Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 13, 2012

X. Xu
S. Jiang
D. Zhang
Huawei Technologies Co.,Ltd
D. Korzun
HIIT
Z. Cao

March 12, 2012

Extensions of Host Identity Protocol (HIP) with Hierarchical Information

draft-xu-hip-hierarchical-03

Abstract

This document explores the benefits brought by extending the Host Identity Protocol (HIP) with hierarchical information. In addition, three types of candidate solutions are introduced.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 13, 2012.

Copyright Notice

Table of Contents

1. Introduction

While having obtained a tremendous success, the current Internet architecture shows its limits in many aspects. For example, the current Internet cannot well support the incorporation of mobile and multi-homed terminals, lacks essential security mechanisms, and suffers from the issues caused by the explosively increased lengths of routing tables. In order to address these challenges, a comprehensive solution, the Host Identity Protocol (HIP), was proposed. A simple principle behind HIP is to separate hosts' identities from their topological locations in the Internet. Currently, the basic architectures and protocols of HIP have been developed, which are security-inherited and provides essential supports for mobility and multi-homing features.

There is no hierarchcial information in the current HITs; hosts in the current HIP architecture are organized as all "flat". This is largely because a flat HIP namespace is simple and easy to implement. This document first discusses the issues with the flat HIP architecture and analyzes the benefits brought by integrating hierarchical information with HIP in terms of security, management, integration with hierarchical overlays and etc. Then, this document introduces several potential solutions which can be used to facilitate the integration of hierarchical information.

2. Benefits introduced by Hierarchical Information

Hierarchy is a practical methodology in the design and organization of non-trivial distributed systems, and has been adopted in many large-scale networks and distributed systems (e.g., Internet). It brought benefits in terms of simplifying system architectures, improving the capability of system management, facilitating audit and security, and etc. To be consistent with the hierarchical features of the Internet, two critical namespaces of the Internet, IP and FQDN, are designed in hierarchical ways. However, based on certain concerns (e.g., easy implementation), the current HIT namespace is flat; HIP itself does not provide any support for hierarchy either.

This section attempts to demonstrate that current HIP, by using hierarchical information, can be more efficient and flexible in many typical scenarios.

Firstly, hierarchical information is essential for the combination of HIP with hierarchical overlays (e.g., hierarchical resolution mechanisms). Compared with flat overlays where resources are maintained at essentially random nodes, hierarchical overlays are able to support reasonable business and trust models where resources

are managed by Administrative Domains (ADs) with distinct boundaries. For example, it is normally not desired for a country to have its resolution infrastructure and the related data resources managed by other countries. In order to correctly route across hierarchical overlays, hierarchical information (e.g., AD identifiers) is required to identify the destination AD where the desired resources are maintained, while the resource identifiers are used to locate the resources.

Secondly, the hierarchical information can be used to address the uniqueness verification issues with HITs in current HIP solutions. In current HIP solutions, the HIT of each host is required to be unique all over the world, which is very difficult to guarantee. However, if the Internet is divided into multiple administration domains, this problem is relatively easier to address. As hierarchical information (i.e., AD identifier) can be used to identify the AD of a HIT, it only needs to be guaranteed that the HIT is unique within the AD. The process of verifying the uniqueness of HITs can be performed when the host registers its HIT with the AD.

Moreover, hierarchical information has been widely employed in advanced authorization systems (e.g., attributes based or role-based authorization systems) to make the access control aggregates. By using AD identifiers, it is possible for security managers to design the access control policies based on the AD of hosts so as to reduce the length of access control lists. In contrast, there is nothing common between flat HITs that were assigned by the same authority or that their represented hosts have the same properties, and thus they are difficult to be categorized.

Apart from the advantages mentioned above, hierarchical information may associate HIP with better HIT administrating and auditing capabilities. The hierarchical information makes HITs more aggregative; they can be grouped according to its belonging authority or domain. Each network operator just needs to manage and maintain HITs and their mapping information in a relatively small range. Such advantages can make HIP easier to be accepted by the countries or organizations which have relatively strict management policies on their networks.

3. Candidate Solutions

There are various ways to integrate hierarchical information into the HIP architecture. In the current version of document, we select three representative candidates, and more solution may be introduced in future versions.

