SDNRG   R. Marin-Lopez
Internet-Draft   G. Lopez-Millan
Intended status: Experimental   University of Murcia
Expires: May 14, 2016   November 11, 2015

Software-Defined Networking (SDN)-based AAA Infrastructures Management
draft-marin-sdnrg-sdn-aaa-mng-00

Abstract

This document describes the management of Authentication, Authorization and Accounting (AAA) infraesctrutures by means of a Software-Defined Network (SDN) controller and raises the requirements to support this service. It considers the management of AAA routing and the establishment of security associations between AAA entities.

Status of This Memo

This Internet-Draft will expire on May 14, 2016.

Copyright Notice

Table of Contents

1. Introduction

Software-Defined Networking (SDN) is an architecture that enables users to directly program, orchestrate, control and manage network resources through software. SDN paradigm relocates the control of network resources to a dedicated network element, namely SDN controller. The SDN controller manages and configures the distributed network resources and provides an abstracted view of the network resources to the SDN applications. The SDN application can customize and automate the operations (including management) of the abstracted network resources in a programmable manner via this interface [RFC7149][ITU-T.Y.3300] [ONF-SDN-Architecture][ONF-OpenFlow].

Authentication, Authorization and Accounting (AAA) [RFC2903][RFC2904] infrastructures manage three basic security services to control the access to network resources: Authentication, in order to determine who the end user is; Authorization, in order to determine under what conditions an end user is granted access to the network resource; and Accounting, in order to account the resource usage of the end user.

Following the terminology in [RFC6733], an AAA infrastructure is formed by AAA nodes. An AAA protocol is used to exchange AAA information between them. RADIUS [RFC2865], [RFC2866] and Diameter [RFC6733]. These AAA nodes can be classified as follows:

- AAA client. It is a AAA node that implement the client part of a AAA protocol.

- AAA server. AAA node that handles authentication, authorization, and accounting requests for a particular realm.

- AAA agent. AAA node that implements one of the following functionalities:

-

  * Relay Agents. They are agents that accept requests and route messages (making use of a routing table) to other nodes based on information found in the DIAMETER messages.

  * Proxy Agents. Similarly to relays, proxy agents also route messages using a routing table. However, they differ since they modify messages to implement policy enforcement.

  * Redirect Agents. Redirect agents do not relay Diameter messages, they return an answer with the information required by the agents to communicate directly, so they do not modify messages. They are useful in scenarios where routing configuration needs to be centralized.

  * Translation Agents. A translation agent is a device that provides translation between two protocols (e.g., RADIUS<->Diameter, TACACS+<->Diameter).

As depicted in Figure 1, the AAA framework [RFC2903] defines a model consisting of the User desiring gain access to a specific network service; the AAA server in the User Home Organization (UHO) which has registered the User's identity and credentials; and the AAA server located in the Service Provider (SP) operating and controlling the access to the network service through a Service Equipment. In non-federated environments, User Home Organization and Service Provider are the same organization. In federated environments, they are two separate organizations.

Between the AAA client and the AAA server in the UHO, the AAA infrastructure can deploy other intermediate AAA agents to forward the information between both entities. RADIUS defines, apart from the RADIUS client and RADIUS server, the role of proxy RADIUS or forwarding RADIUS. Proxy RADIUS receives authentication requests from a RADIUS client, forwards the request to a remote RADIUS server, receives authentication responses from the remote server and forwards the response to the client. In Diameter, apart from the AAA client and the AAA server, it defines a set of these Diameter agents that corresponds with the agents described above.

It is important to note that the AAA node can act as AAA server in a realm but also can act as, for example, Relay/Proxy agent for those types of AAA requests that cannot be processed and need to be forwarded to the next AAA node. It is also important to note that some kind of trust relationship is required to be established between these AAA nodes, in order to protect the AAA traffic

