R. Moskowitz, ICSA Labs

Internet Draft

Document: <draft-moskowitz-hip-impl-01.txt> February 2001

Host Identity Payload

Implementation

Status of this Memo

Table of Contents

1. Abstract....2
2. Conventions used in this document....2
3. Introduction....2
4. HIP Rendezvous Server - support for mobile Responders....2
5. Using LSIs in place of IP addresses....3
5.1. HIP address translation services....3
6. Using the Host Identity....3
7. HIP Fragmentation Support....4
8. HIP Scenarios....4

8.1. HIP Scenario #1.....4
8.2. HIP Scenario #2.....5
8.3. HIP Scenario #3.....5
8.4. HIP Scenario #4.....6
8.5. Refusing a HIP exchange.....7
8.6. Opportunistic HIP.....8
9. Localize Address Translation with HIP.....8
10. Localize Address Translation example of FTP.....8
11. Security Considerations.....8
11.1. HITs used in ACLs.....9

12. IANA Considerations.....10
13. ICANN Considerations.....10
14. References.....10
15. Acknowledgments.....10
16. Author's Address.....11
17. Copyright Statement.....11

1. Abstract

This memo describes implementation aspects of HIP [HIP]. Particular attention is paid to items needed for HIP to span Addressing Realms (e.g. using NATs) and address reassignment (e.g. in mobility).

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC-2119].

3. Introduction

The Host Identity Payload and the HIP protocol have a set of characteristics that make it especially amendable to today's real Internet challenges like end-to-end connectivity and security across addressing realms (via NATs) and mobility support. Beyond host authentication and light-weight IPsec key exchange, HIP's decoupling of the Internetworking and Transport layers allow it this support of NAT transversal and mobility.

The NAT transversal is accomplished at the cost of maintaining parts of the HIP state in the NAT systems. The NAT is not privy to the

keying material exchanged between the systems. NAT transversal DOES rely on plentiful addresses on the private side of the NAT, but can use a single address (without PORT mappings) on the global or public side of the NAT.

4. HIP Rendezvous Server - support for mobile Responders

A mobile responder does not have to rely on the availability of Dynamic DNS with DNSSEC to make its current IP address known to potential Initiators. HIP can make use of a Rendezvous Server. The functions of a Rendezvous Server are minimal; it needs to track the Responder's IP address, and it needs to forward any HIP I1 packets to the Responder.

A Responder designates its Rendezvous Server by using a PTR RR to the Rendezvous Server in place of an A RR. The HI and HIT KEY RR records MUST also be provided. Use of the PTR RR allows the Rendezvous Server to change its IP address without all relying

Responders to make DNS changes. The Rendezvous Server MUST have an enrollment procedure. This procedure MUST include the registration of the Responder's HIT and current IP address. It MAY include the Responder's HI. The procedure MUST support changes to the HIT. Whenever the Responder needs to change its IP address, it sets up a HIP authenticated connection to the Rendezvous Server and sends a HIP readdress payload. The Responder MAY use a very long lifetime for this connection.

Since the Rendezvous Server only needs to support is Responder registration, Responder readdressing, and I1 HIP packet forwarding, the Rendezvous Server has little work to do, and needs minimal bandwidth.

5. Using LSIs in place of IP addresses

LSIs can be used in place of IP addresses in socket calls. This can be accommodated with the aid of a modified resolver module. When a HIP exchange produces, and exchanges the LSIs, either the LSI can be provided to the applications in place of the IP addresses and then the HIP module perform any substitution required, or the HIP module can intercept all address requests and use the LSI where appropriate.

At issue here is when do applications get IP addresses, and where do

they use them. The LSIs are not established until the 3rd and 4th HIP packets, which is probably after the application opened the connection. If the applications are using IP addresses within their data stream (as FTP commands do), the HIP module will have to perform tasks similar to a NAT to find these addresses and replace them with the appropriate LSI.

If the applications are using LSIs (as could be the case on subsequent connections during the lifetime of the HIP state), then the HIP module directs opens to LSIs to the appropriate IP address. The HIP module would not need to perform NAT functions on imbedded IP addresses.

5.1. HIP address translation services

Since the LSI cannot be used directly by applications in many cases, the HIP implementations can provide Address Translation services for registered applications. An application can either provide HIP with the information of where imbedded addresses are, or can supply a piece of code for HIP to use to replace addresses with LSIs. A HIP implementation MUST provide this function for FTP and SHOULD provide it for HTTP.

