Geopriv
Internet-Draft
Expires: January 14, 2006

C. Guenther
Siemens

July 13, 2005

SAML in Authorization Policies
draft-guenther-geopriv-saml-policy-01.txt

Status of this Memo

Copyright Notice

Abstract

Rules of an authorization policy prescribe under which conditions an entity or subject has which permissions. Existing policies support identity-based authorization by matching the authenticated identity of the entity requesting access to a resource with the available policies. This document is about formulating policy rules that express conditions with respect to SAML assertions, thereby supporting non-identity-based authorization and anonymity.

Table of Contents

1. Introduction

The Security Assertion Markup Language, see [SAMLCore], is an XML sublanguage for exchanging security information. It is suitable for expressing assertions concerning previously performed authentication procedures and authorization decisions. For example, a SAML assertion can be used by the assertion issuer to assure that the assertion subject (e.g., a person, a network entity, ...) has been authenticated by means of a specific authentication method. A recipient of such an assertion - if it has trust in the assertion issuer and the integrity of the assertion - can then base its authorization decisions on this assertion.

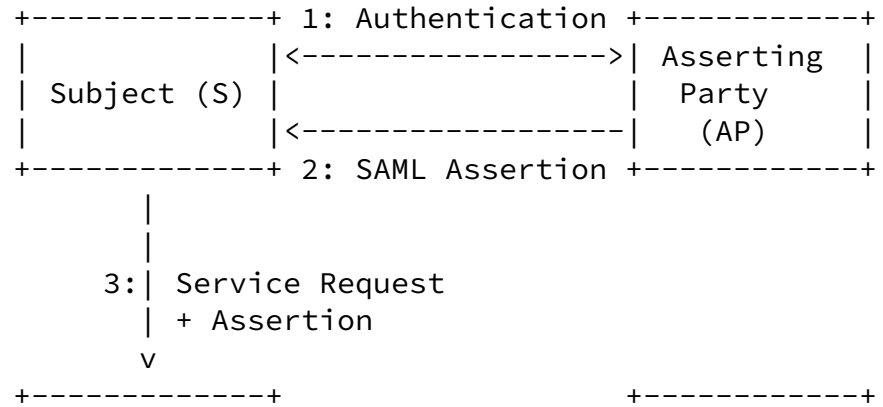
This document is about defining an extension to the Common Policy markup language, see [I-D.ietf-geopriv-common-policy], that allows to express conditions with respect to statements contained in SAML assertions. It shall be possible to express authorization policy rules of the following fashion: If the SAML assertion has been issued by the assertion issuer A and if the assertion assures that the assertion subject S has been authenticated by means of the authenticated method M, then S is permitted to ... .

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 [RFC2119].

3. Basic Scenario

Figure 1 depicts a basic scenario in the scope of this document: a Subject S wishes to have access to a certain resource (e.g., location information of a particular entity). After a successful authentication protocol execution between S and the Asserting Party (AP), see step 1, the AP issues a SAML assertion (step 2), which asserts that S has been authenticated by AP using method M and is associated with a certain set of attributes.



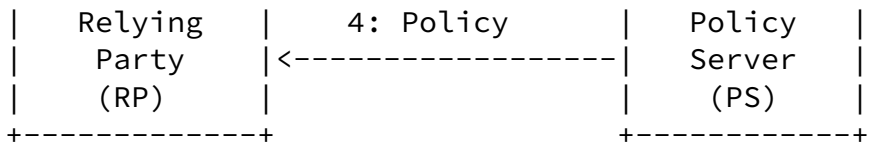


Figure 1: Basic Scenario

After receipt of the assertion, the Relying Party (RP) can base its resource access authorization decision on this assertion. The authorization policy governing access to the requested resource is stored at the Policy Server (PS). Thanks to the language elements introduced in this document, this policy can contain rules whose conditions parts express properties that the SAML assertion must meet in order to make the rule match.

4. SAML Condition Example

Each policy rule of the Common Policy markup language [I-D.ietf-geopriv-common-policy] consists of a <conditions>, an <actions> and a <transformations> element (all of which are optional elements). The Common Policy XML schema defines the <conditions> element in such a way that it allows for any child elements that belong to XML namespaces different from the common policy namespace.