Typically, Proxy RADIUS and Diameter agents (from now on AAA agents) hold and manage AAA routing information. Moreover AAA infrastructures are manually configured, specially the AAA routing information. For example, RADIUS implementations typically require the name or address of servers or clients be manually configured, besides passwords or cryptographic material to establish a security associations between AAA entities. It makes difficult their management and generates a lack of flexibility, specially if the number of entities is high and the AAA infrastructure is complex. With the grow of SDN-based scenarios where network resources are deployed in an autonomous manner, a mechanism to manage AAA infrastructures according to the SDN paradigm becomes more relevant. Thus, the SDN-based service described in this document deals with AAA infrastructures in such as an autonomous manner.

```
                     +------+      +--------------------------+
                     |      |      | User Home Organization   |
                     |      |      |  +--------------------+  |
                     |      |      |  |    AAA Server      |  |
                     |      |      |  |                    |  |
                     |      |      |  +--------------------+  |
                     |      |      |                          |
                     |      |      +--------------------------+
                     |      |
                     |      |
                     |      |
                     | User |      +--------------------------+
                     |      |      | Service Provider         |
                     |      |      |  +--------------------+  |
                     |      |      |  |    AAA Server      |  |
                     |      |      |  |                    |  |
                     |      |      |  +--------------------+  |
                     |      |      |                          |
                     |      |      |  +--------------------+  |
                     |      |      |  |    AAA Client      |  |
                     |      |      |  |--------------------|  |
                     |      |      |  |      Service       |  |
                     |      |      |  |     Equipment      |  |
                     |      |      |  +--------------------+  |
                     |      |      |                          |
                     +------+      +--------------------------+
```

Figure 1: AAA framework

2. Requirements Language

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]. When these words appear in lower case, they have their natural language meaning.

3. Terminology

This document uses the terminology described in [RFC7149], [RFC6733], [ITU-T.Y.3300], [ONF-SDN-Architecture], [ONF-OpenFlow], [ITU-T.X.1252], and [ITU-T.X.800]. In addition, the following terms are defined below:

o Software-Defined Networking: A set of techniques enabling to directly program, orchestrate, control, and manage network resources, which facilitates the design, delivery and operation of network services in a dynamic and scalable manner [ITU-T.Y.3300].

4. Objectives

- Flow-based AAA traffic routing: SDN-based management allows to route AAA traffic (flow) between different AAA agents so that the authentication, authorization and accounting tasks can be preformed. Thus, it can adapt quickly the routing information when a AAA route is not available to redirect the AAA traffic to other nodes.

- Bootstrapping security associations: SDN-based flow protection allows the centralized bootstrapping of credentials to protect the AAA traffic between two adjacent AAA nodes (hop-by-hop) or between two separate AAA nodes without intermediate AAA nodes in between (end-to-end).

5. AAA flow based routing

As mentioned, the AAA infrastructure is formed by a set of AAA nodes. These nodes interconnect each other using the AAA protocol, such as RADIUS or Diameter. When the User desires to access a service which requires authentication and authorization, she typically sends her identity to the Service Equipment. The Service Equipment (e.g. an WiFi access point) deploys the AAA client to interact with the AAA infrastructure. The User's identity contains the realm where she belongs to. For example, an identity can be "alice@um.es". The identifier is "alice" and the realm is "um.es". Based on this realm (realm-based routing), the AAA agents, for example, AAA relay or proxy, route the AAA information to the specific AAA server in the User Home Organization that has a register of "alice@um.es". This AAA server will be in charge of authenticate and authorize the Users in the realm. In order to achieve a correct routing of AAA information each AAA agent maintains a AAA routing table and another table with the adjacent AAA servers, which are considered as next "AAA hop".

As an example of AAA routing table, Diameter defines a format where each table entry contains the following fields:

Realm Name

This is the field that MUST be used as a primary key in the routing table lookups. Note that some implementations perform their lookups based on longest-match-from-the-right on the realm rather than requiring an exact match.

Application Identifier

A Diameter Application (i.e. NAS-REQ, EAP, etc.) is identified by an Application Id. The Diameter message can have a different destination (route entry) based on the Application Id in the message header. This field MUST be used as a secondary key field in routing table lookups.

Local Action

The Local Action field is used to identify how a message should be treated. The following actions are supported:

1. LOCAL - Diameter messages that can be satisfied locally, and do not need to be routed to another Diameter entity.

2. RELAY - All Diameter messages that fall within this category MUST be routed to a next hop Diameter entity that is indicated by the identifier described below. Routing is done without modifying any non-routing AVPs.

3. PROXY - All Diameter messages that fall within this category MUST be routed to a next Diameter entity that is indicated by the identifier described below. The local server MAY apply its local policies to the message by including new AVPs to the message prior to routing.

4. REDIRECT - Diameter messages that fall within this category MUST have the identity of the home Diameter server(s) appended, and returned to the sender of the message.

Additionally, Diameter specification defines the Peer Table, which is used for message forwarding, and referenced by the Routing Table. A Peer Table entry contains the following fields:

Host identity

The name of the peer (i.e. Diameter URI of the peer).

StatusT

This is the state of the peer entry.