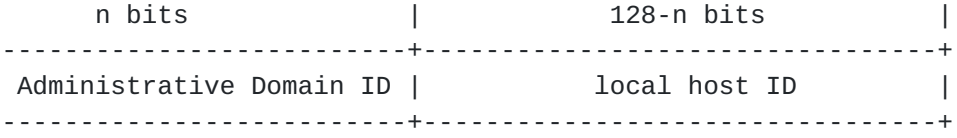
The first type of solution is to embed hierarchical information into HITs directly. For instance, divide a HIT into two parts; the first part indicates the hierarchical information of the host, and the second pare is the identifier of the host. The principle behind this type of solution is similar with IP addresses.

The second type of solution is to transport hierarchical information somewhere outside HITs, e.g., in a certificate or in a parameter. In the preceding case, the certificate can be transported within the CERT parameter of the HIT header.

The third type of solution is a hybrid of the above two types of solutions. This type of solution extends the length of the 128 bits HITs. The extended place is used to contain hierarchical information.

4. Integrating Hierarchical Information into 128 Bits HITs

In this section, we introduce an example hierarchically structured HIT namespace. In this hierarchical HIT namespace, a 128-bit HIT consists of two parts: an n-bit HIP AD ID and a (128-n)-bit local host ID. (n is a subject to be decided in the future.) It can represent maximum 2^n administrative domains and 2^(128-n) hosts within each administrative domain. The Administrative Domain ID has embedded organizational affiliation and global uniqueness. The local host ID is a hash over the AD ID and the public key of the ID owner.



HIT with Hierarchical Information

Since the local host ID is a hash result, the strength of the local host ID in tolerating brute-force attacks is affected by its length. If the hash algorithm cannot be inverted, the expected number of iterations required for a brute force attack is O(2^(128-n)) in order to find a local host ID that matches with a given local host ID. Therefore, for the secure consideration, we recommend to only assign necessary bits for HIP administrative domain ID and leave space for the local host ID as much as possible. It should be noted that this draft does not take into account the ORCHID prefix defined in [RFC4843] for two reasons: firstly, ORCHID is only temporary assigned for experimental usage till 2014 only. The proposal design in the document is targeting to be used continuously after 2014. Secondly, the fixed 28-bit orchid prefix reduces the security properties massively and increase collusion possibility highly.

The HIP administrative domain, as its literal, is a logic region in which the HIs of all nodes are assigned by the same authority. Within a same HIP administrative domain, all the nodes should have the same HIP AD ID or the same leftmost certain bits. Furthermore, the authority may be organized internally hierarchically.

The HIP AD ID should be assigned by a global administrative organization with the principle that every HIP AD ID must be globally unique.

Consequentially, the HIP AD IDs may be organized hierarchically. For example, a big organization may obtain a block of HIP AD IDs with an assigned 16-bit prefix. It then can assign 24-bit HIP AD IDs to its sub-organizations. All these sub-organizations have the same leftmost 16-bit.

One promising allocation solution of HIP AD ID is following current routable IP address allocation system [RFC2050]. At first IANA allocates some HIP AD ID prefixes to RIR (Region Internet Registry) or NIR (National Internet Registry),then RIR or NIR sub-allocates the HIP AD ID prefix to LIR or backbone ISP that subdivides the tag prefix to middle or small ISP. Historical experience of routable IP address allocation indicates that the allocation system can ensure global uniqueness of HIP AD IDs.

An advantage of this solution is that the HHIT architecture can build a distributed catalogue based on current IP address Internet Registry. Each level Internet Registry only needs to maintain its HHIT information. This catalogue is like current IP Whois Server operated by each IP address Internet Registry. But it should include many more attributes about a HHIT, such as organizational affiliation, geographical information, privacy protection rule etc. The catalogue should be independent of current IP Whois system and IP address Internet Registry should provide some mechanism to translate HHIT to its useful attributes on demand of various applications.

