Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 01, 2013

J. Schaad
Soaring Hawk Consulting

May 30, 2013

Trust Router Problem Statement
draft-schaad-trust-router-problem-01.txt

Abstract

There are a number of problems with using the current AAA framework for doing access control, this document looks at a number of these issues and poses some questions about how to create a new trust router system.

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 December 01, 2013.

Table of Contents

1. Introduction

The following document is a product of my diseased mind as opposed to the deceased minds of the Janet Group or Painless Security. Specifically, this document uses terms that are defined in [I-D.howlett-abfab-trust-router-ps] and in [I-D.mrw-abfab-trust-router] but in completely different ways. One therefor needs to be clear about which document one is looking at when referring to these documents as oppose to this one.

The ABFAB architecture as outlined by [I-D.ietf-abfab-arch] provides the necessary information for understanding the basic needs of ABFAB. As outlined in that document, there are still number of worries in terms of trying to implement the ABFAB architecture in the real world. These worries come out of the way that some of the basic components used by ABFAB where orignally designed to be used, and the way that they are actually used in the AFBAB architecture. This document examines some of the perceived short comings of those component and provides a basic paradigm that is believed to address those short comings. In this document a number of issues that any solution adopting that paradigm is also going need to be addressed are outlined.

This write up of the problem statement reflects to some degree a proposed solution to the problem, however it does so in the most general terms possible. Some pieces are fixed by the ABFAB architecture, such as the use of RADIUS clients and servers for user authentication, however other pieces are intentionally left nebulous (what does the inter AAA server routing look like). A basic picture of how the different pieces are put together can be found in Figure 1.

2. Justification

This section outlines a number of different areas where components of the current architecture are considered to be deficient.

2.1. AAA Routing Problems

The fist selection of problems that are examined are related to how routing is done in the AAA architecture. These issues are, in many cases, specific to the RADIUS version of AAA. The problems may or may not also apply to Diameter, however as ABFAB as currently deployed is focused on RADIUS rather than Diameter these issues still need to be addressed.

- There is expected to be a large amount of data that is passed between the EAP server and the EAP client as part of the validation process. This is a logical consequence of the fact that we will be using TEAP [I-D.ietf-emu-eap-tunnel-method] which is based on TLS and thus can contain large certificates, CRLs and other cryptographic information. Hosting this on top of UDP is not going to be successful in the long run, the amount of fragmenting both at the UDP level and at the RADIUS and EAP levels will lead to poor performance in many cases. The use of TLS/TCP [RFC6613][RFC6614], with it's session state and recovery will help this problem.

- Routing needs to be static rather than dynamic from end-to-end. Given that the routing is based on the conditions required for the set of messages under consideration, the policy enforcement needs to be consistent, allowing for routing of each packet in the stream independently means that enforcement of constraints cannot be consistently enforced. Again, the use of TLS/TCP helps if one assumes that sessions are kept up for longer

- Adding and removing destinations from the routing tables is hard as it must be done for every entity on the backbone.

- There is currently no way to apply policy to the routing of items based on such things as the LOA desired by the message.

2.2. AAA Security Issues

The next selection of problems to be examined are the security issues related to the AAA routing infrastructure is deficient to meet the needs of a truly secure system.

The following represents a set of reasons why the current RADIUS security is not adequate.

- Link level security is the current state of the art for RADIUS servers, however it is frequently missing on individual links and it is not possible for either end point to verify that link level security is provided over the entirety of the system.

- Link level security is currently configured on each hop in the link. It would be preferable that the security could be "centrally" configured.

- Different message need to be routed differently based on the level of security needed for the message. This is currently addressed by either having all of the links configured to be at the highest level of security or for there to be multiple links between different entities based on the different levels. Since the configuration is done on each pair of RADIUS entities, there is no easy way for a third party auditing service to either add or remove entities from the backbone based on an evaluation of their security practices.

- RADIUS security is monolithic in concept, this means that one cannot readily have multiple different communities with different needs use a single RADIUS network. The network would need to be configured at the highest needs, but that may not be is not necessary for all of the communities.

