Enterprise Network Packet Filtering for Mobile Cryptographic Identities

Janne Lindqvist and Essi Vehmersalo
Helsinki University of Technology

Miika Komu
Helsinki Insitute of Information Technology

Jukka Manner
Helsinki University of Technology
{janne.lindqvist,essi.vehmersalo,jukka.manner}@tkk.fi miika@iki.fi

1 Introduction

Firewalls are, unfortunately, a critical component of corporate, and personal, networks in the Internet today. Packet filtering is typically based on the 5-tuple of sender and receiver IP addresses and port numbers, and the transport protocol. Sophisticated firewalls can also filter based on the content of application layer protocols. Commonly the filtering rules are quite static, certain services, and a known set of hosts are allowed to pass through the firewall. In more dynamic networks, for example, offering public or subscription-based WLAN access, or nomadic enterprise environments, the firewalls are controlled and rules set up based on some authentication exchange. Typically, a client is authenticated and authorized to use a WLAN service based on a web browser login application. If the login is successful, the firewall opens pre-defined services for the client device MAC and IP address, and the client can start using the Internet, for example, browse the web, or initiate VPN connections.

The current situation has a number of downsides. First, authentication for network access has a number of different implementation choices, which may or may not work with the device of the user, for example, laptop computers, PDAs, or smart mobile phones. Second, even when the client is authenticated and authorized to use the Internet, the exact services allowed are pre-defined. We would need a second separate signaling protocol to dynamically manage the filtering rules associated with a given authenticated client. Third, a third party can still listen to the network communications, collect varying information, and steal the identity of an authenticated client. Fourth, network renumbering becomes a problem, because all static rules on firewalls that are based on IP address must be changed when renumbering occurs. The same problem of updating firewall rules appears in access networks, where the IP address assigned to a client can change during the session, for example, in a mobile access network when the client performs a handover.

The previous issues affect current network firewall management from the inside of the network. Roaming clients of a network, for example, traveling employees of a company, cause additional concerns. A roaming client would need to use the services of his or her home network. Since the firewall cannot know the IP addresses of roaming employees, the rules that protect the network services must be quite liberal, or access is only possible through a separate VPN tunnel.

In this abstract, we present a new firewall architecture [1] that allows efficient, scalable and secure network packet filtering. Our solution solves all the problems discussed above. The firewall is based on the Host Identity Protocol (HIP) [2] architecture and tracking the protocol control messages and IPsec ESP SPI values. Although the standard IPsec architecture could be used to implement firewalls, our architecture provides a simple way to centrally enforce security policies regardless of host IPsec security policies. The HIP architecture introduces a new namespace that consists of Host Identifiers (HI) that are public keys of public and private key pairs. For practical purposes, the public keys - the Host Identifiers - are represented by self-certifying hashes of the keys. The hash is called the Host Identity Tag (HIT). The advantage of using HITs instead of HIs, is that HITs are same size as IPv6 addresses and are compatible with current applications. For example, the HITs can be used to replace IPv6 address fields in other protocols and Applications Programming Interface (API) functions.

One of the key features of using HIP and a HIP-enabled firewall is that the administration of the network does not need to care about IP addresses. Thus, the network can perform renumbering, and support mobile users without changes in the firewall rules. Moreover, when the client is using HIP, it does not need to employ any additional protocol for authentication and firewall control, either inside or outside the enterprise network. Furthermore, the solution also allows encrypting the data transfer end-to-end.

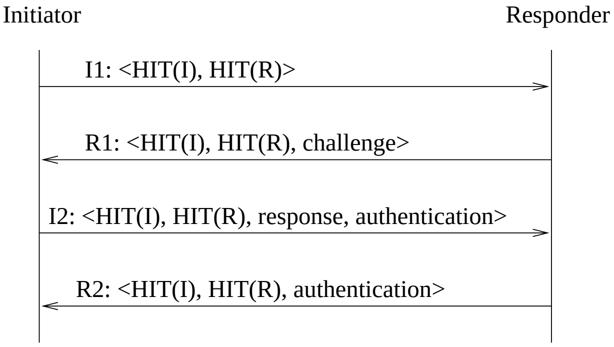
Initiator
Responder
I1: <HIT(I), HIT(R)>
R1: <HIT(I), HIT(R), challenge>
I2: <HIT(I), HIT(R), response, authentication>
R2: <HIT(I), HIT(R), authentication>