6. Using the Host Identity

There are a number of ways that Host Identity can be used in Internet Protocols. The first is to use it in IKE [IKE]. HIT can be used in Main Mode. For this, the Host Identity MUST be a Public Key, and an appropriate Main Mode authentication (e.g. DSA signature) used. The SPI of the HIT can replace the usage of IP addresses in IKE. An appropriate ISAKMP [ISAKMP] payload will be needed to accommodate the Host Identity and HIT. These additions to IKE would produce a mode of operation for IKE that could traverse a NAT. This, coupled with ESP transport mode, would produce a NAT friendly IPsec mode (note that the NATs can alter none of the data within the ESP).

Another, and perhaps more powerful mode is a new, lightweight, protocol that will allow for one host to convey its Host Identity to another host. This Host Identity Protocol will enable two hosts to exchange Host Identity and related information and rapidly establish an ESP Security Association. It will lack the fine-grain controls of IKE and some of IKE's security features (like identity

protection).

7. HIP Fragmentation Support

A HIP implementation MUST support IP fragmentation/reassembly. HIP packets can get large, and may encounter low MTUs along their routed path. Since HIP does not provide a mechanism to use multiple IP datagrams for a single HIP packet, support of path MTU discovery does not bring any value to HIP. HIP aware NAT systems MUST perform any IP reassembly/fragmentation.

8. HIP Scenarios

There are a number of scenarios for using HIP. The variables in these scenarios are the number of address realms (managed by NATs) and the jurisdictional location of the boundaries, and the type of Host Identity (public key or string like an FQDN).

A HIP exchange SHOULD be initiated whenever the DNS lookup returns HIP KEY resource records. Since some hosts may choose not to have information in DNS, hosts SHOULD support opportunistic HIP.

8.1. HIP Scenario #1

Initiator and Responder in common addressing realm (i.e. both public or both private).

Initiator gets IP address, HI, and HIT of Responder somehow.

```
     Hard coded (bad)
     DNS lookup
             DNSSEC important
```



```
I --> DNS (lookup R)
I <-- DNS (R's address, HI, and HIT)
I1      I --> R (Hi. Here is my I1, let's talk HIP)
R1      I <-- R (OK.  Here is my R1, handle this HIP cookie)
I2      I --> R (Compute, compute, here is my counter I2)
R2      I <-- R (OK.  LetÆs finish HIP cookie with my R2)
```

All further packets are without HIP, HIP is implied with the SPIs in ESP

```
I --> R (ESP protected data)
I <-- R (ESP protected data)
```

8.2. HIP Scenario #2

```
Initiator behind NAT, Responder publicly addressed.

Initiator gets IP address of Responder somehow.
     Hard coded (bad)
     DNS lookup
             DNSSEC important
             If no default route to NAT, DNS ALG provides NAT with
                     Responder's IP address and HIT; NAT provides
                     DNS ALG with substitute internal address.

I --> NAT(I) --> DNS (lookup R)
I <-- NAT(I) <-- DNS (R's info, record it in NAT)
I --> NAT(I) --> R (Hi, let's talk HIP)
     NAT records Initiator's HIT and address makes address changes
             in IP header
I <-- NAT(I) <-- R (OK, handle this HIP cookie)
     NAT changes addresses based on HITs in HIP.
I --> NAT(I) --> R (Compute, compute, here is my counter HIP with
        SPI(R))
     NAT adds Initiator's Responder SPI to I/R state and changes
     addresses.
I <-- NAT(I) <-- R (OK, LetÆs finish HIP cookie, and here is SPI(I))
     NAT adds Responder's Initiator SPI to I/R state and changes
     addresses.


All further packets are without HIP, HIP is implied with the SPIs in
ESP.
NAT uses SPIs for address state.
Hosts use SPIs in place of IP addresses inside protocols.
     E.G. Initiator uses Responder's Initiator SPI (packet 4) for
FTP port command.  Responder is able to remap this to state it has
on Initiator.
```

8.3. HIP Scenario #3


```
Initiator publicly addressed, Responder behind NAT.
```

NAT configured with Responder's IP address and HIT. DNS for the responder will have the NAT's address.