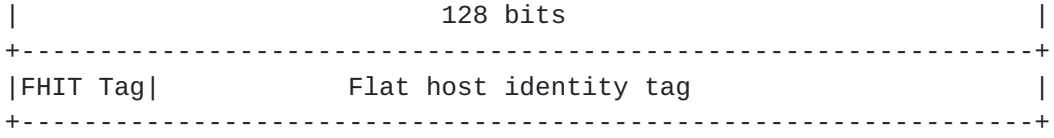
The local host IDs remains the original meaning of HIT - "a hashed encoding of the Host Identity". For each HIP administrative domain, it is mandatory to maintain the uniqueness of all local host IDs. It is guaranteed by the process of generating a HIT, see Section 5.

For resolution purposes, HITs are aggregatable with AD IDs of arbitrary bit-length, similar to IPv4 addresses under Classless Inter-Domain Routing [RFC4632].

4.1. Compatible flat-structured HITs

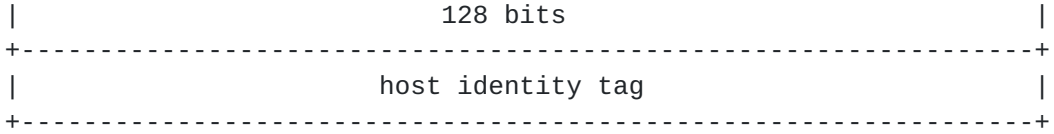
Obviously, not all hosts are willing to use hierarchical HITs in all scenarios for various reasons, such as privacy. Therefore, it is useful that the hierarchical HIT architecture keep compatible with the flat HIT architecture.

The flat HITs can be defined as a specific sub-set of the hierarchical HITs architecture. With the same reserved Flat HIT Tag (3 or 4 bits) at the beginning, for example, the left-most 3 bits is 000, the flat HITs can be used as defined in [RFC4423].



4.2. HITs on nodes

HIP-enabled nodes may have considerable or little knowledge of the internal structure of hierarchical HITs, depending on the role the node plays (for instance, host versus mapping server). At a minimum, a node may consider pre-generated HITs have no internal structure:



Only sophisticated hosts may additionally be aware of the type of their HITS and use the hierarchical structure of HITs to simplify the resolution procedure.

4.3. Generating a hierarchical HIT

The process of generating a new hierarchical HIT takes three input values: an n-bit HIP AD ID, a 2-bit collusion count, (an example, it is a subject to be changed in the future.) the host identity (the public key of an asymmetric key pair). A hierarchical HIT should be generated as follows:

1. Set the 2-bit collision count to zero.

2. Concatenate from left to right the HIP AD ID, the collusion count, and the host identity. Execute the SHA-1 algorithm on the concatenation. Take the (128-2-n) leftmost bits of the SHA-1 hash value.

3. Concatenate from left to right the n-bit HIP AD ID, the 2-bit collusion count and (128-2-n)-bit hash output to form a 128-bit

HIT.

4. Perform duplicate detection within the HIP administrative domain scope. If a HIT collision is detected, increment the collision count by one and go back to step 2. However, after four collisions, stop and report the error. (Note: the duplicate detection mechanism is not discussed in this document. It may be broadcast or central registration.)

The design that includes the HIP AD ID in the hash input is mainly against the re-computation attack: create a database of HITs and matching public keys. With the design, an attacker must create a separate database for each HIP administrative domain.

The design reduces the number of bit of hash output 2 bits lower. It does reduce the safety. However, O(2^(128-2-n)) iterations is large enough to prevent brute-force attacks.

For security reason, the abovementioned SHA-1 hash algorithm may be replaced by any safer algorithm.

5. Transporting Hierarchical information outside HITs

As mentioned previously, there are at least two methods of transporting hierarchical information in HIP headers, i.e., using certificates and using parameters. Compared with the certificate oriented method, it is relatively more efficient to use parameters to transport hierarchical information. For instance, some parameters of a certificate (e.g., the name and the public key of the subject) are already contained in HIT headers. When using a certificate to transport hierarchical information, these parameters may have to be transported again, causing redundancy. In addition, certificates have to be signed by issuers. The signature of a certificate can be used to verify the authenticity of the transported hierarchical information, which is very useful when the certificate is used to transport hierarchical information for the source HIT of a HIP packet. However, when the certificate is used to transport hierarchical information for the destination HIT of a HIP packet, the signature is redundant because the receiver of the packet needs not to verify the authenticity of its hierarchical information. Another concern is performance. A HIT can be attached with multiple certificates which are issued by diverse third parties for the various purposes. The system thus may have to go through all the certificates in order to find the proper certificate issued by the AD and use it to assess the validity of the HIT.

