Network Working Group   R. Moskowitz
Internet-Draft   Verizon
Obsoletes: 4423 (if approved)   February 25, 2011
Intended status: Standards Track
Expires: August 29, 2011

                Host Identity Protocol Architecture
                   draft-ietf-hip-rfc4423-bis-02

Abstract

   This memo describes a new namespace, the Host Identity namespace, and
   a new protocol layer, the Host Identity Protocol, between the
   internetworking and transport layers. Herein are presented the
   basics of the current namespaces, their strengths and weaknesses, and
   how a new namespace will add completeness to them. The roles of this
   new namespace in the protocols are defined.

   This document obsoletes RFC 4423 and addresses the concerns raised by
   the IESG, particularly that of crypto agility. It incorporates
   lessons learned from the implementations of RFC 5201 and goes further
   to explain how HIP works as a secure signalling channel.

Status of this Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on August 29, 2011.

Table of Contents

1. Introduction

The Internet has two important global namespaces: Internet Protocol
(IP) addresses and Domain Name Service (DNS) names. These two
namespaces have a set of features and abstractions that have powered
the Internet to what it is today. They also have a number of
weaknesses. Basically, since they are all we have, we try and do too
much with them. Semantic overloading and functionality extensions
have greatly complicated these namespaces.

The proposed Host Identity namespace fills an important gap between
the IP and DNS namespaces. The Host Identity namespace consists of
Host Identifiers (HI). A Host Identifier is cryptographic in its
nature; it is the public key of an asymmetric key-pair. Each host
will have at least one Host Identity, but it will typically have more
than one. Each Host Identity uniquely identifies a single host,
i.e., no two hosts have the same Host Identity. The Host Identity,
and the corresponding Host Identifier, can either be public (e.g.
published in the DNS), or unpublished. Client systems will tend to
have both public and unpublished Identities.

There is a subtle but important difference between Host Identities
and Host Identifiers. An Identity refers to the abstract entity that
is identified. An Identifier, on the other hand, refers to the
concrete bit pattern that is used in the identification process.

Although the Host Identifiers could be used in many authentication
systems, such as IKEv2 [RFC4306], the presented architecture
introduces a new protocol, called the Host Identity Protocol (HIP),
and a cryptographic exchange, called the HIP base exchange; see also
Section 7. The HIP protocols provide for limited forms of trust
between systems, enhance mobility, multi-homing and dynamic IP
renumbering, aid in protocol translation / transition, and reduce
certain types of denial-of-service (DoS) attacks.

When HIP is used, the actual payload traffic between two HIP hosts is
typically, but not necessarily, protected with IPsec. The Host
Identities are used to create the needed IPsec Security Associations
(SAs) and to authenticate the hosts. When IPsec is used, the actual
payload IP packets do not differ in any way from standard IPsec
protected IP packets.

Much has been learned about HIP since [RFC4423] was published. This
document expands Host Identities beyond use to enable IP connectivity
and security to general interhost secure signalling at any protocol
layer. The signal may establish a security association between the
hosts, or simply pass information within the channel.

2. Terminology

2.1. Terms common to other documents

```
+---------------+----------------------------------------------------+
| Term          | Explanation                                        |
+---------------+----------------------------------------------------+
| Public key    | The public key of an asymmetric cryptographic key  |
|               | pair.  Used as a publicly known identifier for     |
|               | cryptographic identity authentication.             |
|               |                                                    |
| Private key   | The private or secret key of an asymmetric         |
|               | cryptographic key pair.  Assumed to be known only  |
|               | to the party identified by the corresponding       |
|               | public key.  Used by the identified party to       |
|               | authenticate its identity to other parties.        |
|               |                                                    |
| Public key    | An asymmetric cryptographic key pair consisting    |
| pair          | of public and private keys.  For example,          |
|               | Rivest-Shamir-Adelman (RSA) and Digital Signature  |
|               | Algorithm (DSA) key pairs are such key pairs.      |
|               |                                                    |
| End-point     | A communicating entity.  For historical reasons,   |
|               | the term 'computing platform' is used in this      |
|               | document as a (rough) synonym for end-point.       |
+---------------+----------------------------------------------------+
```

2.2. Terms specific to this and other HIP documents

   It should be noted that many of the terms defined herein are
   tautologous, self-referential or defined through circular reference
   to other terms. This is due to the succinct nature of the
   definitions. See the text elsewhere in this document and in RFC 5201
   [RFC5201-bis] for more elaborate explanations.

+---------------+--+
| Term | Explanation |
+---------------+--+
Computing	An entity capable of communicating and computing,
platform	for example, a computer. See the definition of
	'End-point', above.
HIP base	A cryptographic protocol; see also Section 7.
exchange	
HIP packet	An IP packet that carries a 'Host Identity
	Protocol' message.
Host Identity	An abstract concept assigned to a 'computing
	platform'. See 'Host Identifier', below.
Host Identity	A name space formed by all possible Host
namespace	Identifiers.
Host Identity	A protocol used to carry and authenticate Host
Protocol	Identifiers and other information.
Host Identity	A 128-bit datum created by taking a cryptographic
Tag	hash over a Host Identifier.
Host Identity	The cryptograhic hash used in creating the Host
Hash	Identity Tag from the Host Identity.
Host	A public key used as a name for a Host Identity.
Identifier	
Local Scope	A 32-bit datum denoting a Host Identity.
Identifier	
Public Host	A published or publicly known Host Identfier used
Identifier	as a public name for a Host Identity, and the
and Identity	corresponding Identity.
Unpublished	A Host Identifier that is not placed in any
Host	public directory, and the corresponding Host
Identifier	Identity. Unpublished Host Identities are
and Identity	typically short lived in nature, being often
	replaced and possibly used just once.
Rendezvous	A mechanism used to locate mobile hosts based on
Mechanism	their HIT.
+---------------+--+

3. Background

The Internet is built from three principal components: computing
platforms (end-points), packet transport (i.e., internetworking)
infrastructure, and services (applications). The Internet exists to
service two principal components: people and robotic services
(silicon based people, if you will). All these components need to be
named in order to interact in a scalable manner. Here we concentrate
on naming computing platforms and packet transport elements.

