Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 10, 2009

A. Kukec
University of Zagreb
S. Krishnan
Ericsson
S. Jiang
Huawei Technologies Co., Ltd
March 9, 2009

SeND Hash Threat Analysis
draft-ietf-csi-hash-threat-03

Status of this Memo

Copyright Notice

Abstract

This document analysis the use of hashes in SeND, possible threats and the impact of recent attacks on hash functions used by SeND. Current SeND specification [rfc3971] uses the SHA-1 [sha-1] hash algorithm and PKIX certificates [rfc5280] and does not provide support for the hash algorithm agility. The purpose of the document is to provide analysis of possible hash threats and to decide how to encode the hash agility support in SeND.

Table of Contents

1. Introduction

SEND [rfc3971] uses the SHA-1 hash algorithm to generate the contents of the Key Hash field and the Digital Signature field of the RSA Signature option. It also uses a hash algorithm (SHA-1, MD5, etc.) in the PKIX certificates [rfc5280] used for the router authorization in the ADD process. Recently there have been demonstrated attacks against the collision free property of such hash functions [sha1-coll], and attacks on the PKIX X.509 certificates that use the MD5 hash algorithm [x509-coll] This document analyzes the effects of those attacks and other possible hash attacks on the SEND protocol. The document proposes changes to make it resistant to such attacks.

2. Terminology

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].

3. Impact of collision attacks on SeND

Due to the hash attacks demonstrated on the aforesaid hash algorithms a study was performed to assess the threat of these attacks on the cryptographic hash usage in internet protocols [RFC4270]. This document analyzes the hash usage in SEND following the approach recommended by [rfc4270] and [new-hashes].

The basic cryptographic properties of a hash function are that it is both one-way and collision free. There are two attacks against the one-way property, the first-preimage attack and the second-preimage attack. In the first-preimage attack, given a knowledge of a particular value hash(m), an attacker finds an input message m' such that hash(m') yields hash(m). The second-preimage attack deals with the fixed messages. Given a knowledge of a fixed value m used as the input message to the hash function, an attacker finds a different value m' that yields hash(m)=hash(m'). Supposing that the hash function produces an n-bit long output, since each output is equally likely, an attack takes an order of 2^n operations to be successful. Due to the birthday attack, if the hash function is supplied with a random input, it returns one of the k equally-likely values, and the number of operations can be reduced to the number of 1.2*2^(n/2) operations. However, attacks against the one-way property are not yet feasible [rfc4270]. Up to date, all demonstrated attacks are attacks against a collision-free property, in which an attacker produces two different messages m and m' such that hash(m)=hash(m').

We will analyze the impact of hash attacks on SeND case by case in this section. Through our analysis, we also discuss whether we should support the hash agility in SeND.

3.1. Attacks against CGAs in stateless autoconfiguration

Hash functions are used in the stateless autoconfiguration process that is based on CGAs. Impacts of collision attacks on current uses of CGAs are analyzed in the update of the CGA specification [rfc4982], which also enables CGAs to support the hash agility. CGAs provide the proof-of-ownership of the private key corresponding to the public key used to generate the CGA. CGAs do not deal with the non-repudiation feature, while collision attacks are mainly about affecting the non-repudiation feature, i.e. in the collision attack against the CGA both of the CGA Parameters sets are choosen by an attacker, which is not useful in the real-world scenarios. Therefore, as [rfc4982] points out CGA based protocols, including SeND, are not affected by the recent collision attacks. Regarding the pre-image attacks, if pre-image attacks were feasible, an attacker would manage to find the new CGA Parameters based on the associated, victim's CGA, and produce the Key Hash field and the

Digital Signature field afterwards using the new public key. Since the strength of all hashes in SEND depends on the strength of the CGA, the pre-image attack is potentially dangerous, but it is not yet feasible.