RADIUS dynamic discovery [I-D.ietf-radext-dynamic-discovery] has attempted to address some of these issues, however based on the discussions on the RADEXT mailing list this has not always been successful. Some of the issues found are:

- Setting up the correct security infrastructure based on X.509 certificates is still too high of a bar for some RADIUS server operators. This means that people are setting up proxy servers which are discoverer and these then talk to the actual RADIUS server.

- There are problems with how naming of entities should be done and what name checking needs to be done when comparing a RADIUS server with the contents of an X.509 certificate. This becomes even more complex when there are proxy servers in the path.

- The use of unsecured DNS records to do the lookup from the original AAA domain name to a server name is problematic as there is no reason to believe that this cannot be easily attacked by DNS cache poisoning.

2.3. X.509 Security Issues

Finally we look at the some issues with X.509 security. There is no way when using the ABFAB architecture to avoid the possibility of using X.509 certificates. The EAP method TEAP uses TLS, and thus will likely use an X.509 certificate, even if it is self-signed.

With in the community of ABFAB architecture proponents, is it an article of faith that the PKIX architecture [RFC5280] is broken and cannot be trusted. There are many components that go into this statement, however it is as much a statement of religion as it is of reality for the proponents of the ABFAB architecture. That being said, there are a number of reasons why this position can be taken.

- Correct and usable name formats are very difficult to get correct. This is especially true when proxying needs to be done for. As an example see [I-D.ietf-radext-dynamic-discovery] for why this is the case.

- Revocation processing and checking of PKIX systems is frequently missing.

- Trust Anchors are a problem when looking in these scenarios. One cannot generally have a single trust anchor due to political a trust problems, however having a multiplicity of trust points can yield problems when trying to decide who and why trust can be placed. Either one is dependent on a general purpose trust system

(i.e., the current set of commercial Certificate Authorities) or one needs to setup a special purpose CA for the requirements of the current infrastructure.

- Attribute certificates [RFC5755] have never been readily adopted as a way to convey attributes.

- Provisioning and restricting of trust anchors is proving problematic in many cases. This can even be seen in terms of how one should provision a clients computer with the appropriate trust anchor for doing EAP validation in the case of using TEAP.

3. Trust Router Objectives

There are three main objectives for this work:

1. Publish the location of AAA servers in a secure manner to all AAA clients within a trusted network.

2. Create a mechanism to allow for creation of sub-communities within a AAA system.

3. Create a temporary "short-lived" direct link with a DH or ECDH key pair for validation between an AAA client/proxy (near the service) and the AAA server for the user (near the IDP). The link created will carry policy information for governing the release of information from the IDP to the AAA client.

4. Entities in the system

This section documents a number of basic entities that are participating in the trust router system. Some of these entities are included for completeness as they are part of the ABFAB architecture, but are not direct participants in the Trust Router architecture.

4.1. End User

The end user is the entity that is requesting a service from the service provider.

The end user has no pre-existing relationship with the service provider. The end user does have a pre-existing relationship with an IDP. The relationship with the IDP will include the ability for a set of cryptographic credentials to be used to validate the user to the IDP. This validation is done using one or more EAP methods.

4.2. Service Provider

The Service Provider is used by the end user to get some service. The Service Provider has a pre-exsting relationship with a local AAA proxy. The Service Provider itself will be a AAA client.

There is an issue with the current AAA system that needs to be examined, the ABFAB architecture requires that the AAA proxy that the SP connects to validate information about the SP. With a large number of SPs being added to and removed from the AAA network, we need to look for a way of using the AAA architecture itself to do the validation of the SP rather than using the configuration of public keys in the AAA proxy. This can be done with a X.509 certificate architecture, however this would not match the overall principle of not using X.509 certificates. The use of EAP and a AAA server to do the authentication and attribute checking would make the administration of SPs and End Users similar which would reduce administrative problems.

