TOC

Network Working Group	S. Hartman
Internet-Draft	Painless Security
Intended status: Standards Track	J. Howlett
Expires: January 6, 2011	JANET(UK)
	July 5, 2010

Name Attributes for the GSS-API EAP mechanism
draft-hartman-gss-eap-naming-00

Abstract

The naming extensions to the Generic Security Services Application Programming interface provide a mechanism for applications to discover authorization and personalization information associated with GSS-API names. The Extensible Authentication Protocol GSS-API mechanism allows an Authentication/Authorization/Accounting peer to provide authorization attributes along side an authentication response. It also provides mechanisms to process Security Assertion Markup Language (SAML) messages provided in the AAA response. This document describes the necessary information to use the naming extensions API to access that information.

Status of this Memo

Copyright Notice

Table of Contents

---

1. Introduction

TOC

The naming extensions [I-D.ietf-kitten-gssapi-naming-exts] (Williams, N. and L. Johansson, “GSS-API Naming Extensions,” June 2010.)to the Generic Security Services Application Programming interface (GSS-API) [RFC2743] (Linn, J., “Generic Security Service Application Program Interface Version 2, Update 1,” January 2000.) provide a mechanism for applications to discover authorization and personalization information associated with GSS-API names. The Extensible Authentication Protocol GSS-API mechanism [I-D.howlett-eap-gss] (Hartman, S. and J. Howlett, “A GSS-API Mechanism for the Extensible Authentication Protocol,” March 2010.) allows an Authentication/Authorization/Accounting peer to provide authorization attributes along side an authentication response. It also provides mechanisms to process Security Assertion Markup Language (SAML) messages provided in the AAA response. This document describes the necessary information to use the naming extensions API to access that information.

TOC

2. Requirements notation

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 [RFC2119] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

3. Naming Extensions and SAML

TOC

SAML assertions can carry attributes describing properties of the subject of the assertion. For example, an assertion might carry an attribute describing the organizational affiliation or e-mail address of a subject. According to Section 8.2 and 2.7.3.1 of [SAMLCORE], the name of an attribute has two parts. The first is a URI describing the format of the name. The second part, whose form depends on the format URI, is the actual name. As of June 2010, GSS-API name attributes take the form of a single URI.

Administrators may need to type SAML attribute names into configuration files or otherwise tell applications how to find attributes. It is desirable to support accessing these attributes from applications that have no awareness of SAML. So, the GSS-API attribute name should be something that an administrator can reasonably easily construct from a SAML attribute name. In particular, adding or removing URI escapes, base64 encoding or similar transformations would significantly decrease usability.

Instead, it seems desirable to extend GSS-API naming extensions to support concepts such as SAML names where the format is specified separately. The format of GSS-API attribute names should be changed. If no space character is found in the name, then the name is interpreted as a URI describing the attribute. Otherwise, the portion from the beginning of the buffer to the first space is interpreted as a URI describing the form and interpretation of the rest of the buffer; this portion is known as the attribute type URI.

4. RADIUS and Authenticated Attributes

TOC

GSS-API naming extensions have the concept of an authenticated name attribute. The mechanism guarantees that the contents of an authenticated name attribute are an authenticated statement from the trusted source of the peer credential. The fact that an attribute is authenticated does not imply that the trusted source of the peer credential is authorized to assert the attribute.

In the federated context, the trusted source of the peer credential is typically some identity provider. In the GSS EAP mechanism, information is combined from AAA and SAML sources. The SAML IDP and home AAA server are assumed to be in the same trust domain. However, this trust domain is not typically the same as the trust domain of the service. Typically, the IDP is run by another organization in the same federation. The IDP is trusted to make some statements, particularly related to the context of a federation. For example, an academic federation's participants would typically trust an IDP's assertions about whether someone was a student or a professor. However that same IDP would not typically be trusted to make assertions about local entitlements such as group membership. Thus, a service MUST make a policy decision about whether the IDP is permitted to assert a particular attribute and about whether the asserted value is acceptable.

In contrast, attributes in an enterprise context are often verified by a central authentication infrastructure that is trusted to assert most or all attributes. For example, in a Kerberos infrastructure, the KDC typically indicates group membership information for clients to a server using KDC-authenticated authorization data.

