SAVAH: Source Address Validation with Host Identity Protocol

Dmitriy Kuptsov and Andrei Gurtov

Helsinki Institute for Information Technology
Helsinki University of Technology
{dmitriy.kuptsov,gurtov}@hiit.fi

Abstract. Explosive growth of the Internet and lack of mechanisms that validate the authenticity of a packet source produced serious security and accounting issues. In this paper, we propose validating source addresses in LAN using Host Identity Protocol (HIP) deployed in a first-hop router. Compared to alternative solutions such as CGA, our approach is suitable both for IPv4 and IPv6. We have implemented SAVAH in Wi-Fi access points and evaluated its overhead for clients and the first-hop router.

Keywords: Security, Authentication, LAN, HIP.

1 Introduction

Routing of packets in the Internet is based on the destination IP address. It is hard to identify the source of the packet. Attackers can easily spoof the source IP address of a packet; hosts under Denial-of-Service (DoS) attacks cannot trace the originators. Another negative result can be a blocked service for a given source address, which can be the address of a good host compromised by an attacker.

These issues forced the source address validation to become an urgent problem in networking research. Several existing solutions are based on cryptographic authentication, traceback, and filtering. In this paper, we propose a solution called *Source Address Validation Architecture with Host Identity Protocol (HIP)* (SAVAH), which involves cryptographic authentication and filtering based on the host identifiers. This method integrates with a source address validation architecture (SAVA) [1], and acts as an alternative to proposed method of validating source addresses on the edge (or first-hop) router using Cryptographically Generated Addresses (CGA).

Host Identity Protocol (HIP) is a new security protocol which integrates host locator/identifier split, mobility, and multihoming. It was specified by IETF [10,12,11,8,9,14,13,16]. Possessing important properties such as a secure and efficient session key negotiation and self generated host identifiers for authentication, HIP can help to solve the problem of source address validation.

We believe that SAVAH can be deployed in a large-scale network and integrated with existing solutions deployed on different level of granularity. In

A.U. Schmidt and S. Lian (Eds.): MobiSec 2009, LNICST 17, pp. 190–201, 2009.

general, we think that HIP-based source address validation can become a replacement for a CGA approach [1]. We also think that it provides more security than any of the existing solutions for the source address validation in a local network, as it relies on a cryptographically secure protocol.

The paper is organized as follows. Section 2 gives an overview of Host Identity Protocol. Section 3 summarizes related work. In Section 4, design and implementation of SAVAH is described. In Section 5, we discuss integration of our proposal with general SAVA architecture. Performance measurements are given in Section 6 and Section 7 concludes the paper.

2 Background on Host Identity Protocol

The existing Internet architecture was designed for stationary hosts and faces many non-trivial challenges today with the growing number of mobile terminals. Currently, there are two namespaces used globally by the Internet services and applications, domain names and IP addresses. IP addresses serve the dual role in the Internet being both end host identifiers and topological locators. This general principle does not allow hosts to change their location without breaking ongoing transport protocol connections that are strictly bound to IP addresses.

The Host Identity Protocol (HIP) [8,6] was proposed to overcome the problem of using IP addresses for host identification and routing. The idea behind HIP is based on decoupling the network layer from the higher layers in the protocol stack architecture (see Figure 1).

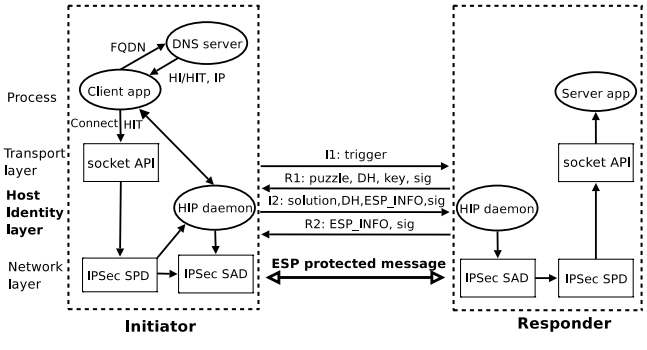

Fig. 1. HIP architecture

HIP defines a new global name space, the Host Identity name space, thereby splitting the double meaning of IP addresses. When HIP is used, upper layers do not anymore rely on IP addresses as host names. Instead, Host Identities are used by the transport protocol for establishing connections. IP addresses at the same time act purely as locators for routing packets towards the destination. For compatibility with IPv6 legacy applications, Host Identity is represented by a 128-bit long hash, the Host Identity Tag (HIT).

