Network Working Group
Internet-Draft
Expires: January 12, 2006

J. Laganier
DoCoMo Euro-Labs
T. Koponen
HIIT
L. Eggert
NEC
July 11, 2005

Host Identity Protocol (HIP) Registration Extension
draft-koponen-hip-registration-01

Status of this Memo

This Internet-Draft will expire on January 12, 2006.

Copyright Notice

Abstract

This document specifies a registration mechanism for the Host Identity Protocol (HIP) that allows hosts to register with services,

such as HIP rendezvous servers or middleboxes.

1. Introduction

This document specifies an extension to the Host Identity Protocol (HIP) [I-D.ietf-hip-arch]. The extension provides a generic means for a host to register with a service. The service may, for example, be a HIP rendezvous server [I-D.ietf-hip-rvs] or a middlebox [RFC3234].

This document makes no further assumptions about the exact type of service. Likewise, this document does not specify any mechanisms to discover the presence of specific services or means to interact with them after registration. Future documents may describe those operations.

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].

2. Terminology

This section defines terminology that is used throughout the remainder of this document. Please note that terminology shared with other documents is defined elsewhere [I-D.ietf-hip-arch].

Requester:
: a HIP node registering with a HIP registrar to request registration for a service.

Registrar:
: a HIP node offering registration for one or more services.

Service:
: a facility that provides requesters with new capabilities or functionalities operating at the HIP layer. Examples include firewalls that support HIP traversal or HIP rendezvous servers.

Registration:
: shared state stored by a requester and a registrar, allowing the requester to benefit from one or more HIP services offered by the registrar. Each registration has an associated finite lifetime. Requesters can extend established registrations through re-registration (i.e., perform a refresh).

Registration Type:
an identifier for a given service in the registration protocol. For example, the rendezvous service is identified by a specific registration type.

3. HIP Registration Extension Overview

This document does not specify the means by which a requester discovers the availability of a service, or how a requester locates a registrar. After a requester has discovered a registrar, it either initiates HIP base exchange or uses an existing HIP association with the registrar. In both cases, registrars use additional parameters that the remainder of this document defines to announce their quality and grant or refuse registration. Requesters use corresponding parameters to register with the service. The following sections describe the differences between this registration handshake and the standard HIP base exchange [I-D.ietf-hip-base] .

3.1 Registrar Announcing its Ability

A host that is capable and willing to act as a registrar SHOULD include a REG_INFO parameter in the R1 packets it sends during all base exchanges. If it is currently unable to provide services due to transient conditions, it SHOULD include an empty REG_INFO, i.e., one with no services listed. If services can be provided later, it SHOULD send UPDATE packets indicating the current set of services available in a new REG_INFO parameter to all hosts it is associated with.

3.2 Requester Requesting Registration

To request registration with a service, a requester constructs and includes a corresponding REG_REQUEST parameter in an I2 or UPDATE packet it sends to the registrar.

If the requester has no HIP association established with the registrar, it SHOULD already send the REG_REQUEST in the I2 packet. This minimizes the number of packets that need to be exchanged with the registrar. A registrar MAY end a HIP association that does not carry a REG_REQUEST by including a NOTIFY with the type REG_REQUIRED in the R2. In this case, no HIP association is created between the hosts. The REG_REQUIRED notification error type is TBD.

3.3 Registrar Granting or Refusing Service(s) Registration

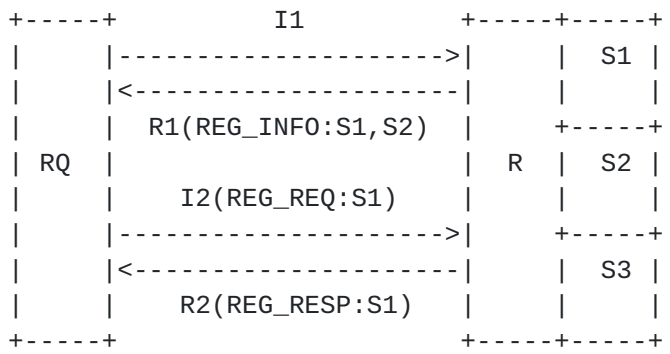
Once registration has been requested, the registrar is able to authenticate the requester based on the host identity included in I2.