There are two principal namespaces in use in the Internet for these
components: IP numbers, and Domain Names. Domain Names provide
hierarchically assigned names for some computing platforms and some
services. Each hierarchy is delegated from the level above; there is
no anonymity in Domain Names. Email, HTTP, and SIP addresses all
reference Domain Names.

IP numbers are a confounding of two namespaces, the names of a host's
networking interfaces and the names of the locations ('confounding'
is a term used in statistics to discuss metrics that are merged into
one with a gain in indexing, but a loss in informational value). The
names of locations should be understood as denoting routing direction
vectors, i.e., information that is used to deliver packets to their
destinations.

IP numbers name networking interfaces, and typically only when the
interface is connected to the network. Originally, IP numbers had
long-term significance. Today, the vast number of interfaces use
ephemeral and/or non-unique IP numbers. That is, every time an
interface is connected to the network, it is assigned an IP number.

In the current Internet, the transport layers are coupled to the IP
addresses. Neither can evolve separately from the other. IPng
deliberations were strongly shaped by the decision that a
corresponding TCPng would not be created.

There are three critical deficiencies with the current namespaces.
Firstly, dynamic readdressing cannot be directly managed. Secondly,
anonymity is not provided in a consistent, trustable manner.
Finally, authentication for systems and datagrams is not provided.
All of these deficiencies arise because computing platforms are not
well named with the current namespaces.

3.1. A desire for a namespace for computing platforms

An independent namespace for computing platforms could be used in
end-to-end operations independent of the evolution of the
internetworking layer and across the many internetworking layers.

This could support rapid readdressing of the internetworking layer
because of mobility, rehoming, or renumbering.

If the namespace for computing platforms is based on public-key
cryptography, it can also provide authentication services. If this
namespace is locally created without requiring registration, it can
provide anonymity.

Such a namespace (for computing platforms) and the names in it should
have the following characteristics:

o The namespace should be applied to the IP 'kernel'. The IP kernel
   is the 'component' between applications and the packet transport
   infrastructure.

o The namespace should fully decouple the internetworking layer from
   the higher layers. The names should replace all occurrences of IP
   addresses within applications (like in the Transport Control
   Block, TCB). This may require changes to the current APIs. In
   the long run, it is probable that some new APIs are needed.

o The introduction of the namespace should not mandate any
   administrative infrastructure. Deployment must come from the
   bottom up, in a pairwise deployment.

o The names should have a fixed length representation, for easy
   inclusion in datagram headers and existing programming interfaces
   (e.g the TCB).

o Using the namespace should be affordable when used in protocols.
   This is primarily a packet size issue. There is also a
   computational concern in affordability.

o Name collisions should be avoided as much as possible. The
   mathematics of the birthday paradox can be used to estimate the
   chance of a collision in a given population and hash space. In
   general, for a random hash space of size n bits, we would expect
   to obtain a collision after approximately $1.2*sqrt(2**n)$ hashes
   were obtained. For 64 bits, this number is roughly 4 billion. A
   hash size of 64 bits may be too small to avoid collisions in a
   large population; for example, there is a 1% chance of collision
   in a population of 640M. For 100 bits (or more), we would not
   expect a collision until approximately $2**50$ (1 quadrillion)
   hashes were generated.

o The names should have a localized abstraction so that it can be
   used in existing protocols and APIs.

o It must be possible to create names locally. This can provide
   anonymity at the cost of making resolvability very difficult.

   * Sometimes the names may contain a delegation component. This
      is the cost of resolvability.

o The namespace should provide authentication services.

o The names should be long lived, but replaceable at any time. This
   impacts access control lists; short lifetimes will tend to result
   in tedious list maintenance or require a namespace infrastructure
   for central control of access lists.

In this document, a new namespace approaching these ideas is called
the Host Identity namespace. Using Host Identities requires its own
protocol layer, the Host Identity Protocol, between the
internetworking and transport layers. The names are based on public-
key cryptography to supply authentication services. Properly
designed, it can deliver all of the above stated requirements.

4. **Host Identity namespace**

A name in the Host Identity namespace, a Host Identifier (HI),
represents a statistically globally unique name for naming any system
with an IP stack. This identity is normally associated with, but not
limited to, an IP stack. A system can have multiple identities, some
'well known', some unpublished or 'anonymous'. A system may self-
assert its own identity, or may use a third-party authenticator like
DNSSEC [RFC2535], PGP, or X.509 to 'notarize' the identity assertion.
It is expected that the Host Identifiers will initially be
authenticated with DNSSEC and that all implementations will support
DNSSEC as a minimal baseline.

In theory, any name that can claim to be 'statistically globally
unique' may serve as a Host Identifier. However, in the authors'
opinion, a public key of a 'public key pair' makes the best Host
Identifier. As will be specified in the Host Identity Protocol Base
EXchange (BEX) [RFC5201-bis] specification, a public-key-based HI can
authenticate the HIP packets and protect them for man-in-the-middle
attacks. Since authenticated datagrams are mandatory to provide much
of HIP's denial-of-service protection, the Diffie-Hellman exchange in
HIP BEX has to be authenticated. Thus, only public-key HI and
authenticated HIP messages are supported in practice.

In this document, the non-cryptographic forms of HI and HIP are
presented to complete the theory of HI, but they should not be
implemented as they could produce worse denial-of-service attacks

than the Internet has without Host Identity. There is on-going
research in challenge puzzles to use non-cryptographic HI, like
RFIDs, in an HIP exchange tailored to the workings of such
challenges.

