Internet Engineering Task Force (IETF)
Request for Comments: 7833
Category: Standards Track
ISSN: 2070-1721

J. Howlett
Jisc
S. Hartman
Painless Security
A. Perez-Mendez, Ed.
University of Murcia

May 2016

A RADIUS Attribute, Binding, Profiles, Name Identifier Format, and Confirmation Methods for the Security Assertion Markup Language (SAML)

Abstract

This document describes the use of the Security Assertion Markup Language (SAML) with RADIUS in the context of the Application Bridging for Federated Access Beyond web (ABFAB) architecture. It defines two RADIUS attributes, a SAML binding, a SAML name identifier format, two SAML profiles, and two SAML confirmation methods. The RADIUS attributes permit encapsulation of SAML Assertions and protocol messages within RADIUS, allowing SAML entities to communicate using the binding. The two profiles describe the application of this binding for ABFAB authentication and assertion Query/Request, enabling a Relying Party to request authentication of, or assertions for, users or machines (clients). These clients may be named using a Network Access Identifier (NAI) name identifier format. Finally, the subject confirmation methods allow requests and queries to be issued for a previously authenticated user or machine without needing to explicitly identify them as the subject. The use of the artifacts defined in this document is not exclusive to ABFAB. They can be applied in any Authentication, Authorization, and Accounting (AAA) scenario, such as network access control.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7833.

Table of Contents

1. Introduction

Within the ABFAB (Application Bridging for Federated Access Beyond web) architecture [RFC7831], it is often desirable to convey Security Assertion Markup Language (SAML) Assertions and protocol messages.

SAML typically only considers the use of HTTP-based transports, known as bindings [OASIS.saml-bindings-2.0-os], which are primarily intended for use with the SAML V2.0 web browser single sign-on profile [OASIS.saml-profiles-2.0-os]. However, the goal of ABFAB is to extend the applicability of federated identity beyond the web to other applications by building on the Authentication, Authorization, and Accounting (AAA) framework. Consequently, there exists a requirement for SAML to integrate with the AAA framework and with protocols such as RADIUS [RFC2865] and Diameter [RFC6733], in addition to HTTP.

In summary, this document specifies:

- Two RADIUS attributes to encapsulate SAML Assertions and protocol messages, respectively.
- A SAML RADIUS binding that defines how SAML Assertions and protocol messages can be transported by RADIUS within a SAML exchange.
- A SAML name identifier format in the form of a Network Access Identifier.
- A profile of the SAML Authentication Request Protocol that uses the SAML RADIUS binding to effect SAML-based authentication and authorization.
- A profile of the SAML Assertion Query and Request Protocol that uses the SAML RADIUS binding to effect the query and request of SAML Assertions.
- Two SAML subject confirmation methods for indicating that a user or machine client is the subject of an assertion.

This document adheres to the guidelines stipulated by [OASIS.saml-bindings-2.0-os] and [OASIS.saml-profiles-2.0-os] for defining new SAML bindings and profiles, respectively, and other conventions applied formally or otherwise within SAML. In particular, this document provides a "Required Information" section for the binding (Section 4.1) and profiles (Sections 7.1 and 8.1) that enumerate:

- A URI that uniquely identifies the protocol binding or profile.
- Postal or electronic contact information for the author.
- A reference to previously defined bindings or profiles that the new binding updates or obsoletes.
- In the case of a profile, any SAML confirmation method identifiers defined and/or utilized by the profile.

1.1. Terminology

This document uses terminology from a number of related standards that tend to adopt different terms for similar or identical concepts. In general, this document uses, when possible, the ABFAB term for the entity, as described in [RFC7831]. For reference, we include the following table, which maps the different terms into a single view. (In this document, "NAS" refers to a network access server, and "AS" refers to an authentication server.)

Protocol	Client	Relying Party	Identity Provider
ABFAB	Client	Relying Party	Identity Provider
SAML	Subject	Service Provider	Identity Provider
	Principal	Requester	Responder
		Consumer	Issuer
RADIUS	User	NAS	AS
		RADIUS client	RADIUS server

Table 1: Terminology

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. RADIUS SAML Attributes

The SAML RADIUS binding defined in Section 4 of this document uses two attributes to convey SAML Assertions and protocol messages [OASIS.saml-core-2.0-os]. Owing to the typical size of these structures, these attributes use the "Long Extended Type" format [RFC6929] to encapsulate their data. RADIUS entities MUST NOT include both attributes in the same RADIUS message, as they represent exclusive alternatives to convey SAML information.

3.1. SAML-Assertion Attribute

This attribute is used to encode a SAML Assertion. Figure 1 represents the format of this attribute.

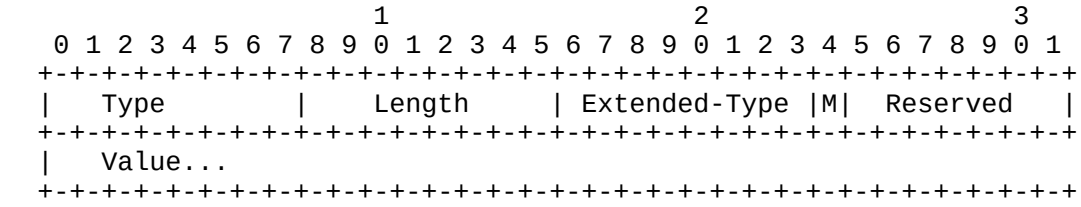

Figure 1: SAML-Assertion Format

Type

245

Length

>= 5

Extended-Type

1

M (More)

As described in [RFC6929].

Reserved

As described in [RFC6929].

Value

One or more octets encoding a SAML Assertion.

