Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 31, 2007

M. Behringer
F. Le Faucheur
Cisco Systems Inc
June 29, 2007

A Framework for RSVP Security Using Dynamic Group Keying
draft-behringer-tsvwg-rsvp-security-groupkeying-00.txt

Status of this Memo

Copyright Notice

Abstract

The Resource reSerVation Protocol (RSVP) allows hop-by-hop authentication of RSVP neighbors. This requires messages to be cryptographically signed using a shared secret between participating nodes. This document compares group keying for RSVP with per neighbor or per interface keying, and discusses the applicability and limitations of these approaches. Draft-weis-gdoi-for-rsvp describes how the Group Domain of Interpretation (GDOI) can be used to distribute group keys to RSVP nodes. The document also discusses

applicability of group keying to RSVP encryption.

Table of Contents

1. Introduction and Problem Statement

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 sender and validate 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. However, to manually configure key per router pair across an entire network is operationally hard, especially for key changes. Effectively, many users of RSVP therefore resort to the same key throughout their network, and change it rarely if ever, because of the operational burden. [RFC 3562] however recommends regular key changes, at least every 90 days.

[I-D.weis-gdoi-for-rsvp] provides an alternative solution, using GDOI ([RFC3547]) for key distribution. This allows dynamic key updates, valid for the entire group of RSVP speakers.

The present document describes the various keying methods and their applicability to different RSVP deployment environments, for both message integrity and encryption. It does not mandate any particular method, but is meant as a comparative guideline to understand where each RSVP keying method is best deployed, and where it cannot be deployed. Furthermore, it discusses the impact on RSVP hop by hop authentication of non-RSVP nodes, as well as subverted nodes, in the reservation path.

2. The RSVP Trust Model

Many protocol security mechanisms used in networks require and use per peer authentication. Each hop authenticates its neighbor with a shared key or certificate. This is also the model used for RSVP. Trust in this model is transitive. Each RSVP node trusts explicitely 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.

The RSVP protocol can operate in the presence of a non-RSVP router in the path from the sender to the receiver. The non-RSVP hop will ignore the RSVP message and just pass it along. The next RSVP node can then process the RSVP message. For RSVP authentication to work

in this case, the key used for computing the RSVP message digest needs to be shared by the two RSVP neighbors, even if they are not IP neighbors. However, in the presence of non-RSVP hops, while an RSVP node always know the next IP hop before forwarding an RSVP Message, it does not always know the RSVP next hop. Thus, the presence of non-RSVP hops impacts operation of RSVP authentication and may influence the keying approaches. This is further discussed in Section 5.3.

3. Key types for RSVP

3.1. Interface based keys

Most current implementations support interface based RSVP keys. All RSVP speakers on a given subnet have to share the same key in this model, which makes it unsuitable for deployment scenarios where different trust groups share a subnet, for example Internet exchange points. In such a case, neighbor based keys are required.

3.2. Neighbor based keys

In this model, an RSVP key is bound to an interface plus a neighbor on that interface. It allows the distinction of different trust groups on a single subnet. (Assuming that layer-2 security is correctly implemented to prevent layer-2 attacks.)

3.3. Group keys

Here, all members of a group of RSVP nodes share the same key. This implies that a node uses the same key regardless of the next RSVP hop that will process the message (within the group of nodes sharing the particular key). It also implies that a node will use the same key on the receiving as on the sending side (when exchanging RSVP messages withn the group).

4. Key Provisioning Methods for RSVP

4.1. Static Key Provisioning

The simplest way to implement RSVP authentication is to use static, preconfigured keys. However, on the operational side key management is heavy, since no secure automated mechanism can be used. This method is therefore mostly useful for small deployments, where key changes can be carried out manually, or for deployments with automated configuration tools which support key changes.

Static key provisioning is therefore not an ideal model in a large network.

Often, the number of interconnection points across two domains where RSVP is allowed to transit is relatively small and well controlled. Also, the different domains may not be in a position to use an infrastructure trusted by both domains to update keys on both sides. Thus, manually configured keys may be applicable to inter-domain RSVP authentication.

Since it is not practical to carry out the key change at the exact same time on both sides, some grace period nees to be implemented during which an RSVP node will accept both the old and the new key. Otherwise, RSVP operation would suffer interruptions.