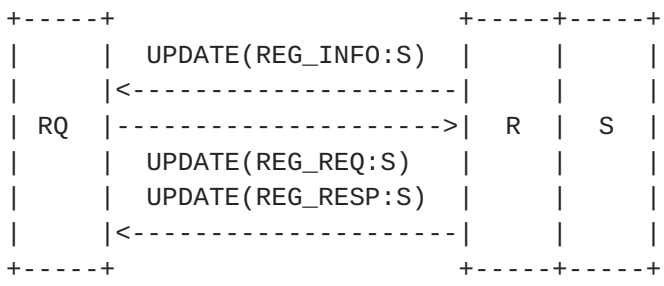
It then verifies the host identity is authorized to register with the requested service(s), based on local policies. The details of this authorization procedure depend on the type of requested service(s) and on the local policies of the registrar, and are therefore not further specified in this document.

After authorization, the registrar includes in its response (i.e., an R2 or an UPDATE, respectively, depending on whether the registration was requested during the base exchange, or using an existing association) a REG_RESPONSE parameter containing the service(s) type(s) for which it has authorized registration, and zero or more REG_FAILED parameter containing the service(s) type(s) for which it has not authorized registration or registration has failed for other reasons. In particular, REG_FAILED with a failure type of zero indicates the service(s) type(s) that require further credentials for registration.

If the registrar requires further authorization and the requester has additional credentials available, the requester SHOULD try to again register with the service after the HIP association has been established.

Successful processing of a REG_RESPONSE parameter creates registration state at the requester. In a similar manner, successful processing of a REG_REQUEST parameter creates registration state at the registrar and possibly at the service. Both the requester and registrar can cancel a registration before it expires, if the services afforded by a registration are no longer needed by the requester, or cannot be provided any longer by the registrar (for instance, because its configuration has changed).





4. Parameter Formats and Processing

This section describes the format and processing of the new parameters introduced by the HIP registration extension.

4.1 Encoding Registration Lifetimes with Exponents

The HIP registration uses an exponential encoding of registration lifetimes. This allows compact encoding of 255 different lifetime values ranging from 4 ms to 178 days into an 8-bit integer field. The lifetime exponent field used throughout this document MUST be interpreted as representing the lifetime value 2^((lifetime - 64)/8) seconds.

4.2 REG_INFO

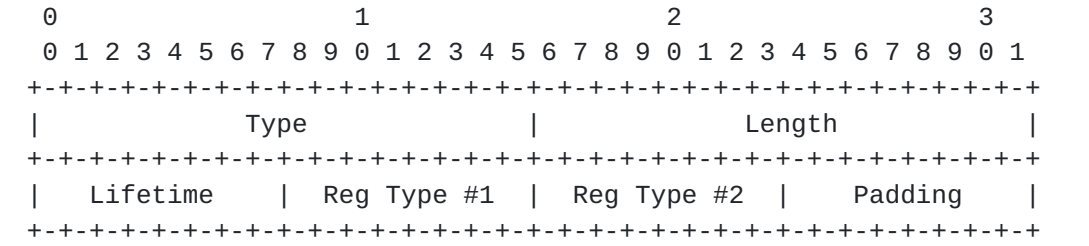
Type	[TBD by IANA (930)]
Length	Length in octets, excluding Type, Length, and Padding.
Min Lifetime	Minimum registration lifetime.
Max Lifetime	Maximum registration lifetime.
Reg Type	The registration types offered by the registrar.

Other documents will define specific values for registration types.

0-200	Reserved by IANA
201-255	Reserved by IANA for private use

Registrars include the parameter in R1 packets in order to announce their registration capabilities. The registrar SHOULD include the parameter in UPDATE packets when its service offering has changed. HIP_SIGNATURE_2 protects the parameter within the R1 packets.

4.3 REG_REQUEST



Type	[TBD by IANA (932)]
Length	Length in octets, excluding Type, Length, and Padding.
Lifetime	Requested registration lifetime.
Reg Type	The preferred registration types in order of preference.

Other documents will define specific values for registration types.

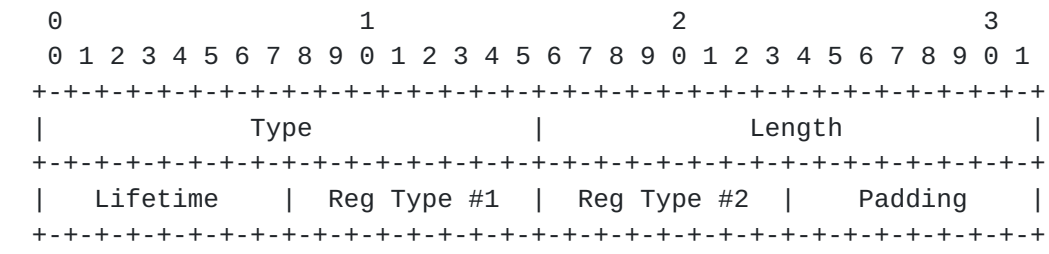
0-200	Reserved by IANA
201-255	Reserved by IANA for private use

A requester includes the REG_REQUEST parameter in I2 or UPDATE packets to register with a registrar's service(s). If the REG_REQUEST parameter is in an UPDATE packet, the registrar MUST NOT modify the registrations of registration types which are not listed in the parameter. Moreover, the requester MUST NOT include the parameter unless the registrar's R1 packet or latest received UPDATE packet has contained a REG_INFO parameter with the requested registration types.

The requester MUST NOT include more than one REG_REQUEST parameter in its I2 or UPDATE packets, while the registrar MUST be able to process one or more REG_REQUEST parameters in received I2 or UPDATE packets.

HIP_SIGNATURE protects the parameter within the I2 and UPDATE packets.

4.4 REG_RESPONSE



Type [TBD by IANA (934)]
Length Length in octets, excluding Type, Length, and Padding.
Lifetime Granted registration lifetime.
Reg Type The granted registration types in order of preference.

Other documents will define specific values for registration types.

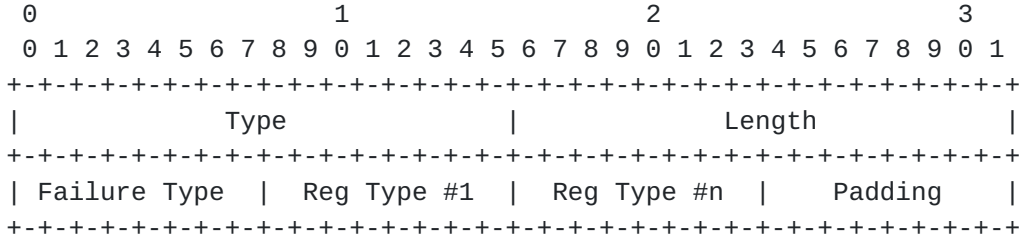
0-200 Reserved by IANA
201-255 Reserved by IANA for private use

The registrar SHOULD includes an REG_RESPONSE parameter in its R2 or UPDATE packet only if a registration has successfully completed.

The registrar MUST NOT include more than one REG_RESPONSE parameter in its R2 or UPDATE packets, while the requester MUST be able to process one or more REG_REQUEST parameters in received R2 or UPDATE packets.

HIP_SIGNATURE protects the parameter within the R2 and UPDATE packets.

4.5 REG_FAILED



Type	[TBD by IANA (936)]
Length	Length in octets, excluding Type, Length, and Padding.
Failure Type	Reason for failure.
Reg Type	The registration types that failed with the specified reason.

Other documents will define specific values for registration types.

0-200	Reserved by IANA
201-255	Reserved by IANA for private use

A failure type of zero means a registrar requires additional credentials to authorize a requester to register with the registration types listed in the parameter. Other failure types than zero have not been defined.

The registrar SHOULD include the REG_FAILED parameter in its R2 or UPDATE packet, if registration with the registration types listed has not completed successfully and a requester is asked to try again with additional credentials.