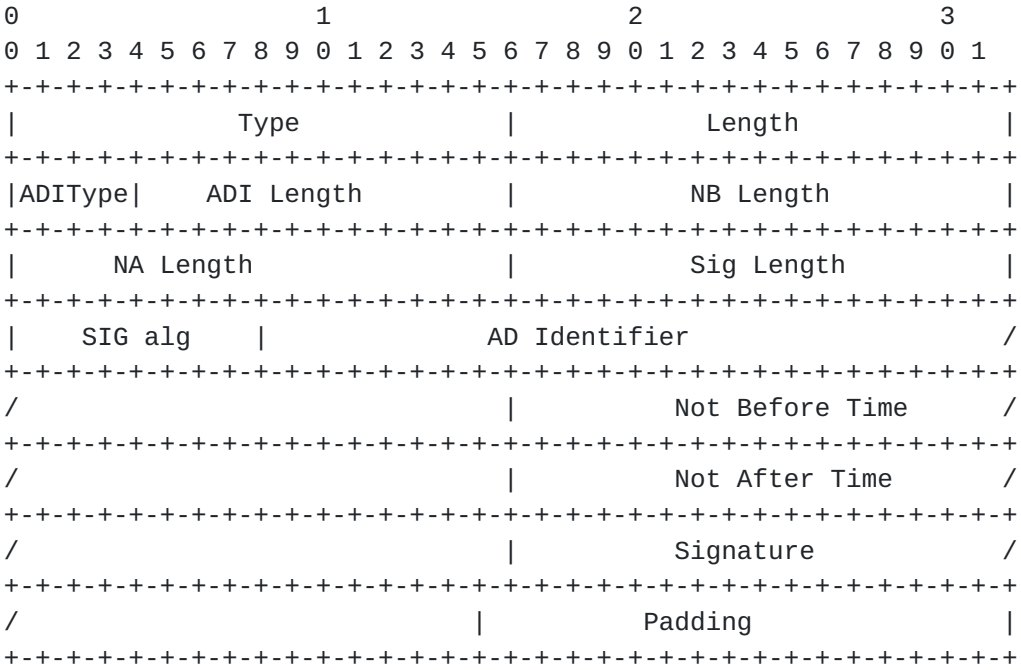
In the remainder of this section, we mainly introduce an example

Hierarchical_HIT Parameter which is used to transport hierarchical information. In addition, several associated extensions are proposed.

5.1. Hierarchical_HIT Parameter

This parameter contains the information about the AD and should be transported in R1 and I2 packets of basic.

Type	61698
Length	length in octets, excluding Type, Length, and Padding
ADI Type	type of the Administration Domain Identifier field
ADI Length	length of the FQDN or NAI in octets
NB Length	length of the Not Before Time field in octets
NA Length	length of the Not After Time field in octets
AD Identifier	the identifier of the AD of the sender
Not Before Time	the beginning of the valid period of the HIT of the sender
Not After Time	the end of the valid period of the HIT of the sender
SIG alg	signature algorithm
Signature	the signature is generated by the AD previously, calculated over the concatenation of Host Identity field of HOST_ID, and AD Identifier, Not Before Time, Not After Time fields of the Hierarchical_HIT parameter.