Static or Dynamic

Specifies whether a peer entry was statically configured or dynamically discovered.

Expiration time

Specifies the time at which dynamically discovered peer table entries are to be either refreshed, or expired.

TLS/TCP and DTLS/SCTP Enabled

Specifies whether TLS/TCP or DTLS/SCTP is to be used when communicating with the peer.

Thus, the general idea presented in the document assumes that a SDN controller can manage and fill this routing information in the different AAA entities under its control. In particular, a set of tasks (not exhaustive) that the SDN controller can perform over the AAA infrastructure is the following:

- The SDN controller can provide this AAA routing information to the AAA agents so having a dynamic AAA routing. In general, the SDN controller can fill the AAA routing and peer table of any AAA agents and servers, but also the AAA client to indicate the next AAA hop. This brings all the existing advantages right now for general IP routing with SDN paradigm to AAA routing. That is, it provides flexibility, scalability, availability, and security.

- AAA entity and its adjacent ones typically keep a security association (hop-by-hop security). For example, RADIUS has a simple and weak model based on shared secrets. Extensions such as RADSec [RFC6614] has been proposed for running TLS between two RADIUS servers. Diameter mandates the usage of TLS and optionally IPsec. To avoid manual configuration of these security associations, the SDN controller can dynamically provide this cryptographic information to both interacting AAA servers.

- When forming AAA federations, security is usually provided hop-by-hop, which means that, between each pair of neighboring AAA entities, the AAA protocol provides message protection. Sometimes, for example in RADIUS, this protection is based on message authentication, but not message encryption. So these intermediate AAA entities can see all the information in clear. To avoid this, end-to-end security MAY be required. With a SDN-approach we foresee achieving the following. When the SP's AAA server observes that the User belongs to a different realm that it

does not know (managed by the UHO's AAA server), it can inform the SDN controller. The SDN controller can then enforce cryptographic material to both the UHO's AAA and SP's AAA server to establish a direct security association between them. Thus, in a simple architecture, the SDN controller would act as the manager of the federation. Nevertheless, more complex cases can be envisages such as each AAA server is managed by a different SDN-controller.

o The SDN controller can modulate the behaviour between a relay, proxy and translation agent. For example, the SDN controller can enforce routing tables but not sending any particular policy to the agent, so that it behaves as a relay agent. However, if a proxy is required the particular behaviour is enforced by the controller based on internal policies. For certain routes the AAA agent can act as a Redirect Agent. In fact, in relationship with Network Function Virtualization (NVF) and Network Security Function (NSF), we envisage that the SDN controller can order the instantiation of a particular AAA agent (VNF) when it is required and "places it" in the path of the chain of AAA agents where the AAA requests and responses have to travel. For example, if the SDN controller realizes the request is a RADIUS message and it has to traverse a Diameter-based AAA infrastructure, it can instantiate a Translation Agent, so that the requests and responses are automatically translated from and to the different AAA protocols. Once it is instantiated, the SDN controller can install a routing entry in the RADIUS proxy so that the AAA request goes through the Translation Agent.

o The SDN controller MAY also manage the User's credentials. In other words, the SDN controller can store all the User information provided by the administrator. When the AAA server is going to act as UHO's AAA server for a set of services and users, it will require this information to complete the authentication and authorization steps. The SDN can proactively push this information to the AAA server. Or, reactively, when the AAA server does not know how to authenticate and authorize a request it can ask for the advice of the SDN controller and the controller can enforce the behaviour and the user's 'information.

o The SDN controller can also select the AAA server in charge of (exclusively) Accounting. This accounting information can be also route to the specific AAA server.

NOTE: In general, the SDN controller MAY make use of NETCONF and YANG model for AAA entities to configure them.

6. Scenarios

This section explains two main use cases as examples for the SDN-based AAA management: first, when a single SDN controller is used; second, when multiple SDN controllers take place in the infrastructure.

