Mitigation of Unsolicited Traffic across Domains with Host Identities and Puzzles

Miika Komu[1], Sasu Tarkoma[2], and Andrey Lukyanenko[1]

[1] Aalto University
[2] University of Helsinki

Abstract. In this paper, we present a general host identity-based technique for mitigating unsolicited traffic across different domains. We propose to tackle unwanted traffic by using a cross-layer technique based on the Host Identity Protocol (HIP). HIP authenticates traffic between two communicating end-points and its computational puzzle introduces a cost to misbehaving hosts. We present a theoretical framework for investigating scalability and effectiveness of the proposal, and also describe practical experiences with a HIP implementation. We focus on email spam prevention as our use case and how to integrate HIP into SMTP server software. The analytical investigation indicates that this mechanism may be used to effectively throttle spam by selecting a reasonably complex puzzle.

1 Introduction

One challenge with the current Internet architecture is that it costs very little to send packets. Indeed, many proposals attempt to introduce a cost to unwanted messages and sessions in order to cripple spammers' and malicious entities' ability to send unsolicited traffic. From the network administration viewpoint, spam and DoS traffic comes in two flavors, *inbound* and *outbound* traffic. Inbound traffic originates from a foreign network and outbound traffic is sent to a foreign network. Typically, spam and packet floods originate from networks infested with zombie machines. A *zombie* machine is a host that has been taken over by spammers or persons working for spammers, e.g., using Trojans or viruses.

We address the problem of unsolicited network traffic. We use two properties unique to the *Host Identity Protocol (HIP)* protocol: First, hosts are authenticated with their public keys which can be used for identifying well-behaving SMTP servers. Second, a computational puzzle introduces a cost to misbehaving hosts. Our approach has a *cross-layer* nature because a lower-layer security protocol is used to the benefit of higher-layer protocols.

2 Host Identity Protocol

The Host Identity Protocol (HIP) [9] addresses mobility, multi-homing, and security issues in the current Internet architecture. HIP requires a new layer in

T. Aura, K. Järvinen, and K. Nyberg (Eds.): NordSec 2010, LNCS 7127, pp. 33–48, 2012.

the networking stack, logically located between the network and transport layers, and provides a new, cryptographic namespace. HIP is based on *identifier-locator split* which separates the *identifier* and *locator* of an Internet host. The identifier uniquely names the host in a cryptographic namespace, and the locator defines a topological location of the node. Communication end points are identified using public cryptographic keys instead of IP addresses. The public keys used for HIP are called *Host Identifiers (HIs)* and each host generates at least one HI for itself.

The HIs can be published as separate HIP-specific records in the DNS [11]. Legacy applications can use HIP transparently without any changes. Typically, the application calls the system resolver to query the DNS to map the host name to its corresponding address. If a HIP record for the host name does not exist, the resolver returns a routable IPv4 or IPv6 address. Otherwise, the resolver returns a Host Identifier fitted into an IPv4 or IPv6 address. *Local-Scope Identifier (LSI)* is a virtual IPv4 address assigned locally by the host and it refers to the corresponding HI. *Host Identity Tag (HIT)* is an IPv6 address derived directly from the HI by hashing and concatenation. An LSI is valid only in the local context of the host whereas a HIT is statistically globally unique.

When an application uses HIP-based identifiers for transport-layer communications, the underlying HIP layer is invoked to authenticate the communication end-points. This process is called the *base exchange*, during which the end points authenticate to each other using their public keys. The host starting the base exchange, the initiator, is typically a client, and the other host, the responder, is typically a server. During the base exchange, the initiator has to use a number of CPU cycles to solve a computational puzzle. The responder can increase the computational difficulty of the puzzle to throttle new incoming HIP sessions. Upon successful completion, both end-hosts create a session state called *HIP association*.

The base exchange negotiates an end-to-end tunnel to encapsulate the consecutive transport-layer traffic between the two communicating end-hosts. The tunnel is required because routers would otherwise discard traffic using virtual, non-routable identifiers. Optionally, the tunnel also protects transport-layer traffic using a shared key generated during the base exchange. By default, the tunnel is based on IPsec [7] but S-RTP [14] can be used as well. It should be noted that a single tunnel can encompass multiple transport-layer connections.

With HIP, transport-layer connections become more resilient against IP address changes because the application and transport layers are bound to the location-independent virtual identifiers, HITs or LSIs. The HIP layer handles IP-address changes transparently from the upper layer using the *UPDATE* procedure [10]. In the first step of the procedure, the end host sends all of its locators to its connected peers. Then, the peers initiate so called *return routability test* to protect against packet-replay attacks, i.e., to make sure that the peer locator is correct. In the test, each node sends a nonce addressed to each of the received peer locators. The peer completes the test by signing each nonce and echoing