HIP offers several benefits including end-to-end security, resistance to CPU and memory exhausting denial-of-service (DoS) attacks, NAT traversal, mobility and multihoming support.

3 Related Work

There are three main methods for source address validation: cryptographic authentication, ingress/egress packet filtering, and various traceback techniques.

Cryptographic authentication appears a promising solution which brings strong security properties for authenticating originator of the network communication. IPSec [7] is one example, which allows secure end-to-end communication. Wu et al. [15] point out that IPSec depends on global deployment of PKI infrastructure. HIP [6,8] in combination with HIP-enabled firewall, in turn, can provide basic functionality for source address authentication and validation. One possibility will be covered in Section 4.

Another approach is SPM [5] designed for authorization of neighboring autonomous systems (AS). This approach provides solution for source address authorization and relies on cryptographic properties.

Filtering of malicious packets containing wrong source addresses can be considered a simple solution. Unlike solutions based on cryptographic properties, it is straightforward to deploy, but less secure approach. Examples can be ingress filtering, SAVE protocol [4] and HCF filtering [3].

4 SAVAH Design and Implementation

We considered three possible approaches when designing the source address validation mechanism using HIP.

1. First, a so-called pure HIP communication with HIP firewall in-between. This requires that all communicating peers support HIP. Then the HIP firewall (which should be deployed on an edge of a local network) tracks base exchange signaling packets from all hosts that try to communicate with the hosts outside the local network and only lets through the packets with valid HITs and IP addresses. Later, data is sent in ESP encapsulated packets so that the firewall can check the SPI values of the packets and drop those that do not match a previously seen base exchange. This approach requires a large-scale deployment of HIP protocol.
2. Secondly, we assume that hosts inside the local network support HIP with our extension. Then the HIP base exchange can replace a CGA approach [15]. This case is the main focus of the article and will be discussed in detail.
3. Finally, HIP tunneling approach can be used. It is less efficient in performance, but can provide better security and mobility support to wireless clients. The client creates a tunnel between itself and the SAVAH router using HIP base exchange and SAVAH extension. Later all traffic is forwarded through this tunnel to the Internet. There were previous studies on tunneling the traffic to home router in PISA [2].

4.1 SAVAH Architecture

SAVAH targets to solve the source address validation problem in the edge router of the local network serving as a default gateway. SAVAH architecture is composed of two main components:

- A SAVAH-enabled client, which is a combination of a HIP daemon and a firewall in a client mode supporting SAVAH extension.
- A SAVAH-enabled router running the HIP daemon and the firewall but in a server mode.

DHCP server can be considered the third component in our architecture. Its main role in the network is to offer particular configuration for SAVAH aware hosts. For instance, DHCP can provide the default gateway IP address as well as a HIT of the SAVAH router. Availability of a proper configured DHCP server can help to solve the problem of opportunistic mode as discussed later in this section. However, we consider DHCP as an optional component in the network.

SAVAH aware clients can be configured manually, meaning that the IP address and the HIT of the SAVAH router can be setup by a system administrator prior to any communication. However, manual configuration can be a tedious task in a large scale network. As the third option, the SAVAH-enabled client can discover the SAVAH router using the opportunistic mode.

The message sequence diagram for SAVAH registration and further authentication process is shown in Figure 2. It involves three entities in the local network to perform the source address validation: the SAVAH client, the SAVAH router, and the DHCP server (optionally). The receiver in Figure 2 is playing the role of a legacy peer, i.e., it may or may not support HIP. Of course, if both communicating peers support HIP, whole scenario requires only a normal HIP-enabled firewall to filter the traffic based on HITs.

