Host Identifier Revocation in HIP
draft-zhang-hip-hi-revocation-00

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on November 27, 2009.

Copyright Notice

Copyright (c) 2009 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 in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.
Abstract

This document mainly analyzes the key revocation issue with host identities (HI) in the Host Identity Protocol (HIP), which has not attracted enough attention from HIP community yet. As a core component of key management mechanism, key revocation is critical for security systems especially which are expected to execute for a long period. Apart from that, this document also discusses the possible challenges that the designers of HI revocation mechanisms have to face and introduces several possible solutions.

Conventions used in this document

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].

Table of Contents

1. Introduction .................................................. 2
2. Key Management.................................................. 3
3. HI Revocation in HIP............................................ 4
   3.1. Implicit HI Revocation in HIP............................ 4
   3.2. Explicit HI Revocation without Third Parties .......... 6
   3.3. Explicit HI Revocation with Third Parties ............. 6
4. Conclusions .................................................... 7
5. IANA Considerations.......................................... 7
6. Acknowledgments............................................... 8
7. References .................................................... 8

Authors’ Addresses ............................................. 8

1. Introduction

Key management aims at guaranteeing the security of cryptographic keys during the period of their application and includes all of the provisions made in a security system design which are related to generation, validation, exchange, storage, safeguard, application, and replacement of cryptographic keys. Appropriate key management is critical to security systems. Generally, a key management mechanism can be broken down into five sub-systems: key generation, key maintenance, key distribution, key assessment, and key revocation.

Key generation systems concern the issues related with key generating processes. Key maintenance systems deal with the issues with key storage. Key distribution systems deal with the issues with
regard to distributing cryptographic keys for users. Key assessment systems take in charge of verifying the legality of cryptographic keys according to certain policies. Key revocation systems cover the issues with regard to updating cryptographic keys. In these sub-systems, key revocation is most complex and covers most functions of the other four sub-systems. For instance, in a typical key revocation process, antique keys need to be abandoned and securely discarded; new keys need to be generated, assessed and distributed; the users supposed to be affected by the change of keys need to be notified.

Generally, there are two types of key revocation approaches: implicit key revocation and explicit key revocation. Indicated by the name, implicit key revocation does not need any additional operations to revoke a cryptographic key. For instance, in PKI, a Certificate Authority (CA) can issue a certificate for a user in order to assert the relationship between the user and its public key. The certificate is associated with a life period. When the period expires, the user’s public key is revoked as well. An explicit key revocation mechanism, in contrast, needs an assigned entity such as the owner a key to explicitly deliver notification when the key is revoked for certain reasons. For instance, in X.500 based systems, an issuer can generates a list of certificates, which have been revoked but not expired yet, for users to consult.

2. Key Management

Key management aims at guaranteeing the security of cryptographic keys during the period of their application and includes all of the provisions made in a security system design which are related to generation, validation, exchange, storage, safeguard, application, and replacement of cryptographic keys. Appropriate key management is critical to security systems. Generally, a key management mechanism can be broken down into five sub-systems: key generation, key maintenance, key distribution, key assessment, and key revocation.

Key generation systems concern the issues related with key generating processes. Key maintenance systems deal with the issues with key storage. Key distribution systems deal with the issues with regard to distributing cryptographic keys for users. Key assessment systems take in charge of verifying the legality of cryptographic keys according to certain policies. Key revocation systems cover the issues with regard to updating cryptographic keys. In these sub-systems, key revocation is most complex and covers most functions of the other four sub-systems. For instance, in a typical key revocation process, antique keys need to be abandoned and securely...
Generally, there are two types of key revocation approaches: implicit key revocation and explicit key revocation. Indicated by the name, implicit key revocation does not need any additional operations to revoke a cryptographic key. For instance, in PKI, a Certificate Authority (CA) can issue a certificate for a user in order to assert the relationship between the user and its public key. The certificate is associated with a life period. When the period expires, the user's public key is revoked as well. An explicit key revocation mechanism, in contrast, needs an assigned entity such as the owner a key to explicitly deliver notification when the key is revoked for certain reasons. For instance, in X.500 based systems, an issuer can generates a list of certificates, which have been revoked but not expired yet, for users to consult.

3. HI Revocation in HIP

The revocation of transient keys has been considered in the design of HIP. In the basic handshaking protocol, communicating hosts employ the authenticated Diffie-Hellman algorithm to distribute a symmetric key which is used to generate new cryptographic keys adopted in following communications. After the handshake, the hosts are able to refresh their transient keys and the corresponding associations, using update messages. The revocation issue with HIs, however, has not been carefully considered yet. In current HIP architecture, hosts are able to update their identifiers arbitrarily without notifying others. This loophole can be taken advantage of by attackers to, for instance, escape tracing or bypass ACLs. In this section, candidate approaches and related issues are discussed.

3.1. Implicit HI Revocation in HIP

The basic idea of implicit HI revocation is to associate a key with a valid period and use cryptographic methodology to prove the combination between the key and its valid period. In practice, the valid period of a key can be transported in HIP headers either using a parameter or a certificate. The issuer of the key can calculate a signature for the key and its valid period. By verifying the signature, users can assert whether the key is still in its valid period.

Although the principle behind implicit key revocation is quite intuitive and there have been lots of successful experience in this
area, there are still many practical challenges. For example, in current HIP, hosts have a full control over its permanent key pairs. A host can arbitrarily assign, reset, or extend the life time of its HIs, which may cause serious security issues. Assume that a lazy manager of a host sets an extremely long life period for the HI of the host to avoid the jobs of updating key pairs. Then attackers may be able to obtain enough time to compromise the key. A practical solution to this problem is to allocate a trusted third party to manage the life period for each HI, and the assertion of the third party should be able to be verified by users.

