KARP Working Group
Internet Draft
Intended status: Informational
Expires: August, 2010

G. Lebovitz
Juniper
M. Bhatia
Alcatel-Lucent

February 2010

Keying and Authentication for Routing Protocols (KARP) Design Guidelines

draft-ietf-karp-design-guide-00.txt

Status of this Memo

Copyright Notice

Abstract

In the March of 2006 the IAB held a workshop on the topic of "Unwanted Internet Traffic". The report from that workshop is documented in RFC 4948 [RFC4948]. Section 8.2 of RFC 4948 calls for [t]ightening the security of the core routing infrastructure." Four main steps were identified for improving the security of the routing infrastructure. One of those steps was "securing the routing protocols' packets on the wire." One mechanism for securing routing protocol packets on the wire is the use of per-packet cryptographic message authentication, providing both peer authentication and message integrity. Many different routing protocols exist and they employ a range of different transport subsystems. Therefore there must necessarily be various methods defined for applying cryptographic authentication to these varying protocols. Many routing protocols already have some method for accomplishing cryptographic message authentication. However, in many cases the existing methods are dated, vulnerable to attack, and/or employ cryptographic algorithms that have been deprecated. This document is one of a series concerned with defining a roadmap of protocol specification work for the use of modern cryptographic mechanisms and algorithms for message authentication in routing protocols. In particular, it defines the framework for a key management protocol that may be used to create and manage session keys for message authentication and integrity. The overall roadmap reflects the input of both the security area and routing area in order to form a jointly agreed upon and prioritized work list for the effort.

Conventions used in this document

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

Table of Contents

1. Introduction

In March 2006 the Internet Architecture Board (IAB) held a workshop on the topic of "Unwanted Internet Traffic". The report from that workshop is documented in RFC 4948 [RFC4948]. Section 8.1 of that document states that "A simple risk analysis would suggest that an ideal attack target of minimal cost but maximal disruption is the core routing infrastructure." Section 8.2 calls for "[t]ightening the security of the core routing infrastructure." Four main steps were identified for that tightening:

- More secure mechanisms and practices for operating routers.

This work is being addressed in the OPSEC Working Group.

- Cleaning up the Internet Routing Registry repository [IRR], and securing both the database and the access, so that it can be used for routing verifications. This work should be addressed through liaisons with those running the IRR's globally.

- Specifications for cryptographic validation of routing message content. This work will likely be addressed in the SIDR Working Group.

- Securing the routing protocols' packets on the wire

This document addresses the last bullet, securing the packets on the wire of the routing protocol exchanges.

2. Categorizing Routing Protocols

For the purpose of this security roadmap definition, we will categorize the routing protocols into groups and have design teams focus on the specification work within those groupings. It is believed that the groupings will have like requirements for their authentication mechanisms, and that reuse of authentication mechanisms will be greatest within these grouping. The work items placed on the roadmap will be defined and assigned based on these categorizations. It is also hoped that, down the road in the Phase 2 work, we can create one Key Management Protocol (KMP) per category (if not for several categories) so that the work can be easily leveraged by the various Routing Protocol teams. KMPs are useful for allowing simple, automated updates of the traffic keys used in a base protocol. KMPs replace the need for humans, or OSS routines, to periodically replace keys on running systems. It also removes the need for a chain of manual keys to be chosen or configured. When configured properly, a KMP will enforce the key freshness policy of two peers by keeping track of the key lifetime and negotiating a new key at the defined interval.

2.1. Category: Message Transaction Type

The first categorization defines four types of messaging transactions used on the wire by the base Routing Protocol. They are:

One-to-One

One peer router directly and intentionally delivers a route update

specifically to one other peer router. Examples are BGP [RFC4271], LDP [RFC5036] [RFC3036], BFD [I-D.ietf-bfd-base] and RSVP [RFC2205]. Point-to-point modes of both IS-IS [RFC1195] and OSPF [RFC2328], when sent over both traditional point-to-point links and when using multi-access layers, may both also fall into this category.

One-to-Many

A router peers with multiple other routers on a single network segment -- i.e. on link local -- such that it creates and sends one route update message which is intended for consumption by multiple peers. Examples would be OSPF and IS-IS in their broadcast, non-point-to-point mode and Routing Information Protocol (RIP) [RFC2453].

Multicast

Multicast protocols have unique security properties because of the fact that they are inherently group-based protocols and thus have group keying requirements at the routing level where link-local routing messages are multicasted. Also, at least in the case of PIM-SM [RFC4601], some messages are sent unicast to a given peer(s), as is the case with router-close-to-sender and the "Rendezvous Point". Some work for application layer message security has been done in the Multicast Security working group (MSEC, http://www.ietf.org/html.charters/msec-charter.html) and may be helpful to review, but is not directly applicable.

2.2. Category: Peer vs Group Keying

The second axis of categorization groups protocols by the keying mechanism that will be necessary for distributing session keys to the actual Routing Protocol transports. They are:

Peer keying

One router sends the keying messages directly and only to one other router, such that a one-to-one, unique keying security association (SA) is established between the two routers. This would be employed by protocols like BGP, BFD, LDP, etc.