4.1. Host Identifiers

Host Identity adds two main features to Internet protocols. The
first is a decoupling of the internetworking and transport layers;
see Section 5. This decoupling will allow for independent evolution
of the two layers. Additionally, it can provide end-to-end services
over multiple internetworking realms. The second feature is host
authentication. Because the Host Identifier is a public key, this
key can be used for authentication in security protocols like IPsec.

The only completely defined structure of the Host Identity is that of
a public/private key pair. In this case, the Host Identity is
referred to by its public component, the public key. Thus, the name
representing a Host Identity in the Host Identity namespace, i.e.,
the Host Identifier, is the public key. In a way, the possession of
the private key defines the Identity itself. If the private key is
possessed by more than one node, the Identity can be considered to be
a distributed one.

Architecturally, any other Internet naming convention might form a
usable base for Host Identifiers. However, non-cryptographic names
should only be used in situations of high trust - low risk. That is
any place where host authentication is not needed (no risk of host
spoofing) and no use of IPsec. However, at least for interconnected
networks spanning several operational domains, the set of
environments where the risk of host spoofing allowed by non-
cryptographic Host Identifiers is acceptable is the null set. Hence,
the current HIP documents do not specify how to use any other types
of Host Identifiers but public keys.

The actual Host Identities are never directly used in any Internet
protocols. The corresponding Host Identifiers (public keys) may be
stored in various DNS or LDAP directories as identified elsewhere in
this document, and they are passed in the HIP base exchange. A Host
Identity Tag (HIT) is used in other protocols to represent the Host
Identity. Another representation of the Host Identities, the Local
Scope Identifier (LSI), can also be used in protocols and APIs.

4.2. Storing Host Identifiers in DNS

The public Host Identifiers should be stored in DNS; the unpublished
Host Identifiers should not be stored anywhere (besides the
communicating hosts themselves). The (public) HI along with the

supported HIHs are stored in a new RR type. This RR type is defined in HIP DNS Extension [I-D.ietf-hip-rfc5205-bis].

Alternatively, or in addition to storing Host Identifiers in the DNS, they may be stored in various kinds of Public Key Infrastructure (PKI). Such a practice may allow them to be used for purposes other than pure host identification.

4.3. Host Identity Tag (HIT)

A Host Identity Tag is a 128-bit representation for a Host Identity. It is created by taking a cryptographic hash over the corresponding Host Identifier. There are two advantages of using a hash over using the Host Identifier in protocols. Firstly, its fixed length makes for easier protocol coding and also better manages the packet size cost of this technology. Secondly, it presents the identity in a consistent format to the protocol independent of the cryptographic algorithms used.

There can be multiple HITs per Host Identifier when multiple hashes are supported. An Initator may have to initially guess which HIT to use for the Responder, typically based on what it perfers, until it learns the appropriate HIT through the HIP exchange.

In the HIP packets, the HITs identify the sender and recipient of a packet. Consequently, a HIT should be unique in the whole IP universe as long as it is being used. In the extremely rare case of a single HIT mapping to more than one Host Identity, the Host Identifiers (public keys) will make the final difference. If there is more than one public key for a given node, the HIT acts as a hint for the correct public key to use.

4.4. Host Identity Hash (HIH)

The Host Identity Hash is the cryptographic hash used in producing the HIT from the HI. It is also the hash used through out the HIP protocol for consistancy and simplicity. It is possible to for the two Hosts in the HIP exchange to use different hashes.

Multiple HIHs within HIP are needed to address the moving target of creation and eventual compromise of cryptographic hashes. This significantly complicates HIP and offers an attacker an additional downgrade attack that is mitigated in the HIP protocol.

4.5. Local Scope Identifier (LSI)

An LSI is a 32-bit localized representation for a Host Identity. The purpose of an LSI is to facilitate using Host Identities in existing

protocols and APIs. LSI's advantage over HIT is its size; its
disadvantage is its local scope.

Examples of how LSIs can be used include: as the address in an FTP
command and as the address in a socket call. Thus, LSIs act as a
bridge for Host Identities into IPv4-based protocols and APIs. LSIs
also make it possible for some IPv4 applications to run over an IPv6
network.

5. New stack architecture

One way to characterize Host Identity is to compare the proposed new
architecture with the current one. As discussed above, the IP
addresses can be seen to be a confounding of routing direction
vectors and interface names. Using the terminology from the IRTF
Name Space Research Group Report [nsrg-report] and, e.g., the
unpublished Internet-Draft Endpoints and Endpoint Names
[chiappa-endpoints], the IP addresses currently embody the dual role
of locators and end-point identifiers. That is, each IP address
names a topological location in the Internet, thereby acting as a
routing direction vector, or locator. At the same time, the IP
address names the physical network interface currently located at the
point-of-attachment, thereby acting as a end-point name.

In the HIP architecture, the end-point names and locators are
separated from each other. IP addresses continue to act as locators.
The Host Identifiers take the role of end-point identifiers. It is
important to understand that the end-point names based on Host
Identities are slightly different from interface names; a Host
Identity can be simultaneously reachable through several interfaces.

