Hierarchical HITs for HIPv2
draft-moskowitz-hip-hierarchical-hit-02

Abstract

This document describes using a hierarchical HIT to facilitate large deployments of managed devices. Hierarchical HITs differ from HIPv2 flat HITs by only using 64 bits for mapping the Host Identity, freeing 32 bits to bind in a hierarchy of Registering Entities that provide services to the consumers of hierarchical HITs.

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 https://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 19 April 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text.
as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ........................................... 2
2. Terms and Definitions ................................... 3
   2.1. Requirements Terminology ........................... 3
   2.2. Definitions ....................................... 3
3. Problem Space ........................................... 3
   3.1. Meeting the future of Mobile Devices in a public space .......................... 3
   3.2. Semi-permanency of Identities ....................... 4
   3.3. Managing a large flat address space .................. 4
   3.4. Defense against fraudulent HITs ...................... 4
4. The Hierarchical Host Identity Tag (HHIT) ..................... 4
   4.1. HHIT prefix ....................................... 5
   4.2. HHIT Suite IDs ..................................... 5
   4.3. The Hierarchy ID (HID) ............................ 5
       4.3.1. The Registered Assigning Authority (RAA) ....... 5
       4.3.2. The Hierarchical HIT Domain Authority (HDA) ... 6
       4.3.3. Example of the HID DNS ........................ 6
       4.3.4. HHIT DNS Retrieval ............................ 6
       4.3.5. Changes to ORCHIDv2 to support Hierarchical HITs .......................... 7
       4.3.6. Collision risks with Hierarchical HITs ............ 7
5. IANA Considerations .................................... 8
6. Security Considerations ................................. 8
7. Acknowledgments ....................................... 8
8. References ............................................. 8
   8.1. Normative References ............................... 8
   8.2. Informative References ............................. 8
Appendix A. Calculating Collision Probabilities .................. 9
Authors’ Addresses ....................................... 9

1. Introduction

This document expands on HIPv2 [RFC7401] to describe the structure of a hierarchical HIT (HHIT). Some of the challenges for large scale deployment addressed by HHITs are presented. The basics for the hierarchical HIT registries are defined here.

Separate documents will further expand on the registry service and how a device can advertise its availability and services provided.
2. Terms and Definitions

2.1. Requirements Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Definitions

HDA (Hierarchical HIT Domain Authority): The 14 bit field identifying the HIT Domain Authority under an RAA.

HID (Hierarchy ID): The 32 bit field providing the HIT Hierarchy ID.

RAA (Registered Assigning Authority): The 18 bit field identifying the Hierarchical HIT Assigning Authority.

3. Problem Space

3.1. Meeting the future of Mobile Devices in a public space

Public safety may impose a "right to know" what devices are in a public space. Public space use may only be permitted to devices that meet an exacting "who are you" query. This implies a device identity that can be quickly validated by public safety personal and even the general public in many situations.

Many proposals for mobile device identities are nothing more than a string of bits. These may provide information about the device but provide no assurance that the identity associated with a device really belongs to a particular device. Further they may impose a slow, complex method to discover the device owner to those with appropriate authorization.

The Host Identity Tag (HIT) from the Host Identity Protocol (HIP) provides a self-asserting Identity through a public key signing operation using the Host Identity’s (HI) private key.

Although the HIT provides a "trust me, I am me" claim, it does not provide an assertion as to why the claim should be trusted and any additional side information about the device. The later could be distributed directly from the device in a secure manner, but again there is no 3rd-party assertion of such a claim.
3.2. Semi-permanency of Identities

A device Identity has some degree of permanency. A device creates its identity and registers it to some 3rd-party that will assert a level of trust for that identity. A device may have multiple identities to use in different contexts, and it may deprecate an identity for any number of reasons. The asserting 3rd-party may withdraw its assertion of an identity for any number of reasons. An identity system needs to facilitate all of this.