Group Keying

One router creates and distributes a single keying message to multiple peers. In this case a group SA will be established and used between multiple peers simultaneously. Group keying exists for protocols like OSPF [RFC2328], and also for multicast protocols like PIM-SM [RFC4601].

3. Consider the future existence of a KMP

When it comes time for the KARP WG to design the re-usable model for a KMP, [RFC4107] should be consulted.

However, when conducting the design work on a manual keyed version of a routing protocol's authentication, consideration must be made for the eventual use of a KMP. In particular, design teams must consider what parameters would need to be handed down to the Routing Protocol by the KMP.

Consider: some sort of security association identifier (e.g. IPsec ESP's SPI, or TCP-AO's KeyID), key life times which may be represented either in bytes or seconds, the cryptographic algorithms being used, the keys themselves, and the direction of the keys (i.e. receiveKey, sendKey).

3.1. Consider Asymmetric Keys

The use of asymmetric keys can be a very powerful way to authenticate machine peers as are found in routing protocol peer exchanges. If generated on the machine, and never moved off the machine, these keys will be very secret, and will not be subject to change if an administrator leaves the organization. Since the keys are totally random, and very long, they are far less susceptible to off-line dictionary and guessing attacks.

An easy and simple way to use asymmetric keys is to start by having the router generate a public/private key pair. At the time of this writing, the keys in the pair SHOULD be no less than 2048bits long (though this length will grow over time). Many routers have the ability to be remotely managed over the SSH [RFC4252] and [RFC4253]. As such, they will also have the ability to generate and store an asymmetric key pair, because this is the commonly used method that users authenticate the SSH service when connecting to the router for management sessions.

Once asymmetric key pair is generated, the KMP generating security association parameters and keys for routing protocol may use the machine's asymmetric keys for the identity proof. The form of the

identity proof could be either raw keys, the more easily administrable self-signed certificate format, or a PKI issued certificate credential.

Regardless which form we eventually standardize, the proof of this identity presentation can be as simple as the SHA-1 fingerprint, which is represented in a very human readable and transferable form of 20 pairs of ASCII characters. More complexly, but also more securely, the identity proof could be verified through the use of a PKI system's revocation checking mechanism, (e.g. Certificate Revocation List (CRL) or OCSP responder). If the SHA-1 fingerprint is used, the solution could be as simple as loading a set of neighbor routers' peer ID strings into a table and listing the associated fingerprint string for each ID string. In most organizations or peering points, this list will not be longer than a thousand or so routers, and often the list will be much much shorter. In other words, the entire list for a given organization's router ID & SHA-1 fingerprints could easily be held in a router's configuration file, uploaded, downloaded and move about at will. And it doesn't matter who sees or gains access to these fingerprint strings, because they are meant to be distributed publicly.

3.2. Cryptographic Keys Life Cycle

Cryptographic keys must have a limited lifetime so that they are vulnerable against cryptanalysis attacks. Each time a key is employed, it generates a cipher text. In case of routing protocols the cipher text is the authentication data that is carried by the protocol

packets. Using the same key repetitively allows an attacker to build up a store of cipher texts which can prove sufficient for a successful cryptanalysis of the key value. It is also worthwhile to note that if the routing protocol is transmitting packets at a high rate then the "long life" may be in order of a few hours. Thus it's the amount of traffic that has been put on the wire using a specific key for authentication and not necessarily the duration for which the key has been in use.

Another reason for limiting the lifetime of a key is to minimize the damage from a compromised key. It is unlikely a user will discover an attacker has compromised his or her key if the attacker remains "passive." Relatively frequent key changes will limit any potential damage from compromised keys.

Thus it is strongly recommended that routing and security protocols do not directly use the long-lived key, but should instead use a key derivation function to derive a short-lived key from the long-lived

key.

The long-lived cryptographic keys used by the routing protocols can be either inserted manually in a database or can make use of an automated key management protocol to do this. In this future environment, we do not anticipate an environment where the automated key management protocol will be used to create short-lived cryptographic session keys for the security of routing protocols.

The cryptographic keying material for individual sessions is derived from the keying material stored in the database of long-lived cryptographic keys [I-D.housley-saag-crypto-key-table]. A key derivation function (KDF) and its inputs are also specified in the database of long-lived cryptographic keys; session specific values based on the routing protocol are input to the KDF. Protocol specific key identifiers may be assigned to the cryptographic keying material for individual sessions if needed.

4. RoadMap

4.1. Work Phases on any Particular Protocol

The desired end state for the KARP work contains several items. First, the people desiring to deploy securely authenticated and integrity validated packets between routing peers have the tools specified, implemented and shipping in order to deploy. These tools should be fairly simple to implement, and not more complex than the security mechanisms to which the operators are already accustomed. (Examples of security mechanisms to which router operators are accustomed include: the use of asymmetric keys for authentication in SSH for router configuration, the use of pre-shared keys (PSKs) in TCP MD5 for BGP

protection, the use of self-signed certificates for HTTPS access to device Web-based user interfaces, the use of strongly constructed passwords and/or identity tokens for user identification when logging into routers and management systems.) While the tools that we intend to specify may not be able to stop a deployment from using "foobar" as an input key for every device across their entire routing domain, we intend to make a solid, modern security system that is not too much more difficult than that. In other words, simplicity and deployability are keys to success. The Routing Protocols will specify modern cryptographic algorithms and security mechanisms. Routing peers will be able to employ unique, pair-wise keys per peering instance, with reasonable key lifetimes, and updating those keys on a somewhat regular basis will be operationally easy, causing no service interruption.

