Network Working Group
Request for Comments: 3983
Category: Standards Track

A. Newton
VeriSign, Inc.
M. Sanz
DENIC eG

January 2005

Using the Internet Registry Information Service (IRIS) over the Blocks Extensible Exchange Protocol (BEEP)

Status of This Memo

Copyright Notice

Abstract

This document specifies how to use the Blocks Extensible Exchange Protocol (BEEP) as the application transport substrate for the Internet Registry Information Service (IRIS).

Table of Contents

1. Introduction and Motivations

The proposal in this document describes the IRIS [6] application transport binding that uses BEEP [2]. Requirements for IRIS and the specification in this document are outlined in CRISP [19].

The choice of BEEP as the transport substrate is primarily driven by the need to reuse an existing, well-understood protocol with all the necessary features to support the requirements. This would give implementers a wealth of toolkits and debugging gear for use in constructing both servers and clients and allow operators to apply existing experience in issues of deployment. The construction of a simple application transport for the specific purpose of IRIS would yield a similar standard, though likely smaller and less complete, after taking into consideration matters such as framing and authentication.

Precedents for using other transport mechanisms in layered applications do not seem to fit with the design goals of IRIS. HTTP [15] offers many features employed for use by similar applications. However, IRIS is not intended to be put to uses such as bypassing firewalls, commingling URI schemes, or any other methods that might lead to confusion between IRIS and traditional World Wide Web applications. Beyond adhering to the guidelines spelled out in RFC 3205 [16], the use of HTTP also offers many other challenges that quickly erode its appeal. For example, the appropriate use of TLS [4] with HTTP is defined by RFC 2817 [14], but the common use, as described in RFC 2818 [18], is usually the only method in most implementations.

Finally, the use of IRIS directly over TCP, such as that specified by EPP-TCP [17], does not offer the client negotiation characteristics needed by a referral application in which a single client, in processing a query, may traverse multiple servers operating with different parameters.

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 BCP 14, RFC 2119 [5].

3. BEEP Profile Identification

The BEEP profile identifier for IRIS is a URI composed of the IRIS schema URN, followed by a slash, followed by an IRIS registry type (which is a URN).

In this profile identifier, the IRIS schema MUST be abbreviated according to the rules of IRIS. This is possible because the IRIS schema URN is compliant with XML_URN [20].

The registry type URN MUST be abbreviated according to the rules of IRIS (see [6]). This is possible because the registry type URN is compliant with XML_URN [20].

The following is an example of an IRIS profile identifier for BEEP. It identifies the version of IRIS to match that specified by "urn:iana:params:xml:ns:iris1" with a registry type URN of "urn:iana:params:xml:ns:dreg1":

```
http://iana.org/beep/iris1/dreg1
```

The full ABNF [8] follows, with certain values included from IRIS [6]:

```
profile             = profile-uri "/" iris-urn-abbrev
                      "/" registry-urn-abbrev
profile-uri         = "http://iana.org/beep/"
iris-urn-abbrev     = // as specified by IRIS
registry-urn-abbrev = // as specified by IRIS
```

This URI is used in the "profile" element in BEEP during channel creation. According to the rules of BEEP, multiple "profile" elements may be offered, thus allowing negotiation of the version of IRIS to be used for every registry type being served.

Once this profile is accepted and the channel is created, the channel is considered ready to exchange IRIS messages. A server MUST honor queries for all advertised registry types on any channel opened with an IRIS profile URI.

4. IRIS Message Packages

The BEEP profile for IRIS transmits XML [1] containing the requests and responses for IRIS registries. These XML instances MUST be encoded as Unicode [9] using the media-type of "application/xml" according to RFC 3023 [11].

XML processors are obliged to recognize both UTF-8 and UTF-16 [9] encodings. XML allows mechanisms to identify and use other character encodings by means of the "encoding" attribute in the declaration. Absence of this attribute or a byte order mark (BOM) indicates a default of UTF-8 encoding. Thus, for compatibility reasons, and per RFC 2277 [12], use of UTF-8 is RECOMMENDED with this transport mapping. UTF-16 is OPTIONAL. Other encodings MUST NOT be used.

A registry type MAY define other message packages that are not IRIS XML instances (e.g., binary images referenced by an IRIS response).

