Internet Engineering Task Force
INTERNET DRAFT
Standards Track: Updates RFC 2608
7 June 2001
Expires in six months

Erik Guttman
Sun Microsystems

Vendor Extensions for Service Location Protocol, Version 2

draft-guttman-svrloc-vendor-ext-04.txt

Status of this Memo

Abstract

The Service Location Protocol, Version 2 [1] allows for vendor extensibility. This document updates the standard, specifying how each of these features can be used safely (with no possibility of name collisions). While proprietary protocol extensions are not encouraged by IETF standards, it is important that when they are undertaken they not hinder interoperability of compliant implementations. This document also defines a new extension to SLPv2: The Vendor Opaque extension.

Table of Contents

1.0 Introduction 2

1.1 Terminology 3

2.0 Enterprise Numbers 3

3.0 Naming Authorities 3

4.0 Vendor Defined Attributes 4

5.0 Vendor Opaque Extension 5

5.1 Vendor Opaque Extension Format 6

5.2 Example: Acme Extension for UA Authentication . . 6

6.0 Extensions Requiring IETF Action 7

7.0 IANA Considerations 7

8.0 Security Considerations 7

References 8

Author's Address 8

1.0 Introduction

The Service Location Protocol, Version 2 [1] defines a number of features which are extensible. This document clarifies exactly which mechanisms can be used to that end (Sections 3-5) and which cannot (Section 6). This document specifies conventions that ensure the protocol extension mechanisms in the SLPv2 specification will not possibly have ambiguous interpretations.

This specification introduces only one new protocol element, the Vendor Opaque Extension. This Extension makes it possible for a vendor to extend SLP independently, once the vendor has registered itself with IANA and obtained an Enterprise Number. This is useful for vendor-specific applications.

Vendor extensions to standard protocols come at a cost.

- Vendor extensions occur without review from the community. They may not make good engineering sense in the context of the protocol they extend, and the engineers responsible may discover this too late.

- Vendor extensions preclude interoperation with compliant but non-extended implementations. There is a real danger of incompatibility if different implementations support different feature sets.

- By extending SLPv2 privately, ubiquitous automatic configuration is impossible, which is the primary benefit of a standard service discovery framework.

For these reasons, registration of service templates with IANA is strongly encouraged! This process is easy and has proved to be rapid (taking less than 2 weeks in most cases).

1.1 Terminology

In this document, the key words "MAY", "MUST", "MUST NOT", "optional", "recommended", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [2].

Service Location Protocol terminology is defined in [1]. IANA registration terminology is defined in [6].

2.0 Enterprise Number

Enterprise Numbers are used to distinguish different vendors in IETF protocols. Vendor Extensions to SLPv2 SHOULD use these values to avoid any possibility of a name space collision. Each vendor is responsible for ensuring that vendor extensions under their own authority are non-conflicting.

RFC 1700 lists the Enterprise Numbers registered at the time of publication as well as rules on how to register new numbers:

> To request an assignment of an Enterprise Number send the complete company name, address, and phone number; and the contact's person complete name, address, phone number, and email mailbox in an email message to <iana-mib@isi.edu>. [3]

The complete up-to-date list is maintained by IANA [4].

3.0 Naming Authorities

Naming Authorities are defined by SLPv2 [1] as an agency or group which catalogues Service Types and attributes.

A Service Type is a string representing a service which can be discovered by SLPv2. Attributes may be associated with a particular Service Type which is advertised by SLPv2.

Service Type strings and service attributes may be registered with IANA by creating a Service Template [5]. The template is included in an internet draft and an email message is sent to srvloc-list@iana.org requesting that the template be included in the Service Template registry. In this case the naming authority for the service type is IANA.

It is also possible for a Vendor to create their own naming authority. In this case, any service type or attributes may be used. SLPv2 allows arbitrary naming authorities to coexist. To use an explicit naming authority, a vendor simply employs their Enterprise Number as a naming authority. For example, for the following (fictitious) Enterprise Number

```
  9999  Acme, Inc.                      Erik Guttman  femur@neato.org
```

the Naming Authority string to use would be "9999". A service: URL which used this Naming Authority to advertise a Roadrunner Detector service could look like

```
   service:roadrunner-detector.9999://neato.org:9341
```

Service types which are defined under a naming authority based on an Enterprise Number are guaranteed not to conflict with other service type strings which mean something entirely different. That is also true of attributes defined for service types defined under a naming authority.

To create a safe naming authority with no possibility of name collisions, a vendor SHOULD use their Enterprise Number as a naming authority.

4.0 Vendor Defined Attributes

SLPv2 [1] suggests that

> Non-standard attribute names SHOULD begin with "x-", because no standard attribute name will ever have those initial characters.

It is possible that two non-standard attributes will conflict that both use the "x-" prefix notation. For that reason, vendors SHOULD use "x-" followed by their Enterprise Number followed by a "-" to guarantee that the non-standard attribute name's interpretation is not ambiguous.