Achieving the above described end-state using manual keys may only be pragmatic in very small deployments. In larger deployments, this end state will be much more operationally difficult to reach with only manual keys. Thus, there will be a need for key life cycle management, in the form of a key management protocol, or KMP. We expect that the two forms, manual key usage and KMP usage, will co-exist in the real world. For example, a provider's edge router at a public exchange peering point will want to use a KMP for ensuring unique and fresh keys with external peers, while a manual key may be used between a provider's access edge router and each of the same provider's customer premise routers with which it peers.

In accordance with the desired end state just described, we define two main work phases for each Routing Protocol:

1. Enhance the Routing Protocol's current authentication mechanism. This work involves enhancing a Routing Protocol's current security mechanisms in order to achieve a consistent, modern level of security functionality within its existing keying framework. It is understood and accepted that the existing keying frameworks are largely based on manual keys. Since many operators have already built operational support systems (OSS) around these manual key implementations, there is some automation available for an operator to leverage in that way, if the underlying mechanisms are themselves secure. In this phase, we explicitly exclude embedding or creating a KMP. Refer to [I-D.ietf-karp-threats-req] for the list of the requirements for Phase 1 work.

2. Develop an automated keying framework. The second phase will focus on the development of an automated keying framework to facilitate unique pair-wise (or perhaps group-wise, where applicable) keys per peering instance. This involves the use of a KMP. A KMP is helpful because it negotiates unique, pair wise, random keys without administrator involvement. It also negotiates several of the SA parameters required for the secure connection, including key life times. It keeps track of those lifetimes using counters, and

negotiates new keys and parameters before they expire, again, without administrator interaction. Additionally, in the event of a breach, changing the KMP key will immediately cause a rekey to occur for the Traffic Key, and those new Traffic Keys will be installed and used in the current connection. In summary, a KMP provides a protected channel between the peers through which they can negotiate and pass important data required to exchange proof of key identifiers, derive Traffic Keys, determine re-keying, synchronize their keying state, signal various keying events, notify with error messages, etc. To address brute force attacks [RFC3562] recommends a key management practice to minimize the possibility of successful attack-- frequent key rotation, limited key sharing, key length restrictions, etc. Advances in computational power due to Moore's law are making that management burden untenable-- keys must be of a size and composition that makes configuration and maintenance difficult or keys must be rotated with an unreasonable frequency. A KMP will help immensely with this growing problem.

The framework for any one Routing Protocol will fall under, and be able to leverage, the generic framework described in [I-D.ietf-karp-framework]

4.2. Work Items Per Routing Protocol

Each Routing Protocol will have a team (the [Routing_Protocol]-KARP team) working on incrementally improving their Routing Protocol's security. These teams will have the following main work items:

PHASE 1:

Characterize the RP

Assess the Routing Protocol to see what authentication mechanisms it has today. Does it needs significant improvement to its existing mechanisms or not? This will include determining if modern, strong security algorithms and parameters are present.

Define Optimal State

List the requirements for the Routing Protocol's session key usage and format to contain to modern, strong security algorithms and mechanisms, per the Requirements document [I-D.ietf-karp-threats-

req]. The goal here is to determine what is needed for the Routing Protocol alone to be used securely with at least manual keys.

Gap Analysis

Enumerate the requirements for this protocol to move from its current security state, the first bullet, to its optimal state, as listed just above.

Transition and Deployment Considerations

Document the operational transition plan for moving from the old to the new security mechanism. Will adjacencies need to bounce? What new elements/servers/services in the infrastructure will be required? What is an example work flow that an operator will take? The best possible case is if the adjacency does not break, but this may not always be possible.

Define, Assign, Design

Create a deliverables list of the design and specification work, with milestones. Define owners. Release a document(s)

PHASE 2:

KMP Analysis

Review requirements for KMPs. Identify any nuances for this particular protocol's needs and its use cases for KMP. List the requirements that this Routing Protocol has for being able to be use in conjunctions with a KMP. Define the optimal state.

Gap Analysis

Enumerate the requirements for this protocol to move from its current security state to its optimal state.

Define, Assign, Design

Create a deliverables list of the design and specification work, with miletsones. Define owners. Do the design and document work for a KMP to be able to generate the Routing Protocol's session keys for the packets on the wire. These will be the arguments passed in

the API to the KMP in order to bootstrap the session keys to the Routing Protocol.

There will also be a team formed to work on the base framework mechanisms for each of the main categories, i.e. the blocks and API's represented in [I-D.ietf-karp-framework].

5. Routing Protocols in Categories

This section groups the Routing Protocols into like categories, according to attributes set forth in Categories Section (Section 2). Each group will have a design team tasked with improving the security of the Routing Protocol mechanisms and defining the KMP requirements for their group, then rolling both into a roadmap document upon which they will execute.