Initiator gets IP address of Responder (NAT) somehow.
    Hard coded (bad)
    DNS lookup
        DNSSEC important
        If no default route to NAT, NAT maps Initiator's HIT and SPI to an internal address. Since the SPI has to be used after HIP exchange, and SPI is on a host basis, the Initiator will use a separate internal address for each internal host.

I --> DNS (lookup R, get NAT's address and R's HIT)
I <-- DNS (R's info)
I --> NAT(R) --> R (Hi, let's talk HIP)
    NAT records Initiator's HIT and address makes address changes in IP header. Uses Responder's HIT to map to responder.
I <-- NAT(R) <-- R (OK, handle this HIP cookie)
    NAT changes addresses based on HITs in HIP.
I --> NAT(R) --> R (Compute, compute, here is my counter HIP with SPI(R))
    NAT adds Initiator's Responder SPI to I/R state and changes addresses. The NAT will have one responder HIT to address state, but potentially many responder SPI to address state.
I <-- NAT(R) <-- R (OK, LetÆs finish HIP cookie, and here is SPI(I))
    NAT adds Responder's Initiator SPI to I/R state and changes addresses.

All further packets are without HIP, HIP is implied with the SPIs in ESP.
NAT uses SPIs for address state.
Hosts use SPIs in place of IP addresses inside protocols.
    E.G. Initiator uses Responder's Initiator SPI (packet 4) for FTP port command. Responder is able to remap this to state it has on Initiator.

8.4. HIP Scenario #4

Initiator and Responder behind separate NATs.

NAT configured with Responder's IP address and HIT. DNS for the responder will have the NAT's address.

Initiator gets IP address of Responder (NAT) somehow.
    Hard coded (bad)
    DNS lookup
        DNSSEC important

If no default route to initiator's NAT, DNS ALG
provides NAT with Responder's IP address and
HIT; NAT provides DNS ALG with substitute
internal address.
If no default route to responder's NAT, NAT maps
Initiator's HIT and SPI to an internal address.
Since the SPI has to be used after HIP
exchange, and SPI is on a host basis, the
Initiator will use a separate internal address
for each internal host.

I --> NAT(I) --> DNS (lookup R, get NAT's address and R's HIT)
I <-- NAT(I) <-- DNS (R's info, record it in NAT)
I --> NAT(I) --> NAT(R) --> R (Hi, let's talk HIP)
Initiator's NAT records Initiator's HIT and address makes
address changes in IP header
Responder's NAT records Initiator's HIT and address makes
address changes in IP header. Uses Responder's HIT to
map to responder.
I <-- NAT(I) <-- NAT(R) <-- R (OK, handle this HIP cookie)
Both NATs changes addresses based on HITs in HIP.
I --> NAT(I) --> NAT(R) --> R (Compute, compute, here is my counter
HIP with SPI(R))
Initiator's NAT adds Initiator's Responder SPI to I/R state and
changes addresses.
Responder's NAT adds Initiator's Responder SPI to I/R state and
changes addresses. The NAT will have one responder HIT to
address state, but potentially many responder SPI to address
state.
I <-- NAT(I) <-- NAT(R) <-- R (OK, LetÆs finish HIP cookie, and here
is SPI(I))
NAT adds Responder's Initiator SPI to I/R state and changes
addresses.

All further packets are without HIP, HIP is implied with the SPIs in
ESP.
Both NATs uses SPIs for address state.
Hosts use SPIs in place of IP addresses inside protocols.
E.G. Initiator uses Responder's Initiator SPI (packet 4) for
FTP port command. Responder is able to remap this to state it has
on Initiator.

8.5. Refusing a HIP exchange

A HIP aware host may choose not to accept a HIP exchange negotiation. If the host's policy is to only be an initiator, it should begin its own HIP exchange. There is a risk of a race condition if each host's policy is to only be an initiator, at which point the HIP exchange will fail.

If the host's policy does not permit it to enter into a HIP exchange with the initiator, it should send an ICMP Host Unreachable,

Administratively Prohibited message. A more complex HIP packet is not used here as it actually opens up more potential DOS attacks than a simple ICMP message.

8.6. Opportunistic HIP

A host MAY attempt to force a system that initiates a regular TCP or UDP connection to restart the connection using HIP. This opportunistic HIP is fraught with denial of service attacks, but is provided for those that wish to risk the DOS attacks to gain this function.