This document defines a new XML element, namely, the <samlcondition> element, whose purpose is to be used as such a child element of the common policy <conditions> element. This paragraph provides an example of an XML document valid with respect to the SAML Condition schema (as shown in Section 5) and the Common Policy schema (as listed in Section 6).

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleset
  xmlns="urn:ietf:params:xml:ns:common-policy"
  xmlns:samlcond="urn:ietf:params:xml:ns:saml-condition"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
  "urn:ietf:params:xml:ns:common-policy common-policy.xsd
   urn:ietf:params:xml:ns:saml-condition saml-condition.xsd">

  <rule id="Hz90op54I">

    <conditions>

      <validity>
        <from>2005-08-02T17:00:00-05:00</from>
        <to>2005-08-04T19:00:00-05:00</to>
      </validity>

      <samlcond:samlcondition>

        <samlcond:issuer>idp.com</samlcond:issuer>

        <samlcond:subject>
          <samlcond:nameid>bob@example.com</samlcond:nameid>
        </samlcond:subject>

        <samlcond:authnstatement>

          <samlcond:authncontext>
            <samlcond:authncontextclassref>
    urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
```



```
            </samlcond:authncontextclassref>
          </samlcond:authncontext>

          <samlcond:authncontext>
            <samlcond:authncontextclassref>
    urn:oasis:names:tc:SAML:2.0:ac:classes:X509
            </samlcond:authncontextclassref>
          </samlcond:authncontext>
```

```
          </saml:authnstatement>

        </saml:samlcondition>


      </conditions>

      <actions></actions>

    </rule>

</ruleset>
```

The rule set in this example consists of one rule only. The <conditions> part of the rule consists of a <validity> condition (defined by the Common Policy schema) and a <samlcondition> (defined by this document in Section 5). The <validity> element specifies the time period during which the rule is applicable. The <samlcondition> element as shown above evaluates to true if and only if the SAML assertion presented to the Relying Party satisfies the following properties:

1) The issuer of the SAML assertion is idp.com.

2) The subject of the SAML assertion is bob@example.com.

3) The authentication context class referenced in the SAML assertion is PasswordProtectedTransport (i.e., the subject of the assertion has authenticated to the Asserting Party through the presentation of a password over a protected session) or X509 (i.e., the subject of the assertion has authenticated to the Asserting Party by means of a digital signature where the key was validated as part of a X.509 public key infrastructure).

To be more precisely, the SAML assertion presented to the Relying Party has to satisfy the following properties to make the <samlcondition> element evaluate to true:

1) The content of the <saml:Issuer> element of the SAML assertion

must equal the string "idp.com".

2) The SAML assertion must contain a <saml:Subject> child element (which is optional by the SAML assertion schema), and this <saml:Subject> element must contain a <saml:NameID> element whose content equals the string "bob@example.com".

3) The SAML assertion must contain an <saml:AuthnStatement> element with an <saml:AuthnContext> child element that possesses an <saml:AuthnContextClassRef> child element whose content is either urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport or urn:oasis:names:tc:SAML:2.0:ac:classes:X509.

The complete list of Authentication Context types defined by SAML can be found in [SAMLAuthnContext].

5. SAML Condition Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="urn:ietf:params:xml:ns:saml-condition"
  xmlns:samlcond="urn:ietf:params:xml:ns:saml-condition"
  xmlns:cp="urn:ietf:params:xml:ns:common-policy"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xs:import namespace="urn:ietf:params:xml:ns:common-policy"
    schemaLocation="common-policy.xsd"/>

  <!-- Definition of element types for saml conditions -->
  <!-- Element names correspond to SAML element names  -->

  <xs:element name="issuer" type="xs:string"/>

  <xs:element name="subject">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="nameid" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="authnstatement">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="samlcond:authncontext"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="authncontext">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="samlcond:authncontextclassref"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="authncontextclassref" type="xs:anyURI"/>
```

```
    <!-- Definition of saml conditions -->
```



```
    <xs:element name="samlcondition">
      <xs:complexType>
        <xs:sequence>

          <xs:element ref="samlcond:issuer"
            minOccurs="0" maxOccurs="unbounded"/>

          <xs:element ref="samlcond:subject"
            minOccurs="0" maxOccurs="unbounded"/>

          <xs:element ref="samlcond:authnstatement"
            minOccurs="0" maxOccurs="unbounded"/>

        </xs:sequence>
      </xs:complexType>
    </xs:element>

  </xs:schema>
```