3.2. SAML-Protocol Attribute

This attribute is used to encode a SAML protocol message. Figure 2 represents the format of this attribute.

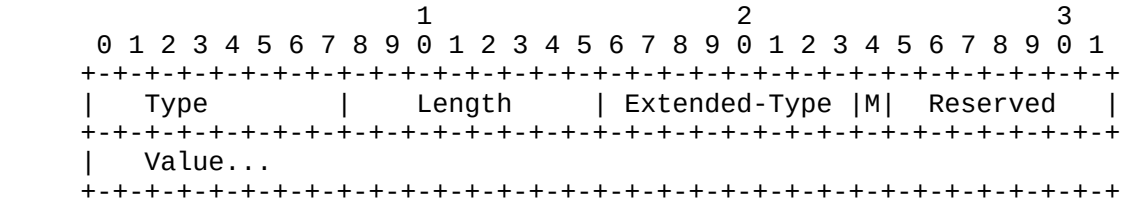

Figure 2: SAML-Protocol Format

Type

245

Length

>= 5

Extended-Type

2

M (More)

As described in [RFC6929].

Reserved

As described in [RFC6929].

Value

One or more octets encoding a SAML protocol message.

4. SAML RADIUS Binding

The SAML RADIUS binding defines how RADIUS [RFC2865] can be used to enable a RADIUS client and server to exchange SAML Assertions and protocol messages.

4.1. Required Information

Identification: urn:ietf:params:abfab:bindings:radius

Contact information: iesg@ietf.org

Updates: None.

4.2. Operation

In this specification, the Relying Party (RP) MUST trust any statement in the SAML messages from the Identity Provider (IdP) in the same way that it trusts information contained in RADIUS attributes. These entities MUST trust the RADIUS infrastructure to provide integrity of the SAML messages.

Hence, it is REQUIRED that the RADIUS exchange be protected using Transport Layer Security (TLS) encryption for RADIUS [RFC6614] to provide confidentiality and integrity protection, unless alternative methods to ensure them are used, such as IPsec tunnels or a sufficiently secure internal network.

Implementations of this profile can take advantage of mechanisms to permit the transport of longer SAML messages over RADIUS transports, such as the support of fragmentation of RADIUS packets [RFC7499] or larger packets for RADIUS over TCP [RADIUS-Large-Pkts].

There are two system models for the use of SAML over RADIUS. The first is a request-response model, using the RADIUS SAML-Protocol attribute defined in Section 3 to encapsulate the SAML protocol messages.

1. The RADIUS client, acting as an RP, transmits a SAML request element within a RADIUS Access-Request message. This message MUST include a single instance of the RADIUS User-Name attribute whose value MUST conform to the Network Access Identifier [RFC7542] scheme. The RP MUST NOT include more than one SAML request element.

2. The RADIUS server, acting as an IdP, returns a SAML protocol message within a RADIUS Access-Accept or Access-Reject message. These messages necessarily conclude a RADIUS exchange, and therefore this is the only opportunity for the IdP to send a response in the context of this exchange. The IdP MUST NOT include more than one SAML response. An IdP that refuses to perform a message exchange with the RP can silently discard the SAML request (this could subsequently be followed by a RADIUS Access-Reject, as the same conditions that cause the IdP to discard the SAML request may also cause the RADIUS server to fail to authenticate).

The second system model permits a RADIUS server acting as an IdP to use the RADIUS SAML-Assertion attribute defined in Section 3 to encapsulate an unsolicited SAML Assertion. This attribute MUST be included in a RADIUS Access-Accept message. When included, the attribute MUST contain a single SAML Assertion.

RADIUS servers MUST NOT include both the SAML-Protocol and the SAML-Assertion attribute in the same RADIUS message. If an IdP is producing a response to a SAML request, then the first system model is used. An IdP MAY ignore a SAML request and send an unsolicited assertion using the second system model (that is, using the RADIUS SAML-Assertion attribute).

In either system model, IdPs SHOULD return a RADIUS State attribute as part of the Access-Accept message so that future SAML queries or requests can be run against the same context of an authentication exchange.

This binding is intended to be composed with other uses of RADIUS, such as network access. Therefore, other arbitrary RADIUS attributes MAY be used in either the request or response.

In the case of a SAML processing error, the RADIUS server MAY include a SAML response message with an appropriate value for the <samlp:Status> element within the Access-Accept or Access-Reject packet to notify the client. Alternatively, the RADIUS server can respond without a SAML-Protocol attribute.

4.3. Processing of Names

SAML entities using profiles making use of this binding will typically possess both the SAML and AAA names of their correspondents. Frequently, these entities will need to apply policies using these names -- for example, when deciding to release attributes. Often, these policies will be security-sensitive, and so it is important that policy is applied on these names consistently.

4.3.1. AAA Names

These rules relate to the processing of AAA names by SAML entities using profiles making use of this binding.

- IdPs SHOULD apply policy based on the RP's identity associated with the RADIUS Access-Request.

- RPs SHOULD apply policy based on the NAI realm associated with the RADIUS Access-Accept.

4.3.2. SAML Names

These rules relate to the processing of SAML names by SAML entities using profiles making use of this binding.

IdPs MAY apply policy based on the RP's SAML entityID. In such cases, at least one of the following methods is required in order to establish a relationship between the SAML name and the AAA name of the RP:

- RADIUS client identity in trusted SAML metadata (as described in Section 4.3.3).

- RADIUS client identity in trusted digitally signed SAML request.

A digitally signed SAML request without the RADIUS client identity is not sufficient, since a malicious RADIUS entity can observe a SAML message and include it in a different RADIUS message without the consent of the issuer of that SAML message. If an IdP were to process the SAML message without confirming that it applied to the RADIUS message, inappropriate policy would be used.