3.2. Attacks against PKIX certificates in ADD process

The second use of hash functions is for the router authorization in the ADD process. Router sends to a host a certification path, which is a path between a router and the hosts's trust anchor, consisting of PKIX certificates. Researchers demonstrated attacks against PKIX certificates with MD5 signature, in 2005 [new-hashes] and in 2007 [X509-COLL]. In 2005 were constructed the original and the false certificate that had the same identity data and the same digital signature, but different public keys [new-hashes]. The problem for the attacker is that two certificates with the same identity are not actually useful in real-world scenarios, because the Certification Authority is not allowed to provide such two certificates. In addition, attacks against the human-readable fields demand much more effort than the attacks against non human-readable fields, such as a public key field. In case of the identity field, an attacker is faced with the problem of the prediction and the generation of the false but meaningful identity data, which at the end might be revealed by the Certification Authority. Thus, in practice, collision attacks do not affect non human-readable parts of the certificate. In 2007 were demonstrated certificates which differ in the identity data and in the public key, but still result in the same signature value. In such attack, even if an attacker produced such two certificates in order to claim that he was someone else, he still needs to predict the content of all fields appearing before the public key, e.g. the serial number and validity periods. Although a relying party cannot verify the content of these fields (each certificate by itself is unsuspicious), the Certification Authority keeps track of those fields and it can reveal the false certificate during the fraud analysis. Regarding certificates in SeND, potentially dangerous are attacks against the X.509 certificate extensions. For example, an attack against the IP address extension would enable the router to advertize the changed IP prefix range, although, not broader than the prefix range of the parent certificate in the ADD chain.

The public-private key pair associated to the Router Authorization Certificate in the ADD process is used both for the CGA generation and for the message signing. Since in the future CGAs might be used only with certificates, attacks against certificates are even more dangerous. Generally, the most dangerous are attacks against middle-certificates in the certification path, where for the cost of the one false certificate, an attacker launches an attack on multiple

routers. Regarding the attacks against certificates in SEND, the only attack that SEND is not suspectable to, is an attack against the Trust Anchor's certificate because it is not exchanged in SeND messages, i.e. it is not the part of the certification path in the ADD process.

3.3. Attacks against the Digital Signature in the SEND Universal Signature option

The SEND Universal Signature option is an updated version of the RSA Signature option, defined in [sig-agility]. In combination with the public key agility support described in [pk-agility], it allows the node to use the public key signing algorithm different then the RSA-based signing algorithm. No matter of the type of the SEND Universal Signature option, the Digital Signature field is computed in the same way as the Digital Signature field of the RSA Signature option descibed in [rfc3971]. The digital signature in the RSA Signature option is produced as the SHA-1 hash over the IPv6 addresses, the ICMPv6 header, the ND message and other fields, e.g. the Message Type Tag and ND options [rfc3971], that is signed with the sender's private key. The sender's private key corresponds to the public key in the CGA parameters structure. It is usually authorized through CGAs. The possible attack on such explicit digital signature is a typical non-repudiation attack, i.e. the Digital Signature field is vulnerable to the collision attack. An attacker produces two different messages, m and m', where hash(m) = hash(m'). He underlays one of the messages to be signed with the key authorized through CGAs, but uses another message afterwards. However, as denoted in [rfc4270], the structure of at least one of two messages in a collision attack is strictly predefined. The previous requirement complicates the collision attack, but we have to take into account that a variant of SHA-1 was already affected by recent collision attacks and we have to prepare for future improved attacks.

3.4. Attacks against the Key Hash in the SEND Universal Signature option

The Key Hash field in the SEND Universal signature option is a SHA-1 hash of the public key from the CGA Parameters structure in the CGA option. The pre-image attack against the Key Hash field is potentially dangerous, as in the case of all other hashes in SEND, because the Key Hash field contains a non human-readable data. However the Key Hash field is not suspectable to the collision attack, since in the collision attack an attacker itself chooses both keys, k and k', where hash(k) = hash(k'). The reason for the former is that the associated public key is already authorized through the use of CGAs, and possibly the certification path in the ADD process.

4. Support for the hash agility in SeND

While all of analyzed hash functions in SeND are theoretically affected by hash attacks, these attacks indicate the possibility of future real-world attacks. In order to increase the future security of SeND, we suggest the support for the hash and algorithm agility in SeND.

