Host Identity Protocol
Internet-Draft
Intended status: Experimental
Expires: August 31, 2009

T. Heer, Ed.
H. Wirtz
Distributed Systems Group, RWTH
Aachen University
S. Varjonen
Helsinki Institute for
Information Technology
February 27, 2009

Service Identifiers for HIP
draft-heer-hip-service-00

Status of this Memo

Copyright Notice

Abstract

The Host Identity Protocol [RFC5201] is a signaling protocol for secure communication, mobility, and multihoming that introduces a cryptographic namespace. This document specifies an extension for HIP that enables HIP end-hosts and HIP-aware middleboxes to announce services to HIP hosts during a HIP Base EXchange (BEX) or HIP update. Service providers are able to specify the type and requirements of a service; clients can then decide to agree on the terms of service. This allows the service provider to verify the accordance of the client with the service conditions while the client is able to verify the authenticity of the used service.

Requirements Language

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].

1. Introduction

The Host Identity Protocol (HIP) introduces a new cryptographic namespace, based on public keys, in order to secure Internet communication. Several HIP-related documents are concerned with the provision and discovery of services, e.g., the HIP registration extension [RFC5203] and the HIP middlebox authentication extension [I-D.heer-hip-middle-auth]. This document specifies a new HIP parameter that lets service providers communicate properties and requirements of a service to the HIP end-hosts and to on-path HIP-aware network entities. Service providers can either be other HIP end-hosts (Initiator or Responder), on-path network entities (HIP-aware middleboxes and other HIP-aware network infrastructure elements), or entities using the HIP registration extension.

2. Terminology

In addition to the terminology defined in [RFC5203], this document defines the following terms:

Service provider: A HIP end-host or HIP-aware on-path entity (middlebox) that offers a service to a HIP end-host. Middleboxes that offer a service can either use the HIP registration extension [RFC5203] or the HIP middlebox authentication extension [I-D.heer-hip-middle-auth].

Client: A HIP end-host (Initiator or Responder) that is offered a service. The client can choose whether to accept or to deny the offered service.

3. Protocol Overview

The service announcement and service acknowledgement procedure defined in this document is a two-way communication process that integrates into the regular HIP control channel packet exchanges (i.e. the HIP BEX and the HIP update mechanism).

During a base exchange or HIP update mechanism, a service provider (the HIP end-host or a HIP-aware service provider on the communication path) can add a SERVICE_OFFER to an I1, R1, I2, R2, or UPDATE packet. The SERVICE_OFFER provides general information about the service and service-specific information for the client. This information is addressed to the receiver of the HIP control packet. Each HIP packet can contain multiple SERVICE_OFFER parameters from one or more service providers.

The client reads the SERVICE_OFFER parameters from the incoming HIP control packet and based on local policies decides to accept or deny the service offer from the service provider. If it decides to accept the service offer, it responds by creating a SERVICE_ACK parameter which it sends in the signed part of the next regular HIP control packet. If the HIP control packet containing the SERVICE_OFFER does not require an immediate response in the next control packet, the receiver of the SERVICE_OFFER generates an additional HIP UPDATE packet that contains the SERVICE_ACK. If a client declines the service offer, it does not respond with a SERVICE_ACK parameter.

The SERVICE_OFFER parameter comes in two flavors: SERVICE_OFFER and SERVICE_OFFER_UNSIGNED. The SERVICE_OFFER parameter is covered by the signature of the HIP control packet that contains it. Therefore, it can only be added by the HIP end-host that generates the HIP control packet. The SERVICE_OFFER_UNSIGNED is not covered by the signature in the HIP control packet, it is added by HIP-aware middleboxes or HIP end-hosts. Consequently, end-hosts can decide whether to use the signed or unsigned version of the parameter. An example in which an end-host may prefer to use the unsigned parameter is the use of pre-created R1 packets which should include a SERVICE_OFFER that depends on properties of the Initiator (e.g. its HI or IP address).

The service provider can determine whether the client acknowledges the service offer by checking the presence of a SERVICE_ACK parameter with a matching SERVICE_ID in the next packet. The SERVICE_ACK

contains the hash of the service offer, allowing the service provider to verify that the user has accepted the terms of service as added by the service provider in the SERVICE_OFFER. Replying with the hash of the complete SERVICE_OFFER ensures that the client adheres to all conditions of the service offer and that the SERVICE_OFFER_UNSIGNED parameter was delivered without modification in transit. Additionally, the service provider SHOULD verify the validity of the signature in the HIP control packet. In order to shelter against Denial-of-Service (DoS) attacks, end-hosts and middleboxes can utilize the puzzle mechanisms specified in [RFC5201] for end-hosts and [I-D.heer-hip-middle-auth] for middleboxes

3.1. HIP Parameters

3.1.1. SERVICE_OFFER and SERVICE_OFFER_UNSIGNED Parameters

The SERVICE_OFFER and the SERVICE_OFFER_UNSIGNED have identical structures and semantics. The two parameters differ only in their type numbers. Therefore, we discuss only about the contents of the SERVICE_OFFER parameter while the following specifications concerning the SERVICE_OFFER parameter also apply to the SERVICE_OFFER_UNSIGNED parameter.

