ABFAB M. Jones
Internet-Draft Bridgewater Systems
Intended status: Informational H. Tschofenig
Expires: September 13, 2012 Nokia Siemens Networks
March 12, 2012

The Diameter 'Application Bridging for Federated Access Beyond Web (ABFAB)' Application
draft-jones-diameter-abfab-01.txt

Abstract

The Application Bridging for Federated Access Beyond Web (ABFAB) architecture provides cross-domain authentication, authorization and accounting functionality by utilizing well-established technologies, such as Diameter, the Extensible Authentication Protocol (EAP), and the Generic Security Services API (GSS-API).

This document defines a Diameter application for usage with the ABFAB architecture to convey authentication information, and authorization decisions from the Diameter server (acting as the identity provider) to the Diameter client (acting as a relying party) encoded in a Security Assertion Markup Language (SAML) encoding.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The Application Bridging for Federated Access Beyond Web (ABFAB) architecture [I-D.ietf-abfab-arch] provides cross-domain authentication, authorization and accounting functionality by utilizing well-established technologies, such as Diameter, the Extensible Authentication Protocol (EAP), and the Generic Security Services API (GSS-API).

The steps taken generally in an ABFAB federated authentication/authorization exchange are as follows:

1. Principal provides NAI to Application: Somehow the client is configured with at least the realm portion of an NAI, which represents the IdP to be discovered.

2. Authentication mechanism selection: this is the step necessary to indicate that the GSS-EAP SASL/GS2 mechanism will be used for authentication/authorization.

3. Client Application provides NAI to RP: At the conclusion of mechanism selection the NAI must be provided to the RP for discovery.

4. Discovery of federated IdP: This is discussed in detail below. Either the RP is configured with authorized IdPs, or it makes use of a federation proxy.

5. Request from Relying Party to IdP: Once the RP knows who the IdP is, it or its agent will forward RADIUS request that encapsulates a GSS/EAP access request to an IdP. This may or may not contain a SAML request as a series of attributes. At this stage, the RP will likely have no idea who the principal is. The RP claims its identity to the IdP in AAA attributes, and it makes whatever SAML Attribute Requests through a AAA attribute.

6. IdP informs the principal of which EAP method to use: The available and appropriate methods are discussed below in this memo.

7. A bunch of EAP messages happen between the endpoints: Messages are exchanged between the principal and the IdP until a result is determined. The number and content of those messages will depend on the EAP method. If the IdP is unable to authenticate the principal, the process concludes here. As part of this process, the principal will, under protection of EAP, assert the identity of the RP to which it intends to authenticate.

8. Successful Authentication: At the very least the IdP (its EAP server) and EAP peer / subject have authenticated one another. As a result of this step, the subject and the IdP hold two cryptographic keys- a Master Session Key (MSK), and an Extended MSK (EMSK). If the asserted identity of the RP by the principal matches the identity the RP itself asserted, there is some confidence that the RP is now authenticated to the IdP.

9. Local IdP Policy Check: At this stage, the IdP checks local policy to determine whether the RP and subject are authorized for a given transaction/service, and if so, what if any, attributes will be released to the RP. Additional policy checks will likely have been made earlier just through the process of discovery.

10. Response from the IdP to the Relying Party: Once the IdP has made a determination of whether and how to authenticate or authorize the principal to the RP, it returns either a negative AAA result to the RP, or it returns a positive result to the RP, along with an optional set of AAA attributes associated with the principal that could include one or more SAML assertions. In addition, an EAP MSK is returned to the subject.

11. RP Processes Results. When the RP receives the result from the IdP, it should have enough information to either grant or refuse a resource access request. It may have information that leads it to make additional attribute queries. It may have information that associates the principal with specific authorization identies. It will apply these results in an application-specific way.

12. RP returns results to principal: Once the RP has a response it must inform the client application of the result. If all has gone well, all are authenticated, and the application proceeds with appropriate authorization levels.