6.1. AAA routing and security association establishment

As depicted in Figure 2, the SDN controller represents the "AAA control plane" where the decisions of AAA routing are taken based on AAA policies provided by the administrator (1). Once, the SDN controller interprets these policies (2), it sends the AAA routing information ("AAA data plane") to the AAA entities (e.g. Relay, Proxies or Redirect) to carry out the forwarding of the AAA request. This information is used to fill the AAA routing table and peer table of AAA agent 1 and AAA agent 2 (3). Associated to this routing information any security credential is also provided (4) by the SDN controller to establish hop-by-hop security (5). In general, the SDN controller can fill all the required information to the different AAA agents that form the AAA routing path to the destination.

```
                        +----------------------------------------+
                        |             SDN Controller             |
                        |                                        |
                     (1)|  +--------------+ (2)+--------------+  |
                AAA ------>| AAA Routing/ |--->| South. Prot. |  |
               Policies |  | Key Distr.   |    |              |  |
                        |  +--------------+    +--------------+  |
                        |    AAA Control Plane     |     |       |
                        |                          |     |       |
                        +--------------------------|-----|-------+
                                                   |     |
                                                   | (3) |
                        |--------------------------+ (4) +---|
           AAA          V AAA Data Plane                     V    AAA
           request  +-------------+                +-----------+request
            ------->|    AAA      |===============>|   AAA     |------>
                    |  agent 1    |  hop-by-hop    |  agent 2  |
                    |             |    security    |           |
                    +-------------+      (5)       +-----------+
```

Figure 2: Example of AAA agent-to-agent routing.

Figure 3 represents the case where end-to-end security (5) is established. In this case, the SDN controller enables credential and AAA routing information so that the AAA request goes directly between

the SP's AAA server and the UHO's AAA server without passing through any intermediate AAA entity.

```
                       +----------------------------------------+
                       |             SDN Controller             |
                       |                                        |
                    (1)|   +--------------+ (2)+--------------+ |
               AAA --------| AAA Routing/ |--->| South. Prot. | |
               Policies.   | Key Distr.   |    |              | |
                       |   +--------------+    +--------------+ |
                       |     AAA Control Plane    |          |  |
                       |                          |          |  |
                       +--------------------------|----------|--+
                                                  |          |
                                       (3) (4)    |          |  (3)(4)
                             |--------------------+          + -----|
                             V  AAA Data Plane                      V
          AAA request  +----------------+                  +-----------+
                ------>|    SP's AAA    |                  | UHO's AAA |
                       |     server     |=================>|   Server  |
                       +----------------+       (5)        +-----------+
                                            end-to-end
                                             security
```

Figure 3: Example of AAA server-to-server routing (end-to-end security).

Finally, Figure 4 describes the case of pushing AAA routing information only when really required (reactive). Let us assume that the administrator has provided general AAA policies (1). When a initial AAA request arrives the first time to the AAA agent 1, it notifies about the request to the SDN Controller (2). The SDN Controller looks for AAA routing information associated with the AAA policies and the information in the AAA request. It decides that the AAA request has to be forwarded to AAA agent 2 (3). The SDN controller installs then AAA routing information in the routing table and peer table in AAA agent 1 (4). Moreover, it fills the routing and peer tables within AAA agent 2 (5). It also sends credentials to both agents to establish a security association (6) in order to provide hop-by-hop or end-to-end security (7).

```
                        +----------------------------------------+
                        |    (1)      SDN Controller             |
            AAA ----------------+                                |
          Policies      |       V                                |
                        |      +----------+ (3) +-------------+  |
                    ---------->| AAA      |<--->|South. Prot. |  |
                    |   |      | Policies |     |             |  |
                    |   |      +----------+     +-------------+  |
                    |   |                        |     |         |
                    |   |                        |     |         |
                (2) |   +------------------------| --- | --------+
                    |                            |     |
                    |                      (4)(6)|     |(5)(6)
                    |   |------------------------+     +--|
            AAA     |   V                                 V       AAA
         request +-------------+                   +-----------+request
         ------->|    AAA      |================= >|   AAA     |------>
                 |  agent 1    |    hop-to-hop/    |  agent 2  |
                 |             |      security     |           |
                 +-------------+     (7)           +-----------+
```

Figure 4: Example of SDN controller pushing AAA routing information (reactive)

NOTE: It is worth noting that for the same AAA request some information can be enforced proactively and other can be retrieved reactively during the travel of the AAA request.

6.2. AAA agents under different SDN controllers

Another case (Figure 5) is when, for example, two organizations, ISP A and ISP B have their own SDN controller (A and B respectively), each one controlling a different set of AAA agents. During the travel of a AAA request, it may need to pass from the AAA agent 1 controlled by SDN Controller A to another AAA agent 2 which is under the control of a different SDN controller B.