RPs MAY apply policy based on the SAML issuer's entityID. In such cases, at least one of the following methods is required in order to establish a relationship between the SAML name and the AAA name of the IdP:

- RADIUS realm in trusted SAML metadata (as described in Section 4.3.3).

- RADIUS realm in trusted digitally signed SAML response or assertion.

A digitally signed SAML response alone is not sufficient, for the same reasons as those described above for SAML requests.

4.3.3. Mapping of AAA Names in SAML Metadata

This section defines extensions to the SAML metadata schema [OASIS.saml-metadata-2.0-os] that are required in order to represent AAA names associated with a particular <EntityDescriptor> element.

In SAML metadata, a single entity may act in many different roles in the support of multiple profiles. This document defines two new roles: RADIUS IdP and RADIUS RP, requiring the declaration of two new subtypes of RoleDescriptorType: RADIUSIDPDescriptorType and RADIUSRPDescriptorType. These subtypes contain the additional elements required to represent AAA names for IdP and RP entities, respectively.

4.3.3.1. RADIUSIDPDescriptorType

The RADIUSIDPDescriptorType complex type extends RoleDescriptorType with elements common to IdPs that support RADIUS. It contains the following additional elements:

<RADIUSIDPService> [Zero or More] Zero or more elements of type EndpointType that describe RADIUS endpoints that are associated with the entity.

<RADIUSRealm> [Zero or More] Zero or more elements of type string that represent the acceptable values of the RADIUS realm associated with the entity, obtained from the realm part of the RADIUS User-Name attribute.

The following schema fragment defines the RADIUSIDPDescriptorType complex type:

```
        <complexType name="RADIUSIDPDescriptorType">
          <complexContent>
            <extension base="md:RoleDescriptorType">
              <sequence>
                <element ref="abfab:RADIUSIDPService"
                              minOccurs="0" maxOccurs="unbounded"/>
                <element ref="abfab:RADIUSRealm"
                              minOccurs="0" maxOccurs="unbounded"/>
              </sequence>
            </extension>
          </complexContent>
        </complexType>
        <element name="RADIUSIDPService" type="md:EndpointType"/>
        <element name="RADIUSRealm" type="string"/>
```

Figure 3: RADIUSIDPDescriptorType Schema

4.3.3.2. RADIUSRPDescriptorType

The RADIUSRPDescriptorType complex type extends RoleDescriptorType with elements common to RPs that support RADIUS. It contains the following additional elements:

<RADIUSRPService> [Zero or More] Zero or more elements of type EndpointType that describe RADIUS endpoints that are associated with the entity.

<RADIUSNasIpAddress> [Zero or More] Zero or more elements of type string that represent the acceptable values of the RADIUS NAS-IP-Address or NAS-IPv6-Address attributes associated with the entity.

<RADIUSNasIdentifier> [Zero or More] Zero or more elements of type string that represent the acceptable values of the RADIUS NAS-Identifier attribute associated with the entity.

<RADIUSGssEapName> [Zero or More] Zero or more elements of type string that represent the acceptable values of the GSS-API Mechanism for the Extensible Authentication Protocol (GSS-EAP) acceptor name associated with the entity. The format for this name is described in Section 3.1 of [RFC7055], while Section 3.4 of [RFC7055] describes how that name is decomposed and transported using RADIUS attributes.

The following schema fragment defines the RADIUSRPDescriptorType complex type:

```
    <complexType name="RADIUSRPDescriptorType">
      <complexContent>
        <extension base="md:RoleDescriptorType">
          <sequence>
            <element ref="md:RADIUSRPService"
                          minOccurs="0" maxOccurs="unbounded"/>
            <element ref="md:RADIUSNasIpAddress"
                          minOccurs="0" maxOccurs="unbounded"/>
            <element ref="md:RADIUSNasIdentifier"
                          minOccurs="0" maxOccurs="unbounded"/>
            <element ref="md:RADIUSGssEapName"
                          minOccurs="0" maxOccurs="unbounded"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
    <element name="RADIUSRPService" type="md:EndpointType"/>
    <element name="RADIUSNasIpAddress" type="string"/>
    <element name="RADIUSNasIdentifier" type="string"/>
    <element name="RADIUSGssEapName" type="string"/>
```

Figure 4: RADIUSRPDescriptorType Schema

4.3.4. Example of SAML Metadata That Includes AAA Names

Figures 5 and 6 illustrate examples of metadata that includes AAA names for an IdP and an RP, respectively. The IdP's SAML name is "https://IdentityProvider.com/", whereas its RADIUS realm is "idp.com". The RP's SAML name is "https://RelyingParty.com/SAML", being its GSS-EAP acceptor name "nfs/fileserver.rp.com@RP.COM".

```
<EntityDescriptor
   xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:abfab="urn:ietf:params:xml:ns:abfab"
   entityID="https://IdentityProvider.com/SAML">
   <RoleDescriptor
      xsi:type="abfab:RADIUSIDPDescriptorType"
      protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
       <RADIUSRealm>idp.com</RADIUSRealm>
   </RoleDescriptor>
</EntityDescriptor>
```

Figure 5: Metadata for the IdP

```
<EntityDescriptor
   xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:abfab="urn:ietf:params:xml:ns:abfab"
   entityID="https://RelyingParty.com/SAML">
   <RoleDescriptor
      xsi:type="abfab:RADIUSRPDescriptorType"
      protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
       <RADIUSGssEapName>nfs/fileserver.rp.com@RP.COM</RADIUSGssEapName>
   </RoleDescriptor>
</EntityDescriptor>
```

Figure 6: Metadata for the RP

4.4. Use of XML Signatures

This binding calls for the use of SAML elements that support XML signatures. To promote interoperability, implementations of this binding MUST support a default configuration that does not require the use of XML signatures. Implementations MAY choose to use XML signatures.