The involved entities are shown in Figure 1.

```
                                    +--------------+
                                    |  AAA Server  |
                                    |  (Identity   |
                                    |  Provider)   |
                                    +-^----------^-+
                                      * EAP      | RADIUS/
                                      *          | Diameter
                                    --v----------v--
                                 ///                \\\
                               //                      \\   ***
                              |        Federation        |  back-
                              |                          |  end
                               \\                      //   ***
                                 \\\                ///
                                    --^----------^--
                                      * EAP      | RADIUS/
                    Application       *          | Diameter
   +-------------+  Data            +-v----------v--+
   |             |<---------------->|               |
   | Client      |  EAP/EAP Method  | Server Side   |
   | Application |<****************>| Application   |
   | @ End Host  |  GSS-API         |(Relying Party)|
   |             |<---------------->|               |
   |             |  Application     |               |
   |             |  Protocol        |               |
   |             |<================>|               |
   +-------------+                  +---------------+
                  *** front-end ***

   Legend:

    <****>: End-to-end exchange
    <---->: Hop-by-hop exchange
    <====>: Protocol through which GSS-API/GS2 exchanges are tunnelled
```

Figure 1: Architecture for Federated Access of non-Web based Applications

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 uses terminology defined [I-D.ietf-abfab-arch].

3. Application Identifiers

This specification defines a new Diameter application and the respective Application Identifier:

```
Diameter ABFAB   (ABFAB)  [[TBD by IANA]]
```

The Diameter ABFAB related accounting information generated by the Diameter client uses the ABFAB Application Identifier in the case of coupled accounting model. The Diameter Base Accounting Application Identifier (value of 3) is used in case of the split accounting model. Refer to Section 4.2 for more information regarding the accounting models.

4. Protocol Description

```
       Relying Party    Client App       IdP

          |              (1)             | Client App gets NAI (somehow)
          |               |              |
          |<-----(2)----->|              | Mechanism Selection
          |               |              |
          |<-----(3)-----<|              | NAI transmitted to RP
          |               |              |
          |<=====(4)====================>| Discovery
          |               |              |
          |>=====(5)====================>| Access request from RP to IdP
          |               |              |
          |               |< - - (6) - -<| EAP method to Principal
          |               |              |
          |               |< - - (7) - ->| EAP Exchange to authenticate
          |               |              | Principal
          |               |              |
          |               |           (8 & 9) Local Policy Check
          |               |              |
          |<====(10)====================<| IdP Assertion to RP
          |               |              |
          |               |              | (11) RP Processes results.
          |               |              |
          |>----(12)----->|              | Results to client app.


        ----- = Between Client App and RP
        ===== = Between RP and IdP
        - - - = Between Client App and IdP
```

Figure 2: Message Interaction Sequence

4.1. Session Management

The Diameter server may maintain state or may be stateless. This is indicated in the Auth-Session-State AVP (or its absence). The Diameter client MUST support the Authorization Session State Machine defined in [RFC3588].

4.1.1. Session-Termination-Request

The Session-Termination-Request (STR) message [RFC3588] is sent by the Diameter client to inform the Diameter server that an authorized session is being terminated.

4.1.2. Session-Termination-Answer

The Session-Termination-Answer (STA) message [RFC3588] is sent by the Diameter server to acknowledge the notification that the session has been terminated.

4.1.3. Abort-Session-Request

The Abort-Session-Request (ASR) message [RFC3588] is sent by the Diameter server to the Diameter client to terminate the authorized session. When the Diameter client receives the ASR message, it MUST take further actions to terminate the established application context.

4.1.4. Abort-Session-Answer

The Abort-Session-Answer (ASA) message [RFC3588] is sent by the Home Agent in response to an ASR message.

4.2. Accounting for ABFAB services

The Diameter client collects accounting records needed for service control and charging MUST support the accounting procedures and the Accounting Session State Machine as defined in [RFC3588].

