Network Working Group
Request for Comments: 4991
Category: Standards Track

A. Newton
VeriSign, Inc.

August 2007

A Common Schema for Internet Registry Information Service Transfer Protocols

Status of This Memo

Copyright Notice

Abstract

This document describes an XML Schema for use by Internet Registry Information Service (IRIS) application transfer protocols that share common characteristics. It describes common information about the transfer protocol, such as version, supported extensions, and supported security mechanisms.

Table of Contents

1. Introduction

IRIS [8] has two transfer protocols, LWZ (lightweight using compression) [9] and XPC (XML pipelining with chunks) [10], that share common negotiation mechanisms. Both transfer protocols have a need for the server to provide rich status information to clients during protocol negotiation. In many cases, this status information would be too complex to describe using simple bit fields and length-specified octet sequences. This document defines an XML Schema for this rich status information and describes the usage of conformant XML for conveying this status information.

This document defines five types of information used in the negotiation of protocol capabilities: version, size, authentication success, authentication failure, and other information. The version information is used to negotiate the versions and extensions to the transfer protocol, the application operations protocol, and data models used by the application operations. Size information is used to indicate request and response size capabilities and errors. Authentication success and failure information is used to indicate the outcome of an authentication action. Other types of information may also be conveyed that is generally a result of an error but cannot be corrected through defined protocol behavior.

As an example, upon initiation of a connection, a server may send version information informing the client of the data models supported by the server and the security mechanisms supported by the server. The client may then respond appropriately. For example, the client may not recognize any of the data models supported by the server, and therefore close the connection. On the other hand, the client may recognize the data models and the security mechanisms and begin the procedure to initialize a security mechanism with the server before proceeding to query data according to a recognized data model.

Both LWZ [9] and XPC [10] provide examples of the usage of the XML Schema defined in this document.

2. Document 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 [6].

3. Formal XML Syntax

The following is the XML Schema used to define transfer protocol status information. See the following specifications: [2], [3], [4], [5]. Updates or changes to this schema require a document that UPDATES or OBSOLETES this document.

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:iristrans="urn:ietf:params:xml:ns:iris-transport"
        targetNamespace="urn:ietf:params:xml:ns:iris-transport"
        elementFormDefault="qualified" >

<annotation>
  <documentation>
    A schema for describing status information
    for use by multiple transfer protocols.
  </documentation>
</annotation>

<element name="versions">
  <complexType>
    <sequence>
      <element name="transferProtocol" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element name="application" minOccurs="0"
              maxOccurs="unbounded">
              <complexType>
                <sequence>
                  <element name="dataModel" minOccurs="0"
                    maxOccurs="unbounded">
                    <complexType>
                      <attribute name="protocolId" type="token"
                        use="required" />
                      <attribute name="extensionIds"
                        type="normalizedString" />
                    </complexType>
                  </element>
                </sequence>
                <attribute name="protocolId" type="token"
                  use="required" />
                <attribute name="extensionIds"
                  type="normalizedString" />
              </complexType>
            </element>
          </sequence>
          <attribute name="protocolId" type="token" use="required"
```

```
              />
            <attribute name="extensionIds" type="normalizedString" />
            <attribute name="authenticationIds"
              type="normalizedString" />
            <attribute name="responseSizeOctets"
              type="positiveInteger" />
            <attribute name="requestSizeOctets"
              type="positiveInteger" />
          </complexType>
        </element>
      </sequence>
    </complexType>
  </element>

  <element name="size">
    <complexType>
      <sequence>
        <element name="request"
          minOccurs="0"
          type="iristrans:octetsType" />
        <element name="response"
          minOccurs="0"
          type="iristrans:octetsType" />
      </sequence>
    </complexType>
  </element>

  <complexType name="octetsType">
    <choice>
      <element name="exceedsMaximum">
        <complexType/>
      </element>
      <element name="octets" type="positiveInteger" />
    </choice>
  </complexType>

  <element name="authenticationSuccess">
    <complexType>
      <sequence>
        <element name="description" minOccurs="0"
          maxOccurs="unbounded">
          <complexType>
            <simpleContent>
              <extension base="string">
                <attribute name="language" type="language"
                  use="required"/>
              </extension>
            </simpleContent>
```

```
          </complexType>
        </element>
        <element name="data" minOccurs="0" maxOccurs="1"
          type="base64Binary"/>
      </sequence>
    </complexType>
  </element>

  <element name="authenticationFailure">
    <complexType>
      <sequence>
        <element name="description" minOccurs="0"
          maxOccurs="unbounded">
          <complexType>
            <simpleContent>
              <extension base="string">
                <attribute name="language" type="language"
                  use="required"/>
              </extension>
            </simpleContent>
          </complexType>
        </element>
      </sequence>
    </complexType>
  </element>

  <element name="other">
    <complexType>
      <sequence>
        <element name="description" minOccurs="0"
          maxOccurs="unbounded">
          <complexType>
            <simpleContent>
              <extension base="string">
                <attribute name="language" type="language"
                  use="required"/>
              </extension>
            </simpleContent>
          </complexType>
        </element>
      </sequence>
      <attribute type="token" name="type" use="required"/>
    </complexType>
  </element>

  </schema>
```

4. Version Information

The <versions> element is used to describe version information about the transfer protocol, the application protocol, and data models used by the application protocol.

The <versions> element has one or more <transferProtocol> child elements. <transferProtocol> elements have zero or more <application> child elements. And <application> elements have zero or more <dataModel> elements. Each of these element types has a 'protocolId' attribute identifying the protocol they represent and an optional 'extensionIds' attribute identifying the protocol extensions they support.

During capabilities negotiation, it is expected that both sides of the negotiation recognize the 'protocolId' value of the <transferProtocol> element and at least one of the <application> and <dataModel> elements. If the negotiation produces a situation where this is not possible, an error SHOULD be given and communication ended. It is not expected that each side must recognize the 'extensionIds' values, and unrecognized 'extensionIds' values MUST be ignored.

Additionally, the <transferProtocol> element has optional 'authenticationIds', 'responseSizeOctets', and 'requestSizeOctets' attributes. The 'authenticationIds' attribute identifies authentication mechanisms supported by the associated transfer protocol. The 'responseSizeOctets' attribute describes the maximum response size in octets the server will give. The 'requestSizeOctets' attribute describes the maximum request size in octets the server will accept.

The protocol, extension, and authentication mechanism identifiers are of no specific type, and this document defines none. Specifications using this XML Schema MUST define the identifiers for use with the <versions> element and its children.

The meaning of octets for the transfer of data is counted in different ways for different transfer protocols. Some transfer protocols need only to specify the octets of the data being transferred, while other transfer protocols need to account for additional octets used to transfer the data. Specifications using this XML Schema MUST describe how these octet counts are calculated.

The following is example XML describing version information.

```
<versions xmlns="urn:ietf:params:xml:ns:iris-transport">
  <transferProtocol protocolId="iris.lwz"
    authenticationIds="PLAIN EXTERNAL">
    <application protocolId="urn:ietf:params:xml:ns:iris1"
      extensionIds="http://example.com/SIMPLEBAG">
      <dataModel protocolId="urn:ietf:params:xml:ns:dchk1"/>
      <dataModel protocolId="urn:ietf:params:xml:ns:dreg1"/>
    </application>
  </transferProtocol>
</versions>
```

Version Information Example

5. Size Information

The <size> element provides a means for a server to communicate to a client that a given request has exceeded a negotiated size (<request>) or that a response to a given request will exceed a negotiated size (<response>).

A server may indicate one of two size conditions by specifying the following child elements:

<exceedsMaximum> - this child element simply indicates that the size exceeded the negotiated size.

<octets> - this child element indicates that the size exceeded the negotiated size and conveys the number of octets that is the maximum for a request if the parent element is a <request> element or the number of octets needed to provide the response if the parent element is a <response> element.

The meaning of octets for the transfer of data is counted in different ways for different transfer protocols. Some transfer protocols need only to specify the octets of the data being transferred, while other transfer protocols need to account for additional octets used to transfer the data. Specifications using this XML Schema MUST describe how these octet counts are calculated.

The following is example XML describing size information.

```
<size xmlns="urn:ietf:params:xml:ns:iris-transport">
  <response>
    <octets>1211</octets>
  </response>
</size>
```

Size Information Example

6. Authentication Success Information

The <authenticationSuccess> element indicates that a client has successfully authenticated to a server. Along with this indication, it can provide text that may be presented to a user with regard to this successful authentication using child <description> elements.

Each <description> element MUST have a 'language' attribute describing the language of the content of the <description> element. Clients are not expected to concatenate multiple descriptions; therefore, servers MUST NOT provide multiple <description> elements with the same language descriptor.

Finally, additional security data may be sent back with the authentication success message using the <data> element. The content of this element is of the base64Binary simple type.

The following is example XML describing authentication success information.

```
<authenticationSuccess
  xmlns="urn:ietf:params:xml:ns:iris-transport">
  <description language="en">
    user 'bob' authenticates via password
  </description>
</authenticationSuccess>
```

Authentication Success Example

7. Authentication Failure Information

The <authenticationFailure> element indicates that a client has failed to properly authenticate to a server. Along with this indication, it can provide text that may be presented to a user with regard to this authentication failure using child <description> elements.

Each <description> element MUST have a 'language' attribute describing the language of the content of the <description> element. Clients are not expected to concatenate multiple descriptions; therefore, servers MUST NOT provide multiple <description> elements with the same language descriptor.

The following is example XML describing authentication failure information.

```
<authenticationFailure
  xmlns="urn:ietf:params:xml:ns:iris-transport">
  <description language="en">
    please consult your admin if you have forgotten your password
  </description>
</authenticationFailure>
```

Authentication Failure Example

8. Other Information

The <other> element conveys status information that may require interpretation by a human to be meaningful. This element has a required 'type' attribute, which contains an identifier regarding the nature of the information. This document does not define any identifiers for use in this attribute, but the intent is that these identifiers are well-known so that clients may take different classes of action based on the content of this attribute. Therefore, specifications making use of this XML Schema MUST define these identifiers.

The <other> element may have zero or more <description> elements. Each <description> element MUST have a 'language' attribute describing the language of the content of the <description> element. Servers may use these child elements to convey textual information to clients regarding the class (or type) of status information being specified by the <other> element. Clients are not expected to concatenate multiple descriptions; therefore, servers MUST NOT provide multiple <description> elements with the same language descriptor.

The following is example XML describing other information.

```
<other xmlns="urn:ietf:params:xml:ns:iris-transport" type="system">
  <description language="en">unavailable, come back
    later</description>
</other>
```

Other Information Example

9. Internationalization Considerations

XML processors are obliged to recognize both UTF-8 and UTF-16 [1] encodings. XML provides for mechanisms to identify and use other character encodings. Application transfer protocols MUST define which additional character encodings, if any, are to be allowed in the use of the XML defined in this document.

10. IANA Considerations

10.1. XML Namespace URN Registration

This document makes use of the XML namespace and schema registry specified in XML_URN [7]. Accordingly, the following registrations have been made by IANA:

- XML Namespace URN/URI:
  - urn:ietf:params:xml:ns:iris-transport
- Contact:
  - Andrew Newton <andy@hxr.us>
- XML:
  - None
- XML Schema URN/URI:
  - urn:ietf:params:xml:schema:iris-transport
- Contact:
  - Andrew Newton <andy@hxr.us>
- XML:
  - The XML Schema specified in Section 3

11. Security Considerations

Transfer protocols using XML conformant to the XML Schema in this document and offering security properties such as authentication and confidentiality SHOULD offer an initial message from the server to the client using the <versions> element. This <versions> element SHOULD contain all relevant authentication identifiers in its 'authenticationId' attribute. The purpose of providing this initial message is to help thwart downgrade attacks.

12. References

12.1. Normative References

[1] The Unicode Consortium, "The Unicode Standard, Version 3", ISBN 0-201-61633-5, 2000, <The Unicode Standard, Version 3>.

[2] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0", W3C XML, February 1998, <http://www.w3.org/TR/1998/REC-xml-19980210>.

[3] World Wide Web Consortium, "Namespaces in XML", W3C XML Namespaces, January 1999, <http://www.w3.org/TR/1999/REC-xml-names-19990114>.

[4] World Wide Web Consortium, "XML Schema Part 2: Datatypes", W3C XML Schema, October 2004, <http://www.w3.org/TR/xmlschema-2/>.

[5] World Wide Web Consortium, "XML Schema Part 1: Structures", W3C XML Schema, October 2004, <http://www.w3.org/TR/xmlschema-1/>.

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

[7] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

12.2. Informative References

[8] Newton, A. and M. Sanz, "IRIS: The Internet Registry Information Service (IRIS) Core Protocol", RFC 3981, January 2005.

[9] Newton, A., "A Lightweight UDP Transfer Protocol for the Internet Registry Information Service", RFC 4993, August 2007.

[10] Newton, A., "XML Pipelining with Chunks for the Internet Registry Information Service", RFC 4992, August 2007.

Appendix A. Contributors

Substantive contributions to this document have been provided by the members of the IETF's CRISP Working Group, especially Robert Martin-Legene, Milena Caires, and David Blacka.

Author's Address

Andrew L. Newton
VeriSign, Inc.
21345 Ridgetop Circle
Sterling, VA 20166
USA

Phone: +1 703 948 3382
EMail: andy@hxr.us
URI: http://www.verisignlabs.com/

Full Copyright Statement

Intellectual Property

Acknowledgement

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