If a host receives a TCP SYN or a UDP packet from a host that it has not done a HIP exchange, the receiving host MAY attempt to force a HIP exchange by sending a HIP packet that contains:

HI, HIT, and SIG

In other words, the host sends the initiator the information that it would have gotten from a DNS lookup. Once the initiator has this HIP information, it should delay its transport negotiation, perform the HIP exchange and then restart the higher layer (with the I2 HIP packet).

9. Localize Address Translation with HIP

Many applications that transmit IP addresses within the data stream cannot be easily changed to accommodate using the LSI in place of the address. A HIP implementation SHOULD provide an address translation service and allow for applications to register and provide for their specific translation requirements. For example, FTP port 21 would be registered with a module that would replace the address in the port command with the LSI.

The HIP implementation SHOULD supply the basic translation services,

for FTP and HTTP URLs. The implementation SHOULD provide a translation registration service for other products to add to the local translation function.

10. Localize Address Translation example of FTP

This service listens on TCP port 21 (FTP Commands) for an FTP server or client port selected by an FTP client. One the client side of the implementation, the service looks for PORT commands, and replaces IP addresses with the matching LSIs. This is reversed on the server side service that replaces LSIs with IP addresses.

11. Security Considerations

HIP is designed to provide secure authentication of hosts and provide a fast key exchange for IPsec ESP. HIP also attempts to limit the exposure of the host to various denial-of-service and man-in-the-middle attacks. In so doing, HIP itself is subject to its own DOS and MITH attacks that potentially could be more damaging to a host's ability to conduct business as usual.

HIP transparently provides mobility for the initiating system. Mobility for a responder requires a Rendezvous Server. There are two potential risks with these servers. They can be compromised and have their forwarding information altered. This will result in the mobile Responder being unreachable. They are a nature place for a MITH attack. Thus the responders are placing a large amount of trust in these systems.

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 build at HI selection time. Since this packet is fixed and easily spoofed the initiator only reacts to it if it has just started a connection to the responder, and it sends the easy to construct I1 HIP packet.

11.1. HITs used in ACLs

It is expected that HITs will be used in ACLs. NATs will use HITs to control egress and ingress to networks, with an assurance difficult to achieve today.

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 here 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 many 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 MANDITORY for systems that

function as responders behind a NAT (see sec 8.3 and 8.4 and the processing of the I1 HIP packets by the responder's 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.

12. IANA Considerations

The IANA considerations for HIP are covered in the Host Identity payload document [HIP].

13. ICANN Considerations

ICANN are covered in the HIP Architecture [HIPARCH] document.

14. References

[HIP], Moskowitz, R., "Host Identity Payload", draft-ietf-moskowitz-

hip-02.txt, January 2001.

[RFC-2119], Bradner, S, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, Harvard University, March 1997

[HIPARCH], Moskowitz, R., "Host Identity Payload Architecture", draft-ietf-moskowitz-hip-arch-02.txt, January 2001.

[IKE], Harkins, D., and Carrel, D., "The Internet Key Exchange", RFC 2409, November 1998.

[ISAKMP], Maughan, D., Schertler, M., Schneider, M., and Turner, J., "Internet Security Association and Key Management Protocol", RFC 2408, November 1998.

15. Acknowledgments

The drive to create HIP came to being after attending the MALLOC meeting at IETF 43. Baiju Patel and Hilarie Orman really gave me the assist to get HIP beyond 5 paragraphs of ideas. It has matured considerably since the early drafts thanks to extensive input from IETFers. Most importantly, its design goals are articulated and are different from other efforts in this direction. Particular mention goes to the members of the NameSpace Research Group of the IRTF. Noel Chiappa provided the framework for LSIs and Kieth Moore the impetuous to provide resolvability. Steve Deering provided encouragement to keep working, as a solid proposal can act as a proof of ideas for a research group.

Many others contributed; extensive security tips were provided by Steve Bellovin. Rob Austein kept the DNS parts on track. Paul

Kocher taught me how to make the cookie exchange expensive for the Initiator to respond, but easy for the Responder to validate. Rodney Thayer and Hugh Daniels provide extensive feedback. John Gilmore kept me challenged to provide something of value. I hope I have.

16. Author's Address

Robert Moskowitz
ICSA Labs
1200 Walnut Bottom Rd.
Carlisle, PA 17013

Email: rgm@icsa.net

17. Copyright Statement