The SERVICE_OFFER parameter is depicted below. It consists of three parts:

1. SERVICE_PROPERTIES (SP): The SERVICE_PROPERTIES field provides the receiving host with a basic classification of the service based on general parameters. The service properties are an aid for the end-hosts for understanding the nature of an unknown service.

2. SERVICE_ID (SID): The SERVICE_ID is a number that identifies a service or a class of services. The SERVICE_DESCRIPTION is interpreted depending on the SERVICE_ID. The SERVICE_ID MUST be known to all hosts that intend to use that particular service. The SID numbers from 0 to 2^31-1 are assigned by IANA. SID numbers from 2^31 to 2^32-1 are unallocated and may be used by service providers without prior request or notice.

3. SERVICE_DESCRIPTION (SD): The SERVICE_DESCRIPTION field is a variable-length data blob that is interpreted based on the information in the SID field. It MUST be understood by all hosts that intend to use the service. The SD field allows a service to provide specific service-related information. The structure and semantics of the SD field are not part of this document but are specified by the service operators.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             Type              |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                SP (SERVICE_PROPERTIES) (32 bit)              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                SID  (SERVICE_ID) (32 bit)                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          SD (SERVICE_DESCRIPTION (variable length)           /
   /                                           +-+-+-+-+-+-+-+-+-+-|
   /                                           |      Padding      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type 65334

Length Variable

SP Service Properties. A bit field characterizing the service (see below).

SID Unique service ID identifying the service or type of service. 0 (zero) to 2^31-1 assigned by IANA, rest unallocated and in free use.

SD Service Description and service conditions specified by the service provider and interpreted by the client.

