Host Identity Protocol: Identifier/Locator Split for Host Mobility and Multihoming

by Andrei Gurtov and Miika Komu, Helsinki Institute for Information Technology, and Robert Moskowitz, ICSAlab

A host and its location are identified using *Internet Protocol* (IP) addresses in the current Internet architecture. However, IP addresses can serve only as short-term identifiers because a considerable amount of hosts are *portable* devices and they change their IP addresses when moved from one network to another. Short-term identifiers disrupt long-term transport layer connections, such as Internet phone calls, and make locating the peer host more difficult. Therefore, mobility and multihoming are hard to implement securely in the present Internet. Upon changing an IP address, the host must prove to its peers that it is the same entity they communicated with before, requiring the use of cryptographic identities.

Another challenge the Internet faces is due to the fact that deployed protocols in the Internet are prone to *Denial-of-Service* (DoS) attacks. Substantial memory state can be created before the communicating peer is authenticated. Impersonation attacks are possible because IP addresses are relatively easy to forge. Because of difficulties in configuring *IP Security* (IPsec) for users, most Internet traffic is still transmitted in plaintext, making it easy for attackers to collect passwords or lists of visited websites, for example, in public *Wireless Local-Area Networks* (WLANs). As the IPv6 protocol is seeing gradual deployment, interoperating traditional IPv4 applications with new IPv6 applications remain a challenge.

The so-called *identifier/locator split* is recognized by the *Internet Engineering Task Force* (IETF) community as a next big change in the Internet architecture. Although the problem has been known for a long time[17], it has only recently started to get sufficient attention. Developments in public key cryptography and increased computational resources of hosts enables the use of cryptographic mechanisms to securely handle identities. Several proposals are under consideration in the IETF, including the *Locator Identifier Separation Protocol* (LISP)[16] for the network-based and the *Host Identity Protocol* (HIP) for the host-based approach. LISP focuses on improving scalability of the routing system, whereas HIP provides secure end-to-end mobility and multihoming. Therefore, the two proposals are complementary rather than competing.

HIP Architecture

The HIP architecture[1,2] uses the identity/locator split advantage to address Internet architecture challenges in an integrated approach. HIP was proposed by Bob Moskowitz in 1999 and since then has been under active development in the IETF Working Group and *Internet Research Task Force* (IRTF) Research Group.

HIP enables host mobility and multihoming across different address families (IPv4 and IPv6), offers end-to-end encryption and protection against certain DoS attacks, allows moving away from IP address-based access control to permanent host identities, and restores end-to-end host identification in the presence of several addressing domains separated by *Network Address Translation* (NAT) devices.