4.3. Community of Registration (COR)

At its most basic level, a Community of Registration (COR) consists of a set of entities and an IDP that can authenticate those entities. A COR operator has a specific set of requirements about how entities are to be initially identified, how they are to be authenticated each time they appear and what information is to released to third parties upon request. A COR operator may have a multiplicity of such requirements based on internal policies and requests from service providers. A Level Of Assurance (LOA) is one of the pieces of information that a COR would have in this set of requirements.

4.3.1. Atomic CORs

An Atomic COR consists of a single database of registration. It may consist of one or more on-line presences, but each on-line presence is required to produce exactly the same results.

4.3.2. Mesh CORs

The following is a potential feature and may not make it into the final output.

It makes sense to allow for CORs to be aggregated together into a single unit, thus for example the University of Washington could run a mesh COR that comprises of the CORs for the undergraduate school, the law school, the medical school, etc..

There is a known privacy issue with allowing the existence of mesh CORs, multiple correlated queries can be constructed that can leak information about which COR an entity is associated, even if that information is not directly provided to the SP.

4.3.3. Release of Information

The main function of a COR in the ABFAB architecture is to release information about the end user to the service provider. The smallest amount of information that can be provided is to say that the COR does or does not recognized the end user. At the larger end of infomration provided, the COR can responde to an SP request with a large number of attributes about the end user as part of a SAML statement.

The decision of releasing attributes about the end user is an important issue, the least possible amount of information should generally be released. If the user can participate in the decision to release information then that should be encouraged, such participation is not always possible. The release of information should always be in accordance with the policy of the IDP and, in many cases, should be an auditible function.

There is a need to look at how policies are to be provided for both external review and auditing. It is not clear that this is a strong requirement ala the CPS framework that PKIX created [RFC3647]. However, the more standardized and machine readable this information is, the simpler it would be for tools to be able to process and look at this information. This may be an issue when starting to look at how things such as attributes are mapped when crossing trust boundaries.

4.4. Communities of Interest (COI)

One of the goals of this work is to allow for the formation of closed subsets of users and services within the overall trust architecture that is created. These closed subsets are called Communities of Interest.

A COI is defined by the following properties:

- The name of the COI. COIs are expected to be uniquely named within some domain. The domain that is used and how that is expressed needs to be determined.

- Version control on the COI. This may be some type of monotonically increasing value or a time of last update indicator. If there are multiple possible configuration locations in the

system then a time value may be better as collisions on a counter could collide. In any event there may be a requirement for detection of update collisions.

- The set of users that can participate in the COI. The set of users are defined by the use of one or more attribute. These attributes can include the name of the user (expressed as an NAI), the COR for a group of users (i.e. everyone at the University of Washington) roles, departments and so forth. The method of expressing the attributes needs to be defined, however one of the issues that needs to be dealt with is that fact that the attributes are frequently dependent on the COR that issues them. This means that attributes will either need to be defined by the COI, the trust backbone, a global attribute definier or have the ability to some type of mapping of attribute names.

- The set of security properties that are required for users. Even when a user might be able to participate in a COI, the location of the user and the methods of authentication used by the user may rule out participating in the COI at a given moment. The security proprities represent a set of dynamic properpties based on how the user is attached to the network rather than relatively static properties that the COR will maintain over time. The security properites may also represent tasks that the COR is requried to perform during the authentication. An example would be that a COI requires a specific LOA to be used in authenticating the user. This is a property of the COI and not a property of the user.

- The set of service providers that are permitted to use the COI. SPs may be specified by name or other attributes in a similar manner to that done for users.

- The set of security properties that are required for SPs. This is similar to the set of security proprieties that are required for users.

- The set of security properties that are required for the Trusted Introducers. This is similar to the set of security properties that are required for users.

COIs are managed in a central location rather than being distributed through the system. It is presumed that the management of COIs is connected to the management of Trust Introducers, but that is an issue that will need to be resolved by the protocol.

It can be seen that ability to make COIs be hierarchical would be a convenient practice. As an example, a COI could be maintained for every physical location of the University of Washington. It would

then be able to group those COIs in a hierarchical manner grouping by larger and larger locations until the entire school is covered. This means that if a new user or service provider were added to one of the leafs in the tree then it automatically propagates into all of the nodes above it in the tree without additional administrative work.