3.3. Managing a large flat address space

For HITs to be successfully used by a large population of mobile devices, it must support an Identity per device; potentially 10 billion Identities. Perhaps a Distributed Hash Table [RFC6537] can scale this large. There is still the operational challenges in establishing such a world-wide DHT implementation and how RVS [RFC8004] works with such a large population. There is also the challenge of how to turn this into a viable business. How can different controlling jurisdictions operate in such an environment?

Even though the probability of collisions with 7B HITs (one HIT per person) in a 96 bit flat address space is 3.9E-10, it is still real. How are collisions managed? It is also possible that weak key uniqueness, as has been shown in deployed TLS certificates [WeakKeys], results in a much greater probability of collisions. Thus resolution of collisions needs to be a feature in a global namespace.

3.4. Defense against fraudulent HITs

How can a host protect against a fraudulent HIT? That is, a second pre-image attack on the HI hash that produces the HIT. A strong defense would require every HIT/HI registered and openly verifiable. This would best be done as part of the R1 and I2 validation. Or any other message that is signed by the HI private key.

4. The Hierarchical Host Identity Tag (HHIT)

The Hierarchical HIT (HHIT) is a small but important enhancement over the flat HIT space. By adding two levels of hierarchical administration control, the HHIT provides for device registration/ownership, thereby enhancing the trust framework for HITs.

HHITs represent the HI in only a 64 bit hash and uses the other 32 bits to create a hierarchical administration organization for HIT domains. Hierarchical HITs are ORCHIDs [RFC7343]. The change in construction rules are in Section 4.3.5.
A HHIT is built from the following fields:

* 28 bit IANA prefix
* 4 bit HIT Suite ID
* 32 bit Hierarchy ID (HID)
* 64 bit ORCHID hash

4.1. HHIT prefix

A unique 28 bit prefix for HHITs would be ideal. It would clearly separate the flat-space HIT processing from HHIT processing. However, the prefix is the domain of ORCHIDs [RFC7343], and cannot be changed directly here. Proposing a unique prefix to simplify Section 4.2 is in scope of this draft.

4.2. HHIT Suite IDs

The HIT Suite IDs specifies the HI and hash algorithms. Any HIT Suite ID can be used for HHITs, provided that the prefix for HHITs is different from flat space HITs. Without a unique prefix Section 4.1, additional HIT Suite IDs would be needed for HHITs. This would risk exhausting the limited Suite ID space of only 15 IDs.

4.3. The Hierarchy ID (HID)

The Hierarchy ID (HID) provides the structure to organize HITs into administrative domains. HIDs are further divided into 2 fields:

* 14 bit Registered Assigning Authority (RAA)
* 18 bit Hierarchical HIT Domain Authority (HDA)

4.3.1. The Registered Assigning Authority (RAA)

An RAA is a business or organization that manages a registry of HDAs. For example, the Federal Aviation Authority (FAA) could be an RAA.

The RAA is a 14 bit field (16,384 RAAs) assigned by a numbers management organization, perhaps ICANN’s IANA service. An RAA must provide a set of services to allocate HDAs to organizations. It must have a public policy on what is necessary to obtain an HDA. The RAA need not maintain any HIP related services. It must maintain a DNS zone minimally for discovering HID RVS servers.
This DNS zone may be a PTR for its RAA. It may be a zone in a HHIT specific DNS zone. Assume that the RAA is 100. The PTR record could be constructed:

```
100.hhit.arpa   IN PTR      raa.bar.com.
```

### 4.3.2. The Hierarchical HIT Domain Authority (HDA)

An HDA may be an ISP or any third party that takes on the business to provide RVS and other needed services for HIP enabled devices.

The HDA is an 18 bit field (262,144 HDAs per RAA) assigned by an RAA. An HDA should maintain a set of RVS servers that its client HIP-enabled customers use. How this is done and scales to the potentially millions of customers is outside the scope of this document. This service should be discoverable through the DNS zone maintained by the HDA’s RAA.

An RAA may assign a block of values to an individual organization. This is completely up to the individual RAA’s published policy for delegation.

