Network Working Group
Request for Comments: 3529
Category: Experimental

W. Harold
IBM

April 2003

Using Extensible Markup Language-Remote Procedure Calling (XML-RPC) in Blocks Extensible Exchange Protocol (BEEP)

Status of this Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Copyright Notice

Abstract

XML-RPC is an Extensible Markup Language-Remote Procedure Calling protocol that works over the Internet. It defines an XML format for messages that are transfered between clients and servers using HTTP. An XML-RPC message encodes either a procedure to be invoked by the server, along with the parameters to use in the invocation, or the result of an invocation. Procedure parameters and results can be scalars, numbers, strings, dates, etc.; they can also be complex record and list structures.

This document specifies a how to use the Blocks Extensible Exchange Protocol (BEEP) to transfer messages encoded in the XML-RPC format between clients and servers.

Table of Contents

1. Introduction

This memo specifies how messages encoded in the XML-RPC [1] format are transmitted using a BEEP profile [2].

Throughout this memo, the terms "request" and "response" refer to the "methodCall" and "methodResponse" elements defined by the XML-RPC specification [1]. Further the terms "peer", "client", "server", and "one-to-one" are used in the context of BEEP. In particular, Sections 2.1 and 2.1.1 of [2] discuss BEEP roles and exchange styles.

2. BEEP Profile Identification

The BEEP profile for XML-RPC is identified as

```
http://iana.org/beep/transient/xmlrpc
```

in the BEEP "profile" element during channel creation.

In BEEP, when the first channel is successfully created, the "serverName" attribute in the "start" element identifies the "virtual host" associated with the peer acting in the server role, e.g.,

```
<start number='1' serverName='stateserver.example.com'>
    <profile uri='http://iana.org/beep/transient/xmlrpc' />
</start>
```

The "serverName" attribute is analogous to HTTP's "Host" request-header field (c.f., Section 14.23 of [3]).

There are two states in the BEEP profile for XML-RPC, "boot", the profile's initial state, and "ready":

- In the "boot" state, the peer requesting the creation of the channel sends a "bootmsg" (either during channel initialization or in a "MSG" message).
  - If the other peer sends a "bootrpy" (either during channel initialization or in a "RPY" message), then the "ready" state is entered
  - Otherwise, the other peer sends an "error" (either during channel initialization or in a "ERR" message), and no state change occurs.
- In the "ready" state, the initiating peer begins an XML-RPC message pattern by sending a "MSG" message containing a request. The other peer completes the message pattern by sending back a "RPY" message containing a response.

2.1 Profile Initialization

The boot message is used to identify the resource accessed by the channel bound to the BEEP profile for XML-RPC.

The DTD syntax for the boot message and its response are:

```
<!ELEMENT bootmsg     EMPTY>
<!ATTLIST bootmsg
          resource    CDATA             #REQUIRED>

<!ELEMENT bootrpy     EMPTY>
```

The boot message contains a single mandatory attribute: "resource", which is analagous to HTTP's "abs_path" Request-URI parameter (c.f., Section 5.1.2 of [3])

If the peer acting in the server role recognizes the requested resource, it replies with a boot response. Otherwise, if the boot message is improperly formed, or if the requested resource isn't recognized, the peer acting in the server role replies with an error message (c.f., Section 7.1 of [2]).

Typically, the boot message and its response are exchanged during channel initialization (c.f., Section 2.3.1.2 of [2]).

For example, here the boot message and its response are exchanged during channel initialization:

```
C: <start number='1' serverName='stateserver.example.com'>
C:     <profile uri='http://iana.org/beep/transient/xmlrpc'>
C:         <![CDATA[<bootmsg resource='/NumberToName' />]]>
C:     </profile>
C: </start>

S: <profile uri='http://iana.org/beep/transient/xmlrpc'>
S:     <![CDATA[<bootrpy />]]>
S: </profile>
```

The channel bound to the BEEP profile for XML-RPC is now in the "ready" state.