Since the SAVAH router is the first-hop router, it should be placed on the edge of the local network and serve as a default gateway. If the default gateway and the SAVAH router are not placed physically on the same node it is meaningless, because all network traffic flowing through the SAVAH-unaware router would not contain the SAVAH option and would be discarded.

4.2 SAVAH Router Discovery

Unless the DHCP server offers a HIT of the SAVAH router during the address assignment phase, the SAVAH client is obliged to discover the presence of the SAVAH service automatically. Otherwise, the client is aware of the HIT and the IP address of the SAVAH router and can register to the service directly. It is also possible to preconfigure the client and specify the IP address and HIT of the default gateway.

Optionally, the presence of a SAVAH-unaware DHCP means that the client will obtain only the IP address of the default gateway. No prior knowledge of the SAVAH router's HIT forces the client to discover the service using the HIP

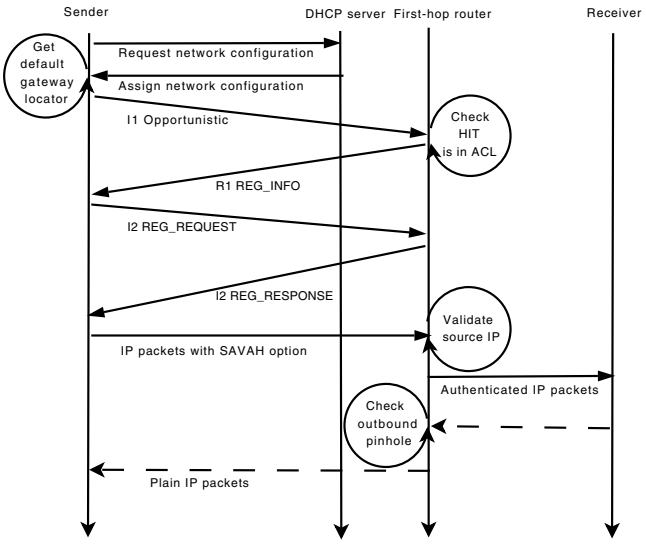

Fig. 2. SAVAH service registration and authentication process

opportunistic mode [6] and a set of usual procedures for HIP service registration [11].

The drawback of such broadcasting is that in the opportunistic mode the client is unaware of the HIT of SAVAH router and any node can thus pretend to be a valid router. This resembles a similar problem with SSH when connecting to unknown hosts. Hence, the opportunistic mode should be used only in a trusted environment.

To trigger a registration in opportunistic mode, the SAVAH client requests from the system the default gateway IP address and sends an I1 packet with the destination HIT as a hashed source IP address. On the other side, the default gateway running SAVAH in a server mode responds with an R1 packet containing an offer for available services in REG_INFO parameter.

Upon receiving the R1 packet the client chooses the supported services and responds with an I2 packet containing a REG_REQUEST parameter to SAVAH server. If REG_INFO parameter does not contain the SAVAH service offer, the client completes base exchange normally and afterward falls back to normal communication, i.e., the SAVAH mode is not supported in this network.

Finally, depending on the setup of the SAVAH router, it either grants or denies the service to a client in an R2 packet with a REG_RESPONSE or REG_FAILED parameter. Receiving a REG_FAILED parameter in an R2 message during the base exchange or experiencing a timeout in an I1 state means that either the default gateway does not support SAVAH extension or that HIP daemon with SAVAH extension is not running at all.

This situation should indicate the SAVAH client to fallback to normal communication, i.e., the packets to be forwarded through the default gateway would not

contain any authentic information. As an opposite result, if the SAVAH router grants the service to the registering client, both parties will posses a shared secret key.

4.3 Packet Authentication

For performance reasons, the keys in use for authentication (i.e., HMAC keys) in IPSec were selected to authenticate the source addresses. Depending on a setup, symmetric cryptography can be selected as well.

Filtering on HITs can be applied to ensure that the peer trying to register to the SAVAH service is legitimate. This filtering can enforce to either grant or deny the SAVAH service to the registrars. Firewall rules in the HIP-enabled firewall control such decisions. By default, all packets with an unknown identifier are dropped.