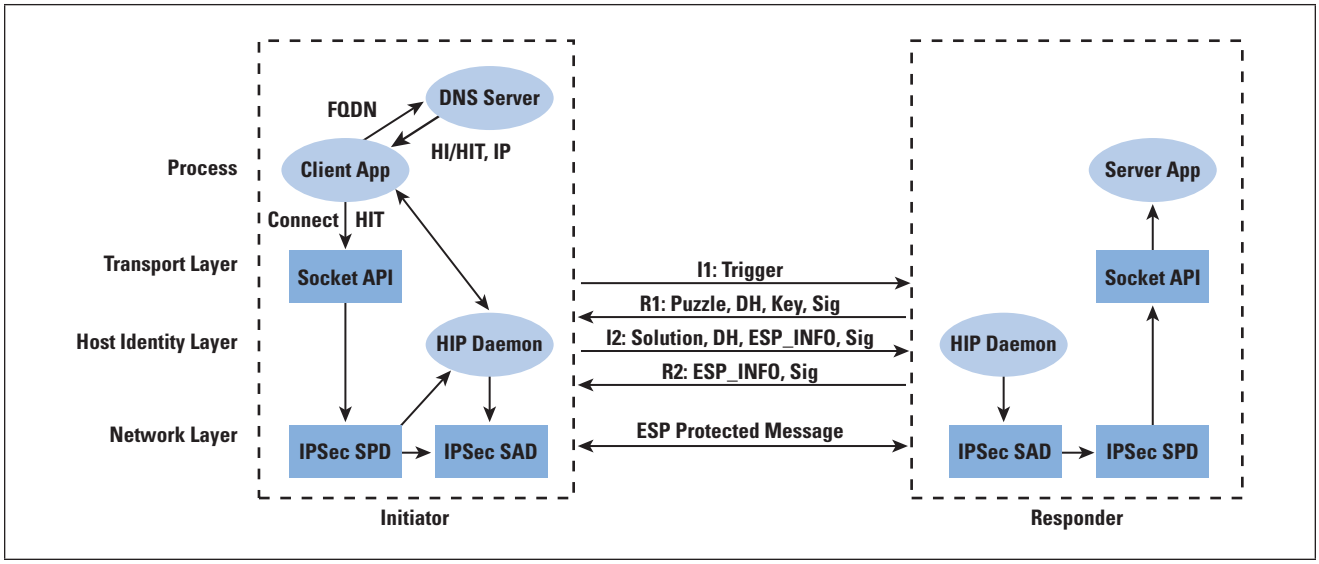
HIP separates the identity of a host from its location. The location of the host is bound to IP addresses and used for routing packets to the host in the same way as in the current Internet architecture. However, transport and application layers use *host identity,* consisting of the public key component of a private-public key pair. Each host is responsible for creating one or more public/private key pairs to provide identities for itself. Because the host identities are based on public key cryptography, they are computationally difficult to forge. Host identities are location-independent identifiers that allow a mobile host to preserve its transport layer connections upon movement. On the other hand, the host identity can be used for looking up the current location of a host because the host identity is a long-term identifier. A client host obtains the host identity of a server typically from the *Domain Name System* (DNS)[7] or a *Distributed Hash Table* (DHT). However, the infrastructure may not support this DHT in certain scenarios, such as in peer-to-peer and temporary environments. In such cases, *opportunistic* HIP can be used for contacting a peer without prior information of the identity of the peer. Opportunistic HIP is based on a "leap-of-faith," meaning that it is prone to man-in-the-middle attacks for the initial connection. It is similar to the *Secure Shell* (SSH) *Protocol,* where the public key of the server is added to the known host list after the first connection.

The problem of certifying the keys in *Public Key Infrastructure* (PKI) or otherwise creating trust relationships between hosts has explicitly been left out of the HIP architecture, because it is expected that each system using HIP may want to address it differently. For mere mobility and multihoming, the systems can work without any explicit trust management, in an opportunistic manner.

All other parties use the host identifier, that is, the *public key,* to identify and authenticate the host. Typically, a host identifier is a 128-bit-long bit string, the *Host Identity Tag* (HIT), as shown in Figure 1. A HIT is constructed by applying a cryptographic hash function over the public key. The introduction of new endpoint identifiers changes the role of IP addresses. When HIP is used, IP addresses become pure topological labels, naming locations in the Internet. One benefit of this identity/locator separation is that hosts in private address realms (behind NATs) can name each other in a unique way with HITs. A second benefit is that the hosts can change their IP address without breaking transport layer connections of applications and rely on HIP to manage host mobility; the relationship between location names and identifiers becomes dynamic.

To start communicating through HIP, two hosts must establish a HIP association. Known as the HIP *Base Exchange* (BEX)[3], this process consists of four messages (I1, R1, I2, and R2) transferred between the initiator and the responder. After BEX is successfully completed, both hosts are confident that private keys corresponding to host identifiers (public keys) are indeed possessed by their peers. Another purpose of the HIP base exchange is to create a pair of IPsec *Encapsulated Security Payload* (ESP) *Security Associations* (SAs), one for each direction. HIP uses IPsec ESP *Bound End-to-End Mode* (BEET)[4,9] to provide data encryption and integrity protection for network applications.

Figure 1: HIP Architecture



Because neither transport layer connections nor security associations created after the HIP base exchange are bound to IP addresses, a mobile client can change its IP address (that is, upon moving, because of a *Dynamic Host Configuration Protocol* [DHCP] lease or IPv6 router advertisement) and continue to transmit ESP-protected packets to its peer. HIP supports such mobility events by implementing an end-to-end three-way UPDATE signaling mechanism[8] between communicating nodes. HIP multihoming uses the same mechanisms as mobility for updating the peer with a current set of host IP addresses.