The difference between the bindings of the logical entities are
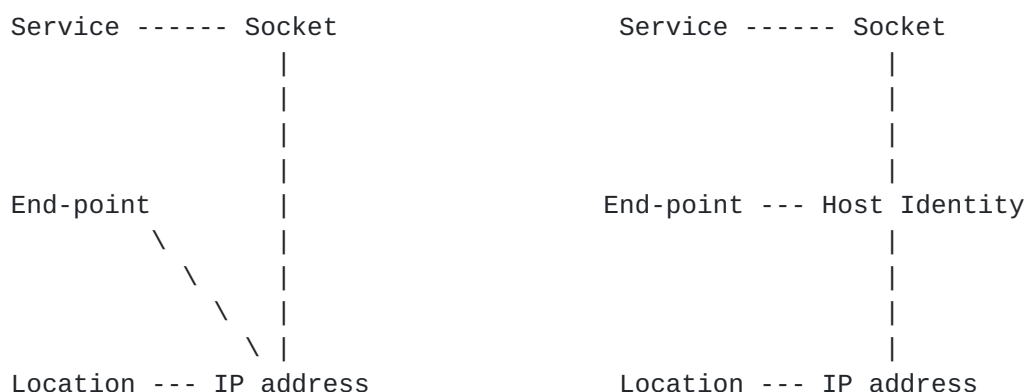illustrated in Figure 1.

```
   Service ------ Socket                Service ------ Socket
                  |                                    |
                  |                                    |
                  |                                    |
                  |                                    |
   End-point      |                     End-point --- Host Identity
           \      |                                    |
            \     |                                    |
             \    |                                    |
              \   |                                    |
   Location --- IP address              Location --- IP address
```

Figure 1

5.1. Transport associations and end-points

   Architecturally, HIP provides for a different binding of transport-
   layer protocols. That is, the transport-layer associations, i.e.,
   TCP connections and UDP associations, are no longer bound to IP
   addresses but to Host Identities.

   It is possible that a single physical computer hosts several logical
   end-points. With HIP, each of these end-points would have a distinct
   Host Identity. Furthermore, since the transport associations are
   bound to Host Identities, HIP provides for process migration and
   clustered servers. That is, if a Host Identity is moved from one
   physical computer to another, it is also possible to simultaneously
   move all the transport associations without breaking them.
   Similarly, if it is possible to distribute the processing of a single
   Host Identity over several physical computers, HIP provides for
   cluster based services without any changes at the client end-point.

6. End-host mobility and multi-homing

   HIP decouples the transport from the internetworking layer, and binds
   the transport associations to the Host Identities (through actually
   either the HIT or LSI). Consequently, HIP can provide for a degree
   of internetworking mobility and multi-homing at a low infrastructure
   cost. HIP mobility includes IP address changes (via any method) to
   either party. Thus, a system is considered mobile if its IP address
   can change dynamically for any reason like PPP, DHCP, IPv6 prefix
   reassignments, or a NAT device remapping its translation. Likewise,
   a system is considered multi-homed if it has more than one globally
   routable IP address at the same time. HIP links IP addresses
   together, when multiple IP addresses correspond to the same Host
   Identity, and if one address becomes unusable, or a more preferred
   address becomes available, existing transport associations can easily
   be moved to another address.

   When a node moves while communication is already on-going, address
   changes are rather straightforward. The peer of the mobile node can
   just accept a HIP or an integrity protected IPsec packet from any
   address and ignore the source address. However, as discussed in
   Section 6.2 below, a mobile node must send a HIP readdress packet to
   inform the peer of the new address(es), and the peer must verify that
   the mobile node is reachable through these addresses. This is
   especially helpful for those situations where the peer node is
   sending data periodically to the mobile node (that is re-starting a
   connection after the initial connection).

6.1. Rendezvous mechanism

   Making a contact to a mobile node is slightly more involved. In
   order to start the HIP exchange, the initiator node has to know how
   to reach the mobile node. Although infrequently moving HIP nodes
   could use Dynamic DNS [RFC2136] to update their reachability
   information in the DNS, an alternative to using DNS in this fashion
   is to use a piece of new static infrastructure to facilitate
   rendezvous between HIP nodes.

   The mobile node keeps the rendezvous infrastructure continuously
   updated with its current IP address(es). The mobile nodes must trust
   the rendezvous mechanism to properly maintain their HIT and IP
   address mappings.

   The rendezvous mechanism is also needed if both of the nodes happen
   to change their address at the same time, either because they are
   mobile and happen to move at the same time, because one of them is
   off-line for a while, or because of some other reason. In such a
   case, the HIP UPDATE packets will cross each other in the network and
   never reach the peer node.

   The HIP rendezvous mechanism is defined in HIP Rendezvous
   [I-D.ietf-hip-rfc5204-bis].

6.2. Protection against flooding attacks

   Although the idea of informing about address changes by simply
   sending packets with a new source address appears appealing, it is
   not secure enough. That is, even if HIP does not rely on the source
   address for anything (once the base exchange has been completed), it
   appears to be necessary to check a mobile node's reachability at the
   new address before actually sending any larger amounts of traffic to
   the new address.

   Blindly accepting new addresses would potentially lead to flooding
   Denial-of-Service attacks against third parties [RFC4225]. In a
   distributed flooding attack an attacker opens high volume HIP
   connections with a large number of hosts (using unpublished HIs), and
   then claims to all of these hosts that it has moved to a target
   node's IP address. If the peer hosts were to simply accept the move,
   the result would be a packet flood to the target node's address. To
   prevent this type of attack, HIP includes an address check mechanism
   where the reachability of a node is separately checked at each
   address before using the address for larger amounts of traffic.

   A credit-based authorization approach Host Mobility with the Host
   Identity Protocol [I-D.ietf-hip-rfc5206-bis] can be used between

hosts for sending data prior to completing the address tests.
Otherwise, if HIP is used between two hosts that fully trust each
other, the hosts may optionally decide to skip the address tests.
However, such performance optimization must be restricted to peers
that are known to be trustworthy and capable of protecting themselves
from malicious software.

7. **HIP and IPsec**

   The preferred way of implementing HIP is to use IPsec to carry the
   actual data traffic. As of today, the only completely defined method
   is to use IPsec Encapsulated Security Payload (ESP) to carry the data
   packets [I-D.ietf-hip-rfc5202-bis]. In the future, other ways of
   transporting payload data may be developed, including ones that do
   not use cryptographic protection.

   In practice, the HIP base exchange uses the cryptographic Host
   Identifiers to set up a pair of ESP Security Associations (SAs) to
   enable ESP in an end-to-end manner. This is implemented in a way
   that can span addressing realms.