After completion of the HIP base exchange, the SAVAH router adds the source IP address of the host to a database. This works if no record with such IP address were already present in the database. If a record with such IP address already exists, it is likely that the host is trying to spoof someone else address and the packet should be dropped nor any state added. Moreover, this incident can be logged and reported for further analysis.

If the host experiences an address change, then the record is replaced with a new IP address. To ensure the validity of the host, a HIP UPDATE packet has to be received and handled properly. This will guarantee that the host indeed the one it is claiming to be. The record should be removed from the database upon a timeout or when the host removes a security association (e.g., a HIP CLOSE packet is received from the corresponding host). To ensure that the client is alive, the SAVAH router can also send heartbeats to the client without waiting for the timeout.

After the secret keys are established by means of the HIP base exchange, the SAVAH client may communicate with the nodes outside of the local network by including an authenticated source IP address in each packet. Current implementation has two options to deliver the authenticated hash value to the router. The first approach is to replace the original source IP address of each packet with truncated result obtained from $HMAC(key|\{P\})$ operation, where $\{P\}$ is the packet to be transmitted. We have chosen the packet value as a feed to the HMAC function to introduce a simple protection mechanism from replay attacks.

This approach has some drawbacks. First of all, for IPv4 networks the size of authentication value will be only 32 bits. Secondly, that would decrease the performance since additional address translation would be required on the router. Finally, that method will require additional changes to the router to recognize locally routed traffic.

A different way to carry the authenticated source IP address to SAVAH router is to store it in an IP option. Since the SAVAH router is the first-hop router and all SAVAH related options are striped out on the forward direction, this ensures that no modifications are required for other routers on the path. Placing the authentication value in the IP option can have certain advantages. First of all,

this approach slightly optimizes the performance of the architecture. Secondly, stronger security can be achieved by increasing the length of the authentication value. We suggest to keep this length within wise bounds, since it affects the size of the actual payload.

4.4 Source Address Validation

To authenticate the packet source address, the following algorithm is used. On the router side, each packet in forward direction is checked for the SAVAH IP option. If found, the router compares the value of the option against truncated 128-bit result from $HMAC(key|\{P\})$ operation, where $\{P\}$ is the packet to be forwarded. If matches, the SAVAH option is striped out and the packet is re-injected to the network. If not, the pinhole is searched for a given source and destination IP addresses. If the pinhole is found, the packet is said to be an inbound packet for previously authenticated outbound communication. Finally, if both fail the packet is dropped.

If the tunneling approach is used, then the authentication succeeds if the SAVAH router can successfully decrypt the ESP packet and resend encapsulated in ESP original packet to its final destination. Unlike in the lightweight approach for inbound traffic, the SAVAH router in a tunnel mode should properly encrypt and tunnel the packet to the corresponding mobile node.

Storing a mapping between the identity and the source IP address of each accessing client on the SAVAH router, allows to keep track for the duplicate and spoofed IP addresses, and account each packet traversing the router.

Whether the network access is controlled by the ACL or not, spoofing of the source IP addresses is eliminated as there can be only one mapping between one particular HIT and the IP address. The mapping can be updated or removed once the corresponding UPDATE or CLOSE packet arrives and validated by the SAVAH router. Even when no ACL is maintained, the client cannot forge the source IP address because for each newly self-assigned IP address and/or generated HIT the user needs to complete a four-way HIP handshake with the SAVAH router.

If the address is spoofed with an address of the same subnetwork, the SAVAH router will detect a duplicate address use (because of a locally stored mapping table), and will prohibit the registration to the service. As an additional security action, such activity can be logged and reported. Moreover, non-repudiation property of HIP allows to take a counter measure against such users. If the client tries to spoof the IP address with an address of a network other than the local, such packets should be filtered out with ingress filtering deployed on the SAVAH router. Hence, spoofed packets will be dropped and the user will be logged as malicious and banned from using the network.