However, such a security infrastructure may become a bottleneck of the Internet. The infrastructure is expected to serve all the end systems attached to the Internet in a global scale. Currently, there have been already billions of hosts attached to the Internet. Considering the other kinds of devices (e.g., mobile phones, network printers, and even human beings) which has been connected or are going to be connected to the Internet, the number of the users that the infrastructure needs to server is even larger. Up till now, no security infrastructure ever deployed has the capability to effectively support such a large amount of users. In addition, the Internet is comprised of multiple networks which belong to different countries or organizations. As there is not a unique trust domain all over the Internet, two hosts intending to communicate may belong to two different trust domains, and each host may not trust the assertion made by the security server in the other domain. Although there have been solutions of generating trust relationship across various trust domains, all these solution imposes additional communications and key exchanges, which will negatively influence the performance of HIP. Therefore, the HIP community argues that two HIP-aware hosts should be able to communicate without any additional security facilities. Actually, the only third party server introduced in the base-line HIP architecture is the Rendezvous Server (RVS). A RVS only relays messages for the hosts which attempts to communicate with mobile hosts and provides little security functionality. The hosts intending to communicate still need to use the basic handshaking protocol to carry out authentication and exchange secrets.

It needs to be clarified that we do not oppose integrating additional security infrastructures with HIP. We just argue that before the integration the issues mentioned above need to be carefully considered.

Compared with explicit key revocation mechanisms, implicit key revocation mechanisms introduce less overhead. No additional
information is needed to notify the revocation of antique keys. However, implicit key revocation mechanisms are incapable in the scenarios where keys need to be revoked for some reasons (e.g., compromise) before they expire. In many security architectures (e.g., X.509 [RFC 2459]), both implicit and explicit key revocation are supported.

3.2. Explicit HI Revocation without Third Parties

In a scenario where there is no assistance from trusted third parties, a host refreshing its HI needs to notify the update to its collaborating peers. Cryptographic methodologies must be used to prove that the notification messages cannot be foiled, modified, replayed so as to prevent attackers from using these messages to perform attacks. An intuitive approach is to define a new type of update packet. When a HI is revoked, the host carries out basic handshakes with its peers, establishes security channels and then transports the update message through the channel to notify the revocation. If a host is communicating with its peers at the time when the revocation is performed, the host can use the established security channels to transport update packets directly.

If a host only has a small group of collaborating partners and the relationship between the host and its partners is stable, the solution discussed above is effective. However, if the number of partners is large or changes frequently, the host has to maintain a long list of partner information, and send a large amount of messages to inform its partners of the revocation, not to even mention that candidate partners are unpredictable on many occasions. In addition, the job of maintaining partner lists normally needs the involvement of human beings which are error-prone. Any omission of candidate partners will cause difficulties or failures in future communications. Moreover, this type of solution requires all the partners to be available during the period of revocation. Otherwise, the host will miss the updating information. This requirement, however, are difficult to fulfill, especially when hosts communicate in asynchronous style.

3.3. Explicit HI Revocation with Third Parties

The notification approach mentioned above uses a "push" model, if a host needs to communicate with a large number of peers, the host may suffer from the overhead introduced by the revocation of its HI. In this case, a "pull" model may show its advantage. That is, deploy a trust third party for hosts to upload their updated HI information and for users to query whenever they need. DNS can be a good
candidate to implement such a model. When a host refreshes its HI, it can update its Resource Record (RR) on a DNS server. Other users then can use the host’s FQDN to find out its latest HI. Security extensions of DNS have been developed, which is able to ensure that only authorized hosts can update their RRs. This solution works well on many occasions. However, in some scenarios, the service of mapping from antique HIs to latest HIs is required. For instance, DNS allows a FQDN to be mapped to multiple hosts in order to facilitate load balance. Each time when receiving a query about the FQDN, the DNS server returns the addressing information about a different host. Therefore, when a user queries a DNS server with the FQDN of the desired host and obtains a HIT different from the previous one, it is difficult for the host to find out whether the HI it gains is the updated HI of the desired host or the HI of another host. The later condition may result in errors as the host indicated by the HI does not have the state information maintained by the desired host. In other resolution mechanisms which facilitate mapping from HIT to IP address (e.g., DHT), the mapping service from the old HI to updated HI is more important as users have no other namespace to rely on.

4. Conclusions

If HIP is only supposed to be an ID/Locator separating solution and security is dispensable, key revocation may not be a critical issue on many occasions. However, if HIP is expected to keep working securely for a relatively long period, proper key management mechanisms have to be provided. In fact, key management has been an active research area for a long period, and lots of successful key-management systems (e.g., PKI) are widely adopted in practice. However, because of many issues (e.g., scalability, lack of trust), it is undesired to employ these key-management systems for HIP directly.

5. IANA Considerations

No such considerations.

6. Acknowledgments

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


Authors’ Addresses

Dacheng Zhang  
Huawei Technologies Co., Ltd  
KuiKe Building, No.9 Xinxi Rd., Hai-Dian District  
Beijing, 100085  
P.R. China  
Email: zhangdacheng@huawei.com

Xiaohu Xu  
Huawei Technologies Co., Ltd  
KuiKe Building, No.9 Xinxi Rd., Hai-Dian District  
Beijing, 100085  
P.R. China  
Email: xuxh@huawei.com