   While it would be possible, at least in theory, to use some existing
   cryptographic protocol, such as IKEv2 together with Host Identifiers,
   to establish the needed SAs, HIP defines a new protocol. There are a
   number of historical reasons for this, and there are also a few
   architectural reasons. First, IKE (and IKEv2) were not designed with
   middle boxes in mind. As adding a new naming layer allows one to
   potentially add a new forwarding layer (see Section 9, below), it is
   very important that the HIP protocols are friendly towards any middle
   boxes.

   Second, from a conceptual point of view, the IPsec Security Parameter
   Index (SPI) in ESP provides a simple compression of the HITs. This
   does require per-HIT-pair SAs (and SPIs), and a decrease of policy
   granularity over other Key Management Protocols, such as IKE and
   IKEv2. In particular, the current thinking is limited to a situation
   where, conceptually, there is only one pair of SAs between any given
   pair of HITs. In other words, from an architectural point of view,
   HIP only supports host-to-host (or endpoint-to-endpoint) Security
   Associations. If two hosts need more pairs of parallel SAs, they
   should use separate HITs for that. However, future HIP extensions
   may provide for more granularity and creation of several ESP SAs
   between a pair of HITs.

   Since HIP is designed for host usage, not for gateways or so called
   Bump-in-the-Wire (BITW) implementations, only ESP transport mode is
   supported. An ESP SA pair is indexed by the SPIs and the two HITs

(both HITs since a system can have more than one HIT). The SAs need
not to be bound to IP addresses; all internal control of the SA is by
the HITs. Thus, a host can easily change its address using Mobile
IP, DHCP, PPP, or IPv6 readdressing and still maintain the SAs.
Since the transports are bound to the SA (via an LSI or a HIT), any
active transport is also maintained. Thus, real-world conditions
like loss of a PPP connection and its re-establishment or a mobile
handover will not require a HIP negotiation or disruption of
transport services [Bel1998].

Since HIP does not negotiate any SA lifetimes, all lifetimes are
local policy. The only lifetimes a HIP implementation must support
are sequence number rollover (for replay protection), and SA timeout.
An SA times out if no packets are received using that SA.
Implementations may support lifetimes for the various ESP transforms.

8. HIP and MAC Security

The IEEE 802 standards have been defining MAC layered security. Many
of these standards use EAP [RFC3748] as a Key Management System (KMS)
transport, but some like IEEE 802.15.4 [IEEE.802-15-4.2006] leave the
KMS and its transport as "Out of Scope".

HIP is well suited as a KMS in these environments.

o HIP is independent of IP addressing and can be directly
   transported over any network protocol.

o Master Keys in 802 protocols are strictly pair-based with group
   keys transported from the group controller using pair-wise keys.

o AdHoc 802 networks can be better served by a peer-to-peer KMS than
   the EAP client/server model.

o Some devices are very memory constrained and a common KMS for both
   MAC and IP security represents a considerable code savings.

9. HIP and NATs

Passing packets between different IP addressing realms requires
changing IP addresses in the packet header. This may happen, for
example, when a packet is passed between the public Internet and a
private address space, or between IPv4 and IPv6 networks. The
address translation is usually implemented as Network Address
Translation (NAT) [RFC3022] or NAT Protocol translation (NAT-PT)
[RFC2766].

In a network environment where identification is based on the IP
addresses, identifying the communicating nodes is difficult when NAT
is used. With HIP, the transport-layer end-points are bound to the
Host Identities. Thus, a connection between two hosts can traverse
many addressing realm boundaries. The IP addresses are used only for
routing purposes; they may be changed freely during packet traversal.

For a HIP-based flow, a HIP-aware NAT or NAT-PT system tracks the
mapping of HITs, and the corresponding IPsec SPIs, to an IP address.
The NAT system has to learn mappings both from HITs and from SPIs to
IP addresses. Many HITs (and SPIs) can map to a single IP address on
a NAT, simplifying connections on address poor NAT interfaces. The
NAT can gain much of its knowledge from the HIP packets themselves;
however, some NAT configuration may be necessary.

NAT systems cannot touch the datagrams within the IPsec envelope,
thus application-specific address translation must be done in the end
systems. HIP provides for 'Distributed NAT', and uses the HIT or the
LSI as a placeholder for embedded IP addresses.

HIP and NAT interaction is defined in [RFC5770].

9.1. HIP and Upper-layer checksums

There is no way for a host to know if any of the IP addresses in an
IP header are the addresses used to calculate the TCP checksum. That
is, it is not feasible to calculate the TCP checksum using the actual
IP addresses in the pseudo header; the addresses received in the
incoming packet are not necessarily the same as they were on the
sending host. Furthermore, it is not possible to recompute the
upper-layer checksums in the NAT/NAT-PT system, since the traffic is
IPsec protected. Consequently, the TCP and UDP checksums are
calculated using the HITs in the place of the IP addresses in the
pseudo header. Furthermore, only the IPv6 pseudo header format is
used. This provides for IPv4 / IPv6 protocol translation.

10. Multicast

Few concrete thoughts exist about how HIP might affect IP-layer or
application-layer multicast.

11. HIP policies

There are a number of variables that will influence the HIP exchanges
that each host must support. All HIP implementations should support
at least 2 HIs, one to publish in DNS and an unpublished one for

anonymous usage. Although unpublished HIs will be rarely used as
responder HIs, they are likely be common for initiators. Support for
multiple HIs is recommended.

Many initiators would want to use a different HI for different
responders. The implementations should provide for a policy of
initiator HIT to responder HIT. This policy should also include
preferred transforms and local lifetimes.

Responders would need a similar policy, describing the hosts allowed
to participate in HIP exchanges, and the preferred transforms and
local lifetimes.