The following ADI Types have been defined:

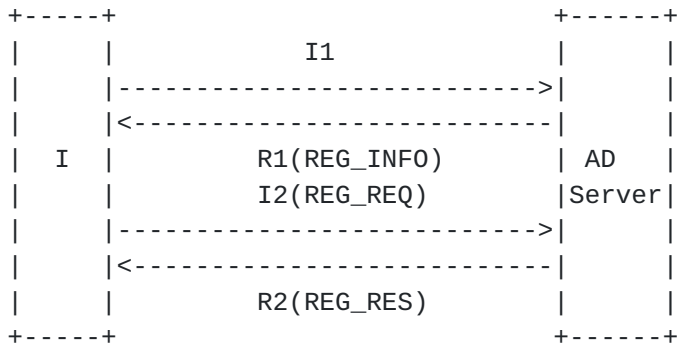
Type	Value
none included	0
FQDN (Fully Qualified Domain Name, in binary format)	1
NAI (Network Access Identifier)	2

The format for the FQDN is defined in[RFC1035] Section 3.1. The format for NAI is defined in [RFC4282]. Not Before Time and Not After Time fields can either UTCTime or GeneralizedTime defined in [RFC2459]. SIG alg is set to 0 when there is no signature included. In this case, Sig Length is set 0 as well.

Note that the parameter introduced in this section only consists of very essential information. The parameter may need to be extended or modified before being applied in future.

5.2. Hierarchical Information Registration

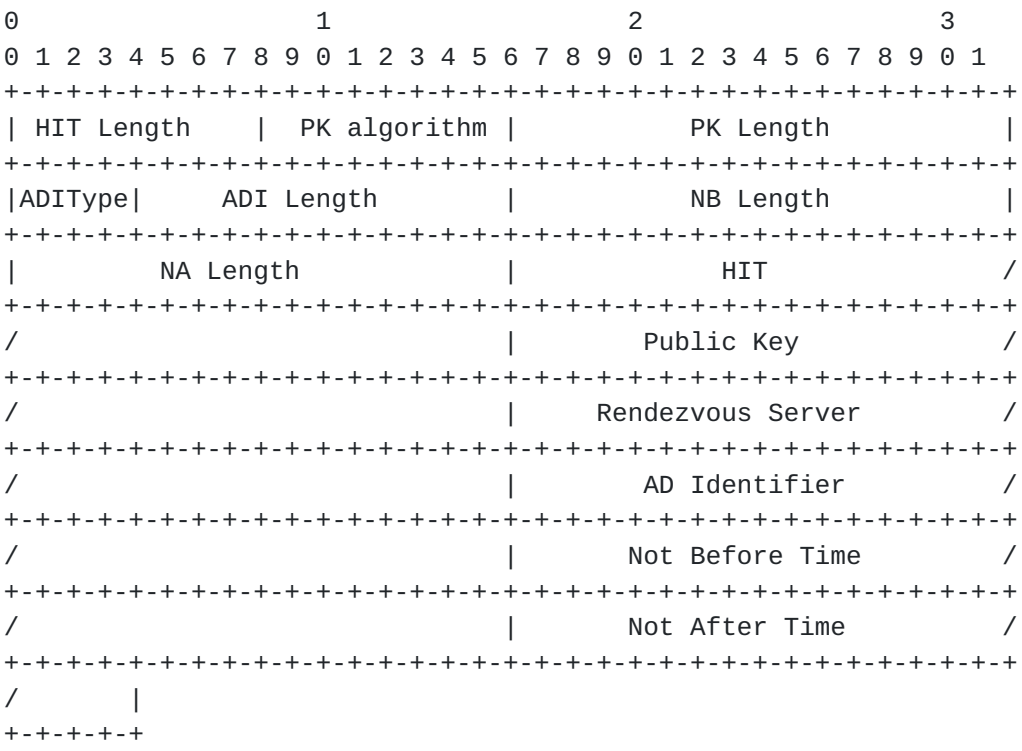
If the authenticity of the hierarchical information of a HIT needs to be proved in practice, the HIT need to register with an AD and obtain the signature. The registration process can be whether in-band or out-of-band. In the following diagram, a protocol for hierarchical information registration is illustrated.



This protocol is an extension of basic by using the HIP Registration Extension [RFC5203]. In R1, AD Server sends the service it provides to Initiator in the REG_INFO element. Initiator then attaches the REG-REQ element and the HHIT parameter with the I2 message. The Signature field in the parameter is left unfilled. The AD server signs the HHIT and its parameters, and sends the signature back in R2.