- The most effective and secure would be to bind the hash function option with something that can not be changed at all, like [rfc4982] does for CGA - encoding the hash function information into addresses. But, there is no possibilty to do that in SeND. We could decide to use by default the same hash function in SeND as in CGA. The security of all hashes in SeND depends on CGA, ie. if an attacker could break CGA, all other hashes are automatically broken. From the security point of view, at the moment, this solution is more reasonable then defining different hash algorithm for each hash. Additionally, if using the hash algorithm from the CGA, no bidding down attacks are possible. On the other hand, this solution introduces the limition for SEND to be used exclusively with CGAs.

- Another solution is to incorporate the Hash algorithm option into the SeND message. However, if the algorithm is defined in the Hash algorithm option in the SeND message, it is vulnerable to the bidding down attack.

- The third possible solution is to encode the algorithm in the CGA. However, this will reduce the strength of the CGAs and make them vulnerable to brute force attacks.

- One of the possible solutions is also the hybrid solution, i.e. to require the hash algorithm to be the same as CGA, if CGA option is present, and to use the Hash agility option if the CGA option is not present.

4.1. The negotiation approach for the SEND hash agility

None of the previous solutions supports the negotiation of the hash function. Therefore we propose the negotiation approach for the SEND hash agility based on the Supported Signature Algorithm option described in [sig-agility]. Based on the processing rules described in [sig-agility] nodes find the intersection between the sender's and the receiver's supported signature algorithms set, as well as the preferred signature algorithm. When producing and validating hashes in SEND, nodes MUST observe the following rules:

- In the ADD process, if any of the certificates in the certification path uses the signature algorithm which is not one of the signature algorithms negotiated in the signature agility process through the use of the Supported Signature Algorithms option, nodes MUST reject the Router Authorization certificate.

- In order to produce the Digital Signature field, nodes MUST use the signature algorithm negotiated in the signature agility process through the use of the Supported Signature Algorithms option.

- In order to produce the Key Hash field, nodes MUST use the hash algorithm defined associated to the signature algorithm negotiated in the signature agility process through the use of the Supported Signature Algorithms option.

5. Security Considerations

This document analyzes the impact of hash attacks in SeND and offeres a higher security level for SeND by providing solution for the hash agility support.

The negotiation approach for the hash agility in SeND based on the Supported Signature Algorithms option is vulnerable to bidding-down attacks, which is usual in the case of any negotiation approach. This issue can be mitigated with the appropriate local policies.

6. References

6.1. Normative References

[new-hashes]
Bellovin, S. and E. Rescorla, "Deploying a New Hash Algorithm", November 2005.

[pk-agility]
Cheneau, T., Maknavicius, M., Sean, S., and M. Vanderveen, "Support for Multiple Signature Algorithms in Cryptographically generated Addresses (CGAs)", draft-cheneau-cga-pk-agility-00 (work in progress), February 2009.

[rfc3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[rfc4270] Hoffman, P. and B. Schneier, "Attacks on Cryptographic Hashed in Internet Protocols", RFC 4270, November 2005.

[rfc4982] Bagnulo, M. and J. Arrko, "Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs)", RFC 4982, July 2007.

[sig-agility]
Cheneau, T. and M. Maknavicius, "Signature Algorithm Agility in the Secure Neighbor Discovery (SEND) Protocol", draft-cheneau-send-sig-agility-00 (work in progress), February 2009.

6.2. Informative References

[rfc2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[rfc5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC rfc5280, May 2008.

[sha-1] NIST, FIBS PUB 180-1, "Secure Hash Standard", April 1995.

[sha1-coll]
Wang, X., Yin, L., and H. Yu, "Finding Collisions in the Full SHA-1. CRYPTO 2005: 17-36", 2005.

[x509-coll]

Stevens, M., Lenstra, A., and B. Weger, "Chosen-Prefix Collisions for MD5 and Colliding X.509 Certificates for Different Identitites. EUROCRYPT 2007: 1-22", 2005.

Authors' Addresses

Ana Kukec
University of Zagreb
Unska 3
Zagreb
Croatia

Email: ana.kukec@fer.hr

Suresh Krishnan
Ericsson
8400 Decarie Blvd.
Town of Mount Royal, QC
Canada

Email: suresh.krishnan@ericsson.com

Sheng Jiang
Huawei Technologies Co., Ltd
KuiKe Building, No.9 Xinxi Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing
P.R. China

Email: shengjiang@huawei.com