12. Benefits of HIP

In the beginning, the network layer protocol (i.e., IP) had the
following four "classic" invariants:

o Non-mutable: The address sent is the address received.

o Non-mobile: The address doesn't change during the course of an
   "association".

o Reversible: A return header can always be formed by reversing the
   source and destination addresses.

o Omniscient: Each host knows what address a partner host can use to
   send packets to it.

Actually, the fourth can be inferred from 1 and 3, but it is worth
mentioning for reasons that will be obvious soon if not already.

In the current "post-classic" world, we are intentionally trying to
get rid of the second invariant (both for mobility and for multi-
homing), and we have been forced to give up the first and the fourth.
Realm Specific IP [RFC3102] is an attempt to reinstate the fourth
invariant without the first invariant. IPv6 is an attempt to
reinstate the first invariant.

Few systems on the Internet have DNS names that are meaningful. That
is, if they have a Fully Qualified Domain Name (FQDN), that name
typically belongs to a NAT device or a dial-up server, and does not
really identify the system itself but its current connectivity.
FQDNs (and their extensions as email names) are application-layer
names; more frequently naming services than a particular system.
This is why many systems on the Internet are not registered in the
DNS; they do not have services of interest to other Internet hosts.

DNS names are references to IP addresses. This only demonstrates the
interrelationship of the networking and application layers. DNS, as
the Internet's only deployed, distributed database is also the
repository of other namespaces, due in part to DNSSEC and application
specific key records. Although each namespace can be stretched (IP
with v6, DNS with KEY records), neither can adequately provide for
host authentication or act as a separation between internetworking
and transport layers.

The Host Identity (HI) namespace fills an important gap between the
IP and DNS namespaces. An interesting thing about the HI is that it
actually allows one to give up all but the 3rd network-layer
invariant. That is to say, as long as the source and destination
addresses in the network-layer protocol are reversible, then things
work ok because HIP takes care of host identification, and
reversibility allows one to get a packet back to one's partner host.
You do not care if the network-layer address changes in transit
(mutable) and you don't care what network-layer address the partner
is using (non-omniscient).

12.1. HIP's answers to NSRG questions

The IRTF Name Space Research Group has posed a number of evaluating
questions in their report [nsrg-report]. In this section, we provide
answers to these questions.

1. How would a stack name improve the overall functionality of the
    Internet?

       HIP decouples the internetworking layer from the transport
       layer, allowing each to evolve separately. The decoupling
       makes end-host mobility and multi-homing easier, also across
       IPv4 and IPv6 networks. HIs make network renumbering easier,
       and they also make process migration and clustered servers
       easier to implement. Furthermore, being cryptographic in
       nature, they provide the basis for solving the security
       problems related to end-host mobility and multi-homing.

2. What does a stack name look like?

       A HI is a cryptographic public key. However, instead of using
       the keys directly, most protocols use a fixed size hash of the
       public key.

3. What is its lifetime?

       HIP provides both stable and temporary Host Identifiers.
       Stable HIs are typically long lived, with a lifetime of years

or more. The lifetime of temporary HIs depends on how long
the upper-layer connections and applications need them, and
can range from a few seconds to years.

4. Where does it live in the stack?

The HIs live between the transport and internetworking layers.

5. How is it used on the end points?

The Host Identifiers may be used directly or indirectly (in
the form of HITs or LSIs) by applications when they access
network services. Additionally, the Host Identifiers, as
public keys, are used in the built in key agreement protocol,
called the HIP base exchange, to authenticate the hosts to
each other.

6. What administrative infrastructure is needed to support it?

In some environments, it is possible to use HIP
opportunistically, without any infrastructure. However, to
gain full benefit from HIP, the HIs must be stored in the DNS
or a PKI, and a new rendezvous mechanism is needed
[I-D.ietf-hip-rfc5205-bis].

7. If we add an additional layer would it make the address list in
SCTP unnecessary?

Yes

8. What additional security benefits would a new naming scheme
offer?

HIP reduces dependency on IP addresses, making the so called
address ownership [Nik2001] problems easier to solve. In
practice, HIP provides security for end-host mobility and
multi-homing. Furthermore, since HIP Host Identifiers are
public keys, standard public key certificate infrastructures
can be applied on the top of HIP.

9. What would the resolution mechanisms be, or what characteristics
of a resolution mechanisms would be required?

For most purposes, an approach where DNS names are resolved
simultaneously to HIs and IP addresses is sufficient.
However, if it becomes necessary to resolve HIs into IP
addresses or back to DNS names, a flat resolution
infrastructure is needed. Such an infrastructure could be

based on the ideas of Distributed Hash Tables, but would
require significant new development and deployment.

13. Changes from RFC 4423

This section summarizes the changes made from [RFC4423].

14. Security considerations

HIP takes advantage of the new Host Identity paradigm to provide
secure authentication of hosts and to provide a fast key exchange for
IPsec. HIP also attempts to limit the exposure of the host to
various denial-of-service (DoS) and man-in-the-middle (MitM) attacks.
In so doing, HIP itself is subject to its own DoS and MitM attacks
that potentially could be more damaging to a host's ability to
conduct business as usual.

Resource exhausting denial-of-service attacks take advantage of the
cost of setting up a state for a protocol on the responder compared
to the 'cheapness' on the initiator. HIP allows a responder to
increase the cost of the start of state on the initiator and makes an
effort to reduce the cost to the responder. This is done by having
the responder start the authenticated Diffie-Hellman exchange instead
of the initiator, making the HIP base exchange 4 packets long. There
are more details on this process in the Host Identity Protocol under
development.

HIP optionally supports opportunistic negotiation. That is, if a
host receives a start of transport without a HIP negotiation, it can
attempt to force a HIP exchange before accepting the connection.
This has the potential for DoS attacks against both hosts. If the
method to force the start of HIP is expensive on either host, the
attacker need only spoof a TCP SYN. This would put both systems into
the expensive operations. HIP avoids this attack by having the
responder send a simple HIP packet that it can pre-build. Since this
packet is fixed and easily replayed, the initiator only reacts to it
if it has just started a connection to the responder.