4.2. Per Neighbor Key Negotiation

To avoid the problem of key rollover in static key deployments, per neighbor key negotiation could be used. However, existing key distribution protocols may not be appropriate in all environments because of the complexity or operational burden they involve.

4.3. Dynamic Key Distribution using GDOI

[I-D.weis-gdoi-for-rsvp] describes a mechanism to distribute group keys to a group of RSVP speakers, using GDOI [RFC3547]. In this model, a key server authenticates each of the RSVP nodes independently, and then distributes a group key to the entire group.

5. Applicability of Various Keying Methods for RSVP

5.1. Per Neighbor or Per Interface Keys for Authentication

Per interface and per peer keys can be used within a single security domain. As mentioned above, per interface keys are only applicable when all the hosts reachable on the specific interface belong to the same security domain.

These key types can also be used between security domains, since they are specific to a particular interface or neighbor. Again, interface level keys can only be deployed safely when all the reachable neighbors on the interface belong to the same security domain.

As discussed in Section 5.3, per neighbor and per interface keys can not be used in the presence of non-RSVP hops.

5.2. Group Keys for Authentication

Group keys apply naturally to intra-domain RSVP authentication, since all RSVP nodes trust each other, and trust the group key server in this model. This is presented Figure 1.

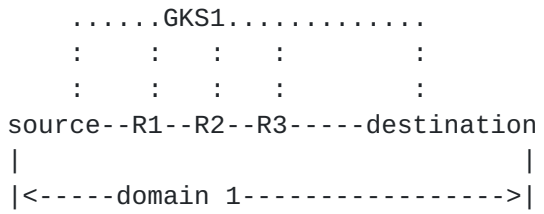

Figure 1: Group Key Server within a single security domain

A single group key cannot normally be used to cover multiple security domains however, because by definition the different domains do not trust each other and would not be willing to trust the same group key server. For a single group key to be used in several security domains, there is a need for a single group key server, which is trusted by both sides. While this is theoretically possible, in practice it is unlikely that there is a single such trusted entity. Figure 2 illustrates this setup.

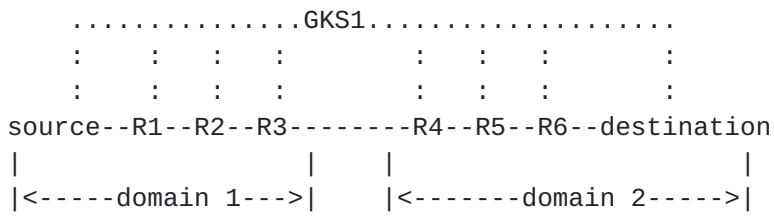

Figure 2: A Single Group Key Server across security domains

A more practical approach for RSVP operation across security domains, to use a separate group key server for each security domain, and to use per interface or per peer authentication between the two domains. Figure 3 shows this set-up.

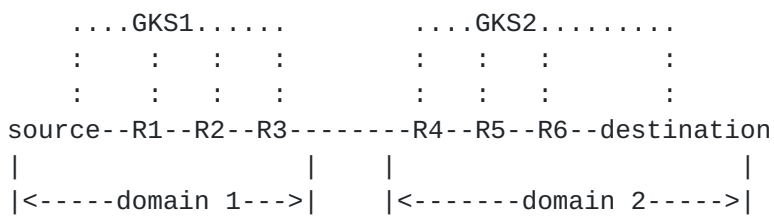

Figure 3: A group Key Server per security domain

5.3. Non-RSVP Hops

In the presence of a non-RSVP router in the path from the sender to the receiver, regular RSVP keeps working. The non-RSVP node ignores the RSVP message, and passes it on transparently to the next node. Figure 4 illustrates this scenario. R2 in this picture does not participate in RSVP, the other nodes do. In this case, R2 will pass on any RSVP messages unchanged, and will ignore them.

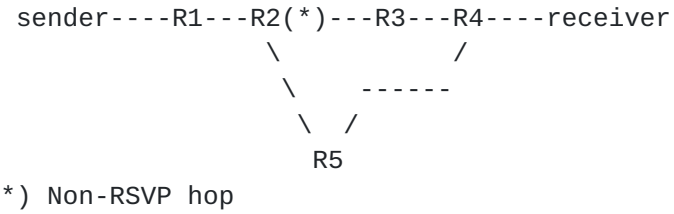

Figure 4: A non-