BGP, LDP and MSDP

The Routing Protocols that fall into the category of the one-to-one peering messages, and will use peer keying protocols. BGP [RFC4271] and MSDP [RFC3618] are transmitted over TCP, while LDP [RFC5036] uses UDP. A team will work on one mechanism to cover these TCP unicast protocols. Much of the work on the Routing Protocol update for its existing authentication mechanism is already occuring in the TCPM Working Group, on the TCP-AO [I-D.ietf-tcpm-tcp-auth-opt] document, as well as its cryptography-helper document, TCP-AO-CRYPTO [I-D.ietf-tcpm-tcp-ao-crypto]. However, this cannot be used for LDP as LDP runs over UDP. A separate team might want to look at LDP. Another exception is the mode where LDP is used directly on the LAN. The work for this may go into the Group keying category (along with OSPF) as mentioned below.

OSPF, ISIS, and RIP

The Routing Protocols that fall into the category Group keying with one-to-many peering messages includes OSPF [RFC2328], ISIS [RFC1195] and RIP [RFC2453]. Not surprisingly, all these routing protocols have two other things in common. First, they are run on a combination of the OSI datalink layer 2, and the OSI network layer 3. By this we mean that they have a component of how the routing protocol works which is specified in Layer 2 as well as in Layer 3. Second, they are all internal gateway protocols, or IGPs. The keying mechanisms and use will be much more complicated to define for these than for a one-to-one messaging protocol.

BFD

Because it is less of a routing protocol, per se, and more of a peer aliveness detection mechanism, Bidirectional Forwarding Detection (BFD) will have its own team. BFD is also different from the other protocols covered here as it works on millisecond timers and would need separate considerations to mitigate the potential for DoS attacks. It also raises interesting issues with respect to the sequence number scheme that is generally deployed to protect against the replay attacks as this space can rollover quite frequently because of the rate at which BFD packets are generated.

RSVP and RSVP-TE

The Resource reSerVation Protocol [RFC2205] allows hop-by-hop authentication of RSVP neighbors, as specified in [RFC2747]. In this mode, an integrity object is attached to each RSVP message to transmit a keyed message digest. This message digest allows the recipient to verify the authenticity of the RSVP node that sent the message, and to validate the integrity of the message. Through the inclusion of a sequence number in the scope of the digest, the digest also offers replay protection.

[RFC2747] does not dictate how the key for the integrity operation is derived. Currently, most implementations of RSVP use a statically configured key, per interface or per neighbor.

RSVP relies on per peer authentication mechanism, where each hop authenticates its neighbor with a shared key or certificate.

Trust in this model is transitive. Each RSVP node trusts explicitly only its RSVP next hop peers, through the message digest contained in the INTEGRITY object. The next hop RSVP speaker in turn trusts its own peers and so on. See also the document "RSVP security properties" [RFC4230] for more background.

The keys used for generating the RSVP messages can, in particular, be group keys (for example distributed via GDOI [RFC3547], as discussed in [I-D.weis-gdoi-mac-tek]).

The trust an RSVP node has to another RSVP node has an explicit and an implicit component. Explicitly the node trusts the other node to maintain the RSVP messages intact or confidential, depending on whether authentication or encryption (or both) is used. This means only that the message has not been altered or seen by another, non-

trusted node. Implicitly each node trusts each other node with which it has a trust relationship established via the mechanisms here to adhere to the protocol specifications laid out by the various standards. Note that in any group keying scheme like GDOI a node trusts all the other members of the group.

RSVP TE [RFC3209] [RFC3473] [RFC4726] [RFC5151] is an extension of the RSVP protocol for traffic engineering. It supports the reservation of resources across an IP network and is used for establishing MPLS LSPs, taking into consideration network constraint parameters such as available bandwidth and explicit hops. RSVP-TE signaling is used to establish both intra and inter-domain TE LSPs.

When signaling an inter-domain RSVP-TE LSP, folks MAY make use of the security features already defined for RSVP-TE [RFC3209]. This may require some coordination between the domains to share the keys (see [RFC2747] and [RFC3097]), and care is required to ensure that the keys are changed sufficiently frequently. Note that this may involve additional synchronization, should the domain border nodes be protected with Fast ReRoute, since the merge point (MP) and point of local repair (PLR) should also share the key.

For inter-domain signaling for MPLS-TE, the administrators of neighboring domains MUST satisfy themselves as to the existence of a suitable trust relationship between the domains. In the absence of such a relationship, the administrators SHOULD decide not to deploy inter-domain signaling, and SHOULD disable RSVP-TE on any inter-domain interfaces.