4.4.1. Security Issues

There are a number of security issues that will need to be addressed:

- Should a COI be able to coop a COR without the consent of the COR.
- Depending on how COIs are defined, they can be turned into oracles about the members of CORs.
- If an SP can use multiple COIs, then it needs a way to select which COI to use for any single transaction. The choice of the COI then needs to be provided to the Trust Backbone.
- There are no provisions for the existence of a COI to be published to either SPs or users. Does there need to be a method for doing so?
- When COIs are propagated around the trust backbone, does the data about them need to be kept confidential. While the existence of a COI is probably not a big security risk, knowledge of the security parameters and entity attributes about the users of the COI may constitute a security risk.

4.5. Trust Backbone

The trust backbone is a generic term that is being used to designate the network of systems which are responsible for connecting the different CORs together. A trust backbone can come in two flavors: Intra backbones are maintained by a single entity and operate at a single level of security. Inter backbones consist of multiple intra backbones with special systems that operate on the boundaries between the different intra backbone to mediate the differences in security practices.

Information flows both within a single backbone and between different backbones. Within a single backbone, all information flows unmodified. However when information crosses between backbones it is frequently modified to deal with differences in policies or simply prevented from passing across the boundary.

A trust backbone will normally have multiple CORs in it. A trust backbone is the location where COIs are introduced into the system.

4.6. Trusted Introducers

The concept of a trusted introducer has been around for a long time. This is the basic method by which webs of trust are created. The basic model that is to be used will be based on a web of trust and trusted introducers.

The trusted introducer subsystem is the method where a direct link is going to be established between the AAA proxy near the SP and the AAA server for the end user.

4.6.1. Trusted Introducer Initiator

The Trusted Introducer Initiator (TII) is expected to be integrated into the AAA proxy that is adjacent to the SP. The TII is the location where the introduction protocol is kicked off. The TII is required to do the necessary enforcement of the SP identity and attributes with the security properties of the backbone and the COI selected by the SP.

4.6.2. Trusted Introducer Router

The Trusted Introducer Router (TIR) is the basic routing element of the trusted introducer network. TIRs come in two flavors, internal and boundary TIRs. The internal TIR is a simple thing that just does routing and the necessary security enforcements. A boundary TIR on the other hand is responsible to dealing with all of the security problems that are needed with crossing a security boundary.

4.6.3. Trusted Introducer Target

A Trusted Introducer Target (TIT) is expected to be integrated into the AAA server. The TIT is the target of the trusted initiator protocol. The TIT is required to do the necessary enforcement of security parameters that are imposed by the AAA server and then return the necessary information for the AAA proxy associated with the TII to setup a direct TLS link between it and the AAA server.

Once a TLS link between the AAA server and the AAA proxy has been established, it may be used for more than one AAA protocol exchange. This means that it is a requirement that the AAA server apply all of the security information setup on the TLS link be enforced on each AAA protocol exchange.

5. Communication Flows

There are going to be three different sets of information that will need to flow through the system. We examine each of those flows and the properties that are needed.

5.1. COI Distribution

The system will need to distribute information about all COIs from the centralized configuration points to all of the AAA entities in the system.

5.2. COR Connectivity

The system will need to distribute the information necessary to build a path from any specific Trust Introducer to any given COR. In point of fact, there may be CORs in the system that will not be reachable from a specific Trust Introducer due to security constraints on the distribution of COR connectivity.

One of the interesting questions that will need to be explored is: Can the COR and COI information be distributed independently and combined on the AAA systems or does it need to be combined by the Trust Brokers that are doing the distribution of this information.

5.3. AAA Entity Introduction

This flow of communication represents the final goal of the protocol. We want to be able to build up a TLS connection between the AAA proxy that resides nearest to the SP and the AAA server that is going to be used to validate the user.

In building the connection between the proxy and the server the following will need to be taken into account:

- The identity of the service provider.
- The security properties of the connection between the service provider and the AAA proxy.
- The COI that will govern the connection.
- The possible routes between the AAA proxy and the AAA server and their security properties.