6. Common Policy Schema

Just for the sake of completeness, this section contains that version of the Common Policy XML schema that defines - along with the schema specified in Section 5 - the XML language to which the example in Section 4 belongs.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="urn:ietf:params:xml:ns:common-policy"
  xmlns:cp="urn:ietf:params:xml:ns:common-policy"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <!-- Rule Set -->

  <xs:element name="ruleset">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="rule" type="cp:ruleType"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!-- Rule -->

  <xs:complexType name="ruleType">

    <xs:sequence>
```

```
        <!-- Conditions -->

        <xs:element name="conditions" minOccurs="0">
          <xs:complexType>
            <xs:sequence>

              <xs:element name="validity" minOccurs="0">
                <xs:complexType>
                  <xs:all>
                    <xs:element name="from" type="xs:dateTime"/>
                    <xs:element name="to" type="xs:dateTime"/>
                  </xs:all>
                </xs:complexType>
              </xs:element>
```



```
              <xs:element name="identity" minOccurs="0">
                <xs:complexType>
                  <xs:choice>

                    <xs:element name="id" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:attribute name="val"
                          type="xs:string" use="required"/>
                      </xs:complexType>
                    </xs:element>

                    <xs:sequence>
                      <xs:element name="domain" type="xs:string"/>
                      <xs:element name="except"
                        minOccurs="0" maxOccurs="unbounded">
                        <xs:complexType>
                          <xs:attribute name="val"
                            type="xs:string" use="required"/>
                        </xs:complexType>
                      </xs:element>
                    </xs:sequence>

                    <xs:element name="anonymous">
                      <xs:complexType>
                        <xs:sequence>
```

```
                    <xs:element name="domain"
                      minOccurs="0" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:attribute name="val"
                          type="xs:string" use="required"/>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>

              <xs:element name="exception">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="domain"
                      minOccurs="0" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:attribute name="val"
                          type="xs:string" use="required"/>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="id"
                      minOccurs="0" maxOccurs="unbounded">
```



```
                      <xs:complexType>
                        <xs:attribute name="val"
                          type="xs:string" use="required"/>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>

              <xs:element name="any-identity" type="xs:string"/>

            </xs:choice>
          </xs:complexType>
        </xs:element>

        <xs:element name="sphere"
          minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
```

```
                  <xs:attribute name="val"
                    type="xs:string" use="required"/>
                </xs:complexType>
              </xs:element>

              <xs:any namespace="##other" processContents="lax"
                minOccurs="0" maxOccurs="unbounded"/>

            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <!-- Actions -->

        <xs:element name="actions" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:any namespace="##other" processContents="lax"
                minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <!-- Transformations -->

        <xs:element name="transformations" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:any namespace="##other" processContents="lax"
                minOccurs="0" maxOccurs="unbounded" />
```



```
            </xs:sequence>
          </xs:complexType>
        </xs:element>

      </xs:sequence>

      <xs:attribute name="id" type="xs:string" use="required"/>

    </xs:complexType>

  </xs:schema>
```

7. Security Considerations

[tbd]

8. IANA Considerations

[tbd]

9. Open Issues

1) SAML assertions with authorization decision statements.

2) SAML assertions with attribute statements.

3) Alignment with Common Policy markup language.

4) Security Considerations.

5) IANA considerations.

10. References

10.1 Normative References

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

[SAMLAuthnContext] OASIS, "Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-authn-context-2.0-os.pdf, March 2005.

[SAMLCore] OASIS, "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-core-2.0-os.pdf, March 2005.

10.2 Informative References

[I-D.ietf-geopriv-common-policy] Schulzrinne, H., Morris, J., Tschofenig, H., Polk, J., and J. Rosenberg, "A Document Format for Expressing Privacy Preferences", draft-ietf-geopriv-common-policy-04 (work in progress), February 2005.

Author's Address

Christian Guenther
Siemens
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Email: christian.guenther@siemens.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.