The Diameter application design guideline [I-D.ietf-dime-app-design-guide] defines two separate models for accounting:

Split accounting model:

> According to this model, the accounting messages use the Diameter Base Accounting Application Identifier (value of 3). Since accounting is treated as an independent application, accounting commands may be routed separately from the rest of application messages and thus the accounting messages generally end up in a central accounting server. Since the Diameter ABFAB application does not define its own unique accounting commands, this is the preferred choice, since it permits use of centralized accounting for several applications.

Coupled accounting model:

In this model, the accounting messages will use the ABFAB Application Identifiers. This means that accounting messages will be routed like any other Diameter ABFAB application messages. This requires the Diameter server in charge of the Diameter ABFAB application to handle the accounting records (e.g., sends them to a proper accounting server).

As mentioned above, the preferred choice is to use the split accounting model and thus to choose Diameter Base Accounting Application Identifier (value of 3) for accounting messages.

4.2.1. Accounting-Request

The Accounting-Request command [RFC3588] is sent by the Diameter client to the Diameter server to exchange accounting information.

4.2.2. Accounting-Answer

The Accounting-Answer command [RFC3588] is sent by the Diameter server to the Diameter client to acknowledge an Accounting-Request.

5. Command Codes

The Diameter ABFAB application defined in this document reuses the Diameter EAP application [RFC4072] commands: Diameter-EAP-Request (DER) and Diameter-EAP-Answer (DEA). This specification extends the existing DER and DEA command ABNFs to offer the necessary ABFAB functionality. Other than new additional AVPs and the corresponding additions to the command ABNFs, the Diameter EAP application command ABNFs remain unchanged. The ABNF language is defined in [RFC3588].

Command-Name	Abbrev.	Code	Reference
Diameter-EAP-Request	DER	268	RFC 4072
Diameter-EAP-Answer	DEA	268	RFC 4072

Figure 3: Command Codes

5.1. Diameter-EAP-Request

The Diameter-EAP-Request (DER) message, indicated by the Command-Code field set to 268 and the 'R' bit set in the Command Flags field, is sent by the Diameter client to the Diameter server to initiate a Diameter ABFAB authentication and authorization procedure. The Application-ID field of the Diameter Header MUST be set to the Diameter ABFAB Application ID (value of TDB).

```
<Diameter-EAP-Request> ::= < Diameter Header: 268, REQ, PXY >
                           < Session-Id >
                           { Auth-Application-Id }
                           { Origin-Host }
                           { Origin-Realm }
                           { Destination-Realm }
                           { Auth-Request-Type }
                           [ Destination-Host ]
                           [ NAS-Identifier ]
                           [ NAS-IP-Address ]
                           [ NAS-IPv6-Address ]
                           [ NAS-Port-Type ]
                           [ User-Name ]
                           ...
                           { EAP-Payload }
                           ...
                           [ SAML-AuthnRequest ]
                           ...
                         * [ AVP ]
```

The SAML-AuthnRequest AVP is only included in the first DER message send by the Diameter client. The user is both authenticated and during the EAP method authentication. Authorization happens immediately after the authentication procedure has been completed. Thus, the Auth-Request-Type AVP MUST be set to the value AUTHORIZE_AUTHENTICATE.

5.2. Diameter-EAP-Answer

The Diameter-EAP-Answer (DEA) message, indicated by the Command-Code field set to 268 and 'R' bit cleared in the Command Flags field, is sent in response to the Diameter-EAP-Request message (DER). The Application-Id field in the Diameter message header MUST be set to the Diameter ABFAB Application-Id (value of TBD).

```
   <Diameter-EAP-Answer> ::= < Diameter Header: 268, PXY >
                             < Session-Id >
                             { Auth-Application-Id }
                             { Auth-Request-Type }
                             { Result-Code }
                             { Origin-Host }
                             { Origin-Realm }
                             [ User-Name ]
                             [ EAP-Payload ]
                             [ EAP-Reissued-Payload ]
                             [ EAP-Master-Session-Key ]
                             [ EAP-Key-Name ]
                             [ Multi-Round-Time ]
                             ...
                             [ SAML-AuthnResponse ]
                             [ SAML-Assertion ]
                             ...
                           * [ AVP ]
```

SAML related attributes are only included in the final message exchange. Either the SAML-AuthnResponse AVP is included in the response or a SAML-Assertion but not both.

