Network Working Group R. Moskowitz
Internet-Draft ICSAlabs, a Division of TruSecure Corporation
Expires: March 1, 2004
P. Nikander
Ericsson Research Nomadic Lab
Sep 2003

Host Identity Protocol Architecture
draft-moskowitz-hip-arch-05

Status of this Memo

Copyright Notice

Abstract

This memo describes the reasoning behind a proposed 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, strengths and weaknesses, and how a new namespace will add completeness to them. The roles of this new namespace in the protocols are defined.

Table of Contents

1. Introduction

The Internet has created two 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 Host Identity namespace fills an important gap between the IP and DNS namespaces. The Host Identity namespace consist of Host Identifiers (HI). A Host Identifier is cryptographic in its nature; it is the public key of an asymmetric key-pair. A Host Identity is assigned to each host, or technically its networking kernel or stack. Each host will have at least one Host Identity and a corresponding Host Identifier, which can either be public (e.g. published in DNS), or anonymous. Client systems will tend to have both public and anonymous Identities.

Although the Host Identities could be used in many authentication systems, the presented architecture introduces a new protocol, called the Host Identity Protocol (HIP), and a cryptographic exchange, called the HIP base exchange [4]. The new protocol provides for limited forms of trust between systems. It enhances mobility, multi-homing and dynamic IP renumbering [7], aids in protocol translation / transition [4], and reduces certain types of denial-of-service (DoS) attacks [4].

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

2. Background

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

There are two principal namespaces in use in the Internet for these components: IP numbers, and Domain Names. Email, HTTP and SIP addresses are really only extensions of Domain Names.

IP numbers are a confounding of two namespaces, the names of the 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 framed by concerns of requiring a TCPng effort as well.

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 addresses provide naming for both humans and autonomous applications. Email addresses are extensions of Domain Names, only in so far as a named service is responsible for managing a person's mail. There is some anonymity in Email addresses.

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 because computing platforms are not well named with the current namespaces.

2.1 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 either from mobility or renumbering.

If the namespace for computing platforms is cryptographically based, 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:

> The namespace should be applied to the IP 'kernel'. The IP kernel is the 'component' between services and the packet transport infrastructure.
>
> 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 TCB). This may require changes to the current APIs. In the long run, it is probable that some new APIs are needed.

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

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

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.

The names must be statistically globally unique. 64 bits is inadequate (1% chance of collision in a population of 640M); thus approximately 100 or more bits should be used.

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

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.

The namespace should provide authentication services. This is a preferred function.

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, such a new namespace 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.

3. 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, but not limited to, an IP stack. A system can have multiple identities, some 'well known', some anonymous. A system may self assert its identity,

or may use a third-party authenticator like DNSSEC, 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.

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.

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 Identifiers. As documented in the Host Identity Protocol specification [4], 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 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.

3.1 Host Identifiers

Host Identity adds two main features to Internet protocols. The first is a decoupling of the internetworking and transport layers; see Section 4. 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 to authenticate security protocols like IPsec.

The only completely defined structure of the Host Identity is that of a public 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. 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 Identities. Another representation of the Host Identities, the Local Scope Identifier (LSI), can also be used in protocols and APIs.

3.2 Storing Host Identifiers in DNS

The Host Identifiers should be stored in DNS. The exception to this is anonymous identities. The HI is stored in a new RR type, to be defined. This RR type is likely to be quite similar to the IPSECKEY RR [5].

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.

3.3 Host Identity Tag (HIT)

A Host Identity Tag is an 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 whatever underlying technology is used.

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. In the extremely rare case that a single HIT happens to map to more than one Host Identities, 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.

3.4 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. The generation of LSIs is defined in the Host Identity Protocol specification [4].

Examples of how LSIs can be used include: as the address in a FTP command and as the address in a socket call. Thus, LSIs act as a bridge for Host Identities into old protocols and APIs.

4. 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 [6] and, e.g., the unpublished Internet-Draft Endpoints and Endpoint Names [9] by Noel Chiappa, the IP addresses currently embody the dual role of locators and endpoint 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 endpoint name.

In the HIP architecture, the endpoint names and locators are separated from each other. IP addresses continue to act as locators. The Host Identifiers take the role of endpoint identifiers. It is important to understand that the endpoint 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.

```
   Process ------ Socket                  Process ------ Socket
                    |                                      |
                    |                                      |
                    |                                      |
                    |                                      |
   Endpoint         |                     Endpoint --- Host Identity
            \       |                                      |
              \     |                                      |
                \   |                                      |
                  \ |                                      |
```

```
Location --- IP address              Location --- IP address
```

Figure 1

4.1 Transport associations and endpoints

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 more bound to IP addresses but to Host Identities.

It is possible that a single physical computer hosts several logical endpoints. With HIP, each of these endpoints 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 endpoint.

5. 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 very 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 allows these IP addresses to be linked with each other, and if one address becomes unusable (e.g. due to a network failure), existing transport associations can be easily 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 totally ignore the source address. However, as discussed in Section 5.2 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).