These protocols will be handled together

PIM-SM and PIM-DM

Finally, the multicast protocols of PIM-SM [RFC4601] and PIM-DM [RFC3973] will be handled together. PIM-SM multicasts routing information (Hello, Join/Prune, Assert) on a link-local basis, using a defined multicast address. In addition, it specifies unicast communication for exchange of information (Register, Register-Stop) between the router closest to a group sender and the "rendezvous point" (RP). The RP is typically not "on-link" for a particular router. While much work has been done on multicast security for application-layer groups, little has been done to address the problem of managing hundreds or thousands of small one-to-many groups with link-local scope. Such an authentication mechanism

should be considered along with the router-to-Rendezvous Point authentication mechanism. The most important issue is ensuring that only the "authorized neighbors" get the keys for (S,G), so that rogue routers cannot participate in the exchanges. Another issue is that some of the communication may occur intra-domain, e.g. the link-local messages in an enterprise, while others for the same (*,G) may occur inter-domain, e.g. the router-to-Rendezvous Point messages may be from one enterprise's router to another. One possible solution proposes a region-wide "master" key server (possibly replicated), and one "local" key server per speaking router. There is no issue with propagating the messages outside the

link, because link-local messages, by definition, are not forwarded. This solution is offered only as an example of how work may progress; further discussion should occur in this work team. Specification of a link-local protection mechanism for PIM-SM occurred in RFC 4601 [RFC4601], and this work is being updated in PIM-SM-LINKLOCAL [I-D.ietf-pim-sm-linklocal]. However, the KMP part is completely unspecified, and will require work outside the expertise of the PIM working group to accomplish, which is why this roadmap is being created.

6. Gap Analysis

The [I-D.ietf-karp-threats-req] document lists the generic requirements for the security and authentication mechanisms that must exist for the various routing and signaling protocols that come under the purview of KARP. There will be different design teams working for each of the categories of routing protocols defined.

To start, design teams must review the "Threats and Requirements for Authentication of Routing Protocols" document [I-D.ietf-karp-threats-req]. This document contains detailed descriptions of the threat analysis for routing protocol authentication in general. Note that it will not contain all the authentication-related threats for any one routing protocol, or category of routing protocol. The design team must conduct a threat analysis to determine if specific threats beyond those in the [I-D.ietf-karp-threats-req] document exist, and to describe those threats.

The [I-D.ietf-karp-threats-req] document also contains many requirements around security matters. The different routing protocol design teams must walk through each section of the requirements and determine one by one how their protocol either does or does not address

each requirement. Examples include modern, strong cryptographic algorithms, with at least one such algorithm listed as a MUST; algorithm agility; secure use of simple PSKs; intra-connection replay protection; inter-connection replay protection, etc.

When doing the gap analysis we must first identify the elements of each routing protocol that we wish to protect. In case of protocols riding on top of IP, we might want to protect the IP header and the protocol headers, while for those that work on top of TCP, it will be the TCP header and the protocol payload. There is patently value in protecting the IP header and the TCP header if the routing protocols rely on these headers for some information (for example, identifying the neighbor which originated the packet).

Then there will be a set of Cryptography requirements that we might want to look at. For example, there MUST be at least on set of

cryptography algorithms or constructions whose use is supported by all implementations and can be safely assumed to be supported by any implementation of the authentication option. The design teams should look for this for the protocol that they are working on. If such algorithms or constructions are not available then some should be defined to support interoperability by having a single default.

Design teams MUST ensure that the default cryptographic algorithms and constructions supported by the routing protocols are accepted by the community. This means that the protocols MUST NOT rely on non-standard or ad-hoc hash functions, keyed-hash constructions, signature schemes, or other functions, and MUST use published and standard schemes.

Care should also be taken to ensure that the routing protocol authentication scheme is capable of supporting algorithms other than its defaults, in order to adapt to future discoveries.

Ideally, authentication MUST be performed on routing protocols packets oblivious to the order in which they have arrived, so that it does not get influenced by packets loss and reordering.

Design teams should ensure that their protocols authentication mechanism is able to accommodate rekeying. This is essential since its well known that keys must periodically be changed. Also what the designers must ensure is that this rekeying event MUST NOT affect the functioning of the routing protocol. For example, OSPF rekeying requires coordination among the adjacent routers, while ISIS requires

coordination among routers in the entire domain.

Design teams while defining the new authentication and security mechanisms MUST design in such a manner that the routing protocol authentication mechanism remains oblivious of how the keying material is derived. This decouples the authentication mechanism from the key management system that is employed.

Design teams should also note that many routing protocols require prioritized treatment of certain protocol packets and authentication mechanisms should honor this.

Not all routing protocol authentication mechanisms provide support for replay attacks, and the design teams should identify such authentication mechanisms and work on them so that this can get fixed. The design teams must look at the protocols that they are working on and see if packets captured from the previous/stale sessions can be replayed.

What might also influence the design is the rate at which the protocol packets are originated. In case of protocols like BFD, where packets

are originated at millisecond intervals, there are some special considerations that must be kept in mind when defining the new authentication and security mechanisms.

It is imperative that the new authentication and security mechanisms defined support incremental deployment, as it is not feasible to deploy a new routing protocol authentication mechanism throughout the network instantaneously. It may also not be possible to deploy such a mechanism to all routers in a large AS at one time. This means that the designers must work on this aspect of authentication mechanism for the routing protocol that they are working on. The mechanisms must provide backward compatibility in the message formatting, transmission, and processing of routing information carried through a mixed security environment.

The designers should also consider whether the current authentication mechanisms impose considerable processing overhead on a router that's doing authentication. Most currently deployed routers do not have hardware accelerators for cryptographic processing and these operations can impose a significant processing burden under some circumstances. The proposed solutions should be evaluated carefully with regard to the processing burden that they will impose, since deployment may be impeded if network operators perceive that a solution will impose a processing burden which either entails substantial capital expenses or

threatens to destabilize the routers.

7. Security Considerations

As mentioned in the Introduction, RFC4948 [RFC4948] identifies additional steps needed to achieve the overall goal of improving the security of the core routing infrastructure. Those include validation of route origin announcements, path validation, cleaning up the IRR databases for accuracy, and operational security practices that prevent routers from becoming compromised devices. The KARP work is but one step in a necessary system of security improvements.

The security of cryptographic-based systems depends on both the strength of the cryptographic algorithms chosen and the strength of the keys used with those algorithms. The security also depends on the engineering of the protocol used by the system to ensure that there are no non-cryptographic ways to bypass the security of the overall system.

7.1. Use Strong Keys

Care should be taken to ensure that the selected key is unpredictable, avoiding any keys known to be weak for the algorithm in use. [RFC4086] contains helpful information on both key generation techniques and cryptographic randomness.

In addition to using a strong key/PSK of appropriate length and randomness, deployers of KARP protocols SHOULD use different keys between different routing peers whenever operationally possible. [RFC3562] provides some very sound guidance. It was meant specifically for the use of TCP MD5 for BGP, but it is more or less applicable to Routing Protocol authentication work that would result from KARP. It states three main points: (1) key lengths SHOULD be between 12 and 24 bytes (this will vary depending on the MAC/KDF in use), with larger keys having effectively zero additional computational costs when compared to shorter keys, (2) key sharing SHOULD be limited so that keys aren't shared among multiple BGP peering arrangements, and (3) Keys SHOULD be changed at least every 90 days (this could be longer for stronger MAC algorithms, but it is generally a wise idea).

This is especially true when the Routing Protocol takes a static Traffic Key as opposed to a Traffic Key derived per-connection by a KDF. The burden for doing so is understandable much higher than for using the same static Traffic Key across all peering routers. This is

why use of a KMP network-wide increases peer-wise security so greatly, because now each set of peers can enjoys a unique Traffic Key, and if an attacker sitting between two routers learns or guesses the Traffic Key for that connection, she doesn't gain access to all the other connections as well.

However, whenever using manual keys, it is best to design a system where a given PSK will be used in a KDF, mixed with connection specific material, in order to generate session unique -- and therefore peer-wise -- Traffic Keys. Doing so has the following advantages: the Traffic Keys used in the per-message MAC operation are peer-wise unique, it provides inter-connection replay protection, and, if the per-message MAC covers some connection counter, intra-connection replay protection.

Note that in the composition of certain key derivation functions (e.g. KDF_AES_128_CMAC, as used in TCP-AO [I-D.ietf-tcpm-tcp-ao-crypto], the pseudorandom function (PRF) used in the KDF may require a key of a certain fixed size as an input. For example, AES_128_CMAC requires a 128 bit (16 byte) key as the seed. However, for convenience to the administrators/deployers, a specification may not want to force the deployer to enter a PSK of exactly 16 bytes. Instead, a specification may call for a sub-key routine that could handle a variable length PSK, one that might be less or more than 16 bytes (see [RFC4615], section 3, as an example). That sub-key routine would act as a key extractor to derive a second key of exactly the required length and thus suitable as a seed to the PRF. This does NOT mean that administrators are safe to use weak keys. Administrators are encouraged to follow [RFC4086] as listed above. We simply attempted to "put a fence around stupidity", in as much as possible.

A better option, from a security perspective, is to use some representation of a device-specific asymmetric key pair as the identity proof, as described in section "UniqueVsSharedKeys" section.

7.2. Internal vs. External Operation

The designers must consider whether the protocol is an internal routing protocol or an external one, i.e. Does it primarily run between peers within a single domain of control or between two different domains of control? Some protocols may be used in both cases, internally and externally, and as such various modes of authentication operation may be required for the same protocol. While it is preferred that all routing exchanges run with the utmost security mechanisms enabled in

all deployments, this exhortation is greater for those protocols running on inter-domain point-to-point links, and greatest for those on shared access link layers with several different domains interchanging together, because the volume of attackers are greater from the outside. Note however that the consequences of internal attacks maybe no less severe -- in fact they may be quite a bit more severe -- than an external attack. An example of this internal versus external consideration is BGP which has both EBGP and IBGP modes. Another example is a multicast protocol where the neighbors are sometimes within a domain of control and sometimes at an inter-domain exchange point. In the case of PIM-SM running on an internal multi-access link, it would be acceptable to give up some security to get some convenience by using a group key between the peers on the link. On the other hand, in the case of PIM-SM running over a multi-access link at a public exchange point, operators may favor security over convenience by using unique pair-wise keys for every peer. Designers must consider both modes of operation and ensure the authentication mechanisms fit both.

Operators are encouraged to run cryptographic authentication on all their adjacencies, but to work from the outside in, i.e. The EBGP links are a higher priority than the IBGP links because they are externally facing, and, as a result, more likely to be targeted in an attack.

