

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 21, 2013

M. Bhatia
Alcatel-Lucent
D. Zhang
Huawei Technologies co., LTD.
M. Jethanandani
Ciena Corporation
October 18, 2012

Analysis of Bidirectional Forwarding Detection (BFD) Security According
to KARP Design Guide
draft-bhatia-zhang-karp-bfd-analysis-03

Abstract

This document analyzes the Bidirectional Forwarding Detection protocol (BFD) according to the guidelines set forth in section 4.2 of KARP Design Guidelines [RFC6518].

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 April 21, 2013.

Copyright Notice

Copyright (c) 2012 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 Simplified BSD License.

1. Introduction

This document performs a gap analysis of the current state of Bidirectional Forwarding Detection [RFC5880] according to the requirements of KARP Design Guidelines [RFC6518]. Previously, the OPSEC working group has provided an analysis of cryptographic issues with BFD in Issues with Existing Cryptographic Protection Methods for Routing Protocols [RFC6039].

The existing BFD specifications provide a basic security solution. Key ID is provided so that the key used in securing a packet can be changed on demand. Two cryptographic algorithms (MD5 and SHA-1) are supported for integrity protection of the control packets; the algorithms are both demonstrated to be subject to collision attacks. Routing protocols like RIPv2 Cryptographic Authentication [RFC4822], IS-IS Generic Cryptographic Authentication [RFC5310] and OSPFv2 HMAC-SHA Cryptographic Authentication [RFC5709] have started to use BFD for liveness check. Moving the routing protocols to a stronger algorithm while using weaker algorithm for BFD would require the attacker to bring down BFD in order to bring down the routing protocol. BFD therefore needs to match the routing protocols in its strength of algorithm.

While BFD uses a non-decreasing per-packet sequence number to protect itself from intra-connection replay attacks, it still leaves the protocol vulnerable to the inter-session replay attacks.

1.1. 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].

2. Requirements to Meet

There are several requirements described in section 3 of The Threat Analysis and Requirements for Cryptographic Authentication of Routing Protocols' Transports [I-D.ietf-karp-threats-reqs] that BFD does not currently meet:

Replay Protection: BFD provides an incomplete intra-session and no inter-session replay attack protection; this creates significant denial-of-service opportunities.

Strong Algorithms: the cryptographic algorithms adopted for message authentication in BFD are MD5 or SHA-1 based. However, both algorithms are known to be vulnerable to collision attacks. BFD Generic Cryptographic Authentication [I-D.ietf-bfd-generic-crypto-auth] and Authenticating BFD using HMAC-SHA-2 procedures [I-D.ietf-bfd-hmac-sha] together propose a solution to support HMAC with the SHA-2 family of hash functions for BFD.

DoS Attacks: BFD packets can be sent at millisecond intervals (the protocol uses timers at microsecond intervals). When malicious packets are sent at short intervals, with the authentication bit set, it can cause a DoS attack.

The remainder of this document explains the details of how these requirements fail to be met and proposes mechanisms for addressing them.

3. Current State of Security Methods

BFD [RFC5880] describes five authentication mechanisms for the integrity protection of BFD control packets: Simple Password, Keyed MD5 The MD5 Message-Digest Algorithm [RFC1321], Meticulous Keyed MD5, Keyed SHA-1 and Meticulous SHA-1. In the simple password mechanism, every control packet is associated with a password transported in plain text; attacks eavesdropping the network traffic can easily learn the password and compromise the security of the corresponding BFD session. In the Keyed MD5 and the Meticulous Keyed MD5 mechanisms, BFD nodes use share secret keys to generate keyed MD5 digests for control packets. Similarly, in the Keyed SHA-1 and the Meticulous Keyed SHA-1 mechanisms, BFD nodes use shared secret keys to generate keyed SHA-1 digests for control packets. Note that in the keyed authentication mechanisms, every BFD control packet is associated with a non-decreasing 32-bit sequence number to resist replay attacks. In the Keyed MD5 and the Keyed SHA-1 mechanisms, the sequence member is only required to increase occasionally. However, in the Meticulous Keyed MD5 and the Meticulous Keyed SHA-1 mechanisms, the sequence member is required to monotonically increase with each successive packet.

Additionally, limited key updating functionality is provided. There is a Key ID in every authenticated BFD control packet, indicating the key used to hash the packet. However, there is no mechanism described to provide a smooth key rollover that the BFD routers can use when moving from one key to the other.

The BFD session timers are defined with the granularity of microseconds, and it is common in practice to send BFD packets at millisecond intervals. Since the cryptographic sequence number space is only 32 bits, a sequence number used in a BFD session may reach its maximum value and roll over within limited period. For instance, if a sequence number is increased by one every 3.3 millisecond, then it will reach its maximum value in less than 24 weeks. This can result in potential inter-session replay attacks especially when BFD uses the non-meticulous authentication modes.

Note that when using authentication mechanisms, BFD requests the sequence of a received BFD packets drops with a limited range (3* Detection time multiplier). Therefore, when meticulous authentication modes are used, a replayed BFD packet will be rejected if it cannot fit into a relatively short window (3 times of the detect interval of the session). This introduces some difficulties for replaying packets. However, in a non-meticulous authentication mode, such windows can be large as sequence numbers are only increased occasionally, thus making it easier to perform replay attacks .

In a BFD session, each node needs to select a 32-bit discriminator to identify itself. Therefore, a BFD session is identified by two discriminators. If a node will randomly select a new discriminator for a new session and use authentication mechanism to secure the control packets, inter-session replay attacks can be mitigated to some extent. However, in existing BFD demultiplexing mechanisms, the discriminators used in a new BFD session may be predictable. In some deployment scenarios, the discriminators of BFD routers may be decided by the destination and source addresses. So, if the sequence number of a BFD router rolls over for some reasons (e.g., reboot), the discriminators used to identify the new session will be identical to the ones used in the previous session. This makes performing a replay attack relatively simple.

BFD allows a mode called the echo mode. Echo packets are not defined in the BFD specification, though they can keep the BFD session up. The format of the echo packet is local to the sending side and there are no guidelines on the properties of these packets beyond the choice of the source and destination addresses. While the BFD specification recommends applying security mechanisms to prevent spoofing of these packets, there are no guidelines on what type of mechanisms are appropriate.

4. Impacts of BFD Replays

As discussed, BFD cannot meet the requirements of inter-session or intra-session replay protection. This section discusses the impacts of BFD replays.

When cryptographic authentication mechanisms are adopted for BFD, a non-decreasing 32-bit long sequence number is used. In the Keyed MD5 and the Keyed SHA-1 mechanisms, the sequence member is not required to increase for every packet. Therefore an attacker can keep replaying the packets with the latest sequence number until the sequence number is updated. This issue is eliminated in the Meticulous Keyed MD5 and the Meticulous Keyed SHA-1 mechanisms. However, note that a sequence number may reach its maximum and be rolled over in a session. In this case, without the support from a automatic key management mechanism, the BFD session will be vulnerable to replay attacks performed by sending the packets before the roll over of the sequence number. For instance, an attacker can replay a packet with a sequence number which is larger than the current one. If the replayed packet is accepted, the victim will reject the legal packets whose sequence members are less than the one in the replayed packet. Therefore, the attacker can get a good chance to bring down the BFD session.

Additionally, the BFD specification allows for the change of authentication state based on the state of a received packet. For instance, according to BFD [RFC5880], if the state of a accepted packet is down, the receiver of the packet needs to transfer its state to down as well. Therefore, an elaborately selected replayed packet can cause a serious denial-of-service attack.

BFD does not provide any solution to deal with inter-session replay attacks. If two subsequent BFD sessions adopt an identical discriminator pair and use the same cryptographic key to secure the control packets, it is intuitive to use a malicious authenticated packet (stored from the past session) to perform inter-connection replay attacks.

Any security issues in the BFD echo mode will directly affect the BFD protocol and session states, and hence the network stability. For instance, any replay attacks would be indistinguishable from normal forwarding of the tested router. An attack would still cause a faulty link to be believed to be up, but there is little that can be done about it. However, if the echo packets are guessable, it may be possible to spoof from an external source and cause BFD to believe that a one-way link is really bidirectional. As a result, it is important that the echo packets contain random material that is also checked upon reception.

5. Impact of New Authentication Requirements

BFD can be run in software or hardware. Hardware implementations run BFD at a much smaller timeout, typically in the order of few milliseconds. For instance with a timeout of 3.3 milliseconds, a BFD session is required to send or receive 3 packets every 10 milliseconds. Software implementations typically run with a timeout in hundreds of milliseconds.

Additionally, it is not common to find hardware support for computing the authentication data for the BFD session in hardware or software. In the keyed MD5 and Keyed SHA-1 implementation where the sequence number does not increase with every packet, software can be used to compute the authentication data. This is true if the time between increasing sequence number is long enough to compute the data in software. The ability to compute the hash in software is difficult with Meticulous Keyed MD5 and Meticulous Keyed SHA-1 if the time interval between transmits or between receives is small.

Implementors should assess the impact of authenticating BFD sessions on their platform.

6. Considerations for improvement

This section suggests changes that can be adopted to improve the protection of BFD.

As mentioned in section 3, a 32 bit sequence number space can wrap around in less than 24 weeks when set for the minimum time interval of 3.3 milliseconds. To prevent a replay attack the sequence number can be tied to notion of real time where part of the sequence number reflects say the UTC time. A replay attack therefore can easily be detected. However, it does require that the two stations exchanging BFD packets are synchorized with respect to time. Alternatively, the sequence number can be a nonce number generated using the shared key. But nonce numbers will also run out in 24 weeks.

Increasing the sequence number space to 64 bits makes the wrap around time be a little less than 2 million years. Combined with nonce or part of the number reflecting real time would make replay attacks difficult if not impossible.

7. IANA Considerations

This document makes no request of IANA.

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

8. Security Considerations

9. Acknowledgements

We would like to thank Alexander Vainshtein for his comments on this document.

10. References

10.1. Normative References

- [RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.
- [RFC6039] Manral, V., Bhatia, M., Jaeggli, J., and R. White, "Issues with Existing Cryptographic Protection Methods for Routing Protocols", RFC 6039, October 2010.

10.2. Informative References

- [I-D.ietf-bfd-generic-crypto-auth]
Bhatia, M., Manral, V., and D. Zhang, "BFD Generic Cryptographic Authentication",
draft-ietf-bfd-generic-crypto-auth-03 (work in progress),
October 2012.
- [I-D.ietf-bfd-hmac-sha]
Zhang, D., Bhatia, M., and V. Manral, "Authenticating BFD using HMAC-SHA-2 procedures", draft-ietf-bfd-hmac-sha-02
(work in progress), October 2012.
- [I-D.ietf-karp-threats-reqs]
Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Overview, Threats, and Requirements", draft-ietf-karp-threats-reqs-06 (work in progress), September 2012.
- [RFC4822] Atkinson, R. and M. Fanto, "RIPv2 Cryptographic Authentication", RFC 4822, February 2007.
- [RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009.
- [RFC5709] Bhatia, M., Manral, V., Fanto, M., White, R., Barnes, M., Li, T., and R. Atkinson, "OSPFv2 HMAC-SHA Cryptographic Authentication", RFC 5709, October 2009.
- [RFC6518] Lebovitz, G. and M. Bhatia, "Keying and Authentication for

Routing Protocols (KARP) Design Guidelines", RFC 6518,
February 2012.

Authors' Addresses

Manav Bhatia
Alcatel-Lucent
Bangalore,
India

Phone:
Email: manav.bhatia@alcatel-lucent.com

Dacheng Zhang
Huawei Technologies co., LTD.
Beijing,
China

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

Mahesh Jethanandani
Ciena Corporation
1741 Technology Drive, #400
San Jose, CA 95110
USA

Phone: 408.436.3313
Fax: 408.436.5582
Email: mjethanandani@gmail.com
URI:

Working Group
Internet-Draft
Intended status: Informational
Expires: April 25, 2013

U. Chunduri
A. Tian
W. Lu
Ericsson Inc.,
October 22, 2012

KARP IS-IS security gap analysis
draft-chunduri-karp-is-is-gap-analysis-03

Abstract

This document analyzes the threats applicable for Intermediate system to Intermediate system (IS-IS) routing protocol and security gaps according to the KARP Design Guide. This document also provides specific requirements to address the gaps with both manual and auto key management protocols.

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 April 25, 2013.

Copyright Notice

Copyright (c) 2012 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 Simplified BSD License.

Table of Contents

| | |
|--|----|
| 1. Introduction | 3 |
| 1.1. Requirements Language | 3 |
| 1.2. Acronyms | 3 |
| 2. Current State | 4 |
| 2.1. Key Usage | 4 |
| 2.1.1. Sub network Independent | 4 |
| 2.1.2. Sub network dependent | 5 |
| 2.2. Key Agility | 5 |
| 2.3. Security Issues | 5 |
| 2.3.1. Replay Attacks | 6 |
| 2.3.1.1. Current Recovery mechanism for LSPs | 7 |
| 2.3.2. Spoofing Attacks | 7 |
| 2.3.3. DoS Attacks | 8 |
| 3. Gap Analysis and Security Requirements | 8 |
| 3.1. Manual Key Management | 8 |
| 3.2. Key Management Protocols | 10 |
| 4. IANA Considerations | 10 |
| 5. Security Considerations | 10 |
| 6. Acknowledgements | 11 |
| 7. References | 11 |
| 7.1. Normative References | 11 |
| 7.2. Informative References | 11 |
| Authors' Addresses | 12 |

1. Introduction

This document analyzes the current state of Intermediate system to Intermediate system (IS-IS) protocol according to the requirements set forth in [RFC6518] for both manual and key management protocols.

With currently published work, IS-IS meets some of the requirements expected from a manually keyed routing protocol. Integrity protection is expanded with more cryptographic algorithms and also limited algorithm agility (HMAC-SHA family) is provided with [RFC5310]. Basic form of Intra-connection re-keying capability is provided by the specification [RFC5310] with some gaps as explained in Section 3.

This draft summarizes the current state of cryptographic key usage in IS-IS protocol and several previous efforts to analyze IS-IS security. This includes base IS-IS specification [RFC1195], [RFC5304], [RFC5310] and the OPSEC working group document [RFC6039]. Authors would like to acknowledge all the previous work done in the above documents.

This document also analyzes applicability of various threats as described in [ietf-karp-threats-reqs] to IS-IS, lists gaps and provides specific recommendations to thwart the applicable threats for both manual keying and for auto key management mechanisms.

1.1. 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].

1.2. Acronyms

| | | |
|------|---|--|
| IGP | - | Interior Gateway Protocol. |
| IIH | - | IS-IS HELLO PDU. |
| IPv4 | - | Internet Protocol version 4. |
| KMP | - | Key Management Protocol (auto key management). |
| LSP | - | IS-IS Link State PDU. |
| MKM | - | Manual Key management Protocols. |

NONCE - Number Once.
SA - Security Association.
SNP - Sequence number PDU.

2. Current State

IS-IS is specified in International Standards Organization (ISO) 10589, with extensions to support Internet Protocol version 4 (IPv4) described in [RFC1195]. The specification includes an authentication mechanism that allows for any authentication algorithm and also specifies the algorithm for clear text passwords. Further [RFC5304] extends the authentication mechanism to work with HMAC-MD5 and also modifies the base protocol for more effectiveness. [RFC5310] provides algorithm agility, with new generic crypto authentication mechanism (CRYPTO_AUTH) for IS-IS. The CRYPTO_AUTH also introduces Key ID mechanism that map to unique IS-IS Security Associations (SAs).

The following sections describe the current authentication key usage for various IS-IS messages, current key change methodologies and the various potential security threats.

2.1. Key Usage

IS-IS can be provisioned with a per interface, peer-to-peer key for IS-IS HELLO PDUs (IIH) and a group key for Link State PDUs (LSPs) and Sequence number PDUs (SNPs). If provisioned, IIH packets potentially can use the same group key used for LSPs and SNPs.

2.1.1. Sub network Independent

Link State PDUs, Complete and partial Sequence Number PDUs come under Sub network Independent messages. For protecting Level-1 SNPs and Level-1 LSPs, provisioned Area Authentication key is used. Level-2 SNPs as well as Level-2 LSPs use the provisioned domain authentication key.

Since authentication is performed on the LSPs transmitted by an IS, rather than on the LSP packets transmitted to a specific neighbor, it is implied that all the ISes within a single flooding domain must be configured with the same key in order for authentication to work correctly. This is also true for SNP packets, though they are limited to link local scope in broadcast networks.

If multiple instances share the circuits as specified in [I-D.ietf-

isis-mil], instance specific authentication credentials can be used to protect the LSPs and SNPs with in an area or domain. It is important to note, [I-D.ietf-isis-mil] also allows usage of topology specific authentication credentials with in an instance for the LSPs and SNPs.

2.1.2. Sub network dependent

IS-IS HELLO PDUs use the Link Level Authentication key, which may be different from that of Link State PDUs (LSPs) and Sequence number PDUs (SNPs). This could be particularly true for point-to-point links. In broadcast networks it is possible to provision the same common key used for LSPs and SNPs, to protect IIH messages. This allows neighbor discovery and adjacency formation with more than one neighbor on the same physical interface.

2.2. Key Agility

Key roll over without effecting the routing protocols operation in general and IS-IS in particular, is necessary for effective key management protocol integration.

Current HMAC-MD5 crypto authentication as defined in [RFC5304], suggests a transition mode, so that ISes use a set of keys when verifying the authentication value, to allow key changes. This approach will allow changing the authentication key manually without bringing down the adjacency and without dropping any control packet. But, this can increase the load on control plane for the key transition duration as each control packet may have to be verified by more than one key and also allows to mount a potential Denial of Service (DoS) attack in the transition duration.

The above situation is improved with the introduction of Key ID mechanism as defined in [RFC5310]. With this, the receiver determines the active security association (SA) by looking at the Key ID field in the incoming PDU and need not try with other keys, when the integrity check or digest verification fails. But, neither Key co-ordination across the group nor exact key change mechanism is clearly defined. [RFC5310] says: " Normally, an implementation would allow the network operator to configure a set of keys in a key chain, with each key in the chain having a fixed lifetime. The actual operation of these mechanisms is outside the scope of this document."

2.3. Security Issues

The following section analyzes various security threats possible, in the current state for IS-IS protocol.

2.3.1. Replay Attacks

Replaying a captured protocol packet to cause damage is a common threat for any protocol. Securing the packet with cryptographic authentication information alone can not mitigate this threat completely. Though this problem is more prevalent in broadcast networks it is important to note, most of the IGP deployments use P2P-over-lan [RFC5309], which makes an adversary replay 'easier' than the traditional P2P networks

In intra-session replay attacks a secured protocol packet of the current session is replayed, can cause damage, if there is no other mechanism to confirm this is a replay packet. In inter-session replay attacks, captured packet from one of the previous session can be replayed to cause the damage. IS-IS packets are vulnerable to both these attacks, as there is no sequence number verification for IIH packets and SNP packets. Also with current manual key management periodic key changes across the group are done rarely. Thus the intra-connection and inter-connection replay requirements are not met.

IS-IS specifies the use of the HMAC-MD5 [RFC5304] and HMAC-SHA-1 family in [RFC5310], to protect IS-IS packets. An adversary could replay old IIHs or replay old SNPs that would cause churn in the network or bring down the adjacencies.

1. At the time of adjacency bring up an IS sends IIH packet with empty neighbor list (TLV 6) and with the authentication information as per provisioned authentication mechanism. If this packet is replayed later on the broadcast network, all ISes in the broadcast network can bounce the adjacency to create a huge churn in the network.
2. Today LSPs have intra-session replay protection as LSP header contains 32-bit sequence number which is verified for every received packet against the local LSP database. On the other hand, if a node in the network is out of service (is undergoing some sort of high availability condition, or an upgrade) for more than LSP refresh time and the rest of the network ages out the LSPs of the node under consideration, an adversary can potentially plunge in inter-session replay attacks in the network. If the key is not changed in the above circumstances, attack can be launched by replaying a old LSP with higher sequence number and fewer prefixes or fewer adjacencies. This may force the receiver to accept and remove the routes from the routing table, which eventually causes traffic disruption to those prefixes. However, as per the IS-IS specification there is a built-in recovery mechanism for LSPs from inter-session replay

attacks and it is further discussed in Section 2.3.1.1.

3. In any IS-IS network (broadcast or otherwise), if a old and an empty Complete Sequence Number packet (CSNP) is replayed this can cause LSP flood in the network. Similarly a replayed Partial Sequence Number packet (PSNP) can cause LSP flood in the broadcast network.

2.3.1.1. Current Recovery mechanism for LSPs

In the event of inter-session replay attack by an adversary, as LSP with higher sequence number gets accepted, it also gets propagated until it reaches the originating node of the LSP. The originator recognizes the LSP is "newer" than in the local database and this prompts the originator to flood a newer version of the LSP with higher sequence number than the received. This newer version can potentially replace any versions of the replayed LSP which may exist in the network.

But in the above process, depending on where in the network the replay is initiated, how quick the nodes in the network react to the replayed LSP and also how different the content in the accepted LSP determines the damage caused by the replayed LSP.

2.3.2. Spoofing Attacks

IS-IS shares the same key between all neighbors in an area or in a domain to protect the LSP, SNP packets and in broadcast networks even IIH packets. False advertisement by a router is not within scope of the KARP work. However, given the wide sharing of keys as described above, there is a significant risk that an attacker can compromise a key from one device, and use it to falsely participate in the routing, possibly even in a very separate part of the network.

If the same underlying topology is shared across multiple instances to transport routing/application information as defined in [I-D.ietf-isis-mil], it is necessary to use different authentication credentials for different instances. In this connection, based on the deployment considerations, if certain topologies in a particular IS-IS instance require more protection from spoofing attacks and less exposure, topology specific authentication credentials can be used for LSPs and SNPs as facilitated in [I-D.ietf-isis-mil].

Currently possession of the key itself is used as authentication check and there is no identity check done separately. Spoofing occurs when an illegitimate device assumes the identity of a legitimate one. An attacker can use spoofing as a means for launching various types of attacks. For example:

1. The attacker can send out unrealistic routing information that might cause the disruption of network services such as block holes.
2. A rogue system having access to the common key used to protect the LSP, can send an LSP, setting the Remaining Lifetime field to zero, and flooding it thereby initiating a purge. Subsequently, this also can cause the sequence number of all the LSPs to increase quickly to max out the sequence number space, which can cause an IS to shut down for MaxAge + ZeroAgeLifetime period to allow the old LSPs to age out in other ISes of the same flooding domain.

2.3.3. DoS Attacks

Denial-of-service (DoS) attacks using the authentication mechanism is possible and an attacker can send packets which can overwhelm the security mechanism itself. An example is initiating an overwhelming load of spoofed but integrity protected protocol packets, so that the receiver needs to process the integrity check, only to discard the packet. This can cause significant CPU usage. DoS attacks are not generally preventable within the routing protocol. As the attackers are often remote, the DoS attacks are more damaging to area-scoped or domain-scoped packet receivers than link-local scoped packet receivers.

3. Gap Analysis and Security Requirements

This section outlines the differences between the current state of the IS-IS routing protocol and the desired state as specified in KARP Design Guidelines [RFC6518]. The section focuses on where IS-IS protocol fails to meet general requirements as specified in the threats and requirements document.

This section also describes security requirements that should be met by IS-IS implementations that are secured by manual as well as auto key management protocols.

3.1. Manual Key Management

1. With CRYPTO_AUTH specification [RFC5310], IS-IS packets can be protected with HMAC-SHA family of cryptographic algorithms. The specification provides the limited algorithm agility (SHA family). By using Key IDs, it also conceals the algorithm information from the protected control messages.

2. Even though both intra and inter session replay attacks are best prevented by deploying key management protocols with frequent key change capability, basic constructs for sequence number should be there in the protocol messages. So, some basic or extended sequence number mechanism should be in place to protect IIH packets and SNP packets. The sequence number should be increased for each protocol packet. This allows mitigation of some of the replay threats as mentioned in Section 2.3.1.
3. Any common key mechanism with keys shared across a group of routers is susceptible to spoofing attacks caused by a malicious router. Separate authentication check (apart from the integrity check to verify the digest) with digital signatures as described in [RFC2154], can effectively nullify this attack. But this approach was never deployed and one can only assume due to operational considerations at that time. The alternative approach to thwart this threat would be by using the keys from the group key management protocol. As the group key(s) are generated by authenticating the member ISes in the group first, and then periodically rekeyed, per packet identity or authentication check may not be needed.
4. In general DoS attacks may not be preventable with mechanism from routing protocols itself. But some form of Admin controlled lists (ACLs) at the forwarding plane can reduce the damage. There are some other forms the DoS attacks common to any protocol are not in scope as per the section 2.2 in [I-D.ietf-karp-threats-reqs].

As discussed in Section 2.2, though Key ID mechanism in [RFC5310] helps, better key co-ordination mechanism for key roll over is desirable even with manual key management. But, it fell short of specifying exact mechanism other than using key chains. The specific requirements:

- a. Keys SHOULD be able to change without affecting the established adjacency and even better without any control packet loss.
- b. Keys SHOULD be able to change without effecting the protocol operations, for example, LSP flooding should not be help for a specific Key ID availability.
- c. Any proposed mechanism SHOULD also be further incrementally deployable with key management protocols.

3.2. Key Management Protocols

In broadcast deployments, the keys used for protecting IS-IS protocols messages can, in particular, be group keys. A mechanism, similar to as described in [I-D.weis-gdoi-mac-tek] can be used to distribute group keys to a group of ISes in Level-1 area or Level-2 domain, using GDOI as specified in [RFC6407]. There are also similar approaches with IKEv2 based group key management solutions, to routing protocols as described in [I-D.yeung-g-ikev2] and [I-D.hartman-karp-mrkmp].

If a group key is used, the authentication granularity becomes group membership of devices, not peer authentication between devices. Group key management protocol deployed SHOULD be capable of supporting rekeying support.

In some deployments, where IS-IS point-to-point (P2P) mode is used for adjacency bring-up, sub network dependent messages (IIHs) can use a different key shared between the two point-to-point peers, while all other messages use a group key. When group keying mechanism is deployed, even the P2P IIHs can be protected with the common group keys. This approach facilitates one key management mechanism instead of both pair-wise keying and group keying protocols to be deployed together. If same circuits are shared across multiple instances, the granularity of the group can become per instance for IIHs and per instance/topology for LSAs and SNPs as specified in the [I-D.ietf-isis-mil].

Effective key change capability within the routing protocol which allows key roll over without impacting the routing protocol operation, is one of the requirements for deploying any group key mechanism. Once such mechanism is in place with deployment of group key management protocol, IS-IS can be protected from various threats not limited to intra and inter session replay attacks and spoofing attacks.

Specific use of crypto tables [I-D.ietf-karp-crypto-key-table] should be defined for IS-IS protocol.

4. IANA Considerations

This document defines no new namespaces.

5. Security Considerations

This document is mostly about security considerations of IS-IS

protocol, lists potential threats and security requirements for solving those threats. This document does not introduce any new security threats for IS-IS protocol. For more detailed security considerations please refer the Security Considerations section of the KARP Design Guide [RFC6518] document as well as KARP threat document [I-D.ietf-karp-threats-reqs]

6. Acknowledgements

Authors would like to thank Joel Halpern for encouraging us to come up with this document and giving valuable review comments. Authors would like to acknowledge Naiming Shen for reviewing and providing feedback on this document.

7. References

7.1. Normative References

- [RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, October 2008.
- [RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009.

7.2. Informative References

- [I-D.hartman-karp-mrkmp]
Hartman, S., Zhang, D., and G. Lebovitz, "Multicast Router Key Management Protocol (MaRK)",
draft-hartman-karp-mrkmp-05 (work in progress),
September 2012.
- [I-D.ietf-isis-mi]
Previdi, S., Ginsberg, L., Shand, M., Roy, A., and D. Ward, "IS-IS Multi-Instance", draft-ietf-isis-mi-08 (work in progress), October 2012.
- [I-D.ietf-karp-crypto-key-table]
Housley, R., Polk, T., Hartman, S., and D. Zhang,

"Database of Long-Lived Symmetric Cryptographic Keys",
draft-ietf-karp-crypto-key-table-03 (work in progress),
June 2012.

[I-D.ietf-karp-threats-reqs]

Lebovitz, G. and M. Bhatia, "Keying and Authentication for
Routing Protocols (KARP) Overview, Threats, and
Requirements", draft-ietf-karp-threats-reqs-06 (work in
progress), September 2012.

[I-D.weis-gdoi-mac-tek]

Weis, B. and S. Rowles, "GDOI Generic Message
Authentication Code Policy", draft-weis-gdoi-mac-tek-03
(work in progress), September 2011.

[I-D.yeung-g-ikev2]

Rowles, S., Yeung, A., Tran, P., and Y. Nir, "Group Key
Management using IKEv2", draft-yeung-g-ikev2-05 (work in
progress), October 2012.

[RFC2154] Murphy, S., Badger, M., and B. Wellington, "OSPF with
Digital Signatures", RFC 2154, June 1997.

[RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic
Key Management", BCP 107, RFC 4107, June 2005.

[RFC5309] Shen, N. and A. Zinin, "Point-to-Point Operation over LAN
in Link State Routing Protocols", RFC 5309, October 2008.

[RFC6039] Manral, V., Bhatia, M., Jaeggli, J., and R. White, "Issues
with Existing Cryptographic Protection Methods for Routing
Protocols", RFC 6039, October 2010.

[RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain
of Interpretation", RFC 6407, October 2011.

[RFC6518] Lebovitz, G. and M. Bhatia, "Keying and Authentication for
Routing Protocols (KARP) Design Guidelines", RFC 6518,
February 2012.

Authors' Addresses

Uma Chunduri
Ericsson Inc.,
300 Holger Way,
San Jose, California 95134
USA

Phone: 408 750-5678
Email: uma.chunduri@ericsson.com

Albert Tian
Ericsson Inc.,
300 Holger Way,
San Jose, California 95134
USA

Phone: 408 750-5210
Email: albert.tian@ericsson.com

Wenhu Lu
Ericsson Inc.,
300 Holger Way,
San Jose, California 95134
USA

Email: wenhu.lu@ericsson.com

Working Group
Internet-Draft
Intended status: Informational
Expires: April 8, 2013

U. Chunduri
A. Tian
Ericsson Inc.
October 5, 2012

KARP KMP: Simplified Peer Authentication
draft-chunduri-karp-kmp-router-fingerprints-01

Abstract

This document describes the usage of Router Fingerprint Authentication (RFA) with public keys as a potential peer authentication method with KARP Key Management Protocol (KMP). The advantage of RFA is, neither it requires out-of-band, mutually agreeable symmetric keys nor a full PKI based system (trust anchor or CA certificates) for mutual authentication of the peers with KARP KMP deployments. Usage of Router Fingerprints give a significant operational improvement from symmetric key based systems and yet provide a secure authentication technique.

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 April 8, 2013.

Copyright Notice

Copyright (c) 2012 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 Simplified BSD License.

Table of Contents

| | |
|---|---|
| 1. Introduction | 3 |
| 1.1. Requirements Language | 4 |
| 1.2. Acronyms | 4 |
| 2. Router Fingerprint | 4 |
| 3. Usage of Router Fingerprints with KARP KMP | 5 |
| 3.1. Impact on the PAD | 6 |
| 4. Publishing Router Fingerprints | 6 |
| 5. Scope of Fingerprints usage with RPs | 6 |
| 6. Fingerprint Revocation | 7 |
| 7. IANA Considerations | 7 |
| 8. Security Considerations | 7 |
| 9. Acknowledgements | 7 |
| 10. References | 8 |
| 10.1. Normative References | 8 |
| 10.2. Informative References | 8 |
| Authors' Addresses | 9 |

1. Introduction

A Key Management Protocol (KMP) framework for TCP-based pair wise routing protocols (BGP [RFC4271], PCEP [RFC5440], MSDP [RFC3618] and LDP [RFC5036]) is detailed in [chunduri-karp-using-ikev2-with-tcp-ao]. Usage of IKEv2[RFC5996] as the KMP is also described in the same document. This draft explores a simple and secure authentication method, which can be used for KARP KMP deployments.

Currently operators don't often change the manual keys deployed for protecting the Routing Protocol (RP) messages because of various reasons as noted in Section 2.3 of KARP threat document [I-D.ietf-karp-threats-reqs]. One of the KARP WG goals is to define mechanisms to support key changes for all RPs which use either Manual Key Management (MKM) or KMP with out much operational overhead.

Apart from Peer's identity verification, authentication and parameter negotiation, deployment of KMP can be more useful, when it comes to rekey the keys used by RPs. Rekeying can be achieved with out the operator's intervention and as per the provisioned rekey policy. But for the operators, usage of IKEv2 KMP opens up numerous possibilities for peer authentication and manual symmetric keys not only to bootstrap KMP but used for peer authentication. Various other peer authentication mechanisms with the advantages/drawbacks of each mechanisms are described in the Appendix of the [chunduri-karp-using-ikev2-with-tcp-ao] document.

If symmetric pre-shared keys are used by IKEv2 KMP to authenticate the peer before generating the shared key(s), apart from the other issues with symmetric keys, the problem still remain the same when it comes to changing these keys.

To reduce the operational costs for changing the keys at peering points with relatively large number of peers for various TCP based RPs, this document describes the use of one of the available IKEv2 KMP peer authentication methods with raw or x.509 encoded public keys (to be called as Router Fingerprints in the rest of the document). Router Fingerprint Authentication (RFA) mechanism in conjunction with KARP KMP require neither out-of-band symmetric keys nor a fully functional PKI based system with trust anchor certificates as explained further in Section 2.

Section 2 describes the Router Fingerprints in the context of various KMPs and specifically for IKEv2 KMP. Generation and usage of the Router Fingerprints is described in Section 3 and Section 4 describes an error free method for publishing the Router Fingerprints.

1.1. 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].

1.2. Acronyms

| | | |
|------|---|---|
| CRL | - | Certificate Revocation List. |
| EBGP | - | External BGP (BGP connection between external peers). |
| EE | - | End Entity. |
| IBGP | - | Internal BGP (BGP connection between internal peers). |
| KMP | - | Key Management Protocol (auto key management). |
| MKM | - | Manual Key management Protocols. |
| PAD | - | Peer Authorization Database. |
| RFA | - | Router Fingerprint Authentication. |
| RP | - | Routing Protocol. |

2. Router Fingerprint

Router Fingerprint is a sequence of bytes used to authenticate the public key before using the same to authenticate the peer in the context of KMP.

Various forms of the fingerprint mechanism based on the public keys are already in use as defined in [RFC4252] and [RFC4253]. Fingerprints are also used primarily for root key authentication in x.509 based PKI [RFC5280]. This document only highlights the usage of raw public key based authentication mechanism already defined in [RFC5996] for KARP deployments.

To generate a fingerprint:

1. A router needs to generate an asymmetric Private/Public key pair. Asymmetric crypto algorithms based on RSA [RFC3447] or for shorter and still secure keys Elliptic Curve Cryptography (ECC) [RFC4492] can be used for generating the Private/Public key pair.

2. Once the Asymmetric key pair is generated, the public key can be encoded with any additional data (specific to the router) and can be in the form of more easily administrable X.509 PKI Certificate profile and to be specific as specified in the SubjectPublicKeyInfo structure in Section 4.1 of [RFC5280]. This does not force use of X.509 or full compliance with [RFC5280] since formatting any public key as a SubjectPublicKeyInfo is relatively straightforward and well supported by libraries.
3. The result should be hashed with a cryptographic hash function, preferably SHA-256 or hash functions with similar strength (see more discussion on choosing preferred hash function in Section 8).

The fingerprint generated is not a secret and can be distributed publicly. This is further discussed in Section 4.

3. Usage of Router Fingerprints with KARP KMP

To use Router Fingerprints authentication with KARP KMP, a Private/Public key-pair MUST be generated by the router as specified in Section 2. Base IKEv2 [RFC5996] standard supports only raw RSA based public keys. The type of the public keys and encoding has to be more generic to deploy this peer authentication method.

With current specification [RFC5996] when sender needs to get the certificate of the receiver, Certificate Request payload (CERTREQ as specified in [RFC5996]) is sent with cert encoding set to "Raw RSA Key" and Certification Authority field is empty. The receiver of this CERTREQ payload, (currently) uses PKCS #1 encoding for the generated RSA Public Key and sends the same in CERT payload as Certificate Data with Certificate Encoding set to "Raw RSA Key" as described in Section 3.6 of IKEv2 [RFC5996]. Once the public key of the sender is received, the verification MUST be done with the already published/stored fingerprints of the sender.

As noted above the current IKEv2[RFC5996] specification only supports raw RSA public keys. [I-D.kivinen-ipsecme-oob-pubkey] enhances support for other types of public keys and also defines new encoding format to carry the public key fingerprint in the CERT payload. For RPs to use Router Fingerprint Authentication in the context of IKEv2 MUST follow the encoding format as specified in [I-D.kivinen-ipsecme-oob-pubkey].

3.1. Impact on the PAD

The Peer Authorization Database (PAD) and the role it plays in peer authentication is defined in section 4.4.3 of [RFC4301]. One of the functions of the PAD is to provide the authentication data for each peer. [RFC4301] supports X.509 certificate or pre-shared secret authentication data types. So, it is necessary (and one more reason) to encode the raw public keys as X.509 certificates before sending the same in CERT payload. Though the public key received is in the form of x.509 certificate, for RFA, the PAD entry need not contain a trust anchor via which the end entity (EE) certificate or the public key for the peer must be verifiable. The PAD entry MUST rather contain the published fingerprint of the peer.

4. Publishing Router Fingerprints

The router fingerprint generated is not a secret and can be exchanged out-of-band or can be distributed publicly. Please refer to Section 5 for the generic usage and scope of the RFA in routing environments. In the case of inter-domain routing using EBGp [RFC4271] and if the routers are outside of the SIDR [I-D.ietf-sidr-bgpsec-overview] environment, fingerprint can also be exchanged out-of-band through Service Level Agreements (SLAs) at the RP peering points. A KARP KMP deployment using router fingerprints need to resort to out-of-band public key validation procedure to verify authenticity of the keys being used. The router fingerprints should be part of the KMP PAD to validate the public key received in the KMP messages. For conveying router fingerprints data bytes in a clear unambiguous way PGP (Pretty Good Privacy) wordlists can be used.

5. Scope of Fingerprints usage with RPs

The fingerprint method described in this document in general is more suitable for intra domain deployments. This includes KMP usage for E.g., for IBGP [RFC4271] and LDP [RFC5036] peers, where KARP KMP can be deployed with out having to configure either manual pre-shared keys to bootstrap KMP or full PKI with trust anchor certificates. This method also can be potentially used between EBGp [RFC4271] speakers outside of the SIDR ([I-D.ietf-sidr-bgpsec-overview]) deployment scope, where full PKI infrastructure is not available to deploy with KARP KMP and at the same time, still operators want to avoid provisioning manual keys.

6. Fingerprint Revocation

The idea of RFA in the context of KARP KMP is to deploy a better authentication mechanism than the mutually shared symmetric keys between two routers. This SHOULD be used especially where number of peers using this method is relatively smaller and operationally manageable. Any changes in the router fingerprints SHOULD be administered manually by the operator.

When there are a large number of peers, the need for router fingerprint changes may increase. This may be for reasons of key compromises or other potential changes to the routers. In such environments, operators SHOULD look to full PKI with trust anchor certificates and CRL profiles as specified in the [RFC5280]. In this context, RFA mechanism should be only seen as substantial improvement from mutually shared manual keying authentication methods.

7. IANA Considerations

This document defines no new namespaces.

8. Security Considerations

If collision attacks are perceived as a threat, the hash function to generate the fingerprints SHOULD also possess the property of collision-resistance. To mitigate preimage attacks, the cryptographic hash function used for a fingerprint SHOULD possess the property of second preimage resistance.

If generated fingerprints are truncated to make those short, the truncated fingerprints MUST be long enough to preserve the relevant properties of the hash function against brute-force search attacks.

Considering the above facts, it's recommended to use SHA-256 or similar hash functions with good security properties to generate the fingerprints.

9. Acknowledgements

The authors would like to thank Jari Arkko for initial and valuable discussions on operationally simplified authentication mechanisms in general and RFA mechanism as described in this document in particular. Authors would like to acknowledge Joel Halpern for supporting this work and providing continuous feedback on the draft, including the usefulness of this approach in routing environments.

Thanks to Tero Kivinen for extended discussions on applicability and usage of authentication method described for KARP KMP.

10. References

10.1. Normative References

- [I-D.chunduri-karp-using-ikev2-with-tcp-ao]
Chunduri, U., Tian, A., and J. Touch, "Using IKEv2 with TCP-AO", draft-chunduri-karp-using-ikev2-with-tcp-ao-01 (work in progress), March 2012.
- [I-D.kivinen-ipsecme-oob-pubkey]
Kivinen, T., Wouters, P., and H. Tschofenig, "More Raw Public Keys for IKEv2", draft-kivinen-ipsecme-oob-pubkey-00 (work in progress), March 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

10.2. Informative References

- [I-D.ietf-karp-threats-reqs]
Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Overview, Threats, and Requirements", draft-ietf-karp-threats-reqs-05 (work in progress), May 2012.
- [I-D.ietf-sidr-bgpsec-overview]
Lepinski, M. and S. Turner, "An Overview of BGPSEC", draft-ietf-sidr-bgpsec-overview-02 (work in progress), May 2012.
- [RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.
- [RFC3618] Fenner, B. and D. Meyer, "Multicast Source Discovery Protocol (MSDP)", RFC 3618, October 2003.
- [RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, June 2005.

- [RFC4252] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Authentication Protocol", RFC 4252, January 2006.
- [RFC4253] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, January 2006.
- [RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, May 2006.
- [RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.
- [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 5280, May 2008.
- [RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC6518] Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", RFC 6518, February 2012.

Authors' Addresses

Uma Chunduri
Ericsson Inc.
300 Holger Way
San Jose, California 95134
USA

Phone: +1 (408) 750-5678
Email: uma.chunduri@ericsson.com

Albert Tian
Ericsson Inc.
300 Holger Way
San Jose, California 95134
USA

Phone: +1 (408) 750-5210
Email: albert.tian@ericsson.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 25, 2013

S. Hartman
Painless Security
D. Zhang
Huawei
October 22, 2012

Operations Model for Router Keying
draft-ietf-karp-ops-model-04.txt

Abstract

Developing an operational and management model for routing protocol security that works across protocols will be critical to the success of routing protocol security efforts. This document discusses issues and begins to consider development of these models.

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 April 25, 2013.

Copyright Notice

Copyright (c) 2012 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 Simplified BSD License.

Table of Contents

| | |
|---|----|
| 1. Introduction | 3 |
| 2. Requirements notation | 4 |
| 3. Breakdown of KARP configuration | 5 |
| 3.1. Integrity of the Key Table | 6 |
| 3.2. Management of Key Table | 6 |
| 3.3. Interactions with Automated Key Management | 7 |
| 3.4. VRFs | 7 |
| 4. Credentials and Authorization | 8 |
| 4.1. Preshared Keys | 9 |
| 4.2. Asymmetric Keys | 11 |
| 4.3. Public Key Infrastructure | 11 |
| 4.4. The role of Central Servers | 12 |
| 5. Grouping Peers Together | 13 |
| 6. Administrator Involvement | 15 |
| 6.1. Enrollment | 15 |
| 6.2. Handling Faults | 16 |
| 7. Upgrade Considerations | 18 |
| 8. Security Considerations | 19 |
| 9. Acknowledgments | 20 |
| 10. References | 21 |
| 10.1. Normative References | 21 |
| 10.2. Informative References | 21 |
| Authors' Addresses | 23 |

1. Introduction

The KARP working group is designing improvements to the cryptographic authentication of IETF routing protocols. These improvements include improvements to how integrity functions are handled within each protocol as well as designing an automated key management solution.

This document discusses issues to consider when thinking about the operational and management model for KARP. Each implementation will take its own approach to management; this is one area for vendor differentiation. However, it is desirable to have a common baseline for the management objects allowing administrators, security architects and protocol designers to understand what management capabilities they can depend on in heterogeneous environments. Similarly, designing and deploying the protocol will be easier with thought paid to a common operational model. This will also help with the design of NetConf schemas or MIBs later.

2. Requirements notation

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 [RFC2119].

3. Breakdown of KARP configuration

There are multiple ways of structuring configuration information. One factor to consider is the scope of the configuration information. Several protocols are peer-to-peer routing protocols where a different key could potentially be used for each neighbor. Other protocols require the same group key to be used for all nodes in an administrative domain or routing area. In other cases, the same group key needs to be used for all routers on an interface, but different group keys can be used for each interface.

Within situations where a per-interface, per-area or per-peer key can be used for manually configured long-term keys, that flexibility may not be desirable from an operational standpoint. For example consider OSPF [RFC2328]. Each OSPF link needs to use the same authentication configuration, including the set of keys used for reception and the set of keys used for transmission, but may use different keys for different links. The most general management model would be to configure keys per link. However for deployments where the area uses the same key it would be strongly desirable to configure the key as a property of the area. If the keys are configured per-link, they can get out of sync. In order to support generality of configuration and common operational situations, it would be desirable to have some sort of inheritance where default configurations are made per-area unless overridden per-interface.

As described in [I-D.ietf-karp-crypto-key-table], the cryptographic keys are separated from the interface configuration into their own configuration store. Each routing protocol is responsible for defining the form of the Peer specification used by that protocol. Thus each routing protocol needs to define the scope of keys. For group keying, the Peer specification names the group. A protocol could define a Peer specification indicating the key had a link scope and also a Peer specification for scoping a key to a specific area. For link-scoped keys it is generally best to define a single Peer specification indicating the key has a link scope and to use interface restrictions to restrict the key to the appropriate link.

Operational Requirements: KARP MUST support configuration of keys at the most general scope for the underlying protocol; protocols supporting per-peer keys MUST permit configuration of per-peer keys, protocols supporting per-interface keys MUST support configuration of per-interface keys, and so on. KARP MUST NOT permit configuration of an inappropriate key scope. For example, configuration of separate keys per interface MUST NOT be supported for a protocol requiring per-area keys. This restriction can be enforced by rules specified by each routing protocol for validating key table entries.

3.1. Integrity of the Key Table

The routing key table [I-D.ietf-karp-crypto-key-table] provides a very general mechanism to abstract the storage of keys for routing protocols. To avoid misconfiguration and simplify problem determination, the router MUST verify the internal consistency of entries added to the table. Routing protocols describe how their protocol interacts with the key table including what validation MUST be performed. At a minimum, the router MUST verify:

- o The cryptographic algorithms are valid for the protocol.
- o The key derivation function is valid for the protocol.
- o The direction is valid for the protocol; for example protocols that require the same session key be used in both directions MUST have a direction of both.
- o The peer specification is consistent with the protocol.

Other checks are possible. For example the router could verify that if a key is associated with a peer, that peer is a configured peer for the specified protocol. However, this may be undesirable. It may be desirable to load a key table when some peers have not yet been configured. Also, it may be desirable to share portions of a key table across devices even when their current configuration does not require an adjacency with a particular peer in the interest of uniform configuration or preparing for fail-over.

3.2. Management of Key Table

Several management operations will be quite common. For service provider deployments the configuration management system can simply update the key table. However, for smaller deployments, efficient management operations are important.

As part of adding a new key it is typically desirable to set an expiration time for an old key. The management interface SHOULD provide a mechanism to easily update the expiration time for a current key used with a given peer or interface. Also when adding a key it is desirable to push the key out to nodes that will need it, allowing use for receiving packets then later enabling transmit. This can be accomplished automatically by providing a delay between when a key becomes valid for reception and transmission. However, some environments may not be able to predict when all the necessary changes will be made. In these cases having a mechanism to enable a key for sending is desirable.

The key table's schema supports these operations. However equipment can improve usability by providing convenient functions to effect these common changes.

3.3. Interactions with Automated Key Management

Consideration is required for how an automated key management protocol will assign key IDs for group keys. All members of the group may need to use the same key ID. This requires careful coordination of global key IDs. Interactions with the peer key ID field may make this easier; this requires additional study.

Automated key management protocols also assign keys for single peers. If the key ID is global and needs to be coordinated between the receiver and transmitter, then there is complexity in key management protocols.

3.4. VRFs

Many core and enterprise routers support multiple routing instances. For example a router serving multiple VPNs is likely to have a forwarding/routing instance for each of these VPNs. We need to decide how the key table and other configuration information for KARP interacts with this. The obvious first-order answer is that each routing instance gets its own key table. However, we need to consider how these instances interact with each other and confirm this makes sense.

4. Credentials and Authorization

Several methods for authentication have been proposed for KARP. The simplest is preshared keys used directly as traffic keys. In this mode, the traffic integrity keys are directly configured. This is the mode supported by most of today's routing protocols.

As discussed in [I-D.polk-saag-rtg-auth-keytable], preshared keys can be used as the input to a key derivation function (KDF) to generate traffic keys. For example the TCP Authentication Option (TCP-AO) [RFC5925] derives keys based on the initial TCP session state. Typically a KDF will combine a long-term key with public inputs exchanged as part of the protocol to form fresh session keys. a KDF could potentially be used with some inputs that are configured along with the long-term key. Also, it's possible that inputs to a KDF will be private and exchanged as part of the protocol, although this will be uncommon in KARP's uses of KDFs.

Preshared keys could also be used by an automated key management protocol. In this mode, preshared keys would be used for authentication. However traffic keys would be generated by some key agreement mechanism or transported in a key encryption key derived from the preshared key. This mode may provide better replay protection. Also, in the absence of active attackers, key agreement strategies such as Diffie-Hellman can be used to produce high-quality traffic keys even from relatively weak preshared keys.

Public keys can be used for authentication. The design guide [I-D.ietf-karp-design-guide] describes a mode in which routers have the hashes of peer routers' public keys. In this mode, a traditional public-key infrastructure is not required. The advantage of this mode is that a router only contains its own keying material, limiting the scope of a compromise. The disadvantage is that when a router is added or deleted from the set of authorized routers, all routers that peer need to be updated. Note that self-signed certificates are a common way of communicating public-keys in this style of authentication.

Certificates signed by a certification authority or some other PKI could be used. The advantage of this approach is that routers may not need to be directly updated when peers are added or removed. The disadvantage is that more complexity and cost is required.

Each of these approaches has a different set of management and operational requirements. Key differences include how authorization is handled and how identity works. This section discusses these differences.

4.1. Preshared Keys

In the protocol, manual preshared keys are either unnamed or named by a small integer (typically 16 or 32 bits) key ID. Implementations that support multiple keys for protocols that have no names for keys need to try all possible keys before deciding a packet cannot be validated [RFC4808]. Typically key IDs are names used by one group or peer.

Manual preshared keys are often known by a group of peers rather than just one other peer. This is an interesting security property: unlike with digitally signed messages or protocols where symmetric keys are known only to two parties, it is impossible to identify the peer sending a message cryptographically. However, it is possible to show that the sender of a message is one of the parties who knows the preshared key. Within the routing threat model the peer sending a message can be identified only because peers are trusted and thus can be assumed to correctly label the packets they send. This contrasts with a protocol where cryptographic means such as digital signatures are used to verify the origin of a message. As a consequence, authorization is typically based on knowing the preshared key rather than on being a particular peer. Note that once an authorization decision is made, the peer can assert its identity; this identity is trusted just as the routing information from the peer is trusted. Doing an additional check for authorization based on the identity included in the packet would provide little value: an attacker who somehow had the key could claim the identity of an authorized peer and an attacker without the key should be unable to claim the identity of any peer. Such a check is not required by the KARP threat model: inside attacks are not in scope.

Preshared keys used with key derivation function similarly to manual preshared keys. However to form the actual traffic keys, session or peer specific information is combined with the key. From an authorization standpoint, the derivation key works the same as a manual key. An additional routing protocol step or transport step forms the key that is actually used.

Preshared keys that are used via automatic key management have not been specified for KARP. Their naming and authorization may differ from existing uses of preshared keys in routing protocols. In particular, such keys may end up being known only by two peers. Alternatively they may also be known by a group of peers. Authorization could potentially be based on peer identity, although it is likely that knowing the right key will be sufficient. There does not appear to be a compelling reason to decouple the authorization of a key for some purpose from authorization of peers holding that key to perform the authorized function.

Care needs to be taken when symmetric keys are used for multiple purposes. Consider the implications of using the same preshared key for two interfaces: it becomes impossible to cryptographically distinguish a router on one interface from a router on another interface. So, a router that is trusted to participate in a routing protocol on one interface becomes implicitly trusted for the other interfaces that share the key. For many cases, such as link-state routers in the same routing area, there is no significant advantage that an attacker could gain from this trust within the KARP threat model. However, distance-vector protocols, such as BGP and RIP, permit routes to be filtered across a trust boundary. For these protocols, participation in one interface might be more advantageous than another. Operationally, when this trust distinction is important to a deployment, different keys need to be used on each side of the trust boundary. Key derivation can help prevent this problem in cases of accidental misconfiguration. However, key derivation cannot protect against a situation where a system was incorrectly trusted to have the key used to perform the derivation. To the extent that there are multiple zones of trust and a routing protocol is determining whether a particular router is within a certain zone, the question of untrusted actors is within the scope of the routing threat model.

Key derivation can be part of a management solution to a desire to have multiple keys for different zones of trust. A master key could be combined with peer, link or area identifiers to form a router-specific preshared key that is loaded onto routers. Provided that the master key lives only on the management server and not the individual routers, trust is preserved. However in many cases, generating independent keys for the routers and storing the result is more practical. If the master key were somehow compromised, all the resulting keys would need to be changed. However if independent keys are used, the scope of a compromise may be more limited.

More subtle problems with key separation can appear in protocol design. Two protocols that use the same traffic keys may work together in unintended ways permitting one protocol to be used to attack the other. Consider two hypothetical protocols. Protocol A starts its messages with a set of extensions that are ignored if not understood. Protocol B has a fixed header at the beginning of its messages but ends messages with extension information. It may be that the same message is valid both as part of protocol A and protocol B. An attacker may be able to gain an advantage by getting a router to generate this message with one protocol under situations where the other protocol would not generate the message. This hypothetical example is overly simplistic; real-world attacks exploiting key separation weaknesses tend to be complicated and involve specific properties of the cryptographic functions involved.

The key point is that whenever the same key is used in multiple protocols, attacks may be possible. All the involved protocols need to be analyzed to understand the scope of potential attacks.

Key separation attacks interact with the KARP operational model in a number of ways. Administrators need to be aware of situations where using the same manual traffic key with two different protocols (or the same protocol in different contexts) creates attack opportunities. Design teams should consider how their protocol might interact with other routing protocols and describe any attacks discovered so that administrators can understand the operational implications. When designing automated key management or new cryptographic authentication within routing protocols, we need to be aware that administrators expect to be able to use the same preshared keys in multiple contexts. As a result, we should use appropriate key derivation functions so that different cryptographic keys are used even when the same initial input key is used.

4.2. Asymmetric Keys

Outside of a PKI, public keys are expected to be known by the hash of a key or (potentially self-signed) certificate. The Session Description Protocol provides a standardized mechanism for naming keys (in that case certificates) based on hashes (section 5 [RFC4572]). KARP SHOULD adopt this approach or another approach already standardized within the IETF rather than inventing a new mechanism for naming public keys.

A public key is typically expected to belong to one peer. As a peer generates new keys and retires old keys, its public key may change. For this reason, from a management standpoint, peers should be thought of as associated with multiple public keys rather than as containing a single public key hash as an attribute of the peer object.

Authorization of public keys could be done either by key hash or by peer identity. Performing authorizations by peer identity should make it easier to update the key of a peer without risk of losing authorizations for that peer. However management interfaces need to be carefully designed to avoid making this extra level of indirection complicated for operators.

4.3. Public Key Infrastructure

When a PKI is used, certificates are used. The certificate binds a key to a name of a peer. The key management protocol is responsible for exchanging certificates and validating them to a trust anchor.

Authorization needs to be done in terms of peer identities not in terms of keys. One reason for this is that when a peer changes its key, the new certificate needs to be sufficient for authentication to continue functioning even though the key has never been seen before.

Potentially authorization could be performed in terms of groups of peers rather than single peers. An advantage of this is that it may be possible to add a new router with no authentication related configuration of the peers of that router. For example, a domain could decide that any router with a particular keyPurposeID signed by the organization's certificate authority is permitted to join the IGP. Just as in configurations where cryptographic authentication is not used, automatic discovery of this router can establish appropriate adjacencies.

Assuming that potentially self-signed certificates are used by routers that wish to use public keys but that do not need a PKI, then PKI and the infrastructureless mode of public-key operation described in the previous section can work well together. One router could identify its peers based on names and use certificate validation. Another router could use hashes of certificates. This could be very useful for border routers between two organizations. Smaller organizations could use public keys and larger organizations could use PKI.

4.4. The role of Central Servers

An area to explore is the role of central servers like RADIUS or directories. As discussed in the design-guide, a system where keys are pushed by a central management system is undesirable as an end result for KARP. However central servers may play a role in authorization and key rollover. For example a node could send a hash of a public key to a RADIUS server.

If central servers do play a role it will be critical to make sure that they are not required during routine operation or a cold-start of a network. They are more likely to play a role in enrollment of new peers or key migration/compromise.

Another area where central servers may play a role is for group key agreement. As an example, [I-D.liu-ospfv3-automated-keying-req] discusses the potential need for key agreement servers in OSPF. Other routing protocols that use multicast or broadcast such as IS-IS are likely to need a similar approach.

5. Grouping Peers Together

One significant management consideration will be the grouping of management objects necessary to determine who is authorized to act as a peer for a given routing action. As discussed previously, the following objects are potentially required:

- o Key objects are required. Symmetric keys may be preshared. Asymmetric public keys may be used directly for authorization as well. During key transitions more than one key may refer to a given peer. Group preshared keys may refer to multiple peers.
- o A peer is a router that this router might wish to communicate with. Peers may be identified by names or keys.
- o Groups of peers may be authorized for a given routing protocol.

Establishing a management model is difficult because of the complex relationships between each set of objects. As discussed there may be more than one key for a peer. However in the preshared key case, there may be more than one peer for a key. This is true both for group security association protocols such as an IGP or one-to-one protocols where the same key is used administratively. In some of these situations, it may be undesirable to explicitly enumerate the peers in the configuration; for example IGP peers are auto-discovered for broadcast links but not for non-broadcast multi-access links.

Peers may be identified either by name or key. If peers are identified by key it is probably strongly desirable from an operational standpoint to consider any peer identifiers or name to be a local matter and not require the names or identifiers to be synchronized. Obviously if peers are identified by names (for example with certificates in a PKI), identifiers need to be synchronized between the authorized peer and the peer making the authorization decision.

In many cases peers will explicitly be identified. In these cases it is possible to attach the authorization information (keys or identifiers) to the peer's configuration object. Two cases do not involve enumerating peers. The first is the case where preshared keys are shared among a group of peers. It is likely that this case can be treated from a management standpoint as a single peer representing all the peers that share the keys. The other case is one where certificates in a PKI are used to introduce peers to a router. In this case, rather than configuring peers, , the router needs to be configured with information on what certificates represent acceptable peers.

Another consideration is what routing protocols share peers. For example it may be common for LDP peers to also be peers of some other routing protocol. Also, RSVP-TE may be associated with some TE-based IGP. In some of these cases it would be desirable to use the same authorization information for both routing protocols.

In order to develop a management model for authorization, the working group needs to consider several questions. What protocols support auto-discovery of peers? What protocols require more configuration of a peer than simply the peer's authorization information and network address? What management operations are going to be common as security information for peers is configured and updated? What operations will be common while performing key transitions or while migrating to new security technologies?

6. Administrator Involvement

One key operational question is what areas will administrator involvement be required. Likely areas where involvement may be useful includes enrollment of new peers. Fault recovery should also be considered.

6.1. Enrollment

One area where the management of routing security needs to be optimized is the deployment of a new router. In some cases a new router may be deployed on an existing network where routing to management servers is already available. In other cases, routers may be deployed as part of connecting or creating a site. Here, the router and infrastructure may not be available until the router has securely authenticated. This problem is similar to the problem of getting initial configuration of routing instances onto the router. However, especially in cases where asymmetric keys or per-peer preshared keys are used, the configuration of other routers needs to be modified to bring up the security association. Also, there has been discussion of generating keys on routers and not allowing them to leave devices. This also impacts what strategies are possible. For example this might mean that routers need to be booted in a secure environment where keys can be generated, and public keys copied to a management server to push out the new public key to potential peers. Then, the router needs to be packaged, moved to where it will be deployed and set up. Alternatives are possible; it is critical that we understand how what we propose impacts operators.

We need to work through examples with operators familiar with specific real-world deployment practices and understand how proposed security mechanisms will interact with these practices.

Initial discussions suggest that this will be one more configuration item that needs to be set up to establish service. There is no significant security value in protecting routing protocol keys more than administrative password or Authentication, Authorization and Accounting (AAA) secrets that can be used to gain login access to a router. These existing secrets can be used to make configuration changes that impact routing protocols as much as disclosure of a KARP key. Operators already have procedures in place for these items. So, it is appropriate to use similar procedures for KARP. It is reasonable to improve these procedures and the KARP-related procedures over time. However it is more desirable to deploy KARP with security similar to that used for managing existing secrets than to delay deploying KARP.

Operators that have existing procedures for managing hardware

encryption devices such as VPN gateways MAY use those procedures for managing KARP keys. This MAY be used for example if cost savings in terms of training and auditing justify the re-use of a procedure.

6.2. Handling Faults

Faults may interact with operational practice in at least two ways. First, security solutions may introduce faults. For example if certificates expire in a PKI, previous adjacencies may no longer form. Operational practice will require a way of repairing these errors. This may end up being very similar to deploying a router that is connecting a new site as the security fault may have partitioned the network. However, unlike a new deployment, the event is unplanned. Strategies such as configuring a router and shipping it to a site may not be appropriate for recovering a fault even though they may be more useful for new deployments.

Notifications will play a critical role in avoiding security faults. Implementations SHOULD use appropriate mechanisms to notify operators as security resources are about to expire. Notifications can include messages to consoles, logged events, SNMP traps, or notifications within a routing protocol. One strategy is to have increasing escalations of notifications.

Monitoring will also play a important role in avoiding security faults such as certificate expiration. However, the protocols MUST still have adequate operational mechanisms to recover from these situations. Also, some faults, such as those resulting from a compromise or actual attack on a facility are inherent and may not be prevented.

A second class of faults is equipment faults that impact security. For example if keys are stored on a router and never moved from that device, failure of a router implies a need to update security provisioning on the replacement router and its peers.

One approach, recommended by work on securing BGP [I-D.ietf-sidr-rtr-keying] is to maintain the router's keying material. One option is to maintain a copy of the router's keys near the router. For example, keys cys could be maintained on a USB storage driver. Another approach is to maintain router keys on a central server. These approaches permit the credentials of a router to be recovered. This provides valuable options in case of hardware fault. The failing router can be recovered without changing credentials on other routers. One disadvantage of this approach is that even if public-key cryptography is used, the private keys still need to leave the router. Exporting private keys increases the chance that an attacker may be able to impersonate a router. In many

environments favoring the ability to quickly replace a router makes sense.

More generally keying is another item of configuration that needs to be restored to restore service when equipment fails. Operators typically perform the minimal configuration necessary to get a router back in contact with the management server. The same would apply for keys. Operators who do not maintain copies of key material for performing key recovery on routers would need to perform a bit more work to regain contact with the management server. It seems reasonable to assume that management servers will be able to cause keys to be generated or distributed sufficiently to fully restore service.

7. Upgrade Considerations

It needs to be possible to deploy automated key management in an organization without either having to disable existing security or disrupting routing. As a result, it needs to be possible to perform a phased upgrade from manual keying to automated key management. This upgrade procedure needs to be easy and have a very low risk of disrupting routing. Today, many operators do not update keys because the perceived risk of an attack is lower than the complexity of and update and risk of routing disruptions.

For peer-to-peer protocols such as BGP, this can be relatively easy. First, code that supports automated key management needs to be loaded on both peers. Then the adjacency can be upgraded. The configuration can be updated to switch to automated key management when the second router reboots. Alternatively, if the key management protocols involved can detect that both peers now support automated key management, then a key can potentially be negotiated for an existing session.

The situation is more complex for organizations that have not upgraded from TCP MD5 [RFC2385] to the TCP Authentication Option [RFC5925]. Today, routers typically need to understand whether a given peer supports TCP MD5 or TCP-AO before opening a TCP connection. In addition, many implementations support grouping configuration of related peers including security configuration together. Implementations make it challenging to move from TCP-MD5 to TCP-AO before all peers in the group are ready. Operators perceive it as high risk to update the configuration of a large number of peers. One particularly risky situation is upgrading the configuration of iBGP peers.

The situation is more complicated for multicast protocols. It's probably not reasonable to bring down an entire link to reconfigure it as using automated key management. Two approaches should be considered. One is to support key table rows supporting the automated key management and manually configured keying for the same link at the same time. Coordinating this may be tricky. Another possibility is for the automated key management protocol to actually select the same traffic key that is being used manually. This could potentially be accomplished by having an option in the key management protocol to export the current manual group key through the automated key management protocol. Then after all nodes are configured with automated key management, manual key entries can be removed. The next re-key after all nodes have manual entries removed will generate a new fresh key.

8. Security Considerations

This document does not define a protocol. It does discuss the operational and management implications of several security technologies.

9. Acknowledgments

Funding for Sam Hartman's work on this memo is provided by Huawei.

The authors would like to thank Bill Atwood , Randy Bush, Wes George, Gregory Lebovitz, and Russ White for valuable reviews.

10. References

10.1. Normative References

- [I-D.ietf-karp-crypto-key-table]
Housley, R., Polk, T., Hartman, S., and D. Zhang,
"Database of Long-Lived Symmetric Cryptographic Keys",
draft-ietf-karp-crypto-key-table-03 (work in progress),
June 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119, March 1997.

10.2. Informative References

- [I-D.ietf-karp-design-guide]
Lebovitz, G. and M. Bhatia, "Keying and Authentication for
Routing Protocols (KARP) Design Guidelines",
draft-ietf-karp-design-guide-10 (work in progress),
December 2011.
- [I-D.ietf-sidr-rtr-keying]
Turner, S., Patel, K., and R. Bush, "Router Keying for
BGPsec", draft-ietf-sidr-rtr-keying-00 (work in progress),
May 2012.
- [I-D.liu-ospfv3-automated-keying-req]
Liu, Y., "OSPFv3 Automated Group Keying Requirements",
draft-liu-ospfv3-automated-keying-req-01 (work in
progress), July 2007.
- [I-D.polk-saag-rtg-auth-keytable]
Polk, T. and R. Housley, "Routing Authentication Using A
Database of Long-Lived Cryptographic Keys",
draft-polk-saag-rtg-auth-keytable-05 (work in progress),
November 2010.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
- [RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5
Signature Option", RFC 2385, August 1998.
- [RFC4572] Lennox, J., "Connection-Oriented Media Transport over the
Transport Layer Security (TLS) Protocol in the Session
Description Protocol (SDP)", RFC 4572, July 2006.
- [RFC4808] Bellovin, S., "Key Change Strategies for TCP-MD5",
RFC 4808, March 2007.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP
Authentication Option", RFC 5925, June 2010.

Authors' Addresses

Sam Hartman
Painless Security

Email: hartmans-ietf@mit.edu

Dacheng Zhang
Huawei

Email: zhangdacheng@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2013

M. Jethanandani
Ciena Corporation
B. Weis
K. Patel
Cisco Systems
D. Zhang
Huawei
S. Hartman
Painless Security
U. Chunduri
A. Tian
Ericsson Inc.
October 22, 2012

TCP Authentication Option Master Key Tuple negotiation in IKEv2
draft-mahesh-karp-rkmp-02

Abstract

This document describes a mechanism to secure TCP-based pairwise Routing Protocol (RP) associations using the IKEv2 Key Management Protocol (KMP) integrated with TCP-AO. Included are extensions to IKEv2 and its Security Associations to enable its key negotiation to support TCP-AO.

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 April 25, 2013.

Copyright Notice

Copyright (c) 2012 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 Simplified BSD License.

Table of Contents

| | |
|---|----|
| 1. Introduction | 3 |
| 1.1. Terminology | 3 |
| 1.2. Acronyms and Abbreviations | 3 |
| 2. Overview | 3 |
| 2.1. Types of Keys | 4 |
| 3. Protocol Exchanges | 5 |
| 3.1. IKE_SA_INIT | 5 |
| 3.2. IKE_AUTH | 6 |
| 3.3. CREATE_CHILD_SA | 6 |
| 3.4. INFORMATIONAL | 7 |
| 4. Header and Payload Formats | 7 |
| 4.1. Security Association Payload | 8 |
| 4.1.1. Transforms Substructures | 8 |
| 4.1.2. Example Proposal Exchange | 9 |
| 4.2. Derivation of TCP-AO Keying Material | 10 |
| 4.3. Notify and Delete Payloads | 10 |
| 5. Operation Details | 11 |
| 5.1. General | 11 |
| 5.2. Initial Key Specific Data Exchange | 12 |
| 5.3. Key Selection, Rollover and Protocol Interaction | 12 |
| 6. Key Management Database (KMDB) | 12 |
| 7. IANA Considerations | 12 |
| 8. Security Considerations | 12 |
| 9. Acknowledgements | 12 |
| 10. References | 13 |
| 10.1. Normative References | 13 |
| 10.2. Informative References | 13 |
| Authors' Addresses | 14 |

1. Introduction

Existing routing protocols using unicast communication model (e.g., BGP, LDP, RSVP-TE) have cryptographic authentication mechanisms that use a key shared between the routers on the both sides of the model to protect routing message exchanges between the routers. Unicast key management today is limited to statically configuring master keys in individual routers. This document defines a mechanism to secure TCP-based pairwise Routing Protocol (RP) associations using IKEv2 [RFC5996], allowing network devices to automatically exchange key material related information between the network devices.

This memo assumes that routers need to be provisioned with some credentials for a one-to-one authentication protocol. Any method specified for use with IKEv2 is applicable

When two routers running a routing protocol have not authenticated each other yet, and before sending out any routing protocol packets the two routers need to perform mutual authentication using their provisioned credentials. If successful, two routers negotiate the key material to secure the routing protocol execution.

1.1. Terminology

Here are some terms that we will be using throughout the document.

TBD

1.2. Acronyms and Abbreviations

The following acronyms and abbreviations are used throughout this document.

IKE Internet Key Exchange Protocol

IKEv2 Internet Key Exchange Protocol Version 2

RP Routing Protocol

SA Security Association

2. Overview

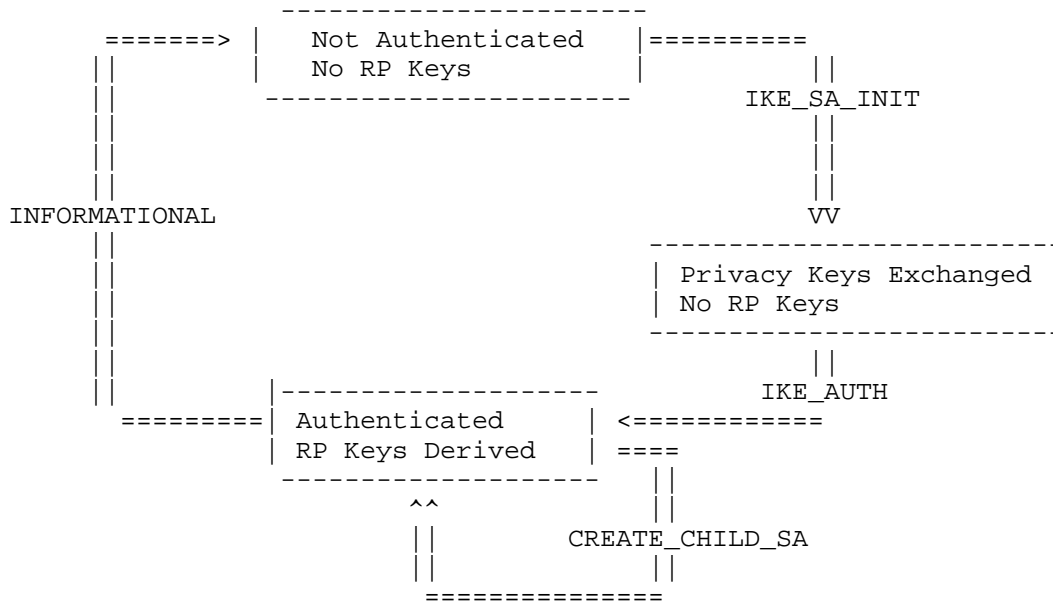


Figure 1: State Diagram

2.1. Types of Keys

The keys adopted in RKMP are listed as follows:

- o PSK (Pre-Shared Key) : PSKs are pair-wise unique keys used for authenticating one router to the other one during the initial exchange. These keys are configured by some mechanism such as manual configuration or a management application outside of the scope of RKMP.
- o Seed key: Refers to value derived from SKEYSEED that is used to derive new keys (e.g., for TCP-AO).
- o Protocol master key: A protocol master key is the key exported by RKMP for use by a routing protocol such as BGP. This is the key that is shared in the key table between the routing protocol and RKMP.
- o Transport key: A transport key is the key used to integrity protect routing messages in a protocol such as BGP. In today's routing protocol cryptographic authentication mechanisms the transport key can be the same as the protocol master key.

3. Protocol Exchanges

The exchange of private keying material between two network devices using a dedicated key management protocol is a requirement as articulated in [I-D.ietf-karp-routing-tcp-analysis]. There is no need to define an entirely new protocol for this purpose, when existing mature protocol exchanges and methods have been vetted. This draft makes use of the IKEv2 protocol exchanges, state machine, and policy definitions to define a dedicated key management protocol.

In the following figures, the notations contained in the message are defined as follows.

| Notation | Payload |
|----------|------------------------------|
| AUTH | Authentication |
| CERT | Certificate |
| CERTREQ | Certificate Request |
| D | Delete |
| HDR | IKEv2 Header (not a payload) |
| IDi | Identification - Initiator |
| IDr | Identification - Responder |
| KE | Key Exchange |
| Ni, Nr | Nonce |
| N | Notify |
| SA | Security Association |
| SK | Encrypted and Authenticated |
| TSi | Traffic Selector - Initiator |
| TSr | Traffic Selector - Responder |

Acronyms Used in Protocol Exchange

3.1. IKE_SA_INIT

A network device desiring to negotiate a TCP-AO MKT to a peer initiates an IKE_SA_INIT exchange defined in Internet Key Exchange Protocol Version 2 [RFC5996]. The IKE_SA_INIT exchange is a two-message exchange that allows the network devices to negotiate cryptographic algorithms, exchange nonces, and do a Diffie-Hellman (DH) [DH] exchange, for their routing protocols, after which protocols on these network devices can communicate privately. Note that at this point the network devices have not identified their peer. For the details of this exchange, refer to IKE_SA_INIT in Internet Key Exchange Protocol Version 2 [RFC5996].

| Peer (Initiator) | | Peer (Responder) |
|--------------------|-----|--------------------------------|
| ----- | | ----- |
| HDR, SAi1, KEi, Ni | --> | |
| | <-- | HDR, SAR1, KEr, Nr, [CERTREQ,] |

Figure 2: IKEv2 IKE_SA_INIT Exchange

3.2. IKE_AUTH

Next, the network devices perform an IKE_AUTH exchange defined in RFC 5996. The SA payloads contain the security policies for a TCP-AO MKT (as defined in Section 4), and the TS payloads contains traffic selectors as defined in [RFC5996]. For the details of the exchange please refer to IKE_AUTH in RFC 5996.

| Peer (Initiator) | | Peer (Responder) |
|--|-----|--|
| ----- | | ----- |
| HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr} | --> | |
| | <-- | HDR, SK {IDr, [CERT,] AUTH, SAr2, TSi, TSr} |

Figure 3: IKEv2 IKE_AUTH Exchange

In the IKE_AUTH exchange, the Initiator proposes one or more sets of policies for a TCP-AO MKT in the SAi2. The SA payload indicates that TCP-AO MKT policy is being proposed, and the TS payloads represent the traffic selectors for the particular routing protocol that will use the TCP-AO MKT (e.g., BGP or LDP). The Responder returns the one policy contained in SAR2 that it accepts. Based on this policy, appropriate keying material is derived from the existing shared keying material. At the successful conclusion of the IKE_AUTH exchange, the initiator and responder have agreed upon a single set of policy and keying material for a particular routing protocol.

3.3. CREATE_CHILD_SA

The network devices may then destroy the state associated with the IKEv2 SA, continuing to use the RP policy and keying material, or they may choose to retain them for the further use. Note that this policy differs from IKEv2/IPsec, where the deletion of the IKEv2 SA necessitates the deletion of the IPsec SAs. If both the network devices choose to retain them, they may use the IKEv2 SA to subsequently agree upon replacement policy for the same RP, or agree upon policy and keying material for another routing protocol. Either case will require the use of the IKEv2 CREATE_CHILD_SA exchange as defined in RFC 5996.

A CREATE_CHILD_SA exchange therefore can be triggered in order to

1. Rekey an antique RP master key and establish a new equivalent one
2. Generate needed key material for a newly executed routing protocol based on an existing SA
3. Rekey an IKEv2 SA and establish a new equivalent IKEv2 SA.

```

Peer (Initiator)                                Peer (Responder)
-----
HDR, SK {[N ], SA, Ni, [KEi ],
[TSi, TSr ]}      -->
<-- HDR, SK {SA, Nr, [KEr ],
[TSi, TSr ]}

```

Figure 4: IKEv2 CREATE_CHILD_SA Exchange

A CREATE_CHILD_SA exchange MAY be initiated by either end of the SA after the initial exchanges are completed. All messages in a CREATE_CHILD_SA exchange are cryptographically protected using the cryptographic algorithms and keys negotiated in the initial exchange.

For details on the exchange, refer to the CREATE_CHILD_SA exchange as defined in RFC 5996.

3.4. INFORMATIONAL

The IKEv2 INFORMATIONAL exchange is also useful for deleting specific IKEv2 SAs or sending status information. The Notify (N) and Delete (D) payloads are as those defined by IKEv2 [IKEV2-PARAMS]. For example, if the Responder refused to accept one of Proposals sent by the Initiator, it would return an INFORMATIONAL exchange of type NO_PROPOSAL_CHOSEN instead of the response to CREATE_CHILD_SA.

```

Peer (Initiator)                                Peer (Responder)
-----
HDR, SK {[N, ] [D, ] ... }      -->
<-- HDR, SK {[N, ] [D, ] ... }

```

Figure 5: IKEv2 INFORMATIONAL Exchange

4. Header and Payload Formats

The protocol defined in this memo uses IKEv2 payload definitions. However, new security policy definitions are described to support security transforms and policy defined by routing protocol documents.

4.1. Security Association Payload

The TCP Authentication Option (TCP-AO) [RFC5925] is primarily intended for BGP and other TCP-based routing protocols. In order for IKEv2 to negotiate TCP-AO policy, a new Security Protocol Identifier needs to be defined in the IANA registry for "IKEv2 Security Protocol Identifiers" [IKEV2-PROTOCOL-IDS]. This memo proposes adding a new Protocol Identifier to the table, with a Protocol Name of "TCP_AO" and a value of TBD1.

The Security Association (SA) payload contains a list of Proposals, which describe one or more sets of policy that a router is willing to use to protect a routing protocol. In the Initiator's message, the SAI2 payload contains a list of Proposal payloads (as defined in the next section), each of which contains a single set of policy that can be applied to the packets described in the Traffic Selector (TS) payloads in the same exchange. Each set of policy is given a particular "Proposal Number" uniquely identifying this set of policy.

The responder includes a single Proposal payload in its SA policy, which denotes the choice it has made amongst the initiator's list of Proposals. Any attributes of a selected transform MUST be returned unmodified as explained in IKEv2 [RFC5996] section 3.3.6. The initiator of an exchange MUST check that the accepted offer is consistent with one of its proposals, and if not MUST terminate the exchange.

4.1.1. Transforms Substructures

Each Proposal has a list of Transform (T) substructures, each of which describe a particular set of cryptographic policy choices. A TCP-AO proposal uses the INTEG transform to negotiate the MKT Message Authentication Code (MAC) algorithm. Cryptographic Algorithms for the TCP Authentication Option (TCP-AO) [RFC5926] describes HMAC-SHA-1-96, AES-128-CMAC-96, which map to the existing INTEG transform IDs of AUTH_HMAC_SHA1_96 and AUTH_AES_CMAC_96 respectively. The use of each INTEG algorithm implies the use of a specific KDF (deriving session keys from a master key) so no the choice of a particular INTEG transform ID also specifies the required KDF transform. This will be true for every transform ID used with TCP-AO, as required in RFC 5926 (see Section 3.2 where the "KDF_Alg" is a fixed element of a MAC algorithm definition for TCP-AO).

A TCP-AO proposal also requires a new type of transform, which describes whether TCP options are to be protected by the integrity algorithm. This memo proposes adding a new Transform Type in the IANA registry for "Transform Type Values" [IKEV2-TRANSFORM-TYPES]

| Number | Name |
|--------|---------------------------------|
| 0 | Options Not Integrity Protected |
| 1 | Options Integrity Protected |

Figure 6: Transform Type TBD2 - TCP Authentication Option Transform IDs

The TCP-AO KeyID that is sent in the SPI field of an IKEv2 proposal. A KeyID for TCP-AO has the same purpose as an IPsec SPI value, so it is natural to place it in this portion of the proposal. If the KeyID in a responder's Proposal is not the same as the initiator's Proposal, then they have chosen to use different KeyID values to represent the same master key and associated proposal policy. This is consistent with how IPsec uses the SPI value, and the semantic of initiator and responder using different SendIDs is supported by RFC 5925.

The following table shows the Transforms that can be negotiated for a TCP-AO protocol.

| Protocol | Mandatory Types | Optional Types |
|----------|-----------------|----------------|
| TCP-AO | INTEG, TCP | D-H |

Figure 7: Mandatory and Optional Transforms

4.1.2. Example Proposal Exchange

Figure 8 shows an example of IKEv2 SA Payload including a single Proposal sent in the first message of an IKE_AUTH or CREATE_CHILD_SA exchange. It indicates a willingness to use either of the two MAC algorithms defined in RFC 5926, and is willing to either protect TCP options or not. The SPI value represents the new SendID it is associating with the TCP-AO Master Key Tuple (MKT) policy being negotiated.

```

SA Payload
|
+--- Proposal #1 ( Proto ID = TCP-AO(TBD1), SPI size = 1,
|                  4 transforms,          SPI = 0x01 )
|
+-- Transform INTEG ( Name = AUTH_HMAC_SHA1_96 )
+-- Transform INTEG ( Name = AUTH_AES_CMAC_96 )
+-- Transform TCP ( Name = PROTECT_OPTIONS )
+-- Transform TCP ( Name = NO_PROTECT_OPTIONS )

```

Figure 8: Example Initiator SA Payload for TCP-AO

The responder will record the SPI value to be the RecvID of the MKT. It chooses its own SendID value, one of each Transform type, and returns this policy in the response message. For example, if the responder chose HMAC-SHA-1-96 and chose to protect the TCP options, the corresponding SA payload would be:

```

SA Payload
|
+--- Proposal #1 ( Proto ID = TCP-AO(TBD1), SPI size = 1,
|                  2 transforms,          SPI = 0x11 )
|
+-- Transform INTEG ( Name = AUTH_HMAC_SHA1_96 )
+-- Transform TCP ( Name = PROTECT_OPTIONS )

```

Figure 9: Example Responder SA Payload for TCP-AO

In this example, the Proposal responder chose to use a different SPI value (0x11) as its SendID. This is possible because Section 2.2 of [RFC5925] declares that "KeyID values MAY be the same in both directions of a connection, but do not have to be and there is no special meaning when they are."

4.2. Derivation of TCP-AO Keying Material

Each TCP-AO MAC algorithm specification in Section 3.2 of [RFC5926] defines the number of bits <n> needed by the MAC algorithm. The first <n> bits of KEYMAT (according to Section 2.17 of [RFC5996]) are used as the key for the negotiated MAC algorithm.

4.3. Notify and Delete Payloads

A Notify Payload ([RFC5996] Section 3.10) or Delete Payload ([RFC5996] Section 3.11) contains a Protocol ID field. The Protocol ID is set to TCP_AO (TBD1) when a notify message is relevant to the TCP-AO KeyID value contained in the SPI field.

5. Operation Details

5.1. General

IKEv2 is used to dynamically derive key material information between the two network devices trying to establish or maintain a routing protocol neighbor adjacency. Typically network devices running the routing protocols establish neighbor adjacencies at the routing protocol level. These routing protocols may run different security algorithms that provide transport level security for the protocol neighbor adjacencies. Depending on the security algorithm used, the routing protocols are configured with security algorithm specific keys that are either long term keys or short term session keys. These keys are specific to the security algorithms used to enforce transport level security for the routing protocols.

A routing protocol causes IKEv2 to execute when it needs key material to establish neighbor adjacency. This can be as a result of the routing protocol neighbor being configured, neighbor changed or updated, a local rekey policy decision, or some other event dictated by the implementation. The key material would allow the network devices to then independently generate the same key and establish an IKEv2 session between them. This is typically done by the Initiator (IKEv2 speaker) initiating an IKEv2 IKE_SA_INIT exchange mentioned in the section 2.1 towards its IKEv2 peer. As part of IKEv2_INIT exchange, IKEv2 will send a message to the peer's IKEv2 port. The format of the message is explained in Section 4. The procedure to exchange key information is explained in Section 4. Once the key material information is successfully exchanged by both of the IKEv2 speakers, the IKEv2 neighbor adjacency may be torn down or kept around as explained in Section 4.

The master key data received from IKEv2 peers is stored in the separate Key Management Database known as KMDB. KMDB follows the guidelines in Database of Long Lived Symmetric Cryptographic Keys [I-D.ietf-karp-crypto-key-table], and each entry consists of Key specific information, Security algorithm to which the Key is applicable to, Routing Protocol Clients of interest, and the announcing RKMP Peer. KMDB is also used to notify the routing protocols about the key updates. Typically key material information is exchanged whenever a routing protocol is about to create a new neighbor adjacency. This is considered as an Initial Key exchange mode. Key material information is also exchanged to refresh existing key data on an already existing neighbor adjacency. This is considered as Key rollover exchange mode. The following sections describes their detail behavior.

5.2. Initial Key Specific Data Exchange

Routing protocols informs IKEv2 of its new neighbor adjacency. It does so by creating a local entry in KMDB which consists of a Security algorithm, Key specific information, routing protocol client and the routing protocol neighbor. Upon a successful creation of such an entry IKEv2 initiates RKMP peering with the neighbor and starts an initial IKE_SA_INIT exchange explained in Section 3.1 followed by the RP_AUTH exchanged explained in Section 3.2. Once the key related information is successfully exchanged, KMDB may invoke the routing protocol client to provide key specific information updates if any.

5.3. Key Selection, Rollover and Protocol Interaction

The procedure for key selection and rollover exchange has been described in Section 3 of Database of Long-Lived Symmetric Cryptographic Keys [I-D.ietf-karp-crypto-key-table]. Details of how RP interact with KMDB and deals with multiple keys during rollover are also described in that section.

6. Key Management Database (KMDB)

Protocol interaction between RKMP and its client routing protocols is typically done using KMDB. Routing protocols update KMDB by installing a new Key related information or purging an existing Key specific information. As part of the KMDB update, IKEv2 initiates peering connections with its appropriate IKEv2 peers to announce the updated key related information. IKEv2 may also receive an updated key related information from its peers which gets installed in KMDB. Whenever IKEv2 updates KMDB with updated key information from its peers, it notifies client routing protocols of its updates.

7. IANA Considerations

TBD

8. Security Considerations

TBD

9. Acknowledgements

During the development of TCP-AO, Gregory Lebovitz noted that a

protocol based on an IKEv2 exchange would be a good automated key management method for deriving a TCP-AO master key. Joe Touch provided many helpful comments.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.
- [RFC5926] Lebovitz, G. and E. Rescorla, "Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)", RFC 5926, June 2010.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

10.2. Informative References

- [DH] Diffie, W. and M. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, V.IT-22 n. 6, June 1977.
- [I-D.ietf-karp-crypto-key-table] Housley, R., Polk, T., Hartman, S., and D. Zhang, "Database of Long-Lived Symmetric Cryptographic Keys", draft-ietf-karp-crypto-key-table-03 (work in progress), June 2012.
- [I-D.ietf-karp-routing-tcp-analysis] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP and MSDP Issues According to KARP Design Guide", draft-ietf-karp-routing-tcp-analysis-05 (work in progress), October 2012.
- [IKEV2-PARAMS] "Internet Key Exchange Version 2 (IKEv2) Parameters", <<http://www.iana.org/assignments/ikev2-parameters/>>

ikev2-parameters.xml>.

[IKEV2-PROTOCOL-IDS]

"'Magic Numbers' for ISAKMP Protocol", <<http://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.xml#ikev2-parameters-18>>.

[IKEV2-TRANSFORM-TYPES]

"'Magic Numbers' for ISAKMP Protocol", <<http://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.xml#ikev2-parameters-3>>.

Authors' Addresses

Mahesh Jethanandani
Ciena Corporation
1741 Technology Drive
San Jose, CA 95110
USA

Phone: +1 (408) 436-3313
Fax:
Email: mjethanandani@gmail.com
URI:

Brian Weis
Cisco Systems
170 W. Tasman Drive
San Jose, California 95134
USA

Phone: +1 (408) 526-4796
Fax:
Email: bew@cisco.com
URI:

Keyur Patel
Cisco Systems
170 Tasman Drive
San Jose, California 95134
USA

Phone: +1 (408) 526-7183
Fax:
Email: keyupate@cisco.com
URI:

Dacheng Zhang
Huawei
Beijing,
China

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

Sam Hartman
Painless Security

Phone:
Fax:
Email: hartmans@painless-security.com
URI:

Uma Chunduri
Ericsson Inc.
300 Holger Way
San Jose, CA 95134
USA

Phone: +1 (408) 750-5678
Fax:
Email: uma.chunduri@ericsson.com
URI:

Albert Tian
Ericsson Inc.
300 Holger Way
San Jose, CA 95134
USA

Phone: +1 (408) 750-5210
Fax:
Email: albert.tian@ericsson.com
URI:

