similar to that of routing protocols [I-D.ietf-karp-threats-reqs]. Defending against a compromised actor announcing a trust link that actor would be permitted to announce were it functioning correctly is out of scope. Similarly, defending against an compromised actor performing some action that actor is authorized to take is out of scope for this threat model.

However, it is a requirement that the system needs to provide tools

to limit the authorization of actors. For example if a particular session between two trust routers is not authorized to announce a trust link to a given realm or with certain properties, then attacks permitting such a link to be announced are in scope. Similarly an attack permitting a temporary identity with properties inconsistent with administrative limits would be in scope.

The system must permit zones of more or less trust to be created. An attack that permits insiders in the zones of less trust to compromise a zone of higher trust beyond what the zone of lesser trust is permitted is within the scope of threats. However, trust can only decrease as distance across the transitive network of trust routers increases. A peer two hops away cannot be permitted to make any statement that a peer one hop away cannot make. In general, it is unknown whether the peer two hops away actually made the statement.

8.2. Security Requirements

TBD

8.3. Data Origin validation and signatures

TBD

9. IANA Considerations

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

10. Acknowledgements

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

11. References

11.1. Normative References

[I-D.howlett-radsec-knp]
Howlett, J. and S. Hartman, "Key Negotiation Protocol for RadSec (KNP)", draft-howlett-radsec-knp-01 (work in progress), March 2011.

[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.

11.2. Informative References

[I-D.ietf-karp-threats-reqs]
Lebovitz, G., Bhatia, M., and R. White, "The Threat Analysis and Requirements for Cryptographic Authentication

of Routing Protocols' Transports",
draft-ietf-karp-threats-reqs-03 (work in progress),
June 2011.

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

Authors' Addresses

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

Hannes Tschofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

Phone: +358 (50) 4871445
Email: Hannes.Tschofenig@gmx.net
URI: http://www.tschofenig.priv.at

Sam Hartman
Painless Security
356 Abbott Street
North Andover, MA 01845

USA

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