A rendezvous server[6] provides a mechanism to locate a host, for example, when two communicating hosts move simultaneously. To employ a rendezvous mechanism, a host first must perform a registration procedure[5], which is an extended version of the HIP base exchange.

The HIP control packets as well as ESP-encapsulated data packets have difficulties in going through NAT applications and firewalls. To traverse NAT, HIP uses *User Datagram Protocol* (UDP)-based encapsulation provided by the *Interactive Connectivity Establishment* (ICE) protocol.

It enables two hosts located behind NAT to communicate through a Rendezvous server. Bob Moskowitz suggests an alternative approach, where HIP always uses IPv6 for end-to-end communication and the *Teredo* protocol is employed to traverse NAT instances in IPv4 networks if native IPv6 connectivity is not available.

Most Internet applications can run unmodified over HIP[10], although only HIP-aware (new) applications using the extended socket interface can take better advantage of the new features that HIP provides. As HIP secures application data traffic with IPsec that is located logically "deep" within the networking stack, the challenge is to provide proper and understandable security indicators to the user to convince the user that the connection, for example, to a banking website, is secured. Such indicators can be developed as extensions to applications (for example, a security plug-in to the *Firefox* browser) or within a hostwide HIP management utility that controls all applications.

HIP provides a network layer alternative to using *Secure Sockets Layer/Transport Layer Security* (SSL/TLS) for application security, which has its benefits and drawbacks. HIP is a generic solution that should work for any transport protocol, whereas until recently TLS supported only TCP. HIP enables host mobility and multihoming, which is not supported by TLS. TLS runs on top of TCP, leaving it vulnerable to various TCP attacks; for example, using spoofed *reset* (RST) packets or DoS attacks with SYNs. Applications must be designed explicitly to use TLS, whereas HIP can provide security as an add-on to existing traditional applications. On the other hand, TLS does not have a problem with traversing traditional middleboxes such as NATs and firewalls that need special attention for HIP. Both protocols share the characteristic of endorsing host identity. TLS relies on certificates issued by one of the known Certification Authorities, whereas HIP can use *Domain Name System Security Extensions* (DNSSEC)[18] or a PKI infrastructure.

There are currently three open-source interoperating HIP implementations. *OpenHIP* from Boeing runs on Linux, Windows, and Mac OS, whereas *HIP on Linux* (HIPL) runs on Linux and Symbian, and *HIP for Inter.net* from Ericsson runs on FreeBSD and Linux. Several testbeds are deployed based on HIP, including the Everett Boeing factory[11], the P2PSIP pilot in Finland[14], and Wi-Fi P2P Internet Sharing Architecture in Germany[12]. Ericsson NomadicLab and TeliaSonera have demonstrated using HIP for transparent IPv4 and IPv6 handovers, mobile router, simultaneous multiaccess, and the use of proxy for traditional hosts[13,15].

Acknowledgements

We are grateful to Pekka Nikander, Tom Henderson, and others in the IETF and the *Internet Research Task Force* (IRTF) community who were encouraging and contributing to the development of HIP. We thank Andrey Khurri for the figure on HIP architecture and Henry Sinnreich for encouraging us to write this article.

We also thank members of InfraHIP II project for comments helping to improve this article.

References

[1] Moskowitz, R. and Nikander, P., "Host Identity Protocol Architecture," RFC 4423, May 2006.

[2] Gurtov, A., *Host Identity Protocol (HIP): Towards the Secure Mobile Internet,* ISBN 978-0-470-99790-1, Wiley and Sons, June 2008.

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

[4] Jokela, P., Moskowitz, R. and Nikander, P., "Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP)," RFC 5202, April 2008.

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

[6] Laganier, J. and Eggert, L., "Host Identity Protocol (HIP) Rendezvous Extension," RFC 5204, April 2008.