7.3. Unique versus Shared Keys

This section discusses security considerations regarding when it is appropriate to use the same authentication key inputs for multiple peers and when it is not. This is largely a debate of convenience versus security. It is often the case that the best secured mechanism is also the least convenient mechanism. For example, an air gap between a host and the network absolutely prevents remote attacks on the host, but having to copy and carry files using the "sneaker net" is quite inconvenient and unscalable.

Operators have erred on the side of convenience when it comes to securing routing protocols with cryptographic authentication. Many do not use it at all. Some use it only on external links, but not on internal links. Those that do use it often use the same key for all peers across their entire network. It is common to see the same key in use for years, and that being the same key that was entered when authentication was originally configured, or the routing gear deployed.

The goal for designers is to create authentication mechanisms that are easy for the operators to deploy and manage, and still use unique keys between peers (or small groups on multi-access links), and within between sessions. Operators have the impression that they NEED one key shared across the network, when in fact they do not. What they need is the relative convenience they experience from deploying cryptographic authentication with one (or few) key, compared to the inconvenience they would experience if they deployed the same authentication mechanism using unique pair-wise keys. An example is BGP Route Reflectors. Here operators often use the same authentication key between each client and the route reflector. The roadmaps defined from this guidance document will allow for unique keys to be used between each client and the peer, without sacrificing much convenience. Designers should strive to deliver peer-wise unique keying mechanisms with similar ease-of-deployment properties as today's one-key method.

Operators must understand the consequences of using the same keys across many peers. Unique keys are more secure than shared keys because they reduce both the attack target size and the attack consequence size. In this context, the attack target size represents the number of unique routing exchanges across a network that an attacker may be able to observe in order to gain security association credentials, i.e. crack the keys. If a shared key is used across the entire internal domain of control, then the attack target size is very large. The larger the attack target, the easier it is for the attacker to gain access to analysis data, and greater the volume of analysis data he can access in a given time frame, both of which make his job easier. Using the same key across the network makes the attack vulnerability surface more penetrable than unique keys. Consider also the attack consequence size, the amount of routing adjacencies that can be negatively affected once a breach has occurred, i.e. once the keys have been acquired by the attacker.

Again, if a shared key is used across the internal domain, then the consequence size is the whole network. Ideally, unique key pairs would be used for each adjacency.

In some cases designers may need to use shared keys in order to solve the given problem space. For example, a multicast packet is sent once but then observed and consumed by several routing neighbors. If unique keys were used per neighbor, the benefit of multicast would be erased

because the casting peer would have to create a different announcement packet/stream for each listening peer. Though this may be desired and acceptable in some small amount of use cases, it is not the norm.

Shared group keys are an acceptable solution here, and much work has been done already in this area (see MSEC working group).

7.4. Out-of-Band vs. In-line Key Management

This section discusses the security and use case considerations for keys placed on devices through out-of-band configurations versus through one routing peer-to-peer key management protocol exchanges. Note, when we say here "Peer-to-Peer KMP" we do not mean in-band to the Routing Protocol. Instead, we mean that the exchange occurs in-line, over IP, between the two routing peers directly. In in-line KMP the peers themselves handle the key and security association negotiation between themselves directly, whereas in an out-of-band system the keys are placed onto the device through some other configuration or management method or interface.

An example of an out-of-band mechanism could be an administrator who makes a remote management connection (e.g. using SSH) to a router and manually enters the keying information -- like the algorithm, the key(s), the lifetimes, etc. Another example could be an OSS system which inputs the same information via a script over an SSH connection, or by pushing configuration through some other management connection, standard (Netconf-based) or proprietary.

The drawbacks of an out-of-band mechanism include: lack of scale-ability, complexity and speed of changing if a breach is suspected. For example, if an employee who had access to keys was terminated, or if a machine holding those keys was believed to be compromised, then the system would be considered insecure and vulnerable until new keys were defined by a human. Those keys then need to be placed into the OSS system, manually, and the OSS system then needs to push the change -- often during a very limited change window -- into the relevant devices. If there are multiple organizations involved in these connections, this process is greatly complicated.

The benefits of out-of-band mechanism is that once the new keys/parameters are set in OSS system they can be pushed automatically to all devices within the OSS's domain of control. Operators have mechanisms in place for this already. In small environments with few routers, a manual system is not difficult to employ.

We further define an in-line key exchange as using cryptographically protected identity verification, session key negotiation, and security association parameter negotiation between the two routing peers. The KMP between the two peers may also include the negotiation of

parameters, like algorithms, cryptographic inputs (e.g. initialization vectors), key life-times, etc.

The benefits an in-line KMP are several. An in-line KMP results in key(s) that are privately generated, and not recorded permanently anywhere. Since the traffic keys used in a particular connection are not a fixed part of a device configuration no steal-able data exists anywhere else in the operator's systems which can be stolen, e.g. in the case of a terminated or turned employee. If a server or other data store is stolen or compromised, the thieves gain no access to current traffic keys. They may gain access to key derivation material, like a PSK, but not current traffic keys in use. In this example, these PSKs can be updated into the device configurations (either manually or through an OSS) without bouncing or impacting the existing session at all. In the case of using raw asymmetric keys or certificates, instead of PSKs, the data theft would likely not even result in any compromise, as the key pairs would have been generated on the routers, and never leave those routers. In such a case no changes are needed on the routers; the connections will continue to be secure, uncompromised. Additionally, with a KMP regular re-keys operations occur without any operator involvement or oversight. This keeps keys fresh.

The drawbacks to using a KMP are few. First, a KMP requires more cryptographic processing for the router at the very beginning of a connection. This will add some minor start-up time to connection establishment versus a purely manual key approach. Once a connection with traffic keys have been established via a KMP, the performance is the same in the KMP and the out-of-band case. KMPs also add another layer of protocol and configuration complexity which can fail or be mis-configured. This was more of an issue when these KMPs were first deployed, but less so as these implementations and operational experience with them has matured.

The desired end goal is in-line KMPs.

8. Acknowledgments

Much of the text for this document came originally from draft-lebovitz-karp-roadmap, authored by Gregory M. Lebovitz.

We would like to thank Russ White, Michael Barnes and Vishwas Manral for their comments on the draft.

9. IANA Considerations

This document places no requests to IANA.

10. References

10.1. Normative References

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

[RFC4948] Andersson, L., et. al, "Report from the IAB workshop on Unwanted Traffic March 9-10, 2006", RFC 4948, August 2007.

10.2. Informative References

[RFC1195] Callon, R. , "Use of OSI IS-IS for Routing in TCP/IP and Dual Environments", RFC 1195, December 1990.

[RFC2205] Braden, R., et. al, "Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification", RFC 2205, September 1197.

[RFC2328] Moy, J., "OSPF Version 2", RFC 2328, April 1998.

[RFC2453] Malkin, G., "RIP Version 2", RFC 2453, November 1998.

[RFC2747] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", RFC 2747, January 2000.

[RFC3036] Andersson, L., et. al, "LDP Specification", RFC 3036, January 2001.

[RFC3097] Braden, R, and Zhang, L., "RSVP Cryptographic Authentication -- Updated Message Type Value", RFC 3097, April 2001

[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.

[RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC3547] Baugher, M., Weis, B., Hardjono, T., and H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003.

[RFC3562] Leech, M., "Key Management Considerations for the TCP MD5 Signature Option", RFC 3562, July 2003.

[RFC3618] Fenner, B. and D. Meyer, "Multicast Source Discovery Protocol (MSDP)", RFC 3618, October 2003.

[RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, January 2005.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

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

[RFC4230] Tschofenig, H. and R. Graveman, "RSVP Security Properties", RFC 4230, December 2005.

[RFC4252] Ylonen, T., et. al, "The Secure Shell (SSH) Authentication Protocol", RFC 4252, January 2006.

[RFC4253] Ylonen, T., et. al, "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, January 2006

[RFC4271] Rekhter, Y., Li, T. and Hares, S.,"A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas,"Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.

[RFC4615] Song, J., Poovendran, R., Lee, J., and T. Iwata, "The Advanced Encryption Standard-Cipher-based Message Authentication Code-Pseudo-Random Function-128 (AES-CMAC-PRF-128) Algorithm for the Internet Key Exchange Protocol (IKE)", RFC 4615, August 2006.

[RFC4726] Farrel, A., et. al.,"A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering", RFC 4726, November 2006.

[RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification",

RFC 5036, October 2007.

[RFC5151] Farrel, A., et. al.,"Inter-Domain MPLS and GMPLS Traffic Engineering -- Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions", February 2008.

[I-D.ietf-bfd-base] Katz, D. and Ward, D., "Bidirectional Forwarding Detection", Work in Progress, January 2010.

[I-D.ietf-tcpm-tcp-ao-crypto] Lebovitz, G., "Cryptographic Algorithms, Use and Implementation Requirements for TCP Authentication Option", Work in Progress, March 2009.

[I-D.ietf-karp-threats-req] Lebovitz, G., "KARP Threats and Requirements", Work in Progress, February 2010.

[I-D.ietf-karp-framework] Lebovitz, G., "Framework for Cryptographic Authentication of Routing Protocol Packets on the Wire", Work in Progress, February 2010.

[I-D.ietf-pim-sm-linklocal] Atwood, W., Islam, S., and M. Siami, "Authentication and Confidentiality in PIM-SM Link-local Messages", Work in Progress, December 2009.

[I-D.ietf-tcpm-tcp-auth-opt] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", Work in Progress), October 2009.

[I-D.housley-saag-crypto-key-table] Housley, R. and Polk, T., "Database of Long-Lived Cryptographic Keys" , Work in Progress, September 2009

[I-D.weis-gdoi-mac-tek] Weis, B. and S. Rowles, "GDOI Generic Message Authentication Code Policy", Work in Progress, July 2008.

[IRR] Merit Network Inc , "Internet Routing Registry Routing Assets Database", 2006, http://www.irr.net/.

Author's Addresses

Gregory M. Lebovitz
Juniper Networks, Inc.
1194 North Mathilda Ave.

Sunnyvale CA 94089-1206
USA

Phone:
Email: gregory.ietf@gmail.com

Manav Bhatia
Alcatel-Lucent
Bangalore
India

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