Middlebox Traversal of HIP Communication

Martin Stiemerling, Jürgen Quittek and Lars Eggert

***Abstract* — The Host Identity Protocol (HIP) fundamentally changes the way two hosts in the Internet communicate. One key advantage over other schemes is that HIP does not require any modifications to the traditional network-layer functionality of the Internet, i.e., its routers. HIP deployment should therefore be transparent. In the current Internet, however, many devices other than routers may affect the network-layer behavior of the Internet. These "middleboxes" are intermediary devices that perform functions other than the normal, standard functions of an IP router on the datagram path between source host and destination hosts. Examples of middleboxes include network address translators, packet classifiers, performance-enhancing proxies, load balancers amongst many. Whereas some types of middleboxes may not interfere with HIP at all, others can affect some aspects of HIP communication and others can render HIP communication impossible. This document examines the specifics of how middleboxes can interfere with HIP and discusses options to enable HIP traffic to traverse some types of middleboxes. It does not promote the use of any of the discussed types of middleboxes.**

I. Introduction

THE current specification of the Host Identity Protocol (HIP) [1] assumes simple Internet paths, where routers forward globally routable IP packets based on their destination address alone. Over such paths, the HIP protocol performs well.

In the current Internet, such pure paths are becoming increasingly rare. For a number of reasons, several types of devices modify or extend the pure forwarding functionality the Internet's network layer used to deliver. RFC 3234 [3] coins the term *middleboxes* for such devices: "A middlebox is (…) any intermediary device performing functions other than the normal, standard functions of an IP router on the datagram path between a source host and destination host."

Middleboxes affect communication in a number of ways. For example, they may inspect the flows of some transport protocols, such as TCP, and selectively drop, insert or modify packets. If such devices encounter a higher-layer protocol they do not support, or even a variant of a supported protocol that they do not know how to handle, communication across the middlebox may become impossible for these kinds of traffic.

There are many different variants of middleboxes. The most common ones may be network address translators and firewalls. RFC 3234 identifies many other types of middleboxes. One broad way of classifying them is by behavior. The first group operates on packets, does not modify application-layer payloads and does not insert additional packets. This group includes NAT, NAT-PT, SOCKS gateways, IP tunnel endpoints, packet classifiers, markers, schedulers, transport relays, IP firewalls, application firewalls, involuntary packet redirectors and anonymizers. Other middleboxes exist, such as TCP performance-enhancing proxies, application-level gateways, gatekeepers and session control boxes, transcoders, proxies, caches, modified DNS servers, content and applications distribution boxes, load balancers that divert or modify URLs, application-level interceptors and application-level multicast systems.

An earlier document investigated the impact of network address translation on HIP deployment [2]. However, it does not discuss the interactions of HIP and other types of middleboxes. Section II of this document describes a simple network scenario involving a middlebox and investigates how different types of middleboxes affect HIP communication in the given scenario. Section III discusses future work and summarizes the document.

Manuscript received October 1, 2004. Parts of this work are a product of the *Ambient Networks* and *Enthrone* projects supported in part by the European Commission under its *Sixth Framework Programme*. It is provided "as is" and without any express or implied warranties, including, without limitation, the implied warranties of fitness for a particular purpose. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the *Ambient Networks* or *Enthrone* projects or the European Commission.

Martin Stiemerling, Jürgen Quittek and Lars Eggert are with NEC Europe Ltd, Network Laboratories, Heidelberg, Germany (phone: +49-6221-905-1143; fax: +49-6221-905-1155; e-mail: {stiemerling, quittek, eggert} @netlab.nec.de.)

II. HIP Across Middleboxes

Figure 1 illustrates a simple network scenario that involves a single middlebox along the path between two networks, *Net 1* and *Net 2*. Packets flowing between the two networks traverse the middlebox, which may drop them, modify their payload or header, diverted them or perform any number of additional operations, depending on the middlebox type.

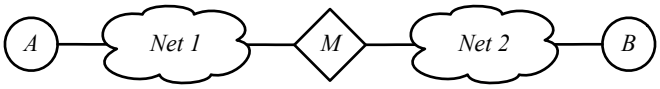

Figure 1. Network Scenario with a Middlebox

The remainder of this section discusses the impact of middleboxes on HIP communication, specifically, the HIP base exchange and HIP's underlying IPsec-based transport. It examines potential issues caused by a number of different middlebox types: a network address translator [4], an application-level gateway and a firewall. Future revisions of this document will examine additional types of middleboxes that would exceed the current length limitations for this abstract.

