Host Identity Indirection Infrastructure (Hi^3)

Pekka Nikander Jari Arkko Börje Ohlman
Ericsson Research Nomadiclab Ericsson Research IP Networks

26th October 2004

Abstract

The Secure Internet Indirection Infrastructure (Secure-i^3) is a proposal for a flexible and secure overlay network that, if universally deployed, would effectively block a number of denial-of-service problems in the Internet. The Host Identity Protocol (HIP), on the other hand, is a proposal for deploying opportunistic, IPsec based end-to-end security, allowing any hosts to communicate in a secure way through the Internet. In this paper, we explore various possibilities for combining ideas from Secure-i^3 and HIP, thereby producing an architecture that is more efficient and secure than Secure-i^3 and more flexible and denial-of-service resistant than HIP.

1 Introduction

The original Internet protocol stack design deliberately did not include solutions for mobility, multi-address multi-homing, address agility in general, or security related to them [2]. In this paper, we consider how to integrate the Secure Internet Indirection Infrastructure, Secure-i^3 [1] and the Host Identity Protocol, HIP [3] in order to address some of these needs.

The Secure Internet Indirection Infrastructure, by Adkins, Lakshminarayanan, Perrig, and Stoica, proposes an i^3-like Distributed Hash Tables (DHT) based overlay connectivity, based on registering *triggers* into the infrastructure, in a secure way. Inheriting from i^3, Secure-i^3 provides infrastructure-based services for mobility, multi-address multi-homing, multicast, and anycast. If a host uses only Secure-i^3 for its communications, it can keep its IP addresses secret and utilize the infrastructure to effectively mitigate even severe distributed flooding denial-of-service attacks. However, as a major drawback, the proposal requires that all traffic flows through an overlay server, thereby almost doubling the amount of network traffic. Similarly, mobile hosts may not be able to use the most efficient route to communicate with each other.

The Host Identity Protocol, originally by Moskowitz, is a proposal to effectively add a new layer to the IP stack. The new layer is located within the IP layer, between the forwarding and the actual end-to-end functions, such as IPsec. Effectively, when HIP is used, the applications no longer connect to IP addresses but to separate Host Identifiers. HIP provides end-to-end oriented opportunistic security, CPU and memory exhausting denial-of-service resistance, mobility, and multi-address multi-homing. It allows any pair of hosts to authenticate the (perhaps anonymous) public key of their peer and establish confidential and integrity protected communication channels. Mobility and multi-address multi-homing are implemented by the end-hosts, allowing each end-host to inform its peers about the current IP addresses in use.

In this paper, we argue that Secure-i^3, with some slight modifications, could provide a secure, integrated *rendezvous infrastructure* for HIP, basically forming a secure *control plane* for the Internet. HIP based data traffic could still be carried directly end-to-end, without involvement from the overlay, between trusted hosts. Additionally, we briefly consider some deployment and operational aspects, arguing that moving into a world combining Secure-i^3 and HIP could be made gradually, giving immediate benefits to the upgrading sites.

2 Hi^3 architecture

We base our proposal for integrating HIP and Secure-i^3 on the observation that DHT extended HIP rendezvous server and the basic Secure-i^3 infrastructure are conceptually fairly close to each other.

The basic idea is to allow direct, IPsec-protected end-to-end traffic while using indirection infrastructure to route the HIP control packets. Additionally, we apply the Secure-i^3DoS protection idea of not revealing the actual IP addresses to HIP, building on our related work [5].

2.1 Minimal integration

Minimally, i^3 could be used as a decentralized instantiation of the HIP rendezvous server. The HITs could directly act as public 128-bit long triggers, and there would be no private triggers. Trigger insertion and removal could be secured with public key cryptography instead of using the secret key construct used in Secure-i^3, see Section 2.3.1. Data traffic could flow, IPsec protected, directly between the hosts just as today.

The main advantage of this initial design is its simplicity. However, many of the more advanced features of i^3 are lost. In the following, we will show a new architecture that that retains many of those properties while keeping the efficiency of HIP. That is, we continue to allow direct IPsec protected traffic between end-hosts but also make it possible for end-hosts to use IPsec-forwarding middle boxes, thereby hiding their actual IP address from untrusted peers.

2.2 Separating data and control

In i^3, public triggers are only used for initial rendezvous. The server host is supposed to create a new private trigger and ask the client to use that for all future communication. In HIP, on the other hand, the HITs are used for all control traffic while the actual data traffic is passed in IPsec envelopes, indexed by SPIs.