4.5. Metadata Considerations

This binding, and the profiles, are mostly intended to be used without metadata. In this usage, RADIUS infrastructure is used to provide integrity and naming of the SAML messages and assertions. RADIUS configuration is used to provide policy, including which attributes are accepted from an RP and which attributes are sent by an IdP.

Nevertheless, if metadata is used, the roles described in Section 4.3.3 MUST be present.

5. Network Access Identifier Name Identifier Format

URI: urn:ietf:params:abfab:nameid-format:nai

Indicates that the content of the element is in the form of a Network Access Identifier (NAI) using the syntax described by [RFC7542].

6. RADIUS State Confirmation Method Identifiers

URI: urn:ietf:params:abfab:cm:user

URI: urn:ietf:params:abfab:cm:machine

Indicates that the subject is the system entity (either the user or machine) authenticated by a previously transmitted RADIUS Access-Accept message, as identified by the value of that RADIUS message's State attribute.

7. ABFAB Authentication Profile

In the scenario supported by the ABFAB Authentication Profile, a client controlling a User Agent requests access to an RP. The RP uses RADIUS to authenticate the client. In particular, the RP, acting as a RADIUS client, attempts to validate the client's credentials against a RADIUS server acting as the client's IdP. If the IdP successfully authenticates the client, it produces an authentication assertion that is consumed by the RP. This assertion MAY include a name identifier that can be used between the RP and the IdP to refer to the client.

7.1. Required Information

Identification: urn:ietf:params:abfab:profiles:authentication

Contact information: iesg@ietf.org

SAML confirmation method identifiers: The SAML V2.0 "RADIUS State" confirmation method identifiers -- either urn:ietf:params:abfab:cm:user or urn:ietf:params:abfab:cm:machine -- are used by this profile.

Updates: None.

7.2. Profile Overview

To implement this scenario, this profile of the SAML Authentication Request Protocol MUST be used in conjunction with the SAML RADIUS binding defined in Section 4.

This profile is based on the SAML V2.0 web browser single sign-on profile [OASIS.saml-profiles-2.0-os]. There are some important differences; specifically:

Authentication: This profile does not require the use of any particular authentication method. The ABFAB architecture does require the use of the Extensible Authentication Protocol (EAP) [RFC3579], but this specification may be used in other non-ABFAB scenarios.

Bindings: This profile does not use HTTP-based bindings. Instead, all SAML protocol messages are transported using the SAML RADIUS binding defined in Section 4. This is intended to reduce the number of bindings that implementations must support to be interoperable.

Requests: The profile does not permit the RP to name the <saml:Subject> of the <samlp:AuthnRequest>. This is intended to simplify implementation and interoperability.

Responses: The profile only permits the IdP to return a single SAML message or assertion that MUST contain exactly one authentication statement. Other statements may be included within this assertion at the discretion of the IdP. This is intended to simplify implementation and interoperability.

Figure 7 below illustrates the flow of messages within this profile.

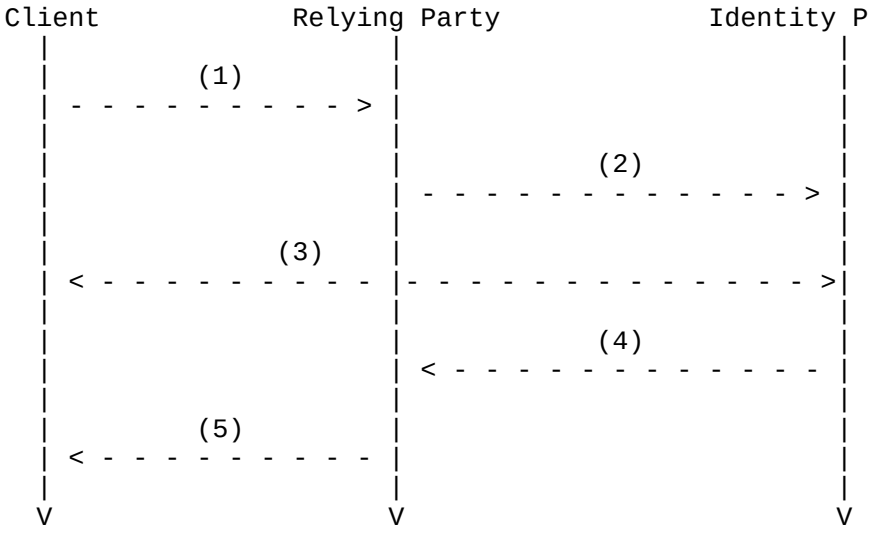

Figure 7: Flow of Messages

The following steps are described by the profile. Within an individual step, there may be one or more actual message exchanges.

1. Client request to RP (Section 7.3.1): In step 1, the client, via a User Agent, makes a request for a secured resource at the RP. The RP determines that no security context for the client exists and initiates the authentication process.

2. RP issues <samlp:AuthnRequest> to IdP (Section 7.3.2). In step 2, the RP may optionally issue a <samlp:AuthnRequest> message to be delivered to the IdP using the SAML-Protocol RADIUS attribute.

3. IdP identifies client (Section 7.3.3). In step 3, the client is authenticated and identified by the IdP, while honoring any requirements imposed by the RP in the <samlp:AuthnRequest> message if provided.

4. IdP issues <samlp:Response> to RP (Section 7.3.4). In step 4, the IdP issues a <samlp:Response> message to the RP using the SAML RADIUS binding. The response either indicates an error or includes a SAML authentication statement in exactly one SAML Assertion. If the RP did not send a <samlp:AuthnRequest>, the IdP issues an unsolicited <samlp:Assertion>, as described in Section 7.4.4.

5. RP grants or denies access to client (Section 7.3.5). In step 5, having received the response from the IdP, the RP can respond to the client with its own error, or can establish its own security context for the client and return the requested resource.

7.3. Profile Description

The ABFAB Authentication Profile is a profile of the SAML V2.0 Authentication Request Protocol [OASIS.saml-core-2.0-os]. Where both specifications conflict, the ABFAB Authentication Profile takes precedence.

7.3.1. Client Request to Relying Party

The profile is initiated by an arbitrary client request to the RP. There are no restrictions on the form of the request. The RP is free to use any means it wishes to associate the subsequent interactions with the original request. The RP, acting as a RADIUS client, attempts to authenticate the client.

7.3.2. Relying Party Issues <samlp:AuthnRequest> to Identity Provider

The RP uses RADIUS to communicate with the client's IdP. The RP MAY include a <samlp:AuthnRequest> within this RADIUS Access-Request message using the SAML-Protocol RADIUS attribute. The "next hop" destination MAY be the IdP or, alternatively, an intermediate RADIUS proxy.

Profile-specific rules for the contents of the <samlp:AuthnRequest> element are given in Section 7.4.1.

7.3.3. Identity Provider Identifies Client

The IdP MUST establish the identity of the client using a RADIUS authentication method, or else it will return an error. If the ForceAuthn attribute in the <samlp:AuthnRequest> element (if sent by the RP) is present and true, the IdP MUST freshly establish this identity rather than relying on any existing session state it may have with the client (for example, TLS state that may be used for session resumption). Otherwise, and in all other respects, the IdP may use any method to authenticate the client, subject to the constraints called out in the <samlp:AuthnRequest> message.

7.3.4. Identity Provider Issues <samlp:Response> to Relying Party

The IdP MUST conclude the authentication in a manner consistent with the RADIUS authentication result. The IdP MAY issue a <samlp:Response> message to the RP that is consistent with the authentication result, as described in [OASIS.saml-core-2.0-os]. This SAML response is delivered to the RP using the SAML RADIUS binding described in Section 4.

Profile-specific rules regarding the contents of the <samlp:Response> element are given in Section 7.4.2.

7.3.5. Relying Party Grants or Denies Access to Client

If a <samlp:Response> message is issued by the IdP, the RP MUST process that message and any enclosed assertion elements as described in [OASIS.saml-core-2.0-os]. Any subsequent use of the assertion elements is at the discretion of the RP, subject to any restrictions contained within the assertions themselves or from any previously established out-of-band policy that governs the interaction between the IdP and the RP.

7.4. Use of Authentication Request Protocol

This profile is based on the Authentication Request Protocol defined in [OASIS.saml-core-2.0-os]. In the nomenclature of actors enumerated in Section 3.4 of that document, the RP is the requester, the User Agent is the attesting entity, and the client is the subject.

7.4.1. <samlp:AuthnRequest> Usage

The RP MUST NOT include a <saml:Subject> element in the request. The authenticated RADIUS identity identifies the client to the IdP.

An RP MAY include any message content described in Section 3.4.1 of [OASIS.saml-core-2.0-os]. All processing rules are as defined in [OASIS.saml-core-2.0-os].

If the RP wishes to permit the IdP to establish a new identifier for the client if none exists, it MUST include a <saml:NameIDPolicy> element with the AllowCreate attribute set to "true". Otherwise, only a client for whom the IdP has previously established an identifier usable by the RP can be authenticated successfully.

The <samlp:AuthnRequest> message MAY be signed. Authentication and integrity are also provided by the SAML RADIUS binding.

7.4.2. <samlp:Response> Message Usage

If the IdP cannot or will not satisfy the request, it MUST respond with a <samlp:Response> message containing an appropriate error status code or codes and/or respond with a RADIUS Access-Reject message.

If the IdP wishes to return an error, it MUST NOT include any assertions in the <samlp:Response> message. Otherwise, if the request is successful (or if the response is not associated with a request), the <samlp:Response> element is subject to the following constraints:

- It MAY be signed.
- It MUST contain exactly one assertion. The <saml:Subject> element of this assertion MUST refer to the authenticated RADIUS user.
- The assertion MUST contain a <saml:AuthnStatement>. Also, the assertion MUST contain a <saml:Subject> element with at least one <saml:SubjectConfirmation> element containing a <saml:ConfirmationMethod> element of urn:ietf:params:abfab:cm:user or urn:ietf:params:abfab:cm:machine that reflects the authentication of the client to the IdP. Since the <samlp:Response> message is in response to a <samlp:AuthnRequest>, the InResponseTo attribute (in both the <saml:SubjectConfirmationData> and <saml:Response> elements) MUST match the request's ID. The <saml:Subject> element MAY use the NAI name identifier format described in Section 5 to establish an identifier between the RP and the IdP.
- Other conditions MAY be included as requested by the RP or at the discretion of the IdP. The IdP is NOT obligated to honor the requested set of conditions in the <samlp:AuthnRequest>, if any.

7.4.3. <samlp:Response> Message Processing Rules

The RP MUST do the following:

- Assume that the client's identifier implied by a SAML <Subject> element, if present, takes precedence over an identifier implied by the RADIUS User-Name attribute.
- Verify that the InResponseTo attribute in the "RADIUS State" <saml:SubjectConfirmationData> equals the ID of its original <samlp:AuthnRequest> message, unless the response is unsolicited, in which case the attribute MUST NOT be present.

- If a <saml:AuthnStatement> used to establish a security context for the client contains a SessionNotOnOrAfter attribute, the security context SHOULD be discarded once this time is reached, unless the RP reestablishes the client's identity by repeating the use of this profile.

- Verify that any assertions relied upon are valid according to processing rules specified in [OASIS.saml-core-2.0-os].

- Any assertion that is not valid or whose subject confirmation requirements cannot be met MUST be discarded and MUST NOT be used to establish a security context for the client.

7.4.4. Unsolicited Responses

An IdP MAY initiate this profile by delivering an unsolicited assertion to an RP. This MUST NOT contain any <saml:SubjectConfirmationData> elements containing an InResponseTo attribute.

7.4.5. Use of the SAML RADIUS Binding

It is RECOMMENDED that the RADIUS exchange be protected using TLS encryption for RADIUS [RFC6614] to provide confidentiality and integrity protection.

7.4.6. Use of XML Signatures

This profile calls for the use of SAML elements that support XML signatures. To promote interoperability, implementations of this profile MUST NOT require the use of XML signatures. Implementations MAY choose to use XML signatures.

7.4.7. Metadata Considerations

There are no metadata considerations particular to this profile, aside from those applying to the use of the RADIUS binding.

8. ABFAB Assertion Query/Request Profile

This profile builds on the SAML V2.0 Assertion Query/Request Profile defined by [OASIS.saml-profiles-2.0-os]. That profile describes the use of the Assertion Query and Request Protocol defined by Section 3.3 of [OASIS.saml-core-2.0-os] with synchronous bindings, such as the SOAP binding defined in [OASIS.saml-bindings-2.0-os].

Although the SAML V2.0 Assertion Query/Request Profile is independent of the underlying binding, it is nonetheless useful to describe the use of the SAML RADIUS binding defined in Section 4 of this document, in the interest of promoting interoperable implementations, particularly as the SAML V2.0 Assertion Query/Request Profile is most frequently discussed and implemented in the context of the SOAP binding.

8.1. Required Information

Identification: urn:ietf:params:abfab:profiles:query

Contact information: iesg@ietf.org

Description: Given below.

Updates: None.

8.2. Profile Overview

As with the SAML V2.0 Assertion Query/Request Profile defined by [OASIS.saml-profiles-2.0-os], the message exchange and basic processing rules that govern this profile are largely defined by Section 3.3 of [OASIS.saml-core-2.0-os], which defines the messages to be exchanged, in combination with the binding used to exchange the messages. The SAML RADIUS binding described in this document defines the binding of the message exchange to RADIUS. Unless specifically noted here, all requirements defined in those specifications apply.

Figure 8 below illustrates the basic template for the Query/Request Profile.

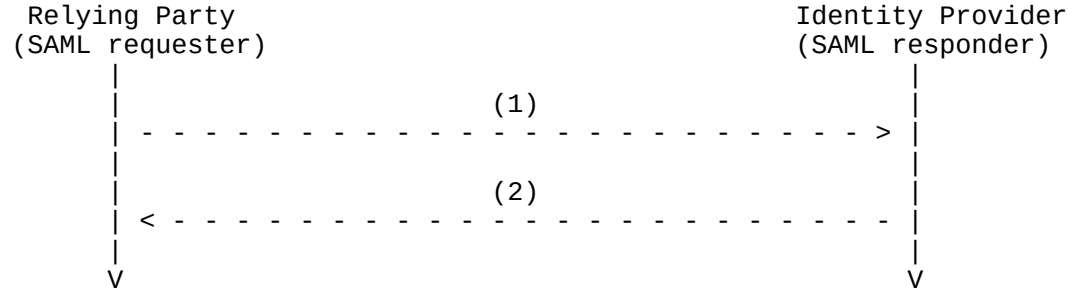

Figure 8: Basic Template for Query/Request Profile

The following steps are described by the profile:

1. Query/Request issued by RP: In step 1, an RP initiates the profile by sending an <AssertionIDRequest>, <SubjectQuery>, <AuthnQuery>, <AttributeQuery>, or <AuthzDecisionQuery> message to a SAML authority.

2. <Response> issued by SAML authority: In step 2, the responding SAML authority (after processing the query or request) issues a <Response> message to the RP.

8.3. Profile Description

8.3.1. Differences from the SAML V2.0 Assertion Query/Request Profile

This profile is identical to the SAML V2.0 Assertion Query/Request Profile, with the following exceptions:

- When processing the SAML request, the IdP MUST give precedence to the client's identifier implied by the RADIUS State attribute, if present, over the identifier implied by the SAML request's <Subject>, if any.

- In respect to Sections 6.3.1 and 6.5 of [OASIS.saml-profiles-2.0-os], this profile does not consider the use of metadata (as in [OASIS.saml-metadata-2.0-os]). See Section 8.3.4.

- In respect to Sections 6.3.2, 6.4.1, and 6.4.2 of [OASIS.saml-profiles-2.0-os], this profile additionally stipulates that implementations of this profile MUST NOT require the use of XML signatures. See Section 8.3.3.

8.3.2. Use of the SAML RADIUS Binding

The RADIUS Access-Request sent by the RP:

- MUST include an instance of the RADIUS Service-Type attribute, having a value of Authorize-Only.

- SHOULD include the RADIUS State attribute, where this Query/Request pertains to a previously authenticated client.

When processing the SAML request, the IdP MUST give precedence to the client's identifier implied by the RADIUS State attribute over the identifier implied by the SAML request's <Subject>, if any.

It is RECOMMENDED that the RADIUS exchange be protected using TLS encryption for RADIUS [RFC6614] to provide confidentiality and integrity protection.

8.3.3. Use of XML Signatures

This profile calls for the use of SAML elements that support XML signatures. To promote interoperability, implementations of this profile MUST NOT require the use of XML signatures. Implementations MAY choose to use XML signatures.

8.3.4. Metadata Considerations

There are no metadata considerations particular to this profile, aside from those applying to the use of the RADIUS binding.

9. Privacy Considerations

The profiles defined in this document allow an RP to request specific information about the client and allow an IdP to disclose information about that client. In this sense, IdPs MUST apply policy to decide what information is released to a particular RP. Moreover, the identity of the client is typically hidden from the RP unless provided by the IdP. Conversely, the RP does typically know the realm of the IdP, as it is required to route the RADIUS packets to the right destination.

The kind of information that is released by the IdP can include generic attributes such as affiliation shared by many clients. But even these generic attributes can help to identify a specific client. Other kinds of attributes may also provide an RP with the ability to link the same client between different sessions. Finally, other kinds of attributes might provide a group of RPs with the ability to link the client between them or with personally identifiable information about the client.

These profiles do not directly provide a client with a mechanism to express preferences about what information is released. That information can be expressed out of band, for example, as part of the enrollment process.

The RP may disclose privacy-sensitive information about itself as part of the request, although this is unlikely in typical deployments.

If RADIUS proxies are used and encryption is not used, the attributes disclosed by the IdP are visible to the proxies. This is a significant privacy exposure in some deployments. Ongoing work is exploring mechanisms for creating TLS connections directly between the RADIUS client and the RADIUS server to reduce this exposure. If proxies are used, the impact of exposing SAML Assertions to the proxies needs to be carefully considered.

The use of TLS to provide confidentiality for the RADIUS exchange is strongly encouraged. Without this, passive eavesdroppers can observe the assertions.

10. Security Considerations

In this specification, the RP MUST trust any statement in the SAML messages from the IdP in the same way that it trusts information contained in RADIUS attributes. These entities MUST trust the RADIUS infrastructure to provide integrity of the SAML messages.

Furthermore, the RP MUST apply policy and filter the information based on what information the IdP is permitted to assert and on what trust is reasonable to place in proxies between them.

XML signatures and encryption are provided as an OPTIONAL mechanism for end-to-end security. These mechanisms can protect SAML messages from being modified by proxies in the RADIUS infrastructure. These mechanisms are not mandatory to implement. It is believed that ongoing work to provide direct TLS connections between a RADIUS client and RADIUS server will provide similar assurances but better deployability. XML security is appropriate for deployments where end-to-end security is required but proxies cannot be removed or where SAML messages need to be verified at a later time or by parties not involved in the authentication exchange.

11. IANA Considerations

11.1. RADIUS Attributes

The Attribute Types and Attribute Values defined in this document have been registered by the Internet Assigned Numbers Authority (IANA) from the RADIUS namespaces as described in the "IANA Considerations" section of [RFC3575], in accordance with BCP 26 [RFC5226]. For RADIUS packets, attributes, and registries created by this document, IANA has placed them at <http://www.iana.org/assignments/radius-types>.

In particular, this document defines two new RADIUS attributes, entitled "SAML-Assertion" and "SAML-Protocol" (see Section 3), with assigned values of 245.1 and 245.2 from the long extended space [RFC6929]:

Type	Ext. Type	Name	Length	Meaning
245	1	SAML-Assertion	>=5	Encodes a SAML Assertion
245	2	SAML-Protocol	>=5	Encodes a SAML protocol message

11.2. ABFAB Parameters

A new top-level registry has been created, entitled "Application Bridging for Federated Access Beyond Web (ABFAB) Parameters".

In this top-level registry, a sub-registry entitled "ABFAB URN Parameters" has been created. Registration in this registry is via IETF Review or Expert Review procedures [RFC5226].

This paragraph gives guidance to designated experts. Registrations in this registry are generally only expected as part of protocols published as RFCs on the IETF stream; other URIs are expected to be better choices for non-IETF work. Expert review is permitted mainly to allow early registration related to specifications under development when the community believes they have reached sufficient maturity. The expert SHOULD evaluate the maturity and stability of such an IETF-stream specification. Experts SHOULD review anything not from the IETF stream for consistency and consensus with current practice. Today, such requests would not typically be approved.

If a parameter named "paramname" is registered in this registry, then its URN will be "urn:ietf:params:abfab:paramname". The initial registrations are as follows:

Parameter	Reference
bindings:radius	Section 4
nameid-format:nai	Section 5
profiles:authentication	Section 7
profiles:query	Section 8
cm:user	Section 6
cm:machine	Section 6

ABFAB Parameters

11.3. Registration of the ABFAB URN Namespace

IANA has registered the "abfab" URN sub-namespace in the IETF URN sub-namespace for protocol parameters defined in [RFC3553].

Registry Name: abfab

Specification: RFC 7833 (this document)

Repository: ABFAB URN Parameters (Section 11.2)

Index Value: Sub-parameters MUST be specified in UTF-8, using standard URI encoding where necessary.

12. References

12.1. Normative References

[OASIS.saml-bindings-2.0-os]
Cantor, S., Hirsch, F., Kemp, J., Philpott, R., and E. Maler, "Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-bindings-2.0-os, March 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf>.

[OASIS.saml-core-2.0-os]
Cantor, S., Kemp, J., Philpott, R., and E. Maler, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-core-2.0-os, March 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>.

[OASIS.saml-metadata-2.0-os]
Cantor, S., Moreh, J., Philpott, R., and E. Maler, "Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-metadata-2.0-os, March 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf>.

