Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 16, 2010

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

February 12, 2010

SEND Hash Threat Analysis
draft-ietf-csi-hash-threat-07

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 X.509 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.

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 August 16, 2010.

Copyright Notice

Copyright (c) 2010 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 (http://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 BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction

SEND [rfc3971] uses the SHA-1 hash algorithm to generate Cryptographically Generated Addresses (CGA) [rfc3972], 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.) within the digital signature in X.509 certificates [rfc5280] for the router authorization in the Authorizaton Delegation Discovery (ADD) process.

There is a great variaty of hash functions, but only MD5 and SHA-1 are in the wide use, which is also the case for SEND. They both derive from MD4, which has been well known for its weaknesses. First hash attacks affected the compression function of MD5, while the latest hash attacks against SHA variants delivered colliding hashes in significantlly smaller number of rounds compared to the brute force attack number of rounds [sha1-coll]. Apart from the aforementioned hash attacks, researchers also demonstrated attacks against X.509 certificates. They demonstrated colliding X.509 certificates with MD5 hash, both with the same and different distinguished names [new-hashes] [x509-coll].

Depending on the way how the Internet protocol uses the hash algorithm, Internet protocol can be affected by the weakness of the underlaying hash function. This document analyzes uses of hash algorithms in SEND, possible vulnerabilities that hash attacks could introduce to SEND, and offers suggestions on how to make SEND 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. This document analyzes the hash usage in SEND following the recommended approach [rfc4270] [new-hashes].

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 hash value h, an attacker finds an input message m such that hash(m) = h. The second-preimage attack deals with 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. Attack against the collision-free property deals with two fixed messages, both produced by an attacker. What happens is that the attacker produces two different messages, m and m', such that hash(m)=hash(m'). Up to date, all demonstrated attacks are attacks against a collision-free property. Attacks against the one-way property are not yet feasible [rfc4270].

The strength of Internet protocol does not have to be necessarily affected by the weakness of the underlaying hash function. The appropriate way of use of the hash algorithm will keep the protocol immune, no matter of the hash algorithm weaknesses. Out of many possible hash algorithm uses, such as non-repudiable digital signatures, certificate digital signatures, message authentication with shared secrets, fingerprints, only the first two can introduce weaknesses to the Internet protocol [rfc4270]. The rest of the section analyzes the impact of hash attacks, mainly collision attacks, on SEND by the cases of use. 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 which is based on CGAs. Impacts of collision attacks on current uses of CGAs and the CGA hash agility are analyzed in the update of the CGA specification [rfc4982]. CGAs provide the proof-of-ownership of the sender's private key corresponding to the public key used to

generate the CGA. Simply stated, their main purpose is to assure that the sender of the message is the same as the sender of the previous message. As such, CGAs do not deal with the non-repudiation feature. The collision attack against the CGA assumes that the attacker generates two different, colliding sets of CGA Parameters that result in the same hash value. Since CGAs do not deal with the non-repudiation feature, and both CGA Parameters sets are chosen by the attacker itself, this attack does not introduce any vulnerabilities to SEND. If pre-image attacks were feasible, an attacker would find colliding CGA Parameters for the 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 X.509 certificates in ADD process

Another 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 host's trust anchor, consisting of X.509 certificates. Researchers demonstrated attacks against X.509 certificates with MD5 signature in 2005 [new-hashes] and in 2007 [x509-coll]. In 2005 researchers constructed colliding certificates with the same distinguished name, different public keys, and identical signatures. Potential problem for the attacker here is that two certificates with the same identity can be easily revealed by the appropriately configured Certification Authority that does not allowe to provide two certificates with the same identities. Human-readable fields significantly complicate the attack. In case of the identity field an attacker is faced with the problem of the prediction and the generation of the two different, false but meaningful identities, which at the end might be revealed by the Certification Authority. Thus, although theoretically possible, real-world circumstances such as the context of the human-readable fields, make these attacks with colliding certificates with the same identities impossible. In 2007 researchers demonstrated colliding certificates which differ in the identity data and in the public key, but still result in the same signature value. Even in this case, the real-world scenarios prevent the hash algorithm weaknesses to introduce vulnerabilities to X.509 certificates or to SEND. Even if an attacker produced such two colliding certificates in order to claim that he was someone else, he still needs to predict the content of all fields (some of them are human-readable 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. Even though real-world

scenarios make SEND immune to recent hash attacks introduced through X.509 certificates, theoretically they are possible. Regarding X.509 certificates in SEND, biggest concer are potential attacks against the RFC3779 IP address extension which would enable the bogus router to advertize the changed IP prefix range (if the IP prefix range used), although, not broader than the prefix range of the parent certificate in the ADD chain. Adding some form of randomness to the such human-readble data such would prevent attacks, which can be considered once when the collision attack improve.

3.3. Attacks against the Digital Signature in the RSA Signature option

The computation of the Digital Signature field is described [rfc3971]. It 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, and signed with the sender's private key. Private key corresponds to the public key in the CGA parameters structure. It is usually authorized through CGAs. The Digital Signature field the example of the non-repudiation digital singature, and it is vulnerable to recent collision attacks. Possible attacks on such explicit digital signature is a typical non-repudiation attack in which the 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, the structure of at least one of two messages in a collision attack is strictly predefined. The previous requirement makes this collision attack to be much more then the simple collision attack. It requires the attacker to know or predict the communication context. Theoretically this attack could harm SEND, but in real-world situation is to achieve it.

3.4. Attacks against the Key Hash field in the RSA Signature option

The Key Hash field in the RSA Signature option is a SHA-1 hash of the public key from the CGA Parameters structure in the CGA option. It is a fingerprint that provides the integrity protection. Fingerprints are generally not affected by the collision attacks because they involve random data as one of the inputs, which prevents recent collision attacks. In addition, context of the SEND message and the protocol makes this attack unable to introduce new vulnerabilities to SEND. An attacker has to produce both keys, k and k', such that hash(k) = hash(k'). Since the key is authorized through CGA, and possibily through the certification in the ADD process, this attack is of no use for the attacker. The pre-image attack against the Key Hash field, if it was possible, would affect SEND since the Key Hash field contains a non human-readable data.

4. Support for the hash agility in SEND

Previous section showed that recent hash attacks against CGAs and fingerprints (Key Hash field of the Send message) do not introduce new vulnerabilities to SEND. Digital signatures in the Digital Signature field of the SEND message and in the X.509 certificate theoretically could introduce new vulnerabilities to SEND, but only in limited circumstances. SEND context prevents those attacks of almost any use in the real-world scenarios.

However, recent attacks indicate the possibility for the future improved real-world attacks. Researchers advise to migrate away from currently used hash algorithms. In November 2007, NIST announced an opened competition for a new SHA-3 function. The selection of a winning function will be in 2012. 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. It encodes the hash function information into addresses. 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, i.e. if an attacker breaks CGA, all other hashes are automatically broken. The use of the hash algorithm embedded in CGA protects from the bidding down attacks. From the security point of view, at the moment, this solution is more reasonable then defining different hash algorithm for each hash. The disadvantage of this solution is that it introduces the limitation for SEND to be used exclusively with CGAs.

- Another solution is to incorporate the Hash algorithm option into the SEND message. This solution is vulnerable to the bidding down attack.

- The third possible solution is to encode the algorithm in the CGA. This would reduce the strength of the CGA and make it vulnerable to brute force attacks.

- Possible solution is also the hybrid solution which would 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. In such way, SEND is not bound exclusively to CGA.

- None of the previous solutions supports the negotiation of the hash function. One of possible solutions is 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.

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.

[rfc3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, 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-01 (work in progress), May 2010.

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