[7] Nikander, P. and Laganier, J., "Host Identity Protocol (HIP) Domain Name System (DNS) Extension," RFC 5205, April 2008.

[8] Nikander, P., Henderson, T., Vogt, C. and Arkko, J. "End-host Mobility and Multihoming with the Host Identity Protocol," RFC 5206, April 2008.

[9] Nikander, P. and Melen, J., "A Bound End-to-End Tunnel (BEET) Mode for ESP," Internet Draft, Work in Progress, **draft-nikander-esp-beet-mode-09**

[10] Henderson, T., Nikander, P. and Komu, M., "Using the Host Identity Protocol with Legacy Applications," RFC 5338, September 2008.

[11] Boeing, "Secure Mobile Architecture (SMA) for Automation Security," **http://www.isa.org/wsummit/presentations/Boeing-NGI_SMA_Automation_Security_Vancouver_ISA_presentationtemplates_7-23-07.ppt**

[12] Heer, T., Götz, S., Weingärtner, E. and Wehrle, K., "Secure Wi-Fi Sharing on Global Scales," in Proceedings of the 15th International Conference on Telecommunication (ICT), St. Petersburg, Russian Federation, IEEE, 2008. **https://www.ds-group.info/members/heer/publications-tobias-heer/pdfs/HeerEtAl2008.pdf**

[13] Jokela, P., Ylitalo, J., and Salmela, P., "HIP Mobile Router Demo," March 2007.
http://www.ietf.org/proceedings/07mar/slides/HIPRG-3.pdf

[14] Koskela, J., Heikkila, J. and Gurtov, A., "A Secure P2PSIP System with SPAM Prevention," Poster at ACM Mobicom, September 2008.

[15] Korhonen, J., Mäkelä, A., and Rinta-aho, T., "HIP Based Network Access Protocol in Operator Network Deployments," in First Ambient Networks Workshop on Mobility, Multiaccess, and Network Management (M2NM'07), Sydney, Australia, October 2007.

[16] Meyer, D., "The Locator Identifier Separation Protocol (LISP)," *The Internet Protocol Journal,* Volume 11, No. 1, March 2008.

[17] Saltzer J., "On The Naming and Binding of Network Destinations," RFC 1498, September 1992.

[18] Gieben, M., "DNSSEC: The Protocol, Deployment, and a Bit of Development," *The Internet Protocol Journal,* Volume 7, No. 2, June 2004.

[19] Sinnreich, H., "Letter to the Editor," *The Internet Protocol Journal,* Volume 11, No. 3, page 37, September 2008.

ANDREI GURTOV received M.Sc and Ph.D. degrees in Computer Science from the University of Helsinki, Finland. He presently is Principal Scientist, leading the Networking Research group at the Helsinki Institute for Information Technology, focusing on distributed system security and next-generation Internet architecture. He co-chairs the IRTF research group on HIP and teaches as an adjunct professor at Helsinki University of Technology. He is a regular visitor of the ICSI Center for Internet Research (ICIR) at Berkeley. Andrei has co-authored more than 50 publications, including a book, research papers, patents, and RFCs. He can be reached through the webpage: **http://www.hiit.fi/~gurtov**

MIIKA KOMU received his M.Sc. from Helsinki University of Technology and continues his studies as a postgraduate student. He is working as a full-time researcher and software engineer at Helsinki Institute for Information Technology. He is an active IETF participant and co-author of RFC 5338. Miika is an open source advocate and martial arts fan. E-mail: **miika.komu@hiit.fi**

ROBERT MOSKOWITZ is senior technical director for ICSA Labs and is an active member in the IAB, IETF, and IEEE. At ICSA Labs, Moskowitz leads the IPsec product and system certification program. Prior to the ICSA, he led the adoption of the world's largest IPsec network deployment servicing the automotive industry. As a former co-chair of the IPsec Working Group, Moskowitz provided a user set of multivendor, multipolicy, and multiuser requirements that galvanized many of the debates on the use of IPsec. A contributing editor for *Network Computing Magazine,* Moskowitz is currently helping define the new security component for the 802.11 standard. E-mail: **rgm@htt-consult.com**