Alternatively, here is an example in which the boot exchange is unsuccessful:

```
C: <start number='1' serverName='stateserver.example.com'>
C:     <profile uri='http://iana.org/beep/transient/xmlrpc'>
C:         <![CDATA[<bootmsg resource='/NameToCapital' />]]>
C:     </profile>
C: </start>

S: <profile uri='http://iana.org/beep/transient/xmlrpc'>
S:     <![CDATA[<error code='550'>resource not
S:                                 supported</error>]]>
S: </profile>
```

Although the channel was created successfully, it remains in the "boot" state.

3. XML-RPC Message Packages

The BEEP profile for XML-RPC transmits requests and responses encoded as UTF-8 using the media type "application/xml" [4], e.g.,

```
I: MSG 1 1 . 0 364
I: Content-Type: application/xml
I:
I: <?xml version="1.0"?>
I:   <methodCall>
I:     <methodName>examples.getStateName</methodName>
I:     <params>
I:       <param>
I:         <value><i4>41</i4></value>
I:       </param>
```

```
I:     </params>
I:   </methodCall>
I: END
```

and its associated response

```
L: RPY 1 1 . 201 100
L: Content-Type: application/xml
L:
L: <?xml version="1.0"?>
L:   <methodResponse>
L:     <params>
L:       <param>
L:         <value><string>South Dakota</string></value>
L:       </param>
L:     </params>
L:   </methodRespose>
L: END
```

4. XML-RPC Message Exchange

A request/response exchange involves sending a request, which results in a response being returned.

The BEEP profile for XML-RPC achieves this using a one-to-one exchange, in which the client sends a "MSG" message containing an request, and the server sends back a "RPY" message containing an response.

The BEEP profile for XML-RPC does not use the "ERR" message for XML-RPC faults when performing one-to-one exchanges. Whatever response is generated by the server is always returned in the "RPY" message.

5. URL Schemes

This memo defines two URL schemes, "xmlrpc.beep" and "xmlrpc.beeps", which identify the use of XML-RPC over BEEP over TCP. Note that, at present, a "generic" URL scheme for XML-RPC is not defined.

5.1 The xmlrpc.beep URL Scheme

The "xmlrpc.beep" URL scheme uses the "generic URI" syntax defined in Section 3 of [5], specifically:

- the value "xmlrpc.beep" is used for the scheme component; and,
- the server-based naming authority defined in Section 3.2.2 of [5] is used for the authority component.

- the path component maps to the "resource" component of the boot message sent during profile initialization (if absent, it defaults to "/").

The values of both the scheme and authority components are case-insensitive.

For example, the URL

```
xmlrpc.beep://stateserver.example.com/NumberToName
```

might result in the example shown in Section 2.1.

5.1.1 Resolving IP/TCP Address Information

The "xmlrpc.beep" URL scheme indicates the use of the BEEP profile for XML-RPC running over TCP/IP.

If the authority component contains a domain name and a port number, e.g.,

```
xmlrpc.beep://stateserver.example.com:1026
```

then the DNS is queried for the A RRs corresponding to the domain name, and the port number is used directly.

If the authority component contains a domain name and no port number, e.g.,

```
xmlrpc.beep://stateserver.example.com
```

the SRV algorithm [6] is used with a service parameter of "xmlrpc-beep" and a protocol parameter of "tcp" to determine the IP/TCP addressing information. If no appropriate SRV RRs are found (e.g., for "_xmlrpc-beep._tcp.stateserver.example.com"), then the DNS is queried for the A RRs corresponding to the domain name and the port number used is assigned by the IANA for the registration in Section 6.4.

If the authority component contains an IP address, e.g.,

```
xmlrpc.beep://10.0.0.2:1026
```

then the DNS is not queried, and the IP address is used directly. If a port number is present, it is used directly; otherwise, the port number used is assigned by the IANA for the registration in Section 6.4.

While the use of literal IPv6 addresses in URLs is discouraged, if a literal IPv6 address is used in a "xmlrpc.beep" URL, it must conform to the syntax specified in [7].