For the reasons discussed in the previous paragraph, several address spoofing and denial-of-service attacks are eliminated. For instance, amplifying attacks including various redirect attacks (e.g., DNS redirect, smurf, fraggle, etc.), DoS attacks (those that rely on spoofing the source address), and SYN flood attacks

(since most of the SYN flood attacks require to generate TCP SYN packets with a spoofed source address) become hard to launch.

5 Deployment and Integration with General SAVA Architecture

We are planning to pilot our architecture in a large-scale IPv6 network in Tsinghua University and CERNET2 in China. Our source address validation is placed on the border between local network and the first autonomous system. Instead of CGA mechanism for source address validation [1], HIP with SAVA extension is used as shown in Figure 3.

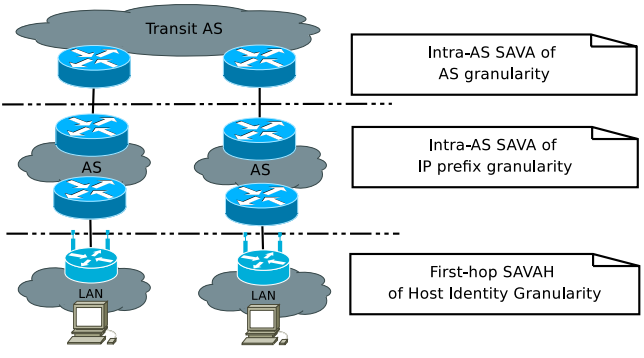

Fig. 3. Integrating SAVAH with inter-AS source address validation

The key strength of our proposal is that in addition to basic source address validation our implementation adds support for mobility, multihoming, data integrity and encryption. Another advantage is that accounting can be maintained easily since the network access is controlled based on valid registered HITs. Although this is not necessary to validate the source address, it can be considered as a plus in large-scale networks. However, as with CGA mechanism, our architecture requires modifications to hosts residing in the local network to pass the authentication procedure.

Figure 4 shows how SAVAH can be deployed and used in a public wireless network, for instance in PanOULU [18] or Tsinghua University campus network, to validate, authenticate, and account network traffic. The figure describes possible step-by-step procedure for adding an unknown HIT to ACL:

1. First, a client tries to register with a SAVAH router using the procedure described in Section 4.
2. In case the HIP firewall does not find the client HIT in the ACL, it drops all packets except for HTTP traffic which is redirected to the SAVAH registration HTTP server. Otherwise, the client successfully completes the HIP base exchange and is permitted to use the network.

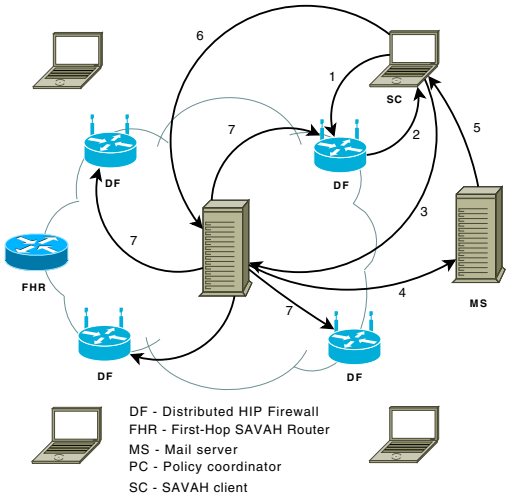

Fig. 4. A HIT registration procedure

3. A web registration form is sent to the client browser. The client provides his email and HIT and submits the form. As an alternative the client can identify itself by some other mechanism, for instance using Internet banking.
4. The SAVAH registration server checks if the submitted email belongs to the list of allowed domains (pre-configured database). It generates the authentication link and sends it to the given email address.
5. The client retrieves the email.
6. The client clicks the link in the email and thus authenticates the previously submitted HIT.
7. The authenticated HIT is distributed to all SAVAH routers and HIP firewalls.

6 Performance Evaluation

In the experiential setup, we used a wireless access point and a laptop. The wireless access point was running a modified version of OpenWRT Linux distribution [17], a HIP daemon, and a firewall in a SAVAH server mode. The laptop was running Ubuntu Linux and HIP in a SAVAH client mode.