6. AVPs

	AVP		AVP Flag rules				
Attribute Name	Code	Value Type	MUST	MAY	SHOULD NOT	MUST NOT	MAY Encr
SAML-Assertion	TBD	UTF8String		P		V	Y
SAML-AuthnRequest	TBD	UTF8String		P		V	Y
SAML-AuthnResponse	TBD	UTF8String		P		V	Y

AVPs for the Diameter ABFAB Application

[Editor's Note: SAML encryption requirements are FFS. The "MAY Encr" column in the above table refers to XML-Enc rather than the defunct Diameter AVP encryption.]

The SAML-Assertion AVP contains the UTF8String encoded SAML assertion.

The SAML-AuthnRequest AVP contains the UTF8String encoded SAML AuthnRequest message.

The SAML-AuthnResponse AVP contains the UTF8String encoded SAML AuthnResponse message.

7. Result-Code AVP Values

This section defines new Result-Code [RFC3588] values that MUST be supported by all Diameter implementations that conform to this specification.

[Editor's Note: ABFAB specific error values may need to be added here.]

8. AVP Occurrence Tables

The following tables present the AVPs defined in this document and their occurrences in Diameter messages.

The table uses the following symbols:

0:

The AVP MUST NOT be present in the message.

0+:

Zero or more instances of the AVP MAY be present in the message.

0-1:

Zero or one instance of the AVP MAY be present in the message.

1:

One instance of the AVP MUST be present in the message.

8.1. DER, DEA AVP/Command-Code Table

	Command-Code	
AVP Name	DER	DEA
SAML-Assertion	0	1
SAML-AuthnRequest	1	0
SAML-AuthnResponse	0	1

[Editor's Note: Is it possible to return more than one SAML-Assertion?]

8.2. Coupled Accounting Model AVP Table

The table in this section is used to represent which AVPs defined in this document are to be present in the Accounting messages, as defined in [RFC3588].

	Command-Code	
Attribute Name	ACR	ACA
Accounting-Input-Octets	0-1	0-1
Accounting-Input-Packets	0-1	0-1
Accounting-Output-Octets	0-1	0-1
Accounting-Output-Packets	0-1	0-1
Acct-Multi-Session-Id	0-1	0-1
Acct-Session-Time	0-1	0-1

[Editor's Note: Do we need any application specific accounting messages for ABFAB?]

9. IANA Considerations

This section contains the namespaces that have either been created in this specification or had their values assigned to existing namespaces managed by IANA.

9.1. AVP Codes

This specification requires IANA to register the following new AVPs from the AVP Code namespace defined in [RFC3588].

- o SAML-Assertion
- o SAML-AuthnRequest
- o SAML-AuthnResponse

The AVPs are defined in Section 8.

9.2. Application Identifier

This specification requires IANA to allocate a new application ID from the Application Identifier namespace defined in [RFC3588].

Application Identifier	Value
Diameter ABFAB (ABFAB)	TBD

10. Security Considerations

11. Acknowledgments

12. Normative References

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

[RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.

[RFC4072] Eronen, P., Hiller, T., and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application", RFC 4072, August 2005.

[I-D.ietf-dime-app-design-guide] Fajardo, V., Tschofenig, H., and L. Morand, "Diameter Applications Design Guidelines", draft-ietf-dime-app-design-guide-13 (work in progress), January 2012.

[I-D.ietf-abfab-arch] Howlett, J., Hartman, S., Tschofenig, H., and E. Lear, "Application Bridging for Federated Access Beyond Web (ABFAB) Architecture", draft-ietf-abfab-arch-01 (work in progress), March 2012.

Authors' Addresses

Mark Jones
Bridgewater Systems

Email: mark@azu.ca

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