For the data traffic, the IPsec envelopes and SPIs can be used to implement DoS protection in a structurally similar way to Secure-i^3. As described in [5], it is easy to design a middle box that forwards traffic based on $<$ destination address, SPI $>$ pairs. The middle boxes can securely learn the appropriate mappings by listening to the signed HIP control packets flowing through them. Such a middle box can also be easily located between distinct IP realms. Hence, instead of selecting an i^3 server that is topologically close to the communication path and using that to forward regular traffic, as suggested in [4], a host could utilize an IPsec-forwarding middle box on the path, either within an IP realm or between multiple IP realms. As the middle box can be on or close to the path, there is no triangular routing. Furthermore, with the HIP mobility and multi-homing mechanisms, a host under attack could even selectively move its legitimate traffic over to alternative middle boxes, similar to a host dynamically changing its private trigger in i^3.

In Hi^3, the SPI mappings can be created dynamically, basically at any location, independent of the trigger and SPI values. This can be compared to Secure-i^3, where the private triggers are distributed based on the DHT topology. Using private triggers necessitates the host to learn the routing location of the servers in order to be able to select a proper one. When SPI based forwarding is used, it is sufficient that the host knows a number of IPsec-forwarding middle boxes that are willing to serve the host. The hosts do not need to know anything about the DHT.

As the SPI-based forwarding and firewalling functionality is separate from the control plane, each host and site can implement the function in a way suitable to them. That makes deployment easier.

2.3 Providing more DoS protection

Above, we separated the control traffic and regular data traffic into different "planes", and protected the data plane with IPsec-forwarding middle boxes. However, this arrangement still leaves the host vulnerable to DoS attacks through the control messages. In other words, it is still possible to flood hosts by sending traffic to their public triggers, i.e., HITs. To block this attack, we next consider a few variations.

As an initial step, we can divide the indirection infrastructure into two parts, more or less like in Secure-i^3. The first part stores only public triggers, and it forwards *only* HIP I1 messages. Compared to the current situation, this arrangement blocks possible I1 storms already at the indirection infrastructure. The second part stores temporary private triggers, and carries the rest of HIP control messages.

As a first approximation, we can imagine the HIP hosts to create temporary triggers as they reply to I1 messages with R1 messages. The created trigger will initially have a very short lifetime, and only a successful R2 message will update its lifetime.

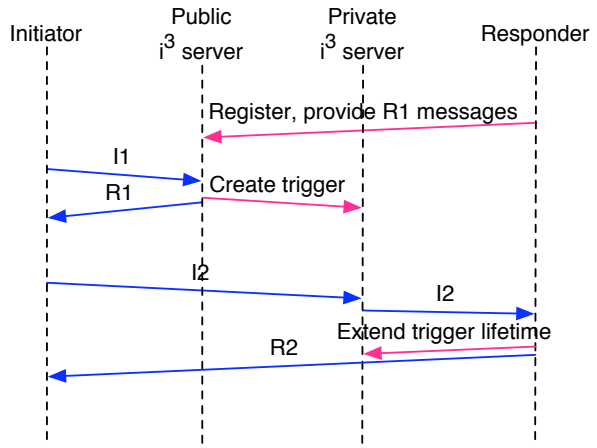

Figure 1: Hi^3 base exchange with delegation

But we can do better. When a HIP host registers its public trigger to the public part of the indirection infrastructure, it can also delegate the process of replying to I1s to the infrastructure. In HIP, replying to an I1 is a mechanical operation that does not create any state. Hence, if a HIP host provides the infrastructure with a number of pre-computed R1 messages, the infrastructure can reply to I1s by constructing the appropriate R1 packets from the precomputed messages. The R1 could include a new private trigger. The private triggers can be distributed over a number of DHT servers, thereby leveling load in the case of an I1 flood. Furthermore, the HIP puzzle can be defined to depend on the private trigger, making the solution valid only for that particular private trigger.

Once the initiator has processed the R1 packet and produced the puzzle solution, it sends an I2. The I2 is now sent to the private trigger. Furthermore, the DHT server that hosts the private trigger can *verify that the puzzle solution is correct* before passing the I2 packet to the end-host. Effectively, this distributes the proposed Secure-i^3DoS-filter function over all DHT server nodes, allowing the puzzle to be formed and verified by different nodes. The packet flow is illustrated in Figure 1.

Note that in the resulting structure we do not need IP source addresses for anything any more. The initial requests are sent to the infrastructure, and the infrastructure answers back based on the requestor's registered identifier. Once the HIP association has been set up, the source addresses are no longer used. Consequently, the source address field can be reused for other purposes, for example, to record the packet's path.

2.3.1 Securing registration with public keys

In both i^3 and Secure-i^3, basic trigger security is provided by keeping parts of the trigger secret. In Secure-i^3, this is additionally strengthened by constructing a part of the trigger by taking a hash over the secret part. In Hi^3, this can be made more flexible by using public key cryptography.

In HIP, the Host Identity Tag (HIT), is formed by taking a hash over a public key. That allows a recipient to verify that a given public key actually matches with a given HIT. Consequently, public key authentication can be used to provide *strong ownership* over HIT based triggers, allowing only the host that knows the private key to register and remove that particular trigger.

To simultaneously preserve the flexibility of the original i^3 and to provide strong security for public triggers, we propose that the trigger space is divided into two parts. In the first part the trigger IDs are bound to public cryptographic keys. In the second part, there are either no restriction on the structure, or something similar to the Secure-i^3 constrains are applied. The exact nature of the constrains are left for further study.

2.4 Cascading and stacked identifiers

In this paper, we have called the i^3 ability of having triggers of the form $< \mathrm{id}_1, \mathrm{id}_2 >$ as cascading triggers. The original motivation for including this feature in i^3 included support for large multicast trees and additional security by employing secret private triggers in cascading trigger chains. However, as discussed in [1] , the feature allows an potential attacker to construct various topologies that potentially may harm the whole infrastructure. To counter this threat, in Secure-i^3 the structure of the triggers is restricted so that it becomes impossible to construct loops.

For the sake of security and simplicity, we propose removing the cascading triggers from the architecture. All triggers would point to an IP address. On the other hand, using *stacked identifiers* for service composition may still be a good idea.

2.5 Mobility

In Hi^3, basic mobility between already communicating hosts can be provided directly at the HIP level, without involving the infrastructure. Only if the hosts lose direct reachability information, they need to revert back to the infrastructure. Even in that case they may use private triggers, being safe from attacks launched by third parties. However, if they do use private triggers, the hosts must keep the infrastructure updated with their current location information. For initial reachability, a mobile host needs to update its public registrations at the infrastructure.

To reduce signalling overhead, we propose that the mobile host only updates its public registration. As the private triggers are created by the infrastructure during the initial session setup (see Section 2.3), the mobile hosts could easily provide the infrastructure with information that allows it to distribute this update to the servers hosting the private triggers.

By combining the end-to-end mobility provided by HIP and the indirect mobility provided by the infrastructure, the resulting mechanism is highly efficient (no triangle routing for regular data) and robust (inherited from i^3).

2.6 Anycast and delegation

A major problem in the original i^3anycast is that an attacker can easily add its own receivers for an anycast identifier. To alleviate this attack, stricter control is needed on anycast triggers.

While sharing a secret and clever use of Secure-i^3 trigger constrains might provide an adequate level of security for some applications, we propose using delegation based on public keys. Basically, the holder of the private key corresponding to a HIT can either directly register multiple anycast triggers or alternatively sign statements that allow HITs to add new anycast registrations.

As anycast aims to find a server that is close in real communication terms, we surmise that any actual overlay based implementation of real anycast service will be somewhat complex. Selecting an anycast server may not be as easy as indicated in [4]. The indirection infrastructure must have some understanding of the real underlying routing topology.

3 Conclusions

HIP and i^3 belong to the most prominant proposals to improve the Internet architecture to provide support for mobility, multi-homing and various forms of denial-of-service resistance. In this paper, we have discussed one possible approach for integrating ideas from i^3 and HIP, thereby providing an architecture that largely preserves the end-to-end nature and efficiency of HIP and the flexibility of i^3. Furthermore, the result appears to be more secure and more denial-of-service resistant than either of the proposals alone.

References

[1] D. Adkins, K. Lakshminarayanan, A. Perrig, and I. Stoica. Towards a more functional and secure network infrastructure. Technical Report UCB/CSD-03-1242, Computer Science Division (EECS), University of California, Berkeley, 2003.

[2] J. Kempf, P. Nikander, D. Crocker, H. Esaki, and J.-I. Hagino. Panel on the future of the internet architecture. in SAINT'2004, Yokohama, Japan, Jan. 2004.

[3] R. Moskowitz and P. Nikander. Host Identity Protocol Architecture. Internet Draft, work in progress, Nov. 2003.

[4] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet Indirection Infrastructure. In *Proc. ACM SIGCOMM 2002*, Aug. 2002. Pittsburgh, PA, USA.

[5] J. Ylitalo and P. Nikander. SPINAT: SPI multiplexed NAT for secure and efficient mobility. Unpublished manuscript.