SERVICE_PROPERTIES field structure:

```
    0                                       1
    0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
0 |REQ COM FOR TER INI ACI ACR CEI CER <---  RESERVED  --->       |
1 |                    <---  RESERVED  --->                       |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

0 REQ - Required: Non adherence to the requested authentication
                        will result in denial of service.

1 COM - Commercial: Use of this service may result in monetary
                        costs.

2 FOR - Forwarding: This service entails forwarding of packets.

3 TER - Terminal: This HIP-aware middlebox is located at the
                        last hop towards the responder.

4 INI - Initial: This HIP-aware middlebox is located at the
                        first hop towards the responder.

5 ACI - ACL Initiator: The HIT of the Initiator must be in the ACL
                        of the service.

6 ACR - ACL Responder: The HIT of the Responder must be in the ACL
                         of the service.

7 CEI - Cert Initiator: Cert from Initiator required. Cert type
                        defined in variable SD field.

8 CER - Cert Responder: Cert from Responder required. Cert type
                        defined in variable SD field.

Bits 9 to 32 are reserved for future purposes.

3.1.2. SERVICE_ACK

A host that accepts a SERVICE_OFFER or SERVICE_OFFER_UNSIGNED replies with a SERVICE_ACK parameter in its next regular HIP packet.

The service acknowledgement contains the SID as reference to the acknowledged service and the hash of the SERVICE_OFFER parameter. The hash is generated by applying SHA-1 hash function to the SERVICE_OFFER or SERVICE_OFFER_UNSIGNED parameter.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             Type              |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                SID (Service IDentifier)  (32 bit)             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                   SH (Service Hash)  (128 bit)                |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type   65334

Length   160 bit

SID   Unique service ID identifying the service or type of service. 0 (zero) to 2^31-1 assigned by IANA, rest unallocated and in free use.

SH   SHA-1 hash of the accepted SERVICE_OFFER parameter belonging to the SID

4. Applications and Use Cases

4.1. Certificates

Middleboxes or end-hosts may require certificates that state that the host is entitled to perform certain actions (e.g. connect to a host, use a certain link, use a certain service) [I-D.ietf-hip-cert]. The HIP CERT parameter allows HIP hosts to transmit certificate information within HIP control packets. However, a host may possess multiple certificates and therefore it must decide which certificate to transmit.

End-hosts and middleboxes can require the client to present a certificate by adding a SERVICE_OFFER parameter to the next packer addressed to the client. Setting the CEI bit set indicates that a certificate is required and should be sent on the consequent control packet in order to get service. The type of certificate can be transmitted in the SD field.

If the end-host fails in providing sufficient credentials to the service provider it can respond with a NOTIFICATION with

BLOCKED_BY_POLICY if the service provider is an end-host or a NOTIFICATION with BLOCKED_BY_POLICY_M if the service provider is a middlebox to signal the error. BLOCKED_BY_POLICY is defined in [RFC5201] and BLOCKED_BY_POLICY_M is defined below.

NOTIFY MESSAGES - ERROR TYPES	Value
BLOCKED_BY_POLICY	42
The Responder is unwilling to set up an association for policy reasons.	
BLOCKED_BY_POLICY_M	8192
The middlebox is not willing to service the client for policy reasons.	

The policy reason for not serving or setting up an association in this case would be a missing or insufficient certificate.

4.2. Quality of Service

Services may offer a free basic service and a commercial premium service. In such cases, the service provider can add a SERVICE_OFFER for the premium service and default to the basic service if the client does not send a matching SERVICE_ACK. Alternatively, the service provider can add multiple SERVICE_OFFER parameters to a hip control packets, leaving it to the client to acknowledge the appropriate offer.

Further service details (e.g. payment and the quality of the offered services) can be negotiated by using the SERVICE_DETAILS field. By signing the SERVICE_ACK, the end-host agrees to the terms of service. The service provider can use the signed HIP packet containing the SERVICE_ACK as proof that the client has requested the service. It can later use this proof for billing.

Service providers MAY send a NOTIFICATION if the client does not respond with a matching SERVICE_ACK by sending either BLOCKED_BY_POLICY (end-host) or BLOCKED_BY_POLICY_M (middlebox) if they decide to deny the service. See section Section 4.1 for the definition of these parameters.

5. Security Considerations

The question of whether a client must subscribe to a service and the question of whether a service is on the shortest direct path between the Initiator and the Responder is out of scope for this document. However, service operators can design the SERVICE_OFFER parameter in a way that allows semantic sanity checks. For example, a host can detect a suspicios situation if two middleboxes claim to be inital or terminal middleboxes (active INI or TER bits in the SD field of the SERVICE_OFFER parameter). In such cases, end-hosts may require to react based on policies or user interaction.

This document makes no assumptions about the authenticity of the SERVICE_OFFER and SERVICE_OFFER_UNSIGNED parameter. Especially the identity of a service provider is not verified. However, should a service require authentication of a service provider, it can implement this in the variable data field in the SERVICE_OFFER and SERVICE_OFFER_UNSIGNED parameter.

6. IANA Considerations

This draft specifies a new namespace for service identifiers (SID numbers). The SID numbers from 0 to 2^31-1 are to be assigned by IANA. SID numbers from 2^31 to 2^32-1 are unallocated and may be used by service providers without prior request or notice. The SID numbers in the unmanaged SID number space should be selected in a random fashion. There is no guarantee that the SID numbers in the unmanaged SID space are free from collisions. Service providers that use SID numbers from the unallocated SID space should, therefore, take precautions for cases of collisions.

In addition to the SID, a service is described by its SP-flags. To guarantee consistent extensibility of service descriptions, assignment of flags and their positions should also be provided by IANA.

7. Normative References

[I-D.heer-hip-middle-auth]
Heer, T., Wehrle, K., and M. Komu, "End-Host Authentication for HIP Middleboxes", draft-heer-hip-middle-auth-01 (work in progress), July 2008.

[I-D.ietf-hip-cert]
Heer, T. and S. Varjonen, "HIP Certificates",

draft-ietf-hip-cert-00 (work in progress), October 2008.

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

[RFC5201] Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", RFC 5201, April 2008.

[RFC5203] Laganier, J., Koponen, T., and L. Eggert, "Host Identity Protocol (HIP) Registration Extension", RFC 5203, April 2008.

Authors' Addresses

Tobias Heer (editor)
Distributed Systems Group, RWTH Aachen University
Ahornstrasse 55
Aachen 52062
Germany

Phone: +49 241 80 20776
Email: heer@cs.rwth-aachen.de
URI: http://ds.cs.rwth-aachen.de/members/heer

Hanno Wirtz
Distributed Systems Group, RWTH Aachen University
Ahornstrasse 55
Aachen 52062
Germany

Phone: +49 241 80 20773
Email: hanno.wirtz@rwth-aachen.de
URI: http://ds.cs.rwth-aachen.de/members/wirtz

Samu Varjonen
Helsinki Institute for Information Technology
Metsnneidonkuja 4
Helsinki
Finland

Fax: +358 969 49768
Email: samu.varjonen@hiit.fi
URI: http://www.hiit.fi