Note that no information about the user except the target COR is used in the path construction as no such information is both available and reliable. Until the authentication between the user and the COR has completed the network will have no idea about the user except for the claimed COR.

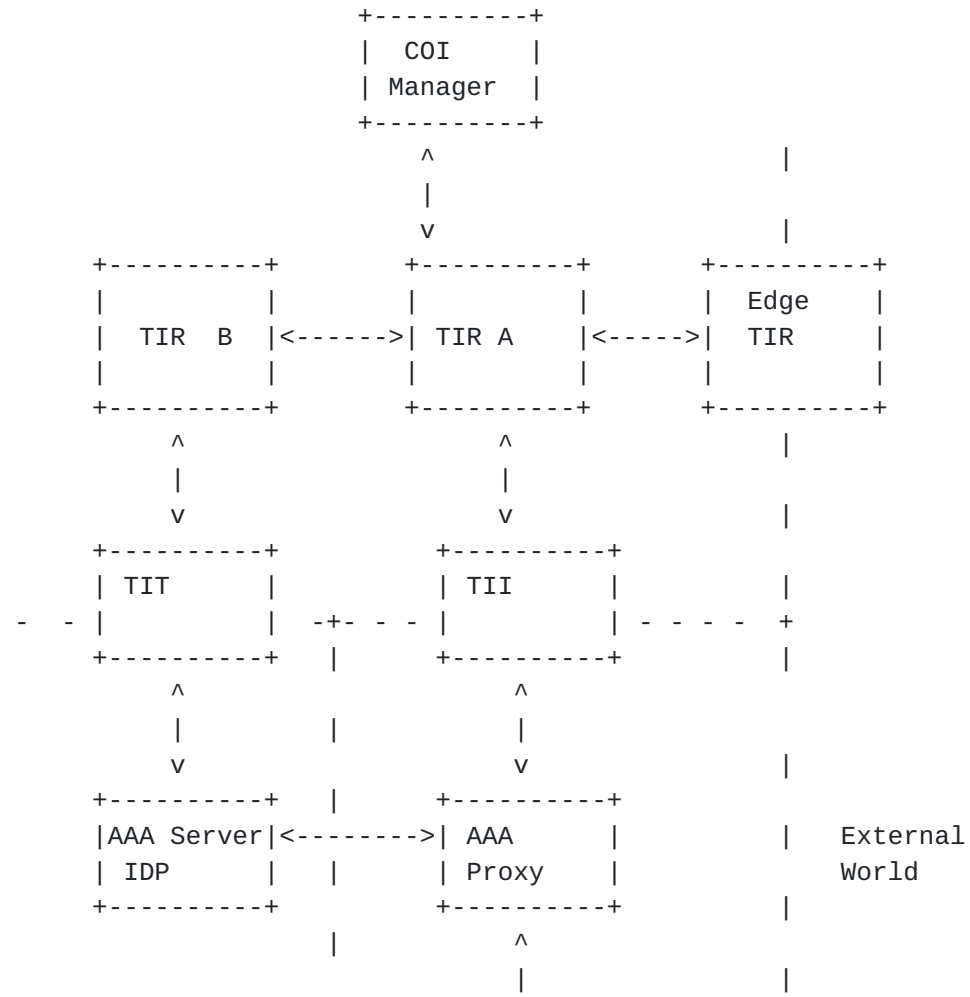
6. Putting it all together

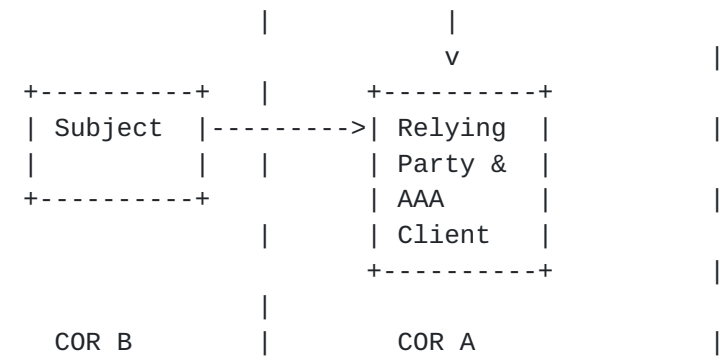
In this section there are a number of pictures of different configurations that are germane to the problem