A. HIP and Network Address Translators

For the purposes of this discussion, assume the middlebox shown in Figure 1 is a network address translator (NAT) and that HIP nodes *A* and *B* are located in two different address realms. *Net 1* is the globally addressed, public Internet and *Net 2* uses private address space. The NAT translated between the two. (This scenario considers only the presence of a single NAT; the effects are identical if both HIP nodes are located behind NATs.)

For packets flowing across the NAT, it changes their IP headers and frequently also the headers of those higher-layer protocols that the NAT supports. Usually, the NAT translates the IP addresses and transport-layer port numbers between the address realms it connects. In the scenario in Figure 1, host *A* addresses all HIP packets to host *B* to a public IP address that

is provided by the NAT. The NAT intercepts packets destined to this address and replaces the destination IP address (and typically also the TCP or UPD port number) before it forwards the packet into the private realm.

Host *A* does not usually know the public IP address (and port number) that the NAT uses for this translation service. Therefore, applications that carry address information in their payloads or applications that rely on static header fields can experience severe problems, if the NAT does not also modify the payloads or header fields accordingly. One common occurrence in such cases is that application-level signaling traffic is able to traverse the NAT, but subsequent data traffic is dropped or corrupted [5].

Currently, NATs are heavily used in IPv4 networks when a significant lack of address space exists in some areas of the global topology. Initially, they were expected not to be present in IPv6 networks due to their much larger address space. It is likely, however, that NATs will also be used in IPv6 networks, because other aspects of their presence have become important to some stakeholders. For example, even organizations that own a sufficiently large block of public IPv4 addresses for all their hosts are installing NAT devices. One reason is that a NAT hides the structure of the internal network from the outside. A second reason is that a NAT decouples internal and external addressing, allowing renumbering of each side without affecting the other.

These reasons suggest that HIP must become able to traverse NAT devices, even within an IPv6 network. HIP communication can be divided into two phases. The first phase is the HIP *base exchange* handshake that creates a HIP association between two nodes. The second phase is the actual application data exchange via IPsec that follows a successful base exchange [9]. This section describes the NAT traversal problems for each of those two phases.

1) HIP Base Exchange

The HIP base exchange uses different transport mechanisms for IPv6 and IPv4 networks [6]. With IPv6, a HIP-specific IPv6 extension header carries all necessary information for the base exchange. Within an IPv4 network, HIP transmits its base exchange messages as UDP payloads.

When HIP is used with IPv6, current implementations use empty IPv6 packets that contain no payload – all HIP information is contained within the IPv6 extension header. This approach causes problems in combination with NATs that translate a large private address space into a much smaller public address space (arguably the most common case.) Usually, translation uses a combination of IP addresses and TCP or UDP port numbers to differentiate the flows of different internal hosts. Because the IPv6 HIP base exchange packets do not carry a TCP or UDP header conventional network address and port translation fails. No specific scheme currently exists for translating IPv6 HIP extension headers in the absence of TCP or UDP headers.

Another problem for HIP within an IPv6 network (that also applies to IPv4) is that HIP provides no explicit means for transmitting the sender's IP address other than the packet's normal source IP address. Even if host *B* in Figure 1 would know the public IP address used by the NAT, it has no means of providing this information to host *A*.

When HIP is used with IPv4, its use of UDP for base exchange messages enables NAT traversal. Instead of using a custom IP option, the UDP encapsulation sends the HIP header as the payload of a regular UDP packet. The NAT traversal issues of UDP are well-documented. One problem is that NATs frequently do not support UDP translation, or explicit policies forbid the translation. This effectively creates a black hole. Even when NATs translate UDP packets correctly, HIP mandates using fixed UDP port numbers of 272 for both the source and destination ports of base exchange packets. NATs that use the source port for demultiplexing concurrent streams and consequently modify it will likely cause the remote peer to drop the translated base exchange packet as invalid.