5.2 The xmlrpc.beeps URL Scheme

The "xmlrpc.beeps" URL scheme is identical, in all ways, to the "xmlrpc.beep" URL scheme specified in Section 5.1, with the exception that prior to starting the BEEP profile for XML-RPC, the BEEP session must be tuned for privacy. In particular, note that both URL schemes use the identical algorithms and parameters for address resolution as specified in Section 5.1.1 (e.g., the same service name for SRV lookups, the same port number for TCP, and so on).

There are two ways to perform privacy tuning on a BEEP session, either:

- a transport security profile may be successfully started; or,
- a user authentication profile that supports transport security may be successfully started.

In either case the client must present the authority component of the URL in the "serverName" attribute of the "start" element it uses to tune the session for privacy.

When TLS is used for privacy the client must verify that the authority component of the URL matches the server's identity as presented in the server's certificate. Section 2.4 of [9] describes the matching process.

For the URL:

```
xmlrpc.beeps://stateserver.example.com/NumberToName
```

the whole process might look like:

```
S: <wait for incoming connection @ stateserver.example.com>
C: <open connection to stateserver.example.com>
C: RPY 0 0 . 0 52
C: Content-Type: application/xml
C:
C: <greeting />
C: END
S: RPY 0 0 . 0 110
S: Content-Type: application/xml
S:
S: <greeting>
```

```
S:   <profile uri='http://iana.org/beep/TLS' />
S:   <profile uri='http://iana.org/beep/SASL/DIGEST-MD5' />
S: </greeting>
S: END
C: MSG 0 1 . 52 158
C: Content-Type: application/xml
C:
C: <start number='1' serverName='stateserver.example.com'>
C:   <profile uri='http://iana.org/beep/TLS'>
C:     <![CDATA[<ready />]]>
C:   </profile>
C: </start>
C: END
S: RPY 0 1 . 110 121
S: Content-Type: application/xml
S:
S: <profile uri='http://iana.org/beep/TLS'>
S:   <![CDATA[<proceed />]]>
S: </profile>
S: END

... TLS negotiations ...

S: RPY 0 0 . 0 88
S: Content-Type: application/xml
S:
S: <greeting>
S:   <profile uri='http://iana.org/beep/transient/xmlrpc'>
S: </greeting>
S: END
C: RPY 0 0 . 0 52
C: Content-Type: application/xml
C:
C: <greeting />
C: END

... use the server's certificate to verify that it is
    in fact stateserver.example.com ...

C: MSG 0 1 . 112 211
C: Content-Type: application/xml
C:
C: <start number='3' serverName='stateserver.example.com'>
C:     <profile uri='http://iana.org/beep/transient/xmlrpc'>
C:         <![CDATA[<bootmsg resource='/NumberToName' />]]>
C:     </profile>
C: </start>
C: END
```

```
S: RPY 0 2 . 341 402
S: Content-Type: application/xml
S:
S: <profile uri='http://iana.org/beep/transient/xmlrpc'>
S:     <![CDATA[<bootrpy />]]>
S: </profile>
S: END
```

6. Initial Registrations

6.1 Registration: The XML-RPC Profile

Profile Identification: http://iana.org/beep/transient/xmlrpc

Messages exchanged during Channel Creation: bootmsg, bootrpy

Messages starting one-to-one exchanges: bootmsg, methodCall

Messages in positive replies: bootrpy, methodResponse

Messages in negative replies: error

Messages in one-to-many exchanges: none

Message Syntax: methodCall, methodResponse as defined in [1]

Message Semantics: c.f., [1]

Contact Information: Ward Harold <wharold@us.ibm.com>

6.2 Registration: The xmlrpc.beep URL Scheme

URL scheme name: xmlrpc.beep

URL scheme syntax: c.f., Section 5.1

Character encoding considerations: c.f., the "generic URI" syntax defined in Section 3 of [5]