Figure 1: HIP base exchange

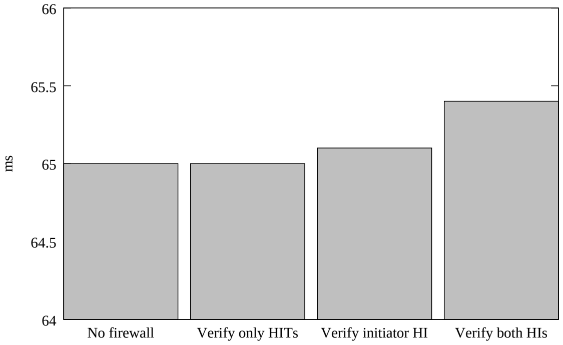

Figure 2: TCP connection establishment time

The firewall solution introduced in this abstract does not require Internet-wide deployment of HIP. An enterprise can deploy HIP gradually to harness the integrated security, mobility, and multihoming capabilities for employees. Services and clients that do not use HIP continue to operate with the old system.

2 Implementation

The firewall uses HITs as access control list identifiers, but also some other properties of network packets can be used in the firewall rules. When a connection is initiated (figure 1), the firewall verifies that the HITs of an I1 message match the access control list and it records the HITs and IP addresses of the Initiator and Responder. It is trivial for an attacker to forge these HITs, since there are no signatures to be verified at this stage. The I1 does not contain any signature, which means that the not firewall, nor the responder, can verify its authenticity. Therefore, a forged I1 can reach the responder through the firewall. However, a connection cannot be established because a verified and completed base exchange is required before data traffic is allowed into the network.

The responder sends an R1 and the firewall checks the HITs from its ACLs. This can be used to enforce access control restrictions to the Responders behind the firewall. The firewall records the HITs of the Initiator and the Responder and their IP addresses from the R1.

Upon receiving the R1, the Initiator solves the puzzle and sends an I2 packet. The I2 contains a public key and a signature calculated using the private key of the Initiator. The firewall can verify the signature either using the public key from the packet or a preconfigured public key. If the verification fails, the firewall discards the packet. Similarly, the firewall checks the response, R2, from the responder. The I2 and R2 messages contain the IPsec ESP SPI values that the firewall needs to establish state to track ESP traffic. Similarly, the firewall uses a message with the LOCATOR parameter to continue the tracking of IPsec ESP flows upon end-host IP address changing handovers. Further, the SPI state expires when there is no traffic for a certain time period. This guarantees that the state is removed when a mobile node disappears, for example, moves further away or shuts down.

We conducted a set of measurements with the firewall to get a rough idea of its performance. The evaluation environment consisted of a server and five clients. The clients were within a single network separated from the server with a combined router and firewall. All of the hosts had a single Pentium 4 processor (3 Ghz) and the IPv6 network operated at 100 Mbit speed. The Linux kernel version was 2.6.17.3. We used the 1024 bit RSA as asymmetric keys. The symmetric keys were AES (128 bits) for HIP encryption, SHA1 (160 bits) for IPsec authentication and 3DES (192 bits) for IPsec encryption.

We measured the time observed by an application to complete UNIX connect() system call, which establishes first a TCP handshake. This time was under 1 ms on the average without HIP. With HIP, the time was roughly 65 ms, independently whether the HIP firewall was active or not. In addition, the use of initiator and responder signatures caused an extra delay of 1 ms at the maximum. The signature check was fast because the verification is quite fast in RSA. The TCP connection times are illustrated in Figure 2.

The full version of this work is available as [1].

References

[1] J. Lindqvist, E. Vehmersalo, M. Komu, and J. Manner. Enterprise Network Packet Filtering for Mobile Cryptographic Identities. Telecommunications Software and Multimedia Laboratory Research Report: TML-B8, Helsinki University of Technology, June 2007.

[2] R. Moskowitz and P. Nikander. Host identity protocol architecture. RFC 4423, IETF, May 2006.