For example, Acme, Inc.'s Enterprise Number is 9999. Say the Service Template for NetHive (a fictitious game) was:

```
template-type=NetHive

template-version=1.0

template-description=
  The popular NetHive game.

template-url-syntax=
  url-path = ; There is no path for a NetHive service URL.

features= string M O
# The list of optional features the NetHive server supports.
secure session, fast mode

current-users= string M
# The list of users currently playing
------------------------------------------------------------
```

Acme's server advertises a feature which is not on the list of standard features, "x-9999-cheat-mode". Only an Acme client would request this attribute to discover servers, since it is not standard.

5.0 Vendor Opaque Extension

SLPv2 [1] defines a protocol extensibility mechanism. SLPv2 Extensions are added at the end of a message and have the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Extension ID          |       Next Extension Offset   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Offset, contd.|                Extension Data                 /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format of the Extension Data depends on the Extension ID. Refer to [6] for a full description of different mechanisms available for registration of values with IANA.

SLPv2 may be extended in any of three ways.

[1] Anyone may request the designated expert for SLP to register a new extension ID with IANA. Send requests to the svrloc-list@iana.org.

     It is recommended that an internet draft specifying this extension be published, with the intention of publishing the document as an Informational RFC. This way others can use the extension as well. This is not a 'vendor extension' - rather this is the preferred way of extending the protocol in a vendor neutral manner.

If no specification is published and the extension is intended for vendor specific use only - the 'Vendor Extension' option below probably makes more sense than assigning an extension ID.

[2] An experimental extension may be done using the range 0x8000 to 0x8FFF. There is always the risk, however, that another vendor will use the same ID, since these IDs are not registered.

[3] A Vendor Extension may be used. This extension allows a Vendor to define their own extensions which are guaranteed to have a unique interpretation. It is OPTIONAL to implement.

5.1. Vendor Opaque Extension Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Extension ID = 0x0003     |       Next Extension Offset   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Offset, contd.|              Enterprise Number                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ent. #, contd.|                Extension Data                 /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Enterprise Number is included in the Extension as a 4 byte unsigned integer value. The Extension Data following is guaranteed to have an unambiguous interpretation determined by the vendor.

5.2 Example: Acme Extension for UA Authentication

For example, the Acme Corporation, whose Enterprise Number is 9999, can define an extensions to SLP to create an application level access control to service information. This would allow replies to be sent only to clients who could authenticate themselves.

The Extension Data has the following form:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Client ID Length        |           Client ID           /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Authenticator                        ...
```

```
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Client ID: The Acme application user ID. Timestamp: # of seconds since 2000. Authenticator is a 16 byte MD5 digest [7] calculated on the following data fields, concatenated together

- UA request bytes, including the header, but not any extensions.
- UA SECRET PASS PHRASE
- Acme UA Authentication Extension - Client ID
- Acme UA Authentication Extension - Timestamp

The SA or DA which receives this extension and supports this extension will check if it (1) recognizes the Client ID, (2) has an associated SECRET PASS PHRASE for it, (3) whether upon calculating an MD5 digest over the same data as listed above it arrives at the same Authenticator value as included in the extension. If all 3 of these steps succeed, the UA has been authenticated.

Note this example is for explanatory purposes only. It would not work well in practice. It requires a shared secret be configured in SAs and DAs, for every UA. Furthermore, the UA secret pass phrase would be susceptible to a dictionary attack.

6.0 Extensions Requiring IETF Action

Terminology and procedures for IETF Actions related to registration of IDs with IANA are defined in [6]. Existing SLPv2 extension assignments are registered with IANA [8].

7.0 IANA Considerations

This document clarifies procedures described in other documents [1] [5]. The Vendor Opaque Extension ID has already been registered [8]. No additional IANA action is required for publication of this document.

8.0 Security Considerations

Vendor extensions may introduce additional security considerations into SLP.

This memo describes mechanisms which are standardized elsewhere [1] [5]. The only protocol mechanism described in this document (see Section 5 above) is no less secure than 'private use' extensions defined in SLPv2 [1].

The example in Section 5.2 above shows how Vendor Opaque Extensions can be used to include an access control mechanism to SLP so that SAs

can enforce an access control policy using an authentication
mechanism. This is merely an example and protocol details were

intentionally not provided. A vendor could, however, create a mechanism similar to this one and provide additional security services to SLPv2 in the manner indicated in the example.

References

[1] Guttman, E., Perkins, C., Veizades, J., Day, M., "Service Location Protocol, Version 2", RFC 2608, July 1999.

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

[3] Reynolds, J., Postel, J., "Assigned Numbers", RFC 1700, October 1994.

[4] ftp://ftp.isi.edu/in-notes/iana/assignments/enterprise-numbers

[5] Guttman, E., Perkins, C., Kempf, J., "Service Templates and URLs", RFC 2609, July 1999.

[6] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[7] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[8] ftp://www.iana.org/assignments/svrloc-extensions

Author's Address

Erik Guttman
Sun Microsystems
Eichhoelzelstr. 7
74915 Waibstadt
Germany

Phone: +49 7263 911 701
Messages: +49 6221 356 202
Email: erik.guttman@sun.com