### 4.3.3. Example of the HID DNS

HID related services should be discoverable via DNS. For example the RVS for a HID could be found via the following. Assume that the RAA is 100 and the HDA is 50. The PTR record is constructed as:

```
50.100.hhit.arpa   IN PTR      rvs.foo.com.
```

The RAA is running its zone, 100.hhit.arpa under the hhit.arpa zone.

### 4.3.4. HHIT DNS Retrieval

The HDA SHOULD provide DNS retrieval per [RFC8005]. Assume that the RAA is 10 and the HDA is 20 and the HHIT is:

```
```

The HHIT FQDN is:

```
```

The NS record for the HDA zone is constructed as:

```
20.10.hhit.arpa   IN NS      registry.foo.com.
```
registry.foo.com returns a HIP RR with the HHIT and matching HI. The HDA sets its policy on TTL for caching the HIP RR. Optionally, the HDA may include RVS information. Including RVS in the HIP RR may impact the TTL for the response.

4.3.5. Changes to ORCHIDv2 to support Hierarchical HITs

ORCHIDv2 [RFC7343] has a number of inputs including a Context ID (unique for HHITs), some header bits, the hash algorithm, and the public key. The output is a 96 bit value. Hierarchical HIT makes the following changes. The HID is added as part of the header bits and the output is a 64 bit value, derived the same way as the 96 bit hash.

Input := HID | HOST_ID
OGA ID := 4-bit Orchid Generation Algorithm identifier
        The HHIT Suite ID
Hash Input := Context ID | Input
              Context ID = 0x00B5 A69C 795D F5D5
              F008 7F56 843F 2C40
Prefix := HIPv2 Prefix
         Note: per section 4.1, this should be a different Prefix
HID := Hierarchy ID
Hash := Hash_function( Hash Input )
Encode_64 := Same as Encode_96, but only 64 bits
ORCHID := Prefix | OGA ID | HID | Encode_64( Hash )

Hierarchical HIT uses the same context as all other HIPv2 HIT Suites as they are clearly separated by the distinct HIT Suite ID.

4.3.6. Collision risks with Hierarchical HITs

The 64 bit hash size does have an increased risk of collisions over the 96 bit hash size used for the other HIT Suites. There is a 0.01% probability of a collision in a population of 66 million. The probability goes up to 1% for a population of 663 million. See Appendix A for the collision probability formula.

However, this risk of collision is within a single HDA. Further, all HDAs are expected to provide a registration process for reverse lookup validation. This registration process would reject a collision, forcing the client to generate a new HI and thus hierarchical HIT and reapplying to the registration process.
5. IANA Considerations

Because HHIT use of ORCHIDv2 format is not compatible with [RFC7343], IANA is requested to allocate a new 28-bit prefix out of the IANA IPv6 Special Purpose Address Block, namely 2001:0000::/23, as per [RFC6890].

6. Security Considerations

A 64 bit hash space presents a real risk of second pre-image attacks. The HHIT Registry services effectively block attempts to "take over" a HHIT. It does not stop a rogue attempting to impersonate a known HHIT. This attack can be mitigated by the Responder using DNS to find the HI for the HHIT or the RVS for the HHIT that then provides the registered HI.

The two risks with hierarchical HITs are the use of an invalid HID and forced HIT collisions. The use of the "hhit.arpa." DNS zone is a strong protection against invalid HIDs. Querying an HDA's RVS for a HIT under the HDA protects against talking to unregistered clients. The Registry service has direct protection against forced or accidental HIT hash collisions.

7. Acknowledgments

The initial versions of this document were developed with the assistance of Xiaohu Xu and Bingyang Liu of Huawei.

Sue Hares contributed to the clarity in this document.

8. References

8.1. Normative References


8.2. Informative References

Appendix A. Calculating Collision Probabilities

The accepted formula for calculating the probability of a collision is:

\[ p = 1 - e^{-k^2/(2n)} \]

<table>
<thead>
<tr>
<th>P</th>
<th>Collision Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>Total possible population</td>
</tr>
<tr>
<td>k</td>
<td>Actual population</td>
</tr>
</tbody>
</table>

Authors’ Addresses

Robert Moskowitz
HTT Consulting
Oak Park, MI 48237