Man-in-the-middle attacks are difficult to defend against, without
third-party authentication. A skillful MitM could easily handle all
parts of the HIP base exchange, but HIP indirectly provides the
following protection from a MitM attack. If the responder's HI is
retrieved from a signed DNS zone or secured by some other means, the
initiator can use this to authenticate the signed HIP packets.
Likewise, if the initiator's HI is in a secure DNS zone, the
responder can retrieve it and validate the signed HIP packets.

However, since an initiator may choose to use an unpublished HI, it
knowingly risks a MitM attack. The responder may choose not to
accept a HIP exchange with an initiator using an unknown HI.

The need to support multiple hashes for generating the HIT from the
HI affords the MitM a potentially powerful downgrade attack due to
the a-priori need of the HIT in the HIP base exchange. The base
exchange has been augmented to deal with such an attack by restarting
on detecting the attack. At worst this would only lead to a
situation in which the base exchange would never finish (or would be
aborted after some retries). As a drawback, this leads to an 6-way
base exchange which may seem bad at first. However, since this only
happens in an attack scenario and since the attack can be handled (so
it is not interesting to mount anymore), we assume the additional
messages are not a problem at all. Since the MitM cannot be
successful with a downgrade attack, these sorts of attacks will only
occur as 'nuisance' attacks. So, the base exchange would still be
usually just four packets even though implementations must be
prepared to protect themselves against the downgrade attack.

In HIP, the Security Association for IPsec is indexed by the SPI; the
source address is always ignored, and the destination address may be
ignored as well. Therefore, HIP-enabled IPsec Encapsulated Security
Payload (ESP) is IP address independent. This might seem to make it
easier for an attacker, but ESP with replay protection is already as
well protected as possible, and the removal of the IP address as a
check should not increase the exposure of IPsec ESP to DoS attacks.

Since not all hosts will ever support HIP, ICMPv4 'Destination
Unreachable, Protocol Unreachable' and ICMPv6 'Parameter Problem,
Unrecognized Next Header' messages are to be expected and present a
DoS attack. Against an initiator, the attack would look like the
responder does not support HIP, but shortly after receiving the ICMP
message, the initiator would receive a valid HIP packet. Thus, to
protect against this attack, an initiator should not react to an ICMP
message until a reasonable time has passed, allowing it to get the
real responder's HIP packet. A similar attack against the responder
is more involved.

Another MitM attack is simulating a responder's administrative
rejection of a HIP initiation. This is a simple ICMP 'Destination
Unreachable, Administratively Prohibited' message. A HIP packet is
not used because it would either have to have unique content, and
thus difficult to generate, resulting in yet another DoS attack, or
just as spoofable as the ICMP message. Like in the previous case,
the defense against this attack is for the initiator to wait a
reasonable time period to get a valid HIP packet. If one does not
come, then the initiator has to assume that the ICMP message is

valid. Since this is the only point in the HIP base exchange where
this ICMP message is appropriate, it can be ignored at any other
point in the exchange.

14.1. HITs used in ACLs

It is expected that HITs will be used in ACLs. Future firewalls can
use HITs to control egress and ingress to networks, with an assurance
level difficult to achieve today. As discussed above in Section 7,
once a HIP session has been established, the SPI value in an IPsec
packet may be used as an index, indicating the HITs. In practice,
firewalls can inspect HIP packets to learn of the bindings between
HITs, SPI values, and IP addresses. They can even explicitly control
IPsec usage, dynamically opening IPsec ESP only for specific SPI
values and IP addresses. The signatures in HIP packets allow a
capable firewall to ensure that the HIP exchange is indeed happening
between two known hosts. This may increase firewall security.

There has been considerable bad experience with distributed ACLs that
contain public key related material, for example, with SSH. If the
owner of a key needs to revoke it for any reason, the task of finding
all locations where the key is held in an ACL may be impossible. If
the reason for the revocation is due to private key theft, this could
be a serious issue.

A host can keep track of all of its partners that might use its HIT
in an ACL by logging all remote HITs. It should only be necessary to
log responder hosts. With this information, the host can notify the
various hosts about the change to the HIT. There has been no attempt
to develop a secure method to issue the HIT revocation notice.

HIP-aware NATs, however, are transparent to the HIP aware systems by
design. Thus, the host may find it difficult to notify any NAT that
is using a HIT in an ACL. Since most systems will know of the NATs
for their network, there should be a process by which they can notify
these NATs of the change of the HIT. This is mandatory for systems
that function as responders behind a NAT. In a similar vein, if a
host is notified of a change in a HIT of an initiator, it should
notify its NAT of the change. In this manner, NATs will get updated
with the HIT change.

14.2. Non-security considerations

The definition of the Host Identifier states that the HI need not be
a public key. It implies that the HI could be any value; for example
a FQDN. This document does not describe how to support such a non-
cryptographic HI. A non-cryptographic HI would still offer the
services of the HIT or LSI for NAT traversal. It would be possible

to carry HITs in HIP packets that had neither privacy nor
authentication. Since such a mode would offer so little additional
functionality for so much addition to the IP kernel, it has not been
defined. Given how little public key cryptography HIP requires, HIP
should only be implemented using public key Host Identities.

If it is desirable to use HIP in a low security situation where
public key computations are considered expensive, HIP can be used
with very short Diffie-Hellman and Host Identity keys. Such use
makes the participating hosts vulnerable to MitM and connection
hijacking attacks. However, it does not cause flooding dangers,
since the address check mechanism relies on the routing system and
not on cryptographic strength.

15. IANA considerations

   This document has no actions for IANA.

16. Acknowledgments

   For the people historically involved in the early stages of HIP, see
   the Acknowledgements section in the Host Identity Protocol
   specification.