Intended usage: identifies a XML-RPC resource made available using the BEEP profile for XML-RPC

Applications using this scheme: c.f., "Intended usage", above

Interoperability considerations: n/a

Security Considerations: c.f., Section 7

Relevant Publications: c.f., [1], and [2]

Contact Information: Ward Harold <wharold@us.ibm.com>

Author/Change controller: the IESG

6.3 Registration: The xmlrpc.beeps URL Scheme

URL scheme name: xmlrpc.beeps

URL scheme syntax: c.f., Section 5.2

Character encoding considerations: c.f., the "generic URI" syntax defined in Section 3 of [5]

Intended usage: identifies a XML-RPC resource made available using the BEEP profile for XML-RPC after the BEEP session has been tuned for privacy

Applications using this scheme: c.f., "Intended usage", above

Interoperability considerations: n/a

Security Considerations: c.f., Section 7

Relevant Publications: c.f., [1], and [2]

Contact Information: Ward Harold <wharold@us.ibm.com>

Author/Change controller: the IESG

6.4 Registration: The System (Well-Known) TCP port number for XML-RPC over BEEP

Protocol Number: TCP

Message Formats, Types, Opcodes, and Sequences: c.f., Section 2.1

Functions: c.f., [1]

Use of Broadcast/Multicast: none

Proposed Name: XML-RPC over BEEP

Short name: xmlrpc-beep

Contact Information: Ward Harold <wharold@us.ibm.com>

7. Security Considerations

Although service provisioning is a policy matter, at a minimum, all implementations must provide the following tuning profiles:

for authentication: http://iana.org/beep/SASL/DIGEST-MD5

for confidentiality: http://iana.org/beep/TLS (using the TLS_RSA_WITH_3DES_EDE_CBC_SHA cipher)

for both: http://iana.org/beep/TLS (using the TLS_RSA_WITH_3DES_EDE_CBC_SHA cipher supporting client-side certificates)

Further, implementations may choose to offer MIME-based security services providing message integrity and confidentiality, such as OpenPGP [8] or S/MIME [10].

Regardless, consult [2]'s Section 9 for a discussion of BEEP-specific security issues.

8. References

[1] Winer, D., "XML-RPC Specification", January 1999, http://www.xmlrpc.com/spec

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

[3] 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.

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

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

[6] Gulbrandsen, A., Vixie, P. and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[7] Hinden, R., Carpenter, B. and L. Masinter, "Format for Literal IPv6 Addresses in URL's", RFC 2732, December 1999.

[8] Elkins, M., Del Torto, D., Levien, R. and T. Roessler, "MIME Security with OpenPGP", RFC 3156, August 2001.

[9] Newman, C., "Using TLS with IMAP, POP3 and ACAP", RFC 2595, June 1999.

[10] Ramsdell, B., "S/MIME Version 3 Message Specification", RFC 2633, June 1999.

[11] O'Tuathail, E. and M. Rose, "Using the Simple Object Access Protocol (SOAP) in Blocks Extensible Exchange Protocol (BEEP)", RFC 3288, June 2002.

Appendix A. Acknowledgements

This document is based, in part, on Using SOAP in BEEP [11] and the author gratefully acknowledges the contributions of Marshall Rose

Appendix B. IANA Considerations

The IANA has registered the profile specified in Section 6.1, and has selected an IANA-specific URI, e.g.,

```
http://iana.org/beep/xmlrpc
```

The IANA has registered "xmlrpc.beep" and "xmlrpc.beeps" as URL schemes, as specified in Section 6.2 and Section 6.3, respectively. (See: http://www.iana.org/assignments/uri-schemes)

The IANA has registered "XML-RPC over BEEP" as a TCP port number (602), as specified in Section 6.4. (See: http://www.iana.org/assignments/port-numbers)

Author's Address

Ward K Harold
IBM
11400 Burnet Road
Austin, Texas 78759
US

Phone: +1 512 838 3622
EMail: wharold@us.ibm.com

Full Copyright Statement

Acknowledgement

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