5.3. Domain Name System (DNS) Extension

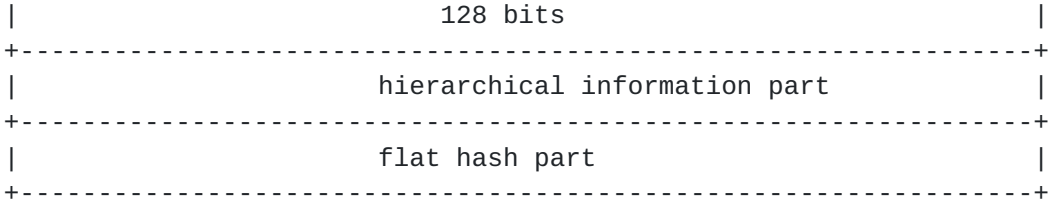
This section introduces a DNS extension which further extends the HIP RR Storage Format proposed in [RFC5205].



Apart from the fields illustrated in [RFC5205], the extension includes following fields: ADI type, ADI Length, NB Length, NA Length, AD Identifier, Not Before Time, Not After Time. Because the meanings of these fields is identical to their counterparts in the Hierarchical_HIT Parameter, they are not introduced here in detail.

6. Extending the length of HITs

In this section, we introduce a hybrid of the above two types of solutions. In this solution, hierarchical information is integrated within HITs. Unlike the solution proposed in section 3, the space of the flat hash part of a HIT does not have to be occupied. Instead, the whole length of the HIT is extended, and the extended space is used to contain the hierarchical information. An example of such hierarchical HITs is presented in the following figure.



The enlarged HIT presented in the figure can be broken into two parts: the hierarchical information part and the flat hash part. In this example, the flat hash part is generated by hashing the

concatenation of hierarchical information part and the associated public key. In order to keep enough capability in tolerating brute force attacks and be compatible with TCP, it is recommended the flat hash part is set 128 bits long. When receiving such a HIT, a user only transfers the flat hash part to the TCP layer, and thus TCP will treat it as an ordinary IPv6 address.

7. Analysis of the Three Solutions

A criticism on the first type of solution is that the capability of an identifier in tolerating brute-force attacks is affected as a part of the space of the identifier that is occupied by the topological information. This issue can be largely addressed by puzzles which have been employed in Cryptographically Generated Addresses (CGA) [RFC3972]. Also, it is possible to extend the length of HITs to enhance their tolerant capability on brute force attacks.

Another concern with hierarchical HITs is that they are not suitable for the scenario where hosts do not intend to disclose their hierarchical information. In section 4, these problems and associated solutions are introduced.

The second type of solution allows a user to flexibly present or hide the hierarchical information in various circumstances. A disadvantage imposed by this type of solution is that more traffic needs to be transported as both certificates and parameters may contain redundant information.

Compared with the first type of solution, the capability of the third type of solution in tolerating brute force attacks is not influenced. Additionally, compared with the second type of solution, the third type of solution avoids transporting the redundant information. However, a disadvantage of the third type of solution is that it modifies the architecture of HIP headers.

8. IANA Considerations

The namespace, HIP AD ID, defined in section 4 is an n-bit long value, which represents a globally unique HIP administrative domain. IANA may found an authority institute to manage the global assignment of HIP AD ID.

Additionally, IANA is expected to allocate a type code for the Hierarchical_HIT Parameter illustrated in section 5.

Note to RFC Editor: this section may be removed on publication as an

RFC.

9. Security Considerations

The hierarchical HIT routing infrastructure provides some mechanisms for defending attacks (e.g., proving HIT ownership, certificate binding hierarchical HITs to a trusted third party, or random AD IDs). Below we list some possible attacks on hierarchical HITs.

Forging hierarchical HITs. An attacker generates a HIT with certain attack-suitable properties for using it for further attacks. Classes of such properties are listed below. For checking the existence of a HIT, the name resolution system (DNS/DHT) can be used.

Intrusion. Generation of HITs belonging to some organization. As a result the attacker can participate in communications between organization's hosts. Organization's AD ID is known to the attacker, and 64-bit hash value is generated. The attacker has to prove the ownership of that HIT since it does not have the private key.