The context of an attribute is an important property of that attribute; trust context is an important part of the context. Applications will often want to treat an attribute in a federated context the same as an attribute in an enterprise context. In order for applications to distinguish the context of attributes, attributes with different context need different names. For example, the name of an attribute containing the initiator's e-mail address in a federated context needs to be different from the name containing the initiator's e-mail address in a different context. The determination of trust from this context information can never be exact: Kerberos typically is used in environments where the KDC is fairly trusted, but an application could have a key in a realm that it does not fully trust. Similarly, SAML is typically in a federated context, but an organization could use SAML for internal authentication as well.

It would be convenient to use the same GSS-API attribute names for the same information regardless of context. However, when considering attribute names it is critical to consider the appropriate interpretation of that name and the distinctions an application will need to make about the name. As a result, it is often the case that attributes from two different mechanisms will have different names. However, the local implementation of the mechanism and layers in the GSS-API implementation above the mechanism can make the job of the application easier. If local policy permits an attribute to be trusted, then the attribute can be copied to a name whose context indicates that local policy has been applied. For example, an implementation could have an attribute for e-mail address that received the value both of a SAML mechanism and Kerberos mechanism's e-mail address attributes after local policy is applied. Such mechanism-level attributes can also be used to normalize the format of attribute values.

In the case of GSS-EAP, the attribute names need to be specific to SAML attributes obtained via AAA transport.

---

5. Name Attributes for GSS-EAP

TOC

This section describes how SAML assertions, SAML attributes and RADIUS attributes received with the GSS-EAP mechanism are named.

5.1. Assertions

TOC

Implementations of GSS-EAP MUST support an attribute with the name "urn:ietf:params:gss-eap:saml-aaa-assertion". The value of this attribute is the assertion carried in the AAA protocol. This attribute is absent from a given acceptor name if no such assertion is present or if the assertion fails local policy checks. This attribute is always authentic when present: authentication only succeeds if the AAA exchange is successfully authenticated. However, users of the GSS-API MUST confirm that the attribute is authenticated because some mechanisms MAY permit an initiator to assert an unauthenticated version of this attribute.

5.2. SAML Attributes

TOC

Each attribute carried in the assertion SHOULD also be a GSS name attribute. The name of this attribute has three parts, all separated by an ASCII space character. The first part is urn:ietf:params:gss-eap:saml-attr. The second part is the URI for the SAML attribute name format. The final part is the name of the SAML attribute. If the mechanism performs an additional attribute query, the retrieved attributes SHOULD be GSS-API name attributes using the same name syntax.

These attributes SHOULD be marked authenticated if they are contained in SAML assertions that have been successfully validated back to the trusted source of the peer credential. In the GSS-EAP mechanism, a SAML assertion carried in an integrity-protected and authenticated AAA protocol SHALL be sufficiently validated. An implementation MAY apply local policy checks to this assertion and discard it if it is unacceptable according to these checks.

Attribute query results made based on this assertion also count as originating with the source of the peer credential. The implementation

MUST validate the authenticity of these results before they are processed.

5.3. RADIUS Attributes

TOC

A mechanism needs to be created to give applications access to AAA AVPs carried along with an access-accept message.

6. Security Considerations

TOC

This needs to be written.

7. IANA Considerations

TOC

This section needs to include URN registrations within the IETF namespace for URNs that are used.

8. Normative References

TOC

[I-D.howlett-eap-gss]	Hartman, S. and J. Howlett, “A GSS-API Mechanism for the Extensible Authentication Protocol,” draft-howlett-eap-gss-00 (work in progress), March 2010 (TXT).
[I-D.ietf-kitten-gssapi-naming-exts]	Williams, N. and L. Johansson, “GSS-API Naming Extensions,” draft-ietf-kitten-gssapi-naming-exts-08 (work in progress), June 2010 (TXT).
[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC2743]	Linn, J., “Generic Security Service Application Program Interface Version 2, Update 1,” RFC 2743, January 2000 (TXT).

Authors' Addresses

TOC

Sam Hartman
Painless Security

Email: hartmans-ietf@mit.edu

Josh Howlett
JANET(UK)
Email: josh.howlett@ja.net