5. IRIS Message Patterns

5.1. Registry Dependent Patterns

Because each registry type is defined by a separate BEEP profile (see [6]), each registry type MAY define a different message pattern. These patterns MUST be within the allowable scope of BEEP [2]. If a registry type does not explicitly define a message pattern, the default pattern is used (see Section 5.2).

However, each registry type MUST be capable of supporting the default pattern (Section 5.2) for use with the <lookupEntity> query in IRIS.

5.2. Default Pattern

The default BEEP profile for IRIS only has a one-to-one request/response message pattern. This exchange involves sending an IRIS XML instance, which results in a response of an IRIS XML instance.

The client sends the request by using an "MSG" message containing a valid IRIS XML instance. The server responds with an "RPY" message containing a valid IRIS XML instance. The "ERR" message is used for sending fault codes. The list of allowable fault codes is listed in BEEP [2].

6. Server Authentication Methods

6.1. Registry Dependent Methods

When the TLS [4] tuning profile in BEEP is used, it is possible to verify the authenticity of the server. However, a convention is needed to conduct this authentication. This convention dictates the name of the authority a client uses to ask for authentication credentials so that the server knows which set of credentials to pass back. Because this is dependent on the authority component of the URI, each registry type SHOULD define a server authentication method.

If a registry type does not explicitly define a server authentication method, the basic server authentication method (Section 6.2) is used.

6.2. Basic Server Authentication Method

The basic server authentication method is as follows:

1. When connecting to a server, the client MUST present the name of the authority to the server by using the BEEP [2] serverName mechanism. For instance, if the URI "iris:dreg1//com/domain/example.com" is being resolved, the client would use the serverName="com" attribute during the BEEP session instantiation.

2. During TLS negotiation, the server presents to the client a certificate for the authority given in serverName. This certificate MUST be an X.509 certificate [10]. This certificate MUST contain the authority in either the subjectDN or the subjectAltName extension of type dNSName.

3. The certificate MUST be cryptographically verified according to the procedures of TLS.

4. The client then checks the subject of the certificate for a case insensitive match in the following order:

    1. Any of the dNSName types in the subjectAltName.
    2. The subjectDN consisting solely of 'dc' components, in which each 'dc' component represents a label from the authority name (e.g., example.com is dc=example, dc=com).
    3. A subjectDN in which the left-most component is a 'cn' component containing the name of the authority. A wildcard character ('*') MAY be used as the left-most label of the name in the 'cn' component.

If the subject of the certificate does not match any of these name components, then the certificate is invalid for representing the authority.

7. IRIS Transport Mapping Definitions

This section lists the definitions required by IRIS [6] for transport mappings.

7.1. URI Scheme

The URI scheme name specific to BEEP over IRIS MUST be "iris.beep".

7.2. Application Protocol Label

The application protocol label MUST be "iris.beep".

7.3. Allowable Character Sets

See Sections 4 and 10.

7.4. BEEP Mapping

The mapping of IRIS in this document is specific to RFC 3080 [2]. This mapping MUST use TCP as specified by RFC 3081 [3].

8. Registrations

8.1. BEEP Profile Registration

Profile Identification: http://iana.org/beep/iris1

Messages exchanged during Channel Creation: none

Messages starting one-to-one exchanges: IRIS XML instance

Messages in positive replies: IRIS XML instance

Messages in negative replies: none

Messages in one-to-many exchanges: none

Message Syntax: IRIS XML instances as defined by IRIS [6]

Message Semantics: request/response exchanges as defined by IRIS [6]

Contact Information: Andrew Newton <andy@hxr.us> and Marcos Sanz <sanz@denic.de>

8.2. URI Scheme Registration

URL scheme name: iris.beep

URL scheme syntax: defined in Section 7.1 and [6]

Character encoding considerations: as defined in RFC 2396 [7]

Intended usage: identifies an IRIS entity made available using the BEEP profile for IRIS

Applications using this scheme: defined in IRIS [6]

Interoperability considerations: n/a

Security Considerations: defined in Section 12.

Relevant Publications: BEEP [2] and IRIS [6]

Contact Information: Andrew Newton <andy@hxr.us> and Marcos Sanz <sanz@denic.de>

Author/Change controller: the IESG

8.3. Well-Known TCP Port Registration

Protocol Number: TCP

Message Formats, Types, Opcodes, and Sequences: defined in Sections 3, 4, and 5.