Substitution. An attacker tries to use the HIT already existed in the organization. As a result, the attacker substitutes good host. Generation of HITs (64 bits of hash value) randomly or by enumeration. For every HIT the attacker tries to join the system.

Cutting. AD ID can code a hierarchy with in a large organization. The hierarchy of AD ID is based on prefixes. As a result, an attacker can generate a HIT that shares a prefix with the AD ID of the organization. Hence the attacker cuts a part of HIT space. Similarly to intrusion and substitution except that the generated HITs share some prefix with the given AD ID.

Accumulation. Valid HITs can be prepared in advance, i.e., collected in a database. Similarly to substitution attack, the attacker generates HITs and tries to join. Is it possible to identify that the HIT is in use and what is the ratio of successful identifications (does HIT exist or not) .

Sybils. Introduction of many forged HITs. They incrementally appear in the system. The attacker has a host with a valid HIT joined to the system. Can this host introduce new participants (with new HITs) easier than a newcomer without a protege.

Denial of Service. The hierarchical HIT infrastructure consists of DNS and DHT servers. In addition, there can be third-party servers, e.g., for license binding. All such servers are a target to DoS attacks, including DDoS.

1. Generation of a high-rate request flow to a DNS server.

2. Generation of a high-rate request flow to a DHT server (or a set of severs of a given organization).

3. Generation of a high-rate request flow to a third-party server. It controls its service rate limiting incoming requests.

Pollution. Similarly to DoS attacks, DNS and DHT servers are target to incorrect data.1. The attacker tries to store bogus AAAA records for hierarchical HIT in DNS. 2. Similarly to the intrusion and substitution attacks, the attacker tries to insert bogus HITs into DHT system of a given organization.

10. Acknowledgements

Thanks Thomas. R. Henderson for his kindly prove-reading and precious comments.

11. References

11.1. Normative References

[RFC2050] Hubbard, K., Kosters, M., Conrad, D., Karrenberg, D., and J. Postel, "INTERNET REGISTRY IP ALLOCATION GUIDELINES", BCP 12, RFC 2050, November 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2459] Housley, R., Ford, W., Polk, T., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 2459, January 1999.

[RFC4282] Aboba, B., Beadles, M., Arkko, J., and P. Eronen, "The Network Access Identifier", RFC 4282, December 2005.

[RFC4423] Moskowitz, R. and P. Nikander, "Host Identity Protocol (HIP) Architecture", RFC 4423, May 2006.

[RFC5201] Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", RFC 5201, April 2008.

[RFC5203] Laganier, J., Koponen, T., and L. Eggert, "Host Identity Protocol (HIP) Registration Extension", RFC 5203, April 2008.

[RFC5205] Nikander, P. and J. Laganier, "Host Identity Protocol (HIP) Domain Name System (DNS) Extensions", RFC 5205, April 2008.

11.2. Informative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

[RFC4843] Nikander, P., Laganier, J., and F. Dupont, "An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers (ORCHID)", RFC 4843, April 2007.

Authors' Addresses

Xiaohu Xu
Huawei Technologies Co.,Ltd
HuaWei Building, No.3 Xinxi Rd., Shang-Di Information Industry Base, Hai-Dian District
Beijing, 100085
P. R. China

Phone:
Fax:
Email: xuxh@huawei.com
URI:

Sheng Jiang
Huawei Technologies Co.,Ltd
HuaWei Building, No.3 Xinxi Rd., Shang-Di Information Industry Base, Hai-Dian District
Beijing, 100085
P. R. China

Phone:
Fax:
Email: shengjiang@huawei.com
URI:

Dacheng Zhang
Huawei Technologies Co.,Ltd
HuaWei Building, No.3 Xinxi Rd., Shang-Di Information Industry Base, Hai-Dian District
Beijing, 100085
P. R. China

Phone:
Fax:
Email: zhangdacheng@huawei.com
URI:

Dmitry Korzun
HIIT

Phone:
Fax:
Email: Dmitry.Korzun@hiit.fi
URI:

Zhen Cao
Xuanwumenxi Ave. No.32
Beijing, 100053
China

Phone: 86-10-52686688
Fax:
Email: zehn.cao@gmail.com, caozhen@chinamobile.com
URI: