

KARP
Internet-Draft
Intended status: Standards Track
Expires: January 16, 2014

W. Atwood
R. Bangalore Somanatha
Concordia University/CSE
S. Hartman
Painless Security
D. Zhang
Huawei
July 15, 2013

Authentication, Authorization and Policy Management for Routing
Protocols
draft-atwood-karp-aapm-rp-00

Abstract

When tightening the security of the core routing infrastructure, one requirement is to ensure that the keying material for the routing protocol exchanges is distributed only to the appropriate routers. This document specifies requirements on the authentication and authorization of routers and proposes the use of policy distribution to achieve those requirements.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). 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 January 16, 2014.

Copyright Notice

Copyright (c) 2013 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.

Table of Contents

1. Introduction	2
1.1. Terminology	3
2. System Overview	4
2.1. System Structure	4
2.2. System Operation	5
2.3. Routing Authentication Policy Database	5
3. Problem Statement	6
3.1. Security Goals	6
3.2. Operational Goals	7
4. System Design	7
4.1. Authentication	7
4.2. Authorization	8
4.3. Management of Cryptographic Material	8
4.4. Router Installation	8
4.5. Router Reboot	9
5. RAPD	9
5.1. Contents of an RAPD entry	10
5.2. RAPD Authentication Information	10
5.3. Organization and lookup	11
5.4. Hierarchy of Policy	11
6. Policy Distribution	11
6.1. System Configuration Information	12
6.2. Router Authentication	12
6.3. Router Authorization	12
6.4. Key Management	12
7. IANA Considerations	12
8. Acknowledgements	13
9. Change History (RFC Editor: Delete Before Publishing)	13
10. Needs Work in Next Draft (RFC Editor: Delete Before Publishing)	13
11. References	13
11.1. Normative References	13
11.2. Informative References	13
Authors' Addresses	14

1. Introduction

Within the Keying and Authentication for Routing Protocols (KARP) working group, there are several goals:

- o Determining how to update the security of existing routing protocols, and guiding this work;

- o Development of automated mechanisms for management of the keying material.

Within the first goal, each routing protocol has its own procedures for protecting a routing protocol message "on the wire", given appropriate parameters such as an appropriate traffic encryption key and the cryptographic transforms to be used. How these parameters are placed is not defined by the routing protocol specification.

Within the second goal, protocols and procedures for creating shared keys for specific environments have been developed [I-D.hartman-karp-mrkmp] [I-D.mahesh-karp-rkmp], under the assumption that the end points of the exchanges (the routers) are entitled to enter into the conversation. However, these protocols rely on the authentication mechanisms of IKEv2 [RFC5996] to ensure the endpoints are who they say they are. No way is offered to provide these mechanisms with expected endpoint identities or to provide information on whether an endpoint is entitled to be a neighbor. Provision of expected endpoint identities and neighbor authorization is in effect provision of a policy on what constitutes an acceptable identity and who is an acceptable neighbor.

In addition, requirements for an operations and management model are specified in [I-D.ietf-karp-ops-model].

This document addresses the issue of policy distribution for authentication and authorization of adjacent routers in secure routing protocols. In particular, it addresses the need to ensure that cryptographic parameters are distributed only to routers that legitimately form part of the "authorized neighbor set" of a particular router. It is not concerned with the contents of the exchanged Routing Protocol messages; this is the responsibility of the Routing Protocol specification documents. It is also not concerned with the validity of the Routing Protocol messages themselves; this is being considered by the SIDR working group. Finally, in accordance with the KARP charter, only source authentication is provided for the Routing Protocol messages; confidentiality of these messages is out-of-scope at this time.

If the proposed authentication and authorization mechanisms are not in place, the mechanism used for authentication is likely to be a preshared key, with the same key used throughout a specific area. It is also likely never to be changed, given the difficulty of making this change. When changes come in the topology of the network, it is difficult to tell whether a new router is legitimate or not.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. System Overview

2.1. System Structure

A network that is managed by a particular System Administrator will have some number of routers in it, each of which will be running some set of routing protocols.

For a particular routing protocol, the network is divided into one or more Administrative Domains (AD). An AD is a set of routers with a common policy. An AD might encompass a collection of BGP routers spanning several Autonomous Systems, or all of the routers inside a particular Autonomous System, or all of the routers in an organization, or all of the routers in a unit within an organization, or simply a pair of routers with a point-to-point link.

We distinguish four participants in the architecture:

System Administrator (SA) This is the human who controls the Administrative Domain.

Administrative Domain Manager (ADM) This is the manager for the entire AD. Its role is to distribute the operational policies to the routers within the AD.

Standby ADM (SADM) This provides for robustness if the ADM is unavailable.

Group Member (GM) Any router within the AD.

[[NOTE: A figure would be helpful here.]]

The common policy for a particular AD is managed by the ADM.

Each router has a unique identity in the context of a particular AD. To deal with the issue of interaction between routers in adjacent ADs, the link between two such routers may either be managed by one of the ADs, or a small AD may be created to manage that specific link. In either case, this implies that a specific router may have more than one identity. Authentication of a router involves presenting this identity and establishing its validity.

For a particular routing protocol, a router needs to know which other routers are allowed to exchange routing messages with it. This set

of legitimate neighbors may, in general, be different for each routing protocol that a particular router is executing. Authorization of a router involves matching the identity of that router against the policy governing the set of permitted neighbors.

Within an AD, there are two levels of interaction. At the first level, there is the interaction between the ADM and each of the client routers (GMs). At the second level, there are the interactions between a specific router and the members of its neighbor set.

To participate in the activities within an AD, a router must be authenticated, i.e., it must be able to prove that it is a legitimate member of the AD.

To be permitted to communicate with an adjacent router (however adjacency is defined for a particular routing protocol), a router must be authorized. A router needs to present its identity when communicating with the ADM and also when communicating with the routers that are adjacent to it.

The ADM will distribute to each router the policy that defines how the router is to assess the authenticity of a prospective neighbor, and how the router is to ascertain that the prospective neighbor is authorized to communicate with it.

2.2. System Operation

The SA interacts with the ADM to set the policies for the AD.

The ADM establishes a mutually authenticated relationship with each client router, i.e., with each GM in the AD.

The ADM then pushes the policy definitions to the GMs.

Based on the policy, each GM establishes a mutually authenticated relationship with each of its authorized neighbors.

Each GM will then negotiate cryptographic parameters with its neighbors, or distribute the parameters that it generates, depending on the policy in place.

2.3. Routing Authentication Policy Database

This specification introduces a new conceptual database on each GM, the Routing Authentication Policy Database (RAPD). The RAPD compliments the key table [I-D.ietf-karp-crypto-key-table]. The key table provides manually configured keys and the RAPD provides policy

for automated key management. The RAPD provides services similar to the IPsec Security Policy Database and Peer Authentication Database [RFC4301]

The RAPD serves the following purposes:

- o Is automated key management expected for a particular routing protocol peer or group
- o What identity and credentials are used to authenticate to a remote key-management peer
- o What identities and credentials are accepted when a remote peer authenticates to us
- o Is a particular peer authorized for a particular routing protocol
- o What parameters and transforms are used for a particular security association
- o What key management protocols does this router need to participate in and on what interfaces

See Section 5 for details on the RAPD.

3. Problem Statement

The aim of this document is to specify an overall system for automated key management, which will eliminate the disadvantages of the manual method of key updating. The basic function of this automated system is to distribute and enforce the key management policies of the Administrative Domain. In accordance with these policies, secure generation and distribution will be effected of the keys or other cryptographic material that will be used for the router-to-router exchanges. The system will also enable key updates at regular intervals so as to protect against both active intruders and passive intruders who could be eavesdropping the traffic after having gained access to the keys secretly.

Along with these basic goals, a key management system should satisfy an additional set of requirements. These requirements ensure among other things, security, easy deployment, robustness and scalability. We have compiled this set after referring to the KARP Design Guide [RFC6518], the KARP Threats and Requirements Guide [RFC6862] and the PIM-SM "security on the wire" specification [RFC5796].

3.1. Security Goals

[[NOTE: The following lists of goals were appropriate in the context of Revathi's thesis, which was on formal validation of the security of her proposed protocols. Since we will probably meet at least some of these goals by using an already-standardized, secure protocol, the list is subject to revision as the details of the framework are established.]]

1. Peer authentication for unicast and authentication of all members of the group for multicast protocols.
2. Message authentication, which includes data origin authentication and message integrity.
3. Protection of the system from replay attacks.
4. Peer liveness.
5. Secrecy of key management messages.
6. Authorization to ensure that only authorized routers get the keys.
7. Resistance to man-in-the-middle attacks.
8. Resistance to DoS attacks.
9. Usage of strong keys; those that are unpredictable and are of sufficient length.

3.2. Operational Goals

1. Possibility for easy and incremental deployment.
2. Smooth key rollover.
3. Robustness across router reboots.
4. Scalable design.
5. Policy for authentication and authorization can be shared between unicast and multicast key management.

4. System Design

4.1. Authentication

Each router is assumed to have an identity, i.e., some way of distinguishing it from other routers. The form of this identity is

determined by the SA of the network. It may be a simple string, or it may be a cryptographically strong identity such as that proposed by Chunduri [draft-chunduri].

Each router is assumed to have a way to assert the validity of this identity. The acceptable form(s) of this assertion mechanism will be determined by the policy set by the SA.

[[NOTE: Insert examples here from Sections 4.1, 4.2 and 4.3 of the ops-model.]]

4.2. Authorization

A router has a neighbor set, which is the set of routers that it is able to exchange packets with. The connection used for this exchange may be physical or virtual.

A router has an authorized neighbor set, for a particular Routing Protocol, which is the set of routers that it is authorized to communicate with using the exchanges of that Routing Protocol.

The verification that a router in the neighbor set is also in the authorized neighbor set for a particular Routing Protocol is governed by a policy that is set by the SA.

[[NOTE: Insert examples here from ops-model, section 4.]]

4.3. Management of Cryptographic Material

When a router discovers one or more members of its authorized neighbor set, it will then generate, negotiate, or acquire the cryptographic parameters that it will use when exchanging Routing Protocol packets with these neighbors. Depending on the procedures defined by the Routing Protocol specification for securing the exchanged packets, these cryptographic parameters may include the key(s) to be used, the IPsec Security Parameter Index (SPI) assigned, etc.

For the case where inter-router communication is based on unicast communication, an approach has been developed, which is presented in [I-D.mahesh-karp-rkmp]. For the case where the inter-router communication is based on multicast exchanges, an approach has been developed, which is presented in [I-D.hartman-karp-mrkmp].

4.4. Router Installation

An important aspect of the design is ease of deployment. When a new router is installed, the following steps must be taken:

1. Establish the existence of a new router identity in the AD, using the SA - ADM interface.
2. Define the policy or policies that are applicable to this new identity, using the SA - ADM interface.
3. For the router that will be the first router on the network path between the new router and the ADM, take whatever action is necessary to force the ADM to push revised configuration information to it.
4. At the new router, manually install sufficient policy to allow it to accept its neighbor as part of its authorized neighbor set, and to allow it to know the location of the ADM. Then, force the ADM to push complete configuration information to it.

4.5. Router Reboot

A router must store the information concerning its governing policies in a form of storage that persists over a reboot.

When a router reboots (and especially when a large number of routers reboot due to a power failure and restoration), a router must use the stored information to re-establish its neighbor relationships. This will minimize the likelihood of an apparent denial of service attack on the ADM.

Once the router has established its neighbor relationships, and after a suitable (random) interval, the router should contact its ADM to refresh its policy database.

5. RAPD

According to the key table, routing protocols specify a peer and protocol in order to request a key to send a message. The peer is either the identity of a unicast peer or of a group. The form of the peer identifier is specified by the specific routing protocol in question.

The peer and protocol are enough to find an existing key. As a result, the RAPD needs to be able to locate the appropriate automated key management policy given a peer and protocol.

The RAPD is also used by key management applications when a peer attempts to authenticate or request a key. In this instance, the key management application has the IKE identity of the peer.

5.1. Contents of an RAPD entry

In order to establish an IKE SA, the following information is needed:

- o Identity of the local system to use
- o Identities acceptable for the remote endpoint
- o Credential to use for the local system
- o Authentication information for the remote system; see Section 5.2
- o Lifetime information
- o Acceptable transforms

In order to establish a routing SA keyed by an IKE SA, the following information is needed:

- o Peer and protocol
- o Acceptable transforms

Additional information is required for multicast policy.

5.2. RAPD Authentication Information

The RAPD entry needs to include enough information that the remote peer can be authenticated. The required information tends to break down along the same lines as the credential types discussed in section 4 of [I-D.ietf-karp-ops-model].

For pre-shared keys, mutual authentication is obtained by using the same key in both directions. In this case the credential for outbound authentication is a pre-shared key. For inbound authentication, multiple acceptable credentials can be provided.

For public keys used outside of authentication, authentication needs to happen in each direction. Each peer needs a private key and potentially a certificate to send as a credential. Each peer also needs a set of acceptable fingerprints for the remote key-management peer's key or certificate.

When a PKI is used, each peer needs a private key and a certificate as a credential. In addition, trust anchors and constraints on how to validate whether an asserted identifier is appropriate for the presented certificate are required.

5.3. Organization and lookup

One open question is how to organize the RAPD. When initiating a connection, policy is found using the peer and protocol information. When receiving an incoming association, the peer and protocol might not be available until the routing protocol SA is requested so policy needs to be found based on the initiator's asserted identity.

One approach would be to separate incoming and outgoing policy and use two different databases. This is highly undesirable from an operational standpoint. In general it is not possible to know ahead of time which router will initiate a key management exchange. For this reason, it is strongly desired from an operational standpoint that the policy be symmetric. That is, an association SHOULD successfully authenticate and be authorized independent of which party initiates the association. There are exceptions; for example, in a multicast association, one router MAY be configured not to take on the role of a Group Controller/Key Server (GCKS) and such a router could not respond to key-management associations.

Another approach is to have a single database indexed by the tuple containing peer and protocol as well as the set of acceptable remote identifiers.

Another approach is to have two databases. One contains the peer, protocol, unicast key management endpoint and a policy identifier. The second database contains the remaining information along with a policy identifier. It is indexed by the policy identifier and by the set of acceptable remote identifiers. This layout is very similar to the breakdown between IPsec's SPD and PAD.

All of these approaches assume that the set of acceptable remote identifiers is enumerated in the policy database. In a PKI this may be undesirable.

5.4. Hierarchy of Policy

6. Policy Distribution

[[NOTE: I give below my initial suggestion on a list of policy items that will need to be distributed. My student Nitin has suggested a different way to organize the information, specifically by looking at the types of interaction: SA - ADM; ADM - GM and GM - GM. I expect that both views will be necessary and will revise the document appropriately.]]

In this section, we give an initial list of the policy items that will need to be distributed. The policy will have several facets, each derived from the operational steps.

6.1. System Configuration Information

The system configuration information consists of the following:

1. ADM information (how to reach it)
2. SADM information (how to reach it)

This information is pushed regularly to allow for changes to the ADM location and the SADM location after the initial (manual) configuration.

6.2. Router Authentication

These entries deal with how to identify a legitimate member of the AD.

Under certain circumstances, the ideas in KARP KMP: Simplified Peer Authentication [I-D.chunduri-karp-kmp-router-fingerprints] are appropriate.

[[NOTE: I need to go through the ideas in section 4 of the ops-model document to clarify this.]]

6.3. Router Authorization

These entries deal with how to authorize a specific group member to communicate with its peers.

At a minimum, this will be a list of "authorized neighbors", along with the necessary cryptographic material to permit identifying them.

6.4. Key Management

- o Key generation/negotiation: acceptable procedures, acceptable transforms
- o Key hygiene: lifetimes, etc.
- o Operational rules (from, for example, Operations Model for Router Keying [I-D.ietf-karp-ops-model])

7. IANA Considerations

This document has no actions for IANA.

8. Acknowledgements

9. Change History (RFC Editor: Delete Before Publishing)

[NOTE TO RFC EDITOR: this section for use during I-D stage only.
Please remove before publishing as RFC.]

atwood-karp-akam-rp-00

- o Original submission.

10. Needs Work in Next Draft (RFC Editor: Delete Before Publishing)

[NOTE TO RFC EDITOR: this section for use during I-D stage only.
Please remove before publishing as RFC.]

List of stuff that still needs work

- o Expand the set of policy descriptions

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

[I-D.chunduri-karp-kmp-router-fingerprints]
Chunduri, U., Tian, A., Keranen, A., and T. Kivinen, "KARP KMP: Simplified Peer Authentication", draft-chunduri-karp-kmp-router-fingerprints-03 (work in progress), March 2013.

[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-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-07 (work in progress), March 2013.

[I-D.ietf-karp-ops-model]

Hartman, S. and D. Zhang, "Operations Model for Router Keying", draft-ietf-karp-ops-model-07 (work in progress), July 2013.

[I-D.mahesh-karp-rkmp]

Jethanandani, M., Weis, B., Patel, K., Zhang, D., Hartman, S., Chunduri, U., Tian, A., and J. Touch, "Negotiation for Keying Pairwise Routing Protocols in IKEv2", draft-mahesh-karp-rkmp-04 (work in progress), February 2013.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5796] Atwood, W., Islam, S., and M. Siami, "Authentication and Confidentiality in Protocol Independent Multicast Sparse Mode (PIM-SM) Link-Local Messages", RFC 5796, March 2010.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

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

[RFC6862] Lebovitz, G., Bhatia, M., and B. Weis, "Keying and Authentication for Routing Protocols (KARP) Overview, Threats, and Requirements", RFC 6862, March 2013.

Authors' Addresses

William Atwood
Concordia University/CSE
1455 de Maisonneuve Blvd, West
Montreal, QC H3G 1M8
Canada

Phone: +1(514)848-2424 ext3046
Email: william.atwood@concordia.ca
URI: <http://users.encs.concordia.ca/~bill>

Revathi Bangalore Somanatha
Concordia University/CSE
1455 de Maisonneuve Blvd, West
Montreal, QC H3G 1M8
Canada

Email: revathi.bs@gmail.com

Sam Hartman
Painless Security

Email: hartmans@painless-security.com

Dacheng Zhang
Huawei

Email: zhangdacheng@huawei.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 13, 2013

M.B. Bhatia
Alcatel-Lucent
D. Zhang
Huawei Technologies co., LTD.
M.J. Jethanandani
Ciena Corporation
March 12, 2013

Analysis of Bidirectional Forwarding Detection (BFD) Security According
to KARP Design Guide
draft-ietf-karp-bfd-analysis-00

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 September 13, 2013.

Copyright Notice

Copyright (c) 2013 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 ($3 \times$ 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 reply 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 an 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 an 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 synchornized 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., Bhatia, M., and B. Weis, "Keying and Authentication for Routing Protocols (KARP) Overview, Threats, and Requirements", draft-ietf-karp-threats-reqs-07 (work in progress), December 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

Email: manav.bhatia@alcatel-lucent.com

Dacheng Zhang
Huawei Technologies co., LTD.
Beijing
China

Email: zhangdacheng@huawei.com

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

INTERNET-DRAFT
Internet Engineering Task Force (IETF)
Intended Status: Standards Track

R. Housley
Vigil Security
T. Polk
NIST
S. Hartman
Painless Security
D. Zhang
Huawei
15 July 2013

Expires: 15 January 2014

Database of Long-Lived Symmetric Cryptographic Keys
<draft-ietf-karp-crypto-key-table-08.txt>

Status of this Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

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

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.

Abstract

This document specifies the information contained in a conceptual database of long-lived cryptographic keys used by many different security protocols. The database is designed to support both manual and automated key management. In addition to describing the schema for the database, this document describes the operations that can be performed on the database as well as the requirements for the security protocols that wish to use the database. In many typical scenarios, the security protocols do not directly use the long-lived key, but rather a key derivation function is used to derive a short-lived key from a long-lived key.

1. Introduction

This document specifies the information that needs to be included in a database of long-lived cryptographic keys in order to key the authentication of security protocols such as cryptographic authentication for routing protocols. This conceptual database is designed to separate protocol-specific aspects from both manual and automated key management. The intent is to allow many different implementation approaches to the specified cryptographic key database, while simplifying specification and heterogeneous deployments. This conceptual database avoids the need to build knowledge of any security protocol into key management protocols. It minimizes protocol-specific knowledge in operational/management interfaces, but it constrains where that knowledge can appear. Textual conventions are provided for the representation of keys and other identifiers. These conventions should be used when presenting keys and identifiers to operational/management interfaces or reading keys/identifiers from these interfaces. It is an operational requirement that all implementations represent the keys and key identifiers in the same way so that cross-vendor configuration instructions can be provided.

Security protocols such as TCP-AO [RFC5925] are expected to use per-connection state. Implementations may need to supply keys to the protocol-specific databases as the associated entries in the conceptual database are manipulated. In many instances, the long-lived keys are not used directly in security protocols, but rather a key derivation function is used to derive short-lived key from the long-lived keys in the database. In other instances, security protocols will directly use the long-lived key from the database. The database design supports both use cases.

2. Conceptual Database Structure

The database is characterized as a table, where each row represents a single long-lived symmetric cryptographic key. Normally, each key should only have one row. Only in the (hopefully) very rare cases where a key is used for more than one purpose, or where the same key is used with multiple key derivation functions (KDFs) will multiple rows contain the same key value. The columns in the table represent the key value and attributes of the key.

To accommodate manual key management, the format of the fields has been purposefully chosen to allow updates with a plain text editor and to provide equivalent display on multiple systems.

The columns that the table consists of are listed as follows:

AdminKeyName
The AdminKeyName field contains a string identifying the key by humans. The same string can be used on the local system and peer systems, but this is not required. Protocols do not make use of this string; protocols use the LocalKeyName and the PeerKeyName. Implementations can use this field to uniquely identify rows in the key table.

LocalKeyName
The LocalKeyName field contains a string identifying the key. It can be used to retrieve the key in the local database when received in a message. As discussed in Section 4, the protocol defines the form of this field. For example, many routing protocols restrict the format of their key names to integers that can be represented in 16 or 32 bits. Typically this field does not contain data in human character sets requiring internationalization. If there ever are any Protocols with key names requiring internationalization, those specifications need to address issues of canonicalization and normalization so that key names can be compared using binary comparison.

PeerKeyName

For unicast communication, the PeerKeyName of a key on a system matches the LocalKeyName of the identical key that is maintained on one or multiple peer systems. Similar to LocalKeyName, a protocol defines the form of this identifier and will often restrict it to be an integer. For group keys, the protocol will typically require this field be an empty string as the sending and the receiving key names need to be the same.

Peers

Typically for unicast keys, this field lists the peer systems that have this key in their database. For group keys this field names the groups for which the key is appropriate. For example, this might name a routing area for a multicast routing protocol. Formally, this field provides a protocol-specific set of restrictions on the scope in which the key is appropriate. The format of the identifiers in the Peers field is specified by the protocol.

Interfaces

The Interfaces field identifies the set of physical and/or virtual interfaces for which it is appropriate to use this key. When the long-lived value in the Key field is intended for use on any interface, this field is set to "all". The interfaces field consists of a set of strings; the form of these strings is specified by the implementation and is independent of the protocol in question. Protocols may require support for the interfaces field or may indicate that support for constraining keys based on interface is not required. As an example, TCP-AO implementations are unlikely to make the decision of what interface to use prior to key selection. In this case, the implementations are expected to use the same keying material across all of the interfaces and then require the "all" setting.

Protocol

The Protocol field identifies a single security protocol where this key may be used to provide cryptographic protection. This specification establishes a registry for this field; the registry also specifies the format of the following field, ProtocolSpecificInfo, for each registered protocol.

ProtocolSpecificInfo

This field contains the protocol-specified information which may be useful for a protocol to apply the key correctly. Note that such information must not be required for a protocol to

locate an appropriate key. When a protocol does not need the information in ProtocolSpecificInfo, it will require this field be empty.

KDF

The KDF field indicates the key derivation function which is used to generate short-lived keys from the long-lived value in the Key field. When the long-lived value in the Key field is intended for direct use, the KDF field is set to "none". A key derivation function is a one-way function that provides cryptographic separation of key material. The KDF MAY use inputs from the row in the key table and the message being sent or received but MUST NOT depend on other configuration state. This document establishes an IANA registry for the values in the KDF field to simplify references in future specifications. The protocol indicates what (if any) KDFs are valid.

AlgID

The AlgID field indicates which cryptographic algorithm to be used with the security protocol for the specified peer or peers. Such an algorithm can be an encryption algorithm and mode (e.g., AES-128-CBC), an authentication algorithm (e.g., HMAC-SHA1-96 or AES-128-CMAC), or any other symmetric cryptographic algorithm needed by a security protocol. If the KDF field contains "none", then the long-lived key is used directly with this algorithm, otherwise the derived short-lived key is used with this algorithm. When the long-lived key is used to generate a set of short-lived keys for use with the security protocol, the AlgID field identifies a ciphersuite rather than a single cryptographic algorithm. This document establishes an IANA registry for the values in the AlgID field to simplify references in future specifications. Protocols indicate which algorithms are appropriate.

Key

The Key field contains a long-lived symmetric cryptographic key in the format of a lower-case hexadecimal string. The size of the Key depends on the KDF and the AlgID. For instance, a KDF=none and AlgID=AES128 requires a 128-bit key, which is represented by 32 hexadecimal digits.

Direction

The Direction field indicates whether this key may be used for inbound traffic, outbound traffic, both, or whether the key has been disabled and may not currently be used at all. The supported values are "in", "out", "both", and "disabled", respectively. The Protocol field will determine which of these values are valid.

SendLifetimeStart

The SendLifetimeStart field specifies the earliest date and time in Coordinated Universal Time (UTC) at which this key should be considered for use when sending traffic. The format is YYYYMMDDHHSSZ, where four digits specify the year, two digits specify the month, two digits specify the day, two digits specify the hour, two digits specify the minute, and two digits specify the second. The "Z" is included as a clear indication that the time is in UTC.

SendLifeTimeEnd

The SendLifeTimeEnd field specifies the latest date and time at which this key should be considered for use when sending traffic. The format is the same as the SendLifetimeStart field.

AcceptLifeTimeStart

The AcceptLifeTimeStart field specifies the earliest date and time in Coordinated Universal Time (UTC) at which this key should be considered for use when processing received traffic. The format is YYYYMMDDHHSSZ, where four digits specify the year, two digits specify the month, two digits specify the day, two digits specify the hour, two digits specify the minute, and two digits specify the second. The "Z" is included as a clear indication that the time is in UTC.

AcceptLifeTimeEnd

The AcceptLifeTimeEnd field specifies the latest date and time at which this key should be considered for use when processing the received traffic. The format of this field is identical to the format of AcceptLifeTimeStart.

3. Key Selection and Rollover

A protocol may directly consult the key table to find the key to use on an outgoing message. The protocol provides a protocol (P) and a peer identifier (H) into the key selection function. Optionally, an interface identifier (I) may also need to be provided. Any key that satisfies the following conditions may be selected:

- (1) the Peers field includes H;
- (2) the Protocol field matches P;
- (3) If an interface is specified, the Interfaces field includes I or "all";
- (4) the Direction field is either "out" or "both"; and

(5) `SendLifetimeStart` \leq current time \leq `SendLifeTimeEnd`.

During key selection, multiple entries may simultaneously exist associated with different cryptographic algorithms or ciphersuites. Systems should support selection of keys based on algorithm preference to facilitate algorithm transition.

In addition, multiple entries with overlapping valid periods are expected to be available for orderly key rollover. In these cases, the expectation is that systems will transition to the newest key available. To meet this requirement, this specification recommends supplementing the key selection algorithm with the following differentiation: select the long-lived key specifying the most recent time in the `SendLifetimeStart` field.

In order to look up a key for verifying an incoming message, the protocol provides its protocol (P), the peer identifier (H), the key identifier (L), and optionally the interface (I). If one key matches the following conditions it is selected:

- (1) the Peer field includes H;
- (2) the Protocol field matches P;
- (3) if the Interface field is provided, it includes I or is "all";
- (4) the Direction field is either "in" or "both";
- (5) the LocalKeyName is L; and
- (6) AcceptLifeTimeStart <= current time <= AcceptLifeTimeEnd.

Note that the key usage is loosely bound by the times specified in the AcceptLifeTimeStart and AcceptLifeTimeEnd fields. New security associations should not be established except within the period of use specified by these fields, while allowing some grace time for clock skew. However, if a security association has already been established based on a particular long-lived key, exceeding the lifetime does not have any direct impact. The implementations of security protocols that involve long-lived security association should be designed to periodically interrogate the database and rollover to new keys without tearing down the security association.

Rather than consulting the conceptual database, a security protocol such as TCP-AO may update its own tables as keys are added and removed. In this case, the protocol needs to maintain its own key information.

4. Application of the Database in a Security Protocol

In order to use the key table database in a protocol specification, a protocol needs to specify certain information. This section enumerates items that a protocol must specify.

- (1) The ways of mapping the information in a key table row to the information needed to produce an outgoing message; specified either as an explanation of how to fill in authentication-related fields in a message based on key table information, or for protocols such as TCP-AO how to construct Master Key Tuples (MKTs) or other protocol-specific structures from a key table row
- (2) The ways of locating the peer identifier (a member of the

Peers set) and the LocalKeyName inside an incoming message

- (3) The methods of verifying a message given a key table row; this may be stated directly or in terms of protocol-specific structures such as MKTs
- (4) The form and validation rules for LocalKeyName and PeerKeyName; if either of these is an integer, the conventions in Section 5.1 are used as a vendor-independent format
- (5) The form and validation rules for members of the Peers set
- (6) The algorithms and KDFs supported
- (7) The form of the ProtocolSpecifics field
- (8) The rules for canonicalizing LocalKeyName, PeerKeyName, entries in the Peers set, or ProtocolSpecifics; this may include normalizations such as lower-casing hexadecimal strings
- (9) The Indication whether the support for Interfaces is required by this protocol

The form of the interfaces field is not protocol-specific but instead is shared among all protocols on an implementation. If a protocol needs to distinguish instances running over the same interface, this is included in the specification of peers. Generally it is desirable to define the specification of peers so that an operator can use the interfaces field to refer to all instances of a protocol on a link without having to specify both generic interfaces information and protocol-specific peer information.

5. Textual Conventions

5.1 Key Names

When a key for a given protocol is identified by an integer key identifier, the associated key name will be represented as lower case hexadecimal integers with the most significant octet first. This integer is padded with leading 0's until the width of the key identifier field in the protocol is reached.

5.2 Keys

A key is represented as a lower-case hexadecimal string with the most significant octet of the key first. As discussed in Section 2, the length of this string depends on the associated algorithm and KDF.

6. Operational Considerations

If the valid periods for long-lived keys do not overlap or the system clocks are inconsistent, it is possible to construct scenarios where systems cannot agree upon a long-lived key. When installing a series of keys to be used one after another, operators should configure the `SendLifetimeStart` field of the key to be several hours after the `AcceptLifetimeStart` field of the key to guarantee there is some overlap. This overlap is intended to address the clock skew issue and allow for basic operational considerations. Operators may choose to specify a longer overlap (e.g., several days) to allow for exceptional circumstances.

7. Security Considerations

Management of encryption and authentication keys has been a significant operational problem, both in terms of key synchronization and key selection. For instance, the current guidance [RFC3562] warns against sharing TCP MD5 keying material between systems, and recommends changing keys according to a schedule. The same general operational issues are relevant for the management of other cryptographic keys.

It has been recognized in [RFC4107] that automated key management is not viable in multiple scenarios. The conceptual database specified in this document is designed to accommodate both manual key management and automated key management. A future specification to automatically populate rows in the database is envisioned.

Designers should recognize the warning provided in [RFC4107]:

Automated key management and manual key management provide very different features. In particular, the protocol associated with an automated key management technique will confirm the liveness of the peer, protect against replay, authenticate the source of the short-term session key, associate protocol state information with the short-term session key, and ensure that a fresh short-term session key is generated. Moreover, an automated key management protocol can improve the interoperability by including negotiation mechanisms for cryptographic algorithms. These valuable features are impossible or extremely cumbersome to accomplish with manual key management.

8. IANA Considerations

This specification defines three registries.

8.1. KeyTable Protocols

This document requests establishment of a registry called "KeyTable Protocols". The following subsection describes the registry; the second subsection provides initial values for IEEE 802.1X CAK.

8.1.1. KeyTable Protocols Registry Definition

All assignments to the KeyTable Protocols registry are made on a specification required basis per Section 4.1 of [RFC5226].

Each registration entry must contain the three fields:

- Protocol Name (unique within the registry);
- Specification; and
- Protocol Specific Info.

The specification needs to describe parameters required for using the conceptual database as outlined in Section 4. This typically means that the specification focuses more on the application of security protocols with the key tables rather than being a new security protocol specification for general purposes. New protocols may of course combine information on how to use the key tables database with the protocol specification.

8.1.2. KeyTable Protocols Registry Initial Values

The registry has three columns. The first column is a string of UTF-8 characters representing the name protocol. The second column is a string of UTF-8 characters providing a brief description of Protocol Specific Info. The third column is a reference to a specification defining the protocol.

Protocol	Protocol Specific Info	Reference
-----	-----	-----

IEEE 802.1X CAK	KMD (A string of up to 253 UTF-8 characters that names the transmitting authenticator's key management domain, or null) and NID (A string of up to 100 UTF-8 characters that identifies a network service or null, indicating the key is associated with a default service.)	[IEEE802.1X-2010]
-----------------	--	-------------------

8.2. KeyTable KDFs

This document requests the establishment of a registry called "KeyTable KDFs". The remainder of this section describes the registry.

All assignments to the KeyTable KDFs registry are made on a First Come First Served basis per Section 4.1 of RFC 5226.

The registry has three columns. The first column is a string of UTF-8 characters representing the name of a KDF. The second column is a string of UTF-8 characters providing a brief description of the KDF. The third column is a reference to a specification defining the KDF, if available.

KDF	Description	Reference
---	-----	-----
none	No KDF is used with this key	
802.1X-01	IEEE 802.1X Table 9.1	[IEEE802.1X-2010]

8.3. KeyTable AlgIDs

This document requests establishment of a registry called "KeyTable AlgIDs". The remainder of this section describes the registry.

All assignments to the KeyTable AlgIDs registry are made on a First Come First Served basis per Section 4.1 of RFC 5226.

The registry has three columns. The first column is a string of UTF-8 characters representing the name of an AlgID. The second column is a string of UTF-8 characters providing a brief description of the

AlgID. The third column is a reference to a specification defining the AlgID, if available.

AlgID	Description	Reference
-----	-----	-----
AES-128-CMAC	AES-CMAC using 128-bit keys	[RFC4493]

9. Acknowledgments

This document reflects many discussions with many different people over many years. In particular, the authors thank Jari Arkko, Ran Atkinson, Ron Bonica, Ross Callon, Lars Eggert, Pasi Eronen, Adrian Farrel, Gregory Lebovitz, Acee Lindem, Sandy Murphy, Eric Rescorla, Mike Shand, Dave Ward, and Brian Weis for their insights. The authors additionally thank Brian Weis for supplying text to address IANA concerns and for help with formatting.

Sam Hartman's work on this draft is funded by Huawei.

10. Informational References

- [IEEE802.1X-2010] IEEE Standard for Local and Metropolitan Area Networks -- Port-Based Network Access Control", February 2010.
- [RFC3562] Leech, M., "Key Management Considerations for the TCP MD5 Signature Option", RFC 3562, July 2003.
- [RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", RFC 4107, BCP 107, June 2005.
- [RFC4493] Song, J., Lee, J., Poovendran, R., and T. Iwata, "The AES-CMAC Algorithm", RFC 4493, June 2006.
- [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.

Authors' Addresses

Russell Housley
Vigil Security, LLC
918 Spring Knoll Drive

Herndon, VA 20170
USA
EMail: housley@vigilsec.com

Tim Polk
National Institute of Standards and Technology
100 Bureau Drive, Mail Stop 8930
Gaithersburg, MD 20899-8930
USA
EMail: tim.polk@nist.gov

Sam Hartman
Painless Security, LLC
USA
Email: hartmans@painless-security.com

Dacheng Zhang
Huawei
China
Email: zhangdacheng@huawei.com

Path Computation Element
Internet-Draft
Intended status: Standards Track
Expires: January 11, 2014

D. Lopez
O. Gonzalez de Dios
Telefonica I+D
July 10, 2013

Secure Transport for PCEP
draft-lopez-pcp-pceps-00

Abstract

The Path Computation Element Communication Protocol (PCEP) defines the mechanisms for the communication between a client and a PCE, or among PCEs. This document describe the usage of Transport Layer Security to enhance PCEP security, hence the PCEPS acronym proposed for it. The additional security mechanisms are provided by the transport protocol supporting PCEP, and therefore they do not affect its flexibility and extensibility.

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 January 11, 2014.

Copyright Notice

Copyright (c) 2013 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. Applying TLS to PCEP	3
2.1. TCP ports	3
2.2. Connection Establishment	4
2.3. Peer Identity	5
3. IANA Considerations	6
4. Security Considerations	6
5. Acknowledgements	7
6. References	7
6.1. Normative References	7
6.2. Informative References	8
Authors' Addresses	8

1. Introduction

PCEP [RFC5440] defines the mechanisms for the communication between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs. These interactions include requests and replies that can be critical for a sustainable network operation and adequate resource allocation, and therefore appropriate security becomes a key element in the PCE infrastructure. As the applications of the PCE framework evolves, and more complex service patterns emerge, the definition of a secure mode of operation becomes more relevant.

[RFC5440] analyzes in its section on security considerations the potential threats to PCEP and their consequences, and discusses several mechanisms for protecting PCEP against security attacks, without making a specific recommendation on a particular one or defining their application in depth. Moreover, [RFC6952] remarks the importance of ensuring PCEP communication privacy, especially when PCEP communication endpoints do not reside in the same AS, as the interception of PCEP messages could leak sensitive information related to computed paths and resources.

Among the possible solutions mentioned in these documents, Transport Layer Security (TLS) [RFC5246] provides support for peer authentication, and message encryption and integrity. TLS supports the usage of well-know mechanisms to support key configuration and exchange, and means to perform security checks on the results of PCE discovery procedures ([RFC5088] and [RFC5089]). Since TLS is a security container for the transport of PCEP requests and replies, it will not interfere with the protocol flexibility and extensibility.

This document describes how to apply TLS in securing PCE interactions, including the handshake mechanisms, the methods for peer authentication, and the applicable TLS ciphersuites for data exchange. In the rest of the document we will refer to this usage of TLS as transport for PCEP as either "PCEP over TLS" or "PCEPS".

2. Applying TLS to PCEP

2.1. TCP ports

The default destination port number for PCEP over TLS is TCP/XXXX.

NOTE: This port has to be agreed and registered as PCEPS with IANA.

2.2. Connection Establishment

PCEPS has no notion of negotiating TLS in an established connection. Both peers in the connection need to be preconfigured to use PCEPS for a given endpoint. The connection establishment SHALL follow the following steps:

1. After completing the TCP handshake, immediately negotiate TLS sessions according to [RFC5246]. The following restrictions apply:
 - * Support for TLS v1.2 [RFC5246] or later is REQUIRED.
 - * Support for certificate-based mutual authentication is REQUIRED.
 - * Negotiation of mutual authentication is REQUIRED.
 - * Negotiation of a ciphersuite providing for integrity protection is REQUIRED.
 - * Negotiation of a ciphersuite providing for confidentiality is RECOMMENDED.
 - * Support for and negotiation of compression is OPTIONAL.
 - * PCEPS implementations MUST, at a minimum, support negotiation of the TLS_RSA_WITH_3DES_EDE_CBC_SHA, and SHOULD support TLS_RSA_WITH_RC4_128_SHA and TLS_RSA_WITH_AES_128_CBC_SHA as well. In addition, PCEPS implementations MUST support negotiation of the mandatory-to-implement ciphersuites required by the versions of TLS that they support.
2. Peer authentication can be performed in any of the following two REQUIRED operation models:
 - * TLS with X.509 certificates using PKIX trust models:
 - + Implementations MUST allow the configuration of a list of trusted Certification Authorities for incoming connections.
 - + Certificate validation MUST include the verification rules as per [RFC5280].
 - + Implementations SHOULD indicate their trusted Certification Authorities (CAs). For TLS 1.2, this is done using [RFC5246], Section 7.4.4, "certificate_authorities" (server side) and [RFC6066], Section 6 "Trusted CA Indication"

(client side).

- + Peer validation always SHOULD include a check on whether the locally configured expected DNS name or IP address of the server that is contacted matches its presented certificate. DNS names and IP addresses can be contained in the Common Name (CN) or subjectAltName entries. For verification, only one of these entries is to be considered. The following precedence applies: for DNS name validation, subjectAltName:DNS has precedence over CN; for IP address validation, subjectAltName:iPAddr has precedence over CN.
- + NOTE: Consider here whether peer validation MAY be extended by means of the DANE procedures, including its specs as informative references.
- + Implementations MAY allow the configuration of a set of additional properties of the certificate to check for a peer's authorization to communicate (e.g., a set of allowed values in subjectAltName:URI or a set of allowed X509v3 Certificate Policies)
- * TLS with X.509 certificates using certificate fingerprints: Implementations MUST allow the configuration of a list of trusted certificates, identified via fingerprint of the DER encoded certificate octets. Implementations MUST support SHA-256 as the hash algorithm for the fingerprint.

3. Start exchanging PCEP requests and replies.

NOTE: TLS re-negotiation left as an open issue.

2.3. Peer Identity

Depending on the peer authentication method in use, PCEPS supports different operation modes to establish peer's identity and whether it is entitled to perform requests or can be considered authoritative in its replies. PCEPS implementations SHOULD provide mechanisms for associating peer identities with different levels of access and/or authoritativeness, and they MUST provide a mechanism for establish a default level for properly identified peers. Any connection established with a peer that cannot be properly identified SHALL be terminated before any PCEP exchange takes place.

In TLS-X.509 mode using fingerprints, a peer is uniquely identified by the fingerprint of the presented client certificate.

There are numerous trust models in PKIX environments, and it is beyond the scope of this document to define how a particular deployment determines whether a client is trustworthy. Implementations that want to support a wide variety of trust models should expose as many details of the presented certificate to the administrator as possible so that the trust model can be implemented by the administrator. As a suggestion, at least the following parameters of the X.509 client certificate should be exposed:

- o Peer's IP address
- o Peer's FQDN
- o Certificate Fingerprint
- o Issuer
- o Subject
- o All X509v3 Extended Key Usage
- o All X509v3 Subject Alternative Name
- o All X509v3 Certificate Policies

NOTE: Additional procedures enabled by DANE methods are TBD

NOTE: Specific connections with PCE discovery procedures is TBD

3. IANA Considerations

NOTE: PCEPS has to be registered as TCP port XXXX.

No new PCEP messages or other objects are defined.

4. Security Considerations

Since computational resources required by TLS handshake and ciphersuite are higher than unencrypted TCP, clients connecting to a PCEPS server can more easily create high load conditions and a malicious client might create a Denial-of-Service attack more easily.

Some TLS ciphersuites only provide integrity validation of their payload, and provide no encryption. This specification does not forbid the use of such ciphersuites, but administrators must weight carefully the risk of relevant internal data leakage that can occur

in such a case, as explicitly stated by [RFC6952].

When using certificate fingerprints to identify PCEPS peers, any two certificates that produce the same hash value will be considered the same peer. Therefore, it is important to make sure that the hash function used is cryptographically uncompromised so that attackers are very unlikely to be able to produce a hash collision with a certificate of their choice. This document mandates support for SHA-256, but a later revision may demand support for stronger functions if suitable attacks on it are known.

5. Acknowledgements

This specification relies on the analysis and profiling of TLS included in [RFC6614].

6. References

6.1. Normative References

- [RFC5088] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008.
- [RFC5089] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, January 2008.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [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.
- [RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.

6.2. Informative References

- [RFC6614] Winter, S., McCauley, M., Venaas, S., and K. Wierenga, "Transport Layer Security (TLS) Encryption for RADIUS", RFC 6614, May 2012.
- [RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, May 2013.

Authors' Addresses

Diego R. Lopez
Telefonica I+D
Don Ramon de la Cruz, 82
Madrid, 28006
Spain

Phone: +34 913 129 041
Email: diego@tid.es

Oscar Gonzalez de Dios
Telefonica I+D
Don Ramon de la Cruz, 82
Madrid, 28006
Spain

Phone: +34 913 129 041
Email: ogondio@tid.es

Routing Working Group
Internet-Draft
Intended status: Informational
Expires: January 15, 2014

M. Jethanandani
Ciena Corporation
July 14, 2013

Analysis of LMP Security According to KARP Design Guide
draft-mahesh-karp-lmp-analysis-00.txt

Abstract

This document analyzes Link Management Protocol (LMP) according to guidelines set forth in section 4.2 of KARP Design Guidelines (RFC 6518).

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 January 15, 2014.

Copyright Notice

Copyright (c) 2013 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	2
1.1. Abbreviations	3
2. Current Assessment of LMP	3
2.1. LMP Procedure	3
2.2. Transport Layer	4
2.3. Message Integrity and Node Authentication	4
2.4. Replay Attack	5
2.5. Out-of-order Protection	5
3. Security Requirements for LMP	6
4. Gap Analysis for LMP	6
4.1. Replay Protection	6
5. IANA Requirements	7
6. Security Consideration	7
7. Acknowledgements	7
8. References	7
8.1. Normative References	7
8.2. Informative References	7
Author's Address	7

1. Introduction

In March 2006, the Internet Architecture Board (IAB) described an attack on core routing infrastructure as an ideal attack that would inflict the greatest amount of damage, in their Report from the IAB workshop on Unwanted Traffic March 9-10, 2006 [RFC4948], and suggested steps to tighten the infrastructure against the attack. Four main steps were identified for that tightening:

1. Create secure mechanisms and practices for operating routers.
2. Clean up the Internet Routing Registry (IRR) repository, and securing both the database and the access, so that it can be used for routing verifications.
3. Create specifications for cryptographic validation of routing message content.
4. Secure the routing protocols' packets on the wire.

In order to secure the routing protocols this document performs an initial analysis of the current state of LMP according to the requirements of KARP Design Guidelines [RFC6518]. This draft builds on several previous analysis efforts into routing security:

- o Issues with existing Cryptographic Protection Methods for Routing Protocols [RFC6039] an analysis of cryptographic issues with routing protocols.
- o Analysis of OSPF Security According to KARP Design Guide [RFC6863].
- o Analysis of BGP, LDP, PCEP, and MSDP Issues According to KARP Design Guide [RFC6952] which is a analysis of the four routing protocols.

Link Management Protocol (LMP) [RFC4204] is used to manage Traffic Engineering (TE) links. According to the document, LMP can be subject to a number of attacks. Some examples include:

- o an adversary may spoof control packets
- o an adversary may modify the control packet in transit
- o an adversary may replay control packets
- o an adversary may study a number of control packets and try to break the key using cryptographic tools.

Section 2 looks at the current security state of LMP. Section 3 suggest an optimal security state and section 4 does an analysis of the gap between the existing and the optimal security state of the protocol and suggest some areas where we need to improve.

1.1. Abbreviations

LMP - Link Management Protocol

TE - Traffic Engineering

2. Current Assessment of LMP

This section looks at LMP procedure, the underlying transport layer and security assessment associated with LMP.

2.1. LMP Procedure

The two core procedures of LMP procedure are control channel management and link property correlation. Control channel management is used to establish and maintain control channels between adjacent nodes. This is done using a Config message exchange and a fast keep-alive mechanism between the nodes. Link property correlation is used to synchronize the TE link properties and verify the TE link configuration.

Two additional procedures include link connectivity verification and fault management. Link connectivity verification is used for data plane discovery, Interface_Id exchange, and physical connectivity verification. This is done by sending Test messages over the data channel and the TestStatus messages coming back over the control plane. The LMP link connectivity verification procedure is coordinated using the BeginVerify message exchanged over the control channel.

The LMP fault management procedure is based on a ChannelStatus message exchange. The ChannelStatus message is sent unsolicited and is used to notify an LMP neighbor about the status of one or more data channels. ChannelStatusAck is used to acknowledge receipt of the ChannelStatus message. Similarly, a ChannelStatusResponse message is used to acknowledge receipt of a ChannelStatusRequest message.

2.2. Transport Layer

Except for Test messages, all LMP packets use UDP to communicate with its peers over a LMP port number. Multiple "LMP adjacencies" may be formed and be active between two nodes. LMP messages are transmitted reliably using Message_Ids and retransmissions.

Unlike TCP which can use TCP-AO [RFC5925] for message authentication, UDP does not have any of authenticating packets.

2.3. Message Integrity and Node Authentication

LMP [RFC4204] recommends the use of IPSec for authentication. That document also states that there is currently no requirement that LMP headers or payload be encrypted. It also states that LMP endpoint identity does not need to be protected.

To authenticate LMP, the document further states that manual keying mode be supported. However, it notes that manual keying cannot effectively support replay protection and automatic re-keying. It therefore recommends that manual keying should only be used for diagnostic purposes and only use automatic re-keying for replay protection and automatic re-keying.

2.4. Replay Attack

MESSAGE_ID and MESSAGE_ID_ACK objects are included in the LMP messages to support reliable message delivery. The Message_Id field of the MESSAGE_ID object contains a generator selected value. This value is supposed to be monotonically increasing. A value is considered to be used when it has been sent in an LMP message with the same CC_Id or LMP adjacency. The Message_Id field of the MESSAGE_ID_ACK contains the Message_Id field of the message being acknowledged.

Unacknowledged messages sent with the MESSAGE_ID object are to be retransmitted until the message is acknowledged or until a retry limit is reached. The Message_Id field is 32 bit wide and may wrap.

The 32-bit Message_Id number space is not large enough to guarantee that the Message_Id number will not wrap around within a reasonable long period. Therefore, the system is susceptible to a replay attack.

In addition, LMP does not provide for a generation of a unique monotonically increasing sequence numbers across a failure or a restart.

2.5. Out-of-order Protection

LMP states that nodes processing incoming messages are supposed to check to see if the newly received message is out of order messages, and if so, they are to be ignored and dropped silently.

Specifically, if the message is a Config message, and the Message_Id value is less than the largest Message_Id value previously received from the sender for the CC_Id, then the message is supposed to be treated as being out-of-order. If the message is a LinkSummary message and the Message_Id value is less than the largest Message_Id value previously received from the sender of the TE link, then the message is supposed to be treated as being out-of-order. Similarly, if the message is a ChannelStatus message and the Message_Id value is less than the largest Message_id value previously received from the sender of the specific TE link, then the receiver is supposed to check for the Message_Id value previously received from the state of each data channel included in the ChannelStatus message. If the Message_Id value associated with at least one of the data channels included in the message, the message is not supposed to be treated as out-of-order. All other messages are not supposed to be treated as out-of-order.

3. Security Requirements for LMP

LMP [RFC4204] states that the following requirements should be applied to secure the protocol.

- o LMP security must be able to provide authentication, integrity and replay protection.
- o Confidentiality is not needed for LMP traffic.
- o The protection of identity of the LMP end-points is not commonly required.
- o The security mechanism should provide for a well defined key management scheme. The key management scheme should be scalable and should provide for automatic key rollover.
- o The algorithm used for authentication must be cryptographically sound and it should provide for algorithm agility.

4. Gap Analysis for LMP

This section outlines the differences between the current state of LMP and the desired state as outlined in sections 4.1 and 4.2 of KARP Design Guidelines [RFC6518].

4.1. Replay Protection

As outlined above, LMP protocol is subject to replay attacks. Solutions to replay protection include:

1. Maintaining Message_Id numbers in stable memory
2. Introducing the data from a local time clock into the generation of Message_Id numbers after a restart
3. Introducing the timing information from a Network Recovered Clock into the generation of Message_Id numbers after a restart.

In addition, a handshake is defined for a receiver to get the latest value of a Message_Id number. Therefore, this solution is effective in addressing the issues caused by the rollback of Message_Id numbers across a system restart or failure. However, when a router uses the approach to generating Message_Id numbers with the time information from NTP, an attacker may try to deceive the router to generate a Message_Id number which is less than the Message_Id numbers it used to have, by sending replayed or foiled NTP information.

5. IANA Requirements

This document makes no IANA requests, and the RFC Editor may consider deleting this section on publication of this document as a RFC.

6. Security Consideration

This document is all about security considerations for LMP.

7. Acknowledgements

8. References

8.1. Normative References

- [RFC4204] Lang, J., "Link Management Protocol (LMP)", RFC 4204, October 2005.
- [RFC6518] Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", RFC 6518, February 2012.

8.2. Informative References

- [RFC4948] Andersson, L., Davies, E., and L. Zhang, "Report from the IAB workshop on Unwanted Traffic March 9-10, 2006", RFC 4948, August 2007.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, 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.
- [RFC6863] Hartman, S. and D. Zhang, "Analysis of OSPF Security According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6863, March 2013.
- [RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, May 2013.

Author's Address

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

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

Routing Working Group
Internet-Draft
Intended status: Informational
Expires: November 22, 2013

M. Jethanandani
Ciena Corporation
D. Zhang
Huawei Technologies co., LTD.
May 21, 2013

Analysis of RSVP-TE Security According to KARP Design Guide
draft-mahesh-karp-rsvp-te-analysis-01.txt

Abstract

This document analyzes Resource reSerVation Protocol-Traffic Engineering (RSVP-TE) according to guidelines set forth in section 4.2 of KARP Design Guidelines (RFC 6518).

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 November 22, 2013.

Copyright Notice

Copyright (c) 2013 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	2
1.1. Abbreviations	3
2. Current Assessment of RSVP-TE	4
2.1. Transport Layer	4
2.1.1. UDP Encapsulation	4
2.2. Keying Mechanism	4
2.3. Message Integrity and Node Authentication	5
2.4. Replay Protection	5
2.5. Out of Order Protection	6
2.6. Denial of Service Attack Protection	6
3. Gap Analysis for RSVP-TE	6
4. IANA Requirements	7
5. Security Consideration	7
6. Acknowledgements	7
7. References	7
7.1. Normative References	7
7.2. Informative References	7
Authors' Addresses	9

1. Introduction

In March 2006, the Internet Architecture Board (IAB) described an attack on core routing infrastructure as an ideal attack that would inflict the greatest amount of damage, in their Report from the IAB workshop on Unwanted Traffic March 9-10, 2006 [RFC4948], and suggested steps to tighten the infrastructure against the attack. Four main steps were identified for that tightening:

1. Create secure mechanisms and practices for operating routers.
2. Clean up the Internet Routing Registry (IRR) repository, and securing both the database and the access, so that it can be used for routing verifications.
3. Create specifications for cryptographic validation of routing message content.
4. Secure the routing protocols' packets on the wire.

In order to secure the routing protocols this document performs an initial analysis of the current state of RSVP-TE according to the requirements of KARP Design Guidelines [RFC6518]. This draft builds on several previous analysis efforts into routing security:

- o Issues with existing Cryptographic Protection Methods for Routing Protocols [RFC6039] an analysis of cryptographic issues with routing protocols.
- o Analysis of OSPF Security According to KARP Design Guide [RFC6863].
- o Analysis of BGP, LDP, PCEP, and MSDP Issues According to KARP Design Guide [I-D.ietf-karp-routing-tcp-analysis] which is a analysis of the four routing protocols.

Resource reSerVation Protocol (RSVP) [RFC2205] is a resource reservation setup protocol designed for an integrated services. RSVP Security Properties [RFC4230] indicates the unfeasibility of using IPsec to secure RSVP signaling messages. RSVP Cryptographic Authentication [RFC2747] describes the format and use of RSVP's INTEGRITY objects to provide hop-by-hop integrity and authentication of RSVP messages. RSVP-TE: Extensions to RSVP for LSP Tunnels [RFC3209] is an extension of the RSVP protocol to establish Multi-Protocol Label Switching (MPLS) Label Switch Paths (LSPs). RSVP-TE signaling messages are used to establish both intra- and inter-domain TE LSPs. The security mechanisms for RSVP, RSVP Cryptographic Authentication [RFC2747] can be used by RSVP-TE to provide the security protection for the RSVP-TE message transportation. Therefore, the rest of the document will focus on the current state of security efforts for RSVP and assume that will apply to RSVP-TE also.

Section 2 looks at the current security state of RSVP-TE. Section 3 does an analysis of the gap between the existing and the optimal security state of the protocol and suggest some areas where we need to improve.

1.1. Abbreviations

BGP - Border Gateway Protocol

DoS - Denial of Service

KARP - Key and Authentication for Routing Protocols

KDF - Key Derivation Function

KEK - Key Encrypting Key

KMP - Key Management Protocol

LDP - Label Distribution Protocol

LSP - Label Switch Path

MAC - Message Authentication Code

MKT - Master Key Tuple

MPLS - Multi Protocol Label Switching

MSDP - Multicast Source Distribution Protocol

MD5 - Message Digest algorithm 5

PCEP - Path Computation Element Protocol

RSVP - Resource reSerVation Protocol

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

2. Current Assessment of RSVP-TE

This section looks at RSVP-TE and the underlying transport protocol and key mechanisms built for the protocol.

2.1. Transport Layer

RSVP operates on top of IPv4 or IPv6, occupying the place of a transport protocol in the protocol stack. However, RSVP does not transport application data but is rather an Internet control protocol, like ICMP, IGMP, or routing protocols.

2.1.1. UDP Encapsulation

An RSVP implementation generally requires the ability to perform "raw" network I/O. However, some systems may not support raw network I/O. To use RSVP, such hosts must encapsulate RSVP messages in UDP.

2.2. Keying Mechanism

Section 7 of RSVP Cryptographic Authentication discusses the possibility of using Kerberos to generate and distribute RSVP authentication keys. However, the design of Automated Key Management (AKM) mechanism for RSVP is still incomplete. There is no other AKM solution proposed at this time. If anything, manual key management is used.

The protocol states that manual keying should be supported and states the need for a key management protocol to distribute keys. It even states that the Key Identifier be the hook between RSVP and the key management protocol. But it deliberately excludes defining a integrated key management protocol technique in the document. It does define a key lifetime that should be recorded for all systems although how they are presented e.g. using the start time and the end time of the key life period, is not specified. It even advises that the keys should be changed on a regular basis and that multiple keys should be used to transition from one key to another.

2.3. Message Integrity and Node Authentication

RSVP-TE makes use of RSVP Cryptographic Authentication [RFC2747]. Note that there is currently no RSVP-TE specific security mechanism. It is required that RSVP-TE headers and payload be authenticated, but there is no requirement that RSVP-TE headers be encrypted.

RSVP Cryptographic Authentication [RFC2747] defines the use HMAC-MD5 for both message integrity and node authentication. The length of the keyed digests is 128 bits. In these cases RSVP checksum can be disabled in lieu of message digest. In addition, no algorithm agility is supported.

2.4. Replay Protection

RSVP uses 64 bit monotonically increasing sequence numbers to prevent against replay attacks. The sequence number space is large enough to guarantee that a sequence number will never reach its maximum and roll back within a reasonable long period.

The solution provides three approaches to generate unique monotonically increasing sequence numbers across a failure or a restart. The solutions include:

1. Maintaining sequence numbers in stable memory
2. Introducing the data from a local time clock into the generation of sequence numbers after a restart
3. Introducing the timing information from a Network Recovered Clock into the generation of sequence numbers after a restart.

In addition, a handshake is defined for a receiver to get the latest value of a sequence number. Therefore, this solution is effective in addressing the issues caused by the rollback of sequence numbers across a system restart or failure. However, when a router uses the approach to generating sequence numbers with the time information

from NTP, an attacker may try to deceive the router to generate a sequence number which is less than the sequence numbers it used to have, by sending replayed or foiled NTP information.

2.5. Out of Order Protection

To address the issue of out-of-order message delivery, the solution proposed in RSVP Cryptographic Authentication [RFC2747] allows administrators to specify a sequence number window corresponding to the worst case reordering behavior. Instead of requiring the sequence number of an incoming packet to be strictly larger than the ones previously received, a packet will be accepted if its sequence number is within the window.

2.6. Denial of Service Attack Protection

RSVP does not explicitly mention Denial of Service (DoS) attacks and how to prevent against it. However, a RSVP-TE node does know the peers that it should be communicating with and can therefore accept packets from known hosts only. This feature can largely mitigate the security risks caused by DoS attacks.

3. Gap Analysis for RSVP-TE

This section outlines the differences between the current state of RSVP-TE and the desired state as outlined in sections 4.1 and 4.2 of KARP Design Guidelines [RFC6518].

In RSVP Cryptographic Authentication [RFC2747], only the usage of MD5 to generate digests for RSVP-TE messages is defined. In order to fulfill the requirement of supporting strong algorithms and cryptographic algorithm agility, at least the support of SHA-2 and the ability to indicate additional algorithms needs to be provided..

In addition, in RSVP Cryptographic Authentication [RFC2747], three approaches to generating unique monotonically increasing sequence numbers across a failure and restart are introduced, but no approach is mandated. However, as mentioned above, when using Network Recovered Clocks into the generation of sequence numbers, the capability of RSVP-TE in tolerating inter-connection replay attacks will largely rely on the security of network timing protocols. Therefore, in future this approach should not be recommended.

4. IANA Requirements

This document makes no IANA requests, and the RFC Editor may consider deleting this section on publication of this document as a RFC.

5. Security Consideration

This document is all about security considerations for RSVP-TE.

6. Acknowledgements

The authors would like to thank Sean Turner for his review and comments on the draft.

7. References

7.1. Normative References

- [RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.
- [RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.
- [RFC5926] Lebovitz, G. and E. Rescorla, "Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)", RFC 5926, June 2010.
- [RFC6518] Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", RFC 6518, February 2012.

7.2. Informative References

- [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-07 (work in progress), April 2013.
- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.
- [RFC2747] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", RFC 2747, January 2000.
- [RFC3547] Baugher, M., Weis, B., Hardjono, T., and H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003.
- [RFC4230] Tschofenig, H. and R. Graveman, "RSVP Security Properties", RFC 4230, December 2005.
- [RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
- [RFC4948] Andersson, L., Davies, E., and L. Zhang, "Report from the IAB workshop on Unwanted Traffic March 9-10, 2006", RFC 4948, August 2007.
- [RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.
- [RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007.
- [RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.
- [RFC5961] Ramaiah, A., Stewart, R., and M. Dalal, "Improving TCP's Robustness to Blind In-Window Attacks", RFC 5961, August 2010.
- [RFC6039] Manral, V., Bhatia, M., Jaeggli, J., and R. White, "Issues with Existing Cryptographic Protection Methods for Routing Protocols", RFC 6039, October 2010.
- [RFC6862] Lebovitz, G., Bhatia, M., and B. Weis, "Keying and Authentication for Routing Protocols (KARP) Overview, Threats, and Requirements", RFC 6862, March 2013.

[RFC6863] Hartman, S. and D. Zhang, "Analysis of OSPF Security According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6863, March 2013.

Authors' Addresses

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

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

Dacheng Zhang
Huawei Technologies co., LTD.
Beijing
China

Email: zhangdacheng@huawei.com