We assume that our network is trusted, hence we use an opportunistic mode to discover the SAVAH router presence in the network. Hardware characteristics of the SAVAH router and client are shown in Table 1. Our testbed supports both IPv4 and IPv6 network stacks and we can evaluate SAVAH extension in both configurations.

We performed measurements of packet processing time in the router and the client. Collected results are shown in Figure 5 and Figure 6 for the router and client correspondingly. The expected time for packet processing in both ends

Table 1. Test hardware

Parameter	First-hop router	Laptop
CPU	533 MHz	Dual core 2.4 GHz
RAM	128 MB	3GB
Wireless	Atheros BG	Intel AGN

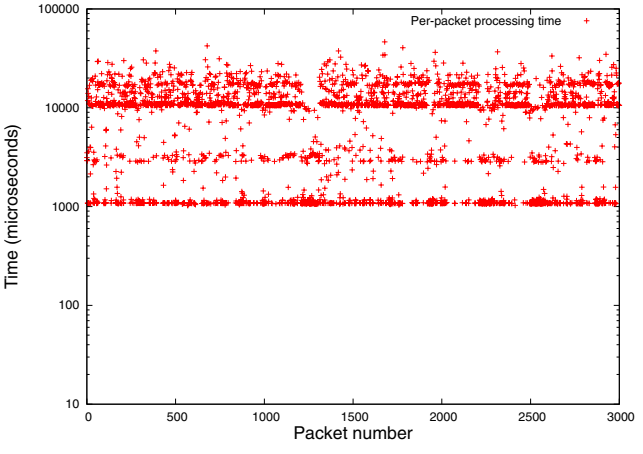

Fig. 5. SAVAH packet processing time in the first-hop router (Avila board)

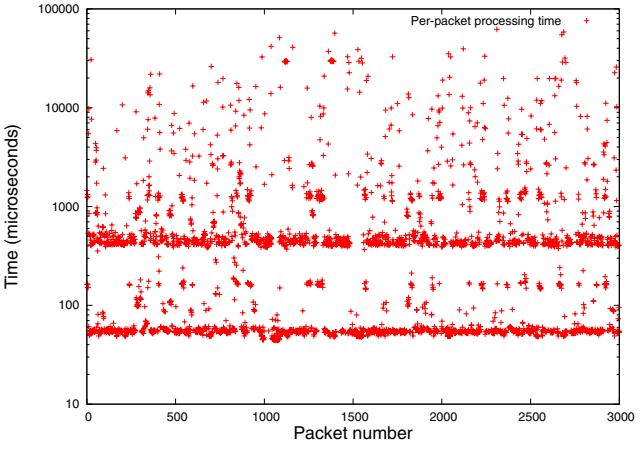

Fig. 6. SAVAH packet processing time in a client (laptop)

(i.e., for router and client) is $9ms$ and $1ms$. In theory, these results should allow to process on the average 110 packets per second in the router and around 1000 packets in the client. Hence, we estimate throughput of 1.3Mbps and 13Mbps in the router and in the client (assuming the Ethernet MTU 1500B).

Such performance is not sufficient for a router on the edge of a network with heavy traffic. Since our implementation is only a proof-of-concept, its optimization should help to overcome this hurdle. On the other hand, implementation can be faster in using RSA or DSA signatures [2]. For instance, the time required to check one DSA signature was about 63.7 *ms*. However, by compiling the prototype without debugging information and with several optimizations, the performance can be increased by $150 - 200\%$.

The memory costs are insignificant in the SAVAH mode. We have tested SAVAH extension during long period of time by streaming video files. We have noticed that memory usage on the SAVAH router stayed below 26MB boundary. On a device such as the Avila board (used as a router), this is not significant.

The CPU usage showed 100% load when heavy traffic is streamed through the router. The scheduler releases all CPU cycles to a demanding application. In our case, the HIP daemon and firewall were the only applications in an active state. Thus, this does not necessarily leads to a decrease in performance of other applications. Instead, 100% usage of CPU tells that the application is a resource hungry. However, memory copy operations required to strip out the SAVAH option are currently inefficient and can be optimized. On the other hand, calculating HMAC does stress the CPU and a packet is processed faster than if any other (symmetric or asymmetric) cryptography would have been used.