A fourth problem that exists when HIP is used with either IPv4 or IPv6 is that NATs are frequently configured to perform limited firewalling functionality. In this case, a NAT will drop any apparently unsolicited inbound traffic on the public side. Only inbound return traffic is translated. This means that the NAT will block any inbound base exchange packets on the public side; it will only translate base exchange packets that originate within the private realm. This significantly limits communication.

2) IPsec Data Exchange

After a successful base exchange, HIP uses IPsec for transmitting application data. A second set of issues with regard to NAT traversal is related to this use of IPsec and already well documented [7]. One approach for enabling IPsec traffic to traverse NATs is UDP encapsulation [8]. A second alternative is extending NATs to become IPsec-aware [2].

B. HIP and Application-Level Gateways

The last section discussed the case where the middlebox in Figure 1 is a NAT. This section looks at the same scenario, but assumes that the middlebox is an *application-level gateway*. An application level gateway intercepts data traffic and provides gateway functionality at the application level. It terminates transport-layer connections and forwards application data units towards their final destination across a second set of transport-layer connections. All packet forwarding occurs at the application-level. Usually, when an application-level gateway is in place, it is the only permitted means of communication between two nodes.

Application-level gateways are a severe obstacle for HIP communication. Such gateways can only parse and forward a very limited set of application-layer protocols, such as HTTP. Unknown or lower-layer traffic, such as HIP or IPsec, cannot use the connectivity an application-level gateway provides.

C. HIP and IP Firewalls

This section considers the case where the middlebox in Figure 1 is an *IP firewall*. Here, hosts *A* and *B* are located within the same addressing realm, separated by a firewall along the path. IP firewalls, also called packet filters, inspect each IP packet individually and decide whether to forward or discard it, based on the current set of configure policies.

This type of middlebox is not inherently an obstacle for HIP communication. However, such firewalls are frequently configured with very restrictive policies that prohibit unknown traffic. When such policies block base exchange or IPsec packets, maybe because HIP's IPv6 extension header is unknown, HIP communication is often impossible.

A second issue is similar to one aspect of NAT traversal, discussed above. Frequently, firewalls are configured to block unsolicited inbound traffic on the public side. This is

This is an issue for IPsec traffic, because correlation of outbound and inbound IPsec traffic (based on their respective source and destination addresses and SPI values) becomes impossible, unless the firewall learns these correlations on the fly.

III. Conclusion and Future Work

This document examined the specifics of how middleboxes can interfere with HIP and discusses options to enable HIP traffic to traverse some types of middleboxes. It does not promote the use of any of the investigated types of middlebox. This document argues that middlebox traversal is a key challenge for successful experimentation and future deployment of HIP. The HIP interactions of other types of middleboxes will be investigated in the future.

References

[1] Robert Moskowitz and Pekka Nikander. Host Identity Protocol Architecture. Work in Progress (draft-moskowitz-hip-arch-06), June 2004.

[2] Martin Stiemerling and Jürgen Quittek. Problem Statement: HIP operation over Network Address Translators. Work in Progress (draft-stiemerling-hip-nat-01), July 2004.

[3] Brian Carpenter and Scott Brim. Middlebox: Taxonomy and Issues. RFC 3234, February 2004.

[4] Pyda Srisuresh and Matt Holdrege. IP Network Address Translator (NAT) Terminology and Considerations. RFC 2663, August 1999.

[5] Brian Ford, Pyda Srisuresh and David Kegel. Peer-to-Peer (P2P) communication across middleboxes. Work in Progress (draft-ford-midcom-p2p-03), June 2004.

[6] Robert Moskowitz, Pekka Nikander, Petri Jokela (*ed.*) and Thomas Henderson. Host Identity Protocol, Work in Progress (draft-ietf-hip-base-00), June 2004.

[7] Bernd Aboba and William Dixon. IPsec-NAT Compatibility Requirements. RFC 3715, March 2004.

[8] Ari Huttunen, Victor Volpe, Larry DiBurro and Markus Stenberg. UDP Encapsulation of IPsec ESP Packets. Work in Progress (draft-ietf-ipsec-udp-encaps-09), May 2004.

[9] Stephen Kent and Randall Atkinson. Security Architecture for the Internet Protocol. RFC 2401, November 1998.