In this case, both SDN controllers may coordinate each other and determine whether the AAA request is allowed to traverse through both realms or not (1). If they agree the conditions (2), the AAA policies that represents this agreement are enforced by the SDN controllers as AAA routing information and credentials into the AAA agents (3). Then the AAA request can move forward (4).

```
                   +-------------+                   +-------------+
        AAA        |     SDN     |<=================>|     SDN     |
        Policy. -->| Controller A|        (2)        |Controller B |
               (1) |             |                   |             |
                   +-------------+                   +-------------+
                          |                                 |
                          | (3)                         (3) |
             AAA          V                                 V     AAA
          request  +-------------+               +-------------+ request
            ------>| AAA agent 1 |<=============>| AAA agent 2 |------>
                   +-------------+      (4)      +-------------+
```

Figure 5: Gateway-to-gateway multi controller flow in case 1

7. Relationship with I2NSF

According to [I-D.merged-i2nsf-framework-04] I2NSF needs to provide identity information, along with additional data that Authentication, Authorization, and Accounting (AAA) frameworks can use. This enables those frameworks to perform AAA functions on the I2NSF traffic. In this sense, we assume that each AAA entity is, in the end, a NSF that may need to be configured with AAA-related information and where the administrator can obtain some valuable information such as, number authentications executed, number of active users accessing the service, accounting records, etc.

We envisage that SDN controller MAY also instantiate a vNSF acting as one of the AAA agents (relay, proxy, redirect, translation, server, etc...) when necessary. Even more, an administrator MAY instantiate a AAA server that works as UHO's AAA server. For that, apart from create the vNSF, it needs to register the users that the AAA server needs.

```
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
               |             Security Application          | App.
               |       (e.g.Identity and AAA Management)   | Layer
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
               |
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
               |             Application Support           | SDN/
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ Security
               |AAA Control Plane(routing,key distribution.| Controller
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
               |
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ AAA server
               |               AAA Data Plane              |(NSF)
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        ---->  |         Policy and Event Repository (PER) | --->
               +-------------------------------------------+
               |             AAA routing table  (AAA-RT)   |
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: High-level Architecture for SDN-based AAA management

Figure 6 describes the mapping with our use cases. In the right side of the figure each entity follows the same terminology than [I-D.merged-i2nsf-framework-04]

8. Security Considerations

TBD.

9. Acknowledgements

TBD.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <http://www.rfc-editor.org/info/rfc2865>.

[RFC2866] Rigney, C., "RADIUS Accounting", RFC 2866, DOI 10.17487/RFC2866, June 2000, <http://www.rfc-editor.org/info/rfc2866>.

[RFC2903] de Laat, C., Gross, G., Gommans, L., Vollbrecht, J., and D. Spence, "Generic AAA Architecture", RFC 2903, DOI 10.17487/RFC2903, August 2000, <http://www.rfc-editor.org/info/rfc2903>.

[RFC2904] Vollbrecht, J., Calhoun, P., Farrell, S., Gommans, L., Gross, G., de Bruijn, B., de Laat, C., Holdrege, M., and D. Spence, "AAA Authorization Framework", RFC 2904, DOI 10.17487/RFC2904, August 2000, <http://www.rfc-editor.org/info/rfc2904>.

[RFC6614] Winter, S., McCauley, M., Venaas, S., and K. Wierenga, "Transport Layer Security (TLS) Encryption for RADIUS", RFC 6614, DOI 10.17487/RFC6614, May 2012, <http://www.rfc-editor.org/info/rfc6614>.

[RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <http://www.rfc-editor.org/info/rfc6733>.

10.2. Informative References

[I-D.merged-i2nsf-framework-04] Lopez, E., Lopez, D., Zhuang, X., Dunbar, L., Parrott, J., Krishnan, R., and SR. Durbha, "Framework for Interface to Network Security Functions", draft-merged-i2nsf-framework-04.txt (work in progress), October 2015.

[ITU-T.X.1252] "Baseline Identity Management Terms and Definitions", April 2010.

[ITU-T.X.800] "Security Architecture for Open Systems Interconnection for CCITT Applications", March 1991.

[ITU-T.Y.3300] "Recommendation ITU-T Y.3300", June 2014.

[ONF-OpenFlow] ONF, "OpenFlow Switch Specification (Version 1.4.0)", October 2013.

[ONF-SDN-Architecture]
"SDN Architecture", June 2014.

[RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <http://www.rfc-editor.org/info/rfc7149>.

Authors' Addresses

Rafa Marin-Lopez
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia 30100
Spain

Phone: +34 868 88 85 01
Email: rafa@um.es

Gabriel Lopez-Millan
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia 30100
Spain

Phone: +34 868 88 85 04
Email: gabilm@um.es