6.1. Scenario - One Trust Backbone

In this scenario, we are looking at what is the basic roll out that will be done. In this scenario there are four different security zones:

- The trust backbone,
- The COR for organization A,
- The COR for organization B,
- The external world





Multiple CORs - One Trust Backbone

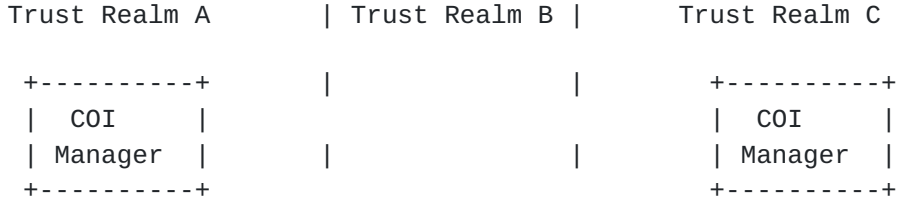
There may be a need to create a different scenario for discussion about what happens if there is a single COR. Specifically, on needs to look at the question: Does one still need to go through a AAA proxy and the trusted introducer protocol or can the SP go directly to the AAA Server? If one goes directly, then there are some security gateways that might not get checked.

6.2. Scenario - Three Trust Backbones

In this scenario, we are looking at a more complex roll out. In this scenario there are five different security zones:

- The trust backbone for A,
- The trust backbone for B,
- The trust backbone for C,
- The COR D,
- The COR E.

In the picture, there is a distinction between the internal and the edge trust routers. In an actual roll out there may not be distinct components. However for conversation purposes it is easier to keep then separate.