   During the later stages of this document, when the editing baton was
   transfered to Pekka Nikander, the comments from the early
   implementors and others, including Jari Arkko, Tom Henderson, Petri
   Jokela, Miika Komu, Mika Kousa, Andrew McGregor, Jan Melen, Tim
   Shepard, Jukka Ylitalo, and Jorma Wall, were invaluable. Finally,
   Lars Eggert, Spencer Dawkins and Dave Crocker provided valuable input
   during the final stages of publication, most of which was
   incorporated but some of which the authors decided to ignore in order
   to get this document published in the first place.

   The authors want to express their special thanks to Tom Henderson,
   who took the burden of editing the document in response to IESG
   comments at the time when both of the authors were busy doing other
   things. Without his perseverance original document might have never
   made it as RFC4423.

   This latest effort to update and move HIP forward within the IETF
   process owes its impetuous to the three HIP development teams:
   Boeing, HIIT (Helsinki Institute for Information Technology), and
   NomadicLab of Ericsson. Without their collective efforts HIP would
   have withered as on the IETF vine as a nice concept.

17. References

17.1. Normative References

[RFC5201-bis]
          Moskowitz, R., Jokela, P., Henderson, T., and T. Heer,
          "Host Identity Protocol", draft-ietf-hip-rfc5201-bis-04
          (work in progress), January 2011.

[I-D.ietf-hip-rfc5202-bis]
          Jokela, P., Moskowitz, R., Nikander, P., and J. Melen,
          "Using the Encapsulating Security Payload (ESP) Transport
          Format with the Host Identity Protocol (HIP)",
          draft-ietf-hip-rfc5202-bis-00 (work in progress),
          September 2010.

[I-D.ietf-hip-rfc5204-bis]
          Laganier, J. and L. Eggert, "Host Identity Protocol (HIP)
          Rendezvous Extension", draft-ietf-hip-rfc5204-bis-00 (work
          in progress), August 2010.

[I-D.ietf-hip-rfc5205-bis]
          Laganier, J., "Host Identity Protocol (HIP) Domain Name
          System (DNS) Extension", draft-ietf-hip-rfc5205-bis-00
          (work in progress), August 2010.

[I-D.ietf-hip-rfc5206-bis]
          Nikander, P., Henderson, T., Vogt, C., and J. Arkko, "Host
          Mobility with the Host Identity Protocol",
          draft-ietf-hip-rfc5206-bis-01 (work in progress),
          October 2010.

17.2. Informative references

[RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound,
          "Dynamic Updates in the Domain Name System (DNS UPDATE)",
          RFC 2136, April 1997.

[RFC2535] Eastlake, D., "Domain Name System Security Extensions",
          RFC 2535, March 1999.

[RFC2766] Tsirtsis, G. and P. Srisuresh, "Network Address
          Translation - Protocol Translation (NAT-PT)", RFC 2766,
          February 2000.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network
          Address Translator (Traditional NAT)", RFC 3022,
          January 2001.

[RFC3102]   Borella, M., Lo, J., Grabelsky, D., and G. Montenegro,
            "Realm Specific IP: Framework", RFC 3102, October 2001.

[RFC3748]   Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H.
            Levkowetz, "Extensible Authentication Protocol (EAP)",
            RFC 3748, June 2004.

[RFC4025]   Richardson, M., "A Method for Storing IPsec Keying
            Material in DNS", RFC 4025, March 2005.

[RFC4225]   Nikander, P., Arkko, J., Aura, T., Montenegro, G., and E.
            Nordmark, "Mobile IP Version 6 Route Optimization Security
            Design Background", RFC 4225, December 2005.

[RFC4306]   Kaufman, C., "Internet Key Exchange (IKEv2) Protocol",
            RFC 4306, December 2005.

[RFC4423]   Moskowitz, R. and P. Nikander, "Host Identity Protocol
            (HIP) Architecture", RFC 4423, May 2006.

[RFC5770]   Komu, M., Henderson, T., Tschofenig, H., Melen, J., and A.
            Keranen, "Basic Host Identity Protocol (HIP) Extensions
            for Traversal of Network Address Translators", RFC 5770,
            April 2010.

[nsrg-report]
            Lear, E. and R. Droms, "What's In A Name:Thoughts from the
            NSRG", draft-irtf-nsrg-report-10 (work in progress),
            September 2003.

[IEEE.802-15-4.2006]
            "Information technology - Telecommunications and
            information exchange between systems - Local and
            metropolitan area networks - Specific requirements - Part
            15.4: Wireless Medium Access Control (MAC) and Physical
            Layer (PHY) Specifications for Low-Rate Wireless Personal
            Area Networks (WPANs)", IEEE Standard 802.15.4,
            September 2006, <http://standards.ieee.org/getieee802/
            download/802.15.4-2006.pdf>.

[chiappa-endpoints]
            Chiappa, J., "Endpoints and Endpoint Names: A Proposed
            Enhancement to the Internet Architecture",
            URL http://www.chiappa.net/~jnc/tech/endpoints.txt, 1999.

[Nik2001]   Nikander, P., "Denial-of-Service, Address Ownership, and
            Early Authentication in the IPv6 World", in Proceesings
            of Security Protocols, 9th International Workshop,

Cambridge, UK, April 25-27 2001, LNCS 2467, pp. 12-26,
Springer, 2002.

[Bel1998] Bellovin, S., "EIDs, IPsec, and HostNAT", in Proceedings
of 41th IETF, Los Angeles, CA,
URL http://www1.cs.columbia.edu/~smb/talks/hostnat.pdf,
March 1998.

Author's Address

Robert Moskowitz
Verizon Telcom and Business
1000 Bent Creek Blvd, Suite 200
Mechanicsburg, PA
USA

Email: robert.moskowitz@verizonbusiness.com