5.1 Rendezvous server

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 Dynamic DNS could be used for this function for infrequently moving nodes, an alternative to using DNS in this fashion is to use a piece of new static infrastructure called a HIP rendezvous server. Instead of registering its current dynamic address to the DNS server, the mobile node registers the address(es) of its rendezvous server(s). The mobile node keeps the rendezvous server(s) continuously updated with its current IP address(es). A rendezvous server simply forwards the initial HIP packet from an initiator to the mobile node at its current location. All further packets flow between the initiator and the mobile node. There is typically very little activity on a rendezvous server, address updates and initial HIP packet forwarding. Thus, one server can support a large number of potential mobile nodes. The mobile nodes must trust the rendezvous server to properly maintain their HIT and IP address mappings.

The rendezvous server is also needed if both of the nodes are mobile

and happen to move at the same time. In that case, the HIP readdress packets will cross each other in the network and never reach the peer node. To solve this situation, the nodes should remember the rendezvous server address, and re-send the HIP readdress packet to the rendezvous server if no reply is received.

The mobile node keeps its address current on the rendezvous server by setting up a HIP association with the rendezvous server and sending HIP readdress packets to it. A rendezvous server will permit two mobile systems to use HIP without any extraneous infrastructure (in addition to the rendezvous server itself), including DNS if they have a method other than a DNS query to get each other's HI and HIT.

5.2 Protection against Flooding Attacks

While 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 [8]. In a distributed flooding attack an attacker opens (anonymous) high volume HIP connections with a large number of hosts, 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 close this 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.

Whenever 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.

6. 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. In the future, other ways of transporting payload data may be developed, including ones that do not use cryptographic protection.

In practise, 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.

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. 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, 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.

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.

7. 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) [2] or NAT Protocol translation (NAT-PT) [1].

In a network environment where the identification is based on the IP addresses, identifying the communicating nodes is difficult when NAT is used. With HIP, the transport layer endpoints 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; the IP addresses may be changed freely during packet traversal.

For a HIP based flow, a NAT or NAT-PT system tracks the mapping of HITs and the corresponding IPsec SPIs to an IP address. Many HITs 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.

The 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 place holder for embedded IP addresses.

7.1 HIP and TCP Checksum

There is no way for a host to know if any of the IP addresses in the 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.

8. 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 one for anonymous usage. Although anonymous 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, representing which hosts they accept HIP exchanges, and the preferred transforms and local lifetimes.

9. Benefits of HIP

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

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

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

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

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 trying intentionally 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 [3] 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 to them. That is, if they have a Fully Qualified Domain Name (FQDN), that typically belongs to a NAT device or a dial-up server, and does not really identify the system itself but its current connectivity. FQDN names (and their extensions as email names) are Application Layer names; more frequently naming processes than a particular system. This is why many systems on the internet are not registered in DNS; they do not have processes of interest to other Internet hosts.

DNS names are indirect 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 don't 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).

Since all systems can have a Host Identity, every system can have an entry in the DNS. The mobility features in HIP make it attractive to trusted 3rd parties to offer rendezvous servers.

9.1 HIP's Answers to NSRG questions

The IRTF Name Space Research Group has posed a number of evaluating questions in their report [6]. In this section, we provide answers to these questions.

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

   At the fundamental level, HI decouples the internetworking layer from the transport layer, allowing each to evolve separately. At the same time, the decoupling makes end-host mobility and multi-homing easier. It also allows mobility and multi-homing across the IPv4 and IPv6 networks. HIs make network renumbering easier. At the conceptual level, 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, in the form of HITs or LSIs, are used by legacy applications as if they were IP addresses. 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?

   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 infrastructure of rendezvous servers is needed.

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 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, hash based 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.

10. 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

specification [4].

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 anonymous HI, it knowingly risks a MitM attack. The responder may choose not to accept a HIP exchange with an anonymous initiator.

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.

10.1 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 6, once a HIP session has been established, the SPI value in an IPsec packet may be used as an index, indicating the HITs. In practise, the firewalls can inspect the 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 the HIP packets allow a capable firewall to make sure 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 the 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 (like in CMP and CMC) to issue the HIT revocation notice.

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.

10.2 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 an 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 carry the 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.

11. Acknowledgments

For the people historically involved in the early stages of HIP, see the Acknowledgements section in the Host Identity Protocol specification [4].

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.

References (informative)

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

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

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

[4] Moskowitz, R., Nikander, P. and P. Jokela, "Host Identity Protocol", draft-moskowitz-hip-07 (work in progress), June 2003.

[5] Richardson, M., "A method for storing IPsec keying material in DNS", draft-ietf-ipseckey-rr-07 (work in progress), September 2003.

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

[7] Nikander, P., "End-Host Mobility and Multi-Homing with Host Identity Protocol", draft-nikander-hip-mm-00 (work in progress), June 2003.

[8] Nikander, P., "Mobile IP version 6 Route Optimization Security Design Background", draft-nikander-mobileip-v6-ro-sec-01 (work in progress), July 2003.

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

Authors' Addresses

Robert Moskowitz
ICSAlabs, a Division of TruSecure Corporation
1000 Bent Creek Blvd, Suite 200

Mechanicsburg, PA
USA

EMail: rgm@icsalabs.com

Pekka Nikander
Ericsson Research Nomadic Lab

JORVAS FIN-02420
FINLAND

Phone: +358 9 299 1
EMail: pekka.nikander@nomadiclab.com

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Acknowledgement

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