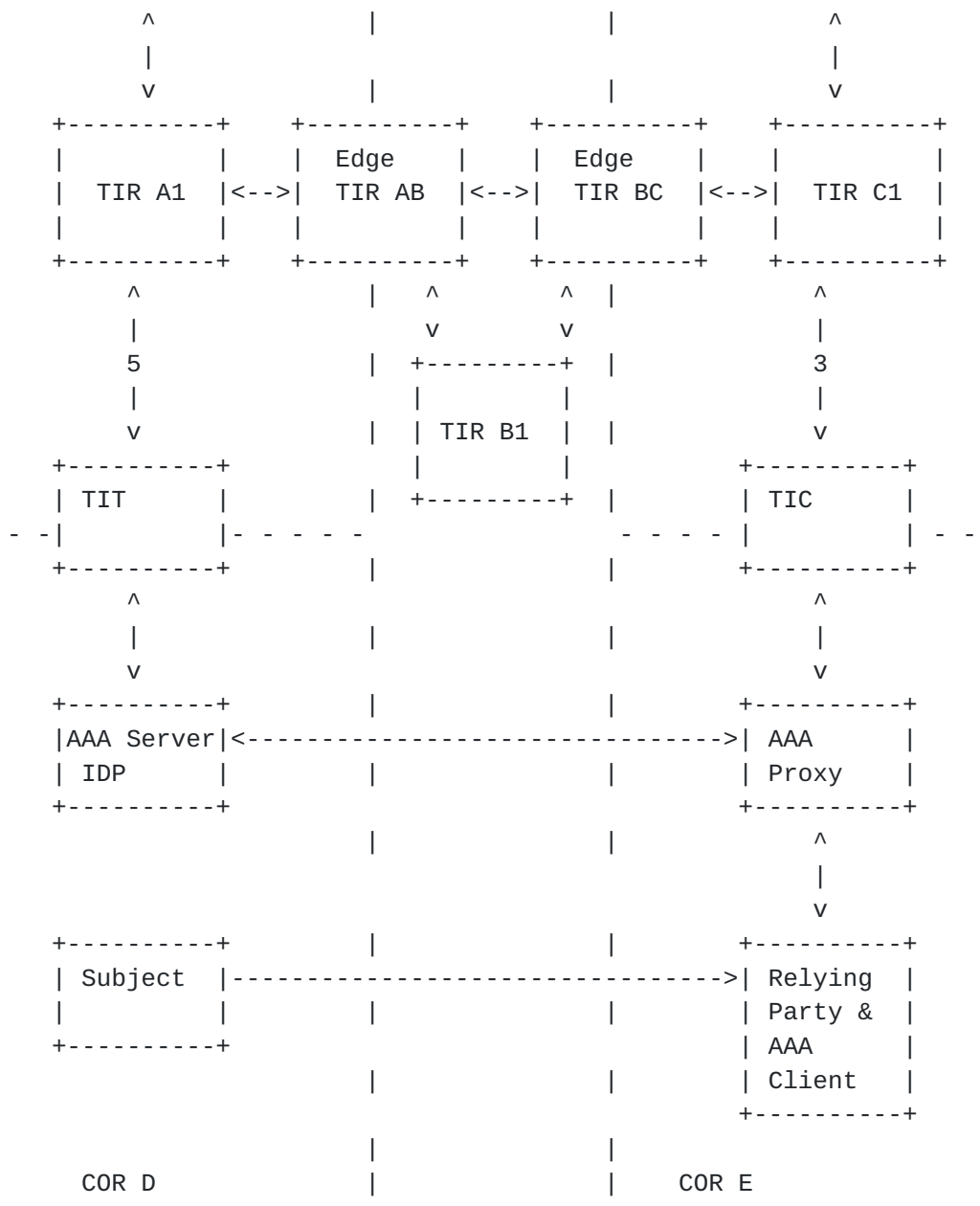


Figure 1: Architecture

7. Security Considerations

This document does not define a protocol and as such has no direct security considerations. The document does pose a number of questions dealing with security that will need to be addressed by a protocol that implements the problem stated here.

8. IANA Considerations

This document has no IANA considerations.

9. Acknowledgments

This document is an expansion of an email message that was originally sent to Alan DeKork and was probably passed on to others.

10. References

10.1. Normative References

[I-D.ietf-abfab-arch]
Howlett, J., Hartman, S., Tschofenig, H., Lear, E., and J. Schaad, "Application Bridging for Federated Access Beyond Web (ABFAB) Architecture", draft-ietf-abfab-arch-06 (work in progress), April 2013.

10.2. Informative References

[I-D.howlett-abfab-trust-router-ps]
Howlett, J., Smith, R., and M. Wasserman, "Trust Requirements in a Federated World", draft-howlett-abfab-trust-router-ps-03 (work in progress), March 2013.

[I-D.mrw-abfab-trust-router]
Wasserman, M., Hartman, S., and J. Howlett, "Application Bridging for Federation Beyond the Web (ABFAB) Trust Router Protocol", draft-mrw-abfab-trust-router-02 (work in progress), February 2013.

[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.

[I-D.ietf-radext-dynamic-discovery]
Winter, S. and M. McCauley, "NAI-based Dynamic Peer Discovery for RADIUS/TLS and RADIUS/DTLS", draft-ietf-radext-dynamic-discovery-06 (work in progress), February 2013.

[I-D.ietf-emu-eap-tunnel-method]
Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel EAP Method (TEAP) Version 1", draft-ietf-emu-eap-tunnel-method-05 (work in progress), February 2013.

[RFC5755] Farrell, S., Housley, R., and S. Turner, "An Internet Attribute Certificate Profile for Authorization", RFC 5755, January 2010.

[RFC6613] DeKok, A., "RADIUS over TCP", RFC 6613, May 2012.

[RFC6614] Winter, S., McCauley, M., Venaas, S., and K. Wierenga, "Transport Layer Security (TLS) Encryption for RADIUS", RFC 6614, May 2012.

[RFC3647] Chokhani, S., Ford, W., Sabett, R., Merrill, C., and S. Wu, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", RFC 3647, November 2003.

Author's Address

Jim Schaad
Soaring Hawk Consulting

Email: ietf@augustcellars.com