HIP_SIGNATURE protects the parameter within the R2 and UPDATE packets.

5. Establishing and Maintaining Registrations

Establishing and/or maintaining a registration may require additional information not available in the transmitted REG_REQUEST or REG_RESPONSE parameters. Therefore, registration type definitions MAY define dependencies for HIP parameters that are not defined in this document. Their semantics are subject to the specific registration type specifications.

The minimum lifetime both registrars and requesters MUST support is 10 seconds, while they SHOULD support a maximum lifetime of 120 seconds, at least.

A zero lifetime is reserved for canceling purposes. Requesting a

zero lifetime for a registration type equals to canceling the registration of that type. A requester MAY cancel a registration before it expires by sending a REG_REQ to the registrar with a zero lifetime. A registrar SHOULD respond and grant a registration with a zero lifetime. A registrar (and an attached service) MAY cancel a registration before it expires, at its own discretion. However, if it does so, it SHOULD send a REG_RESPONSE with a zero lifetime to all registered requesters.

6. Security Considerations

This section discusses the threats on the HIP registration protocol, and their implications on the overall security of HIP. In particular, it argues that the extensions described in this document do not introduce additional threats to HIP.

The extensions described in this document rely on the HIP base exchange and do not modify its security characteristics, e.g., digital signatures or HMAC. Hence, the only threat introduced by these extensions are related to the creation of soft registration state at the registrar.

Registrars act on a voluntary basis and are willing to accept to be a responder and to then create HIP associations with a number of previously unknown hosts. Because they have to store HIP association state anyway, adding a certain amount of time-limited HIP registration state should not introduce and serious additional threats, especially because HIP registrars may cancel registrations at any time at their own discretion, e.g., because of resource constraints during an attack.

7. IANA Considerations

This section is to be interpreted according to [RFC2434].

This document updates the IANA Registry for HIP Parameters Types by assigning new HIP Parameter Types values for the new HIP Parameters defined in this document:

- REG_INFO (defined in Section 4.2)
- REG_REQUEST (defined in Section 4.3)
- REG_RESPONSE (defined in Section 4.4)
- REG_FAILED (defined in Section 4.5)

IANA needs to open a new registry for registration types. No types

are defined in this document. Adding a new type requires new IETF specifications.

8. Acknowledgments

The following people have provided thoughtful and helpful discussions and/or suggestions that have improved this document: Pekka Nikander, Hannes Tschofenig and Mika Kousa.

Julien Laganier and Lars Eggert are partly funded by Ambient Networks, a research project supported by the European Commission under its Sixth Framework Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Ambient Networks project or the European Commission.

9. References

9.1 Normative References

[I-D.ietf-hip-arch]
Moskowitz, R., "Host Identity Protocol Architecture", draft-ietf-hip-arch-02 (work in progress), January 2005.

[I-D.ietf-hip-base]
Moskowitz, R., "Host Identity Protocol", draft-ietf-hip-base-03 (work in progress), June 2005.

[I-D.ietf-hip-rvs]
Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", draft-ietf-hip-rvs-02 (work in progress), June 2005.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

9.2 Informative References

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.

Authors' Addresses

Julien Laganier
DoCoMo Communications Laboratories Europe GmbH
Landsberger Strasse 312
Munich 80687
Germany

Phone: +49 89 56824 231
Email: julien.ietf@laposte.net
URI: http://www.docomolab-euro.com/

Teemu Koponen
Helsinki Institute for Information Technology
Advanced Research Unit (ARU)
P.O. Box 9800
Helsinki FIN-02015-HUT
Finland

Phone: +358 9 45 1
Email: teemu.koponen@hiit.fi
URI: http://www.hiit.fi/

Lars Eggert
NEC Network Laboratories
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 90511 43
Fax: +49 6221 90511 55
Email: lars.eggert@netlab.nec.de
URI: http://www.netlab.nec.de/

Appendix A. Document Revision History

Revision	Comments
00	Initial submission.
01	Editorial and boilerplate fixes. Modified terminology. Added security considerations. Changed requirement keyword on new parameters processing.

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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