7 Conclusions

In this paper, we proposed a secure mechanism for validating source addresses and authenticating hosts in the local network. In SAVAH, local hosts use the extended Host Identity Protocol (HIP) to connect to legacy Internet hosts through a first-hop router which authenticates users.

SAVAH architecture involves three network entities, a network configuration server (DHCP), a client, and a router supporting extended HIP for host authenticating and source address validation. We implemented the system by re-flashing firmware of a wireless access point with OpenWRT Linux and HIP protocol supporting SAVAH extension.

Performance evaluation showed that SAVAH wireless access points can validate source addresses and provide reasonable protection against address spoofing. In addition, SAVAH offers stronger cryptographic properties including authentication, authorization, accountability, non-repudiation, and consistency. Our approach enables host mobility and multihoming. In contrast to CGA-based approach, SAVAH supports both IPv4 and IPv6, and is a patent-free technology.

References

1. Wu, J., Ren, G., Li, X.: Source Address Validation: Architecture and Protocol Design. In: IEEE International Conference on Network Protocols, pp. 276–283. IEEE Computer Society Press, Los Alamitos (2007)
2. Heer, T., Li, S., Wehrle, K.: PISA: P2P Wi-Fi Internet Sharing Architecture. In: 7th IEEE International Conference on Peer-to-Peer Computing, pp. 251–252. IEEE Computer Society Press, Los Alamitos (2007)

3. Jin, C., Wang, H., Shin, G.K.: Hop-count filtering: an effective defense against spoofed DDoS traffic. In: 10th ACM conference on Computer and communications security, pp. 30–41. ACM Press, New York (2003)
4. Li, J., Mirkovic, J., Wang, M., Reiher, P., Zhang, L.: SAVE: Source address validity enforcement protocol. In: 21st Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 1557–1566. IEEE Press, Los Alamitos (2002)
5. Bremler-Barr, A., Levy, H.: Spoofing Prevention Method. In: 24th Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 536–547. IEEE Press, Los Alamitos (2005)
6. Gurtov, A.: Host Identity Protocol (HIP): Towards the Secure Mobile Internet. John Wiley and Sons Publishing, Chichester (2008)
7. Kent, S., Atkinson, R.: Security Architecture for the Internet Protocol, IETF, RFC 2401 (1998)
8. Moskowitz, R., Nikander, P.: Host Identity Protocol Architecture, IETF, RFC 4423 (2006)
9. Moskowitz, R., Nikander, P., Jokela, P., Henderson, T.: Experimental Host Identity Protocol (HIP), IETF, RFC 5201 (2008)
10. Jokela, P., Moskowitz, R., Nikander, P.: Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP), IETF, RFC 5202 (2008)
11. Laganier, J., Koponen, T., Eggert, L.: Host Identity Protocol (HIP) Registration Extension, IETF, RFC 5203 (2008)
12. Laganier, J., Eggert, L.: Host Identity Protocol (HIP) Rendezvous Extension, IETF, RFC 5204 (2008)
13. Nikander, P., Laganier, J.: Host Identity Protocol (HIP) Domain Name System (DNS) Extension, IETF, RFC 5205 (2008)
14. Nikander, P., Henderson, T., Vogt, C., Arkko, J.: End-Host Mobility and Multihoming with the Host Identity Protocol (HIP), IETF, RFC 5206 (2008)
15. Bi, J., Wu, J., Yao, G.: A CGA based Source Address Authorization and Authentication (CSA) Mechanism for First IPv6 Layer-3 Hop: draft-bi-savi-csa-00, IETF, Internet Draft (2007)
16. Nikander, P., Melen, J.: A Bound End-to-End Tunnel (BEET) mode for ESP: draft-nikander-esp-beet-mode-09, ITEF, Internet Draft (2008)
17. OpenWRT Web site, `http://www.openwrt.org`
18. PanOULU Public WLAN Network, `http://www.panoulu.net`