[OASIS.saml-profiles-2.0-os]
Hughes, J., Cantor, S., Hodges, J., Hirsch, F., Mishra, P., Philpott, R., and E. Maler, "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard OASIS.saml-profiles-2.0-os, March 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf>.

[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>.

[RFC3575] Aboba, B., "IANA Considerations for RADIUS (Remote Authentication Dial In User Service)", RFC 3575, DOI 10.17487/RFC3575, July 2003, <http://www.rfc-editor.org/info/rfc3575>.

[RFC3579] Aboba, B. and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)", RFC 3579, DOI 10.17487/RFC3579, September 2003, <http://www.rfc-editor.org/info/rfc3579>.

[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>.

[RFC6929] DeKok, A. and A. Lior, "Remote Authentication Dial In User Service (RADIUS) Protocol Extensions", RFC 6929, DOI 10.17487/RFC6929, April 2013, <http://www.rfc-editor.org/info/rfc6929>.

[RFC7542] DeKok, A., "The Network Access Identifier", RFC 7542, DOI 10.17487/RFC7542, May 2015, <http://www.rfc-editor.org/info/rfc7542>.

12.2. Informative References

[RADIUS-Large-Pkts]
Hartman, S., "Larger Packets for RADIUS over TCP", Work in Progress, draft-ietf-radext-bigger-packets-07, April 2016.

[RFC3553] Mealling, M., Masinter, L., Hardie, T., and G. Klyne, "An IETF URN Sub-namespace for Registered Protocol Parameters", BCP 73, RFC 3553, DOI 10.17487/RFC3553, June 2003, <http://www.rfc-editor.org/info/rfc3553>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[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>.

[RFC7055] Hartman, S., Ed., and J. Howlett, "A GSS-API Mechanism for the Extensible Authentication Protocol", RFC 7055, DOI 10.17487/RFC7055, December 2013, <http://www.rfc-editor.org/info/rfc7055>.

[RFC7499] Perez-Mendez, A., Ed., Marin-Lopez, R., Pereniguez-Garcia, F., Lopez-Millan, G., Lopez, D., and A. DeKok, "Support of Fragmentation of RADIUS Packets", RFC 7499, DOI 10.17487/RFC7499, April 2015, <http://www.rfc-editor.org/info/rfc7499>.

[RFC7831] Howlett, J., Hartman, S., Tschofenig, H., and J. Schaad, "Application Bridging for Federated Access Beyond Web (ABFAB) Architecture", RFC 7831, DOI 10.17487/RFC7831, May 2016, <http://www.rfc-editor.org/info/rfc7831>.

[W3C.REC-xmlschema-1]
Thompson, H., Beech, D., Maloney, M., and N. Mendelsohn, "XML Schema Part 1: Structures Second Edition", W3C REC-xmlschema-1, October 2004, <http://www.w3.org/TR/xmlschema-1/>.

Appendix A. XML Schema

The following schema formally defines the "urn:ietf:params:xml:ns:abfab" namespace used in this document, in conformance with [W3C.REC-xmlschema-1]. Although XML validation is optional, the schema that follows is the normative definition of the constructs it defines. Where the schema differs from any prose in this specification, the schema takes precedence.

```
<schema
  targetNamespace="urn:ietf:params:xml:ns:abfab"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
  xmlns:abfab="urn:ietf:params:xml:ns:abfab"
  elementFormDefault="unqualified"
  attributeFormDefault="unqualified"
  blockDefault="substitution"
  version="1.0">

  <import namespace="urn:oasis:names:tc:SAML:2.0:metadata"/>

  <complexType name="RADIUSIDPDescriptorType">
    <complexContent>
      <extension base="md:RoleDescriptorType">
        <sequence>
          <element ref="abfab:RADIUSIDPService"
                   minOccurs="0" maxOccurs="unbounded"/>
          <element ref="abfab:RADIUSRealm"
                   minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <element name="RADIUSIDPService" type="md:EndpointType"/>
  <element name="RADIUSRealm" type="string"/>
```

```
  <complexType name="RADIUSRPDescriptorType">
    <complexContent>
      <extension base="md:RoleDescriptorType">
        <sequence>
          <element ref="md:RADIUSRPService"
                        minOccurs="0" maxOccurs="unbounded"/>
          <element ref="md:RADIUSNasIpAddress"
                        minOccurs="0" maxOccurs="unbounded"/>
          <element ref="md:RADIUSNasIdentifier"
                        minOccurs="0" maxOccurs="unbounded"/>
          <element ref="md:RADIUSGssEapName"
                        minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <element name="RADIUSRPService" type="md:EndpointType"/>
  <element name="RADIUSNasIpAddress" type="string"/>
  <element name="RADIUSNasIdentifier" type="string"/>
  <element name="RADIUSGssEapName" type="string"/>
</schema>
```

Acknowledgments

The authors would like to acknowledge the OASIS Security Services (SAML) Technical Committee, and Scott Cantor in particular, for their help with the SAML-related material.

The authors would also like to acknowledge the collaboration of Jim Schaad, Leif Johansson, Klaas Wierenga, Stephen Farrell, Gabriel Lopez-Millan, and Rafa Marin-Lopez, who have provided valuable comments on this document.

Authors' Addresses

Josh Howlett
Jisc
Lumen House, Library Avenue, Harwell
Oxford OX11 0SG
United Kingdom

Phone: +44 1235 822363
Email: Josh.Howlett@ja.net

Sam Hartman
Painless Security

Email: hartmans-ietf@mit.edu

Alejandro Perez-Mendez (editor)
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia 30100
Spain

Phone: +34 868 88 46 44
Email: alex@um.es