Functions: defined in IRIS [6]

Use of Broadcast/Multicast: none

Proposed Name: IRIS over BEEP

Short name: iris.beep

Contact Information: Andrew Newton <andy@hxr.us> and Marcos Sanz <sanz@denic.de>

8.4. S-NAPTR Registration

Application Protocol Label: iris.beep

Intended usage: identifies an IRIS server using BEEP

Interoperability considerations: n/a

Security Considerations: defined in Section 12

Relevant Publications: BEEP [2] and IRIS [6]

Contact Information: Andrew Newton <andy@hxr.us> and Marcos Sanz <sanz@denic.de>

Author/Change controller: the IESG

9. Registry Definition Checklist

Specifications of registry types MUST include the following explicit definitions:

- message pattern -- a definition of the message pattern for use with BEEP, or a declaration to use the default message pattern in Section 5.2.

- server authentication method -- a definition of the method to use for server authentication with TLS, a declaration to use the basic server authentication method in Section 6.2, or a declaration to use no server authentication at all.

10. Internationalization Considerations

See Section 4.

11. IANA Considerations

Registrations with the IANA are described in Section 8.

12. Security Considerations

Implementers should be fully aware of the security considerations given by IRIS [6], BEEP [2], and TLS [4]. With respect to server authentication with the use of TLS, see Section 6.

Clients SHOULD be prepared to use the following BEEP tuning profiles:

- http://iana.org/beep/SASL/DIGEST-MD5 -- for user authentication without the need of session encryption.
- http://iana.org/beep/SASL/OTP -- for user authentication without the need of session encryption.
- http://iana.org/beep/TLS using the TLS_RSA_WITH_3DES_EDE_CBC_SHA cipher -- for encryption.
- http://iana.org/beep/TLS using the TLS_RSA_WITH_3DES_EDE_CBC_SHA cipher with client-side certificates -- for encryption and user authentication.
- http://iana.org/beep/TLS using the TLS_RSA_WITH_AES_128_CBC_SHA cipher -- for encryption. See [13].
- http://iana.org/beep/TLS using the TLS_RSA_WITH_AES_128_CBC_SHA cipher with client-side certificates -- for encryption and user authentication. See [13].
- http://iana.org/beep/TLS using the TLS_RSA_WITH_AES_256_CBC_SHA cipher -- for encryption. See [13].
- http://iana.org/beep/TLS using the TLS_RSA_WITH_AES_256_CBC_SHA cipher with client-side certificates -- for encryption and user authentication. See [13].

Anonymous client access SHOULD be considered in one of two methods:

1. When no authentication tuning profile has been used.
2. Using the SASL anonymous profile: http://iana.org/beep/SASL/ANONYMOUS

IRIS contains a referral mechanism as a standard course of operation. However, care should be taken that user authentication mechanisms do not hand user credentials to untrusted servers. Therefore, clients SHOULD NOT use the http://iana.org/beep/SASL/PLAIN tuning profile. As specified by SASL/PLAIN, clients MUST NOT use the http://iana.org/beep/SASL/PLAIN tuning profile without first encrypting the TCP session (e.g. such as with the http://iana.org/beep/TLS tuning profile).

13. References

13.1. Normative References

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

[2] Rose, M., "The Blocks Extensible Exchange Protocol Core", RFC 3080, March 2001.

[3] Rose, M., "Mapping the BEEP Core onto TCP", RFC 3081, March 2001.

[4] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

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

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

[7] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

[8] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

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

[10] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[11] Murata, M., Laurent, S. St., and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[12] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, January 1998.

[13] Chown, P., "Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)", RFC 3268, June 2002.

13.2. Informative References

[14] Khare, R. and S. Lawrence, "Upgrading to TLS Within HTTP/1.1", RFC 2817, May 2000.

[15] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[16] Moore, K., "On the use of HTTP as a Substrate", BCP 56, RFC 3205, February 2002.

[17] Hollenbeck, S., "EPP TCP Transport", Work in Progress, January 2002.

[18] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[19] Newton, A., "Cross Registry Internet Service Protocol (CRISP) Requirements", RFC 3707, February 2004.

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

14. Authors' Addresses

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

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

Marcos Sanz
DENIC eG
Wiesenhuettenplatz 26
D-60329 Frankfurt
Germany

EMail: sanz@denic.de
URI: http://www.denic.de/

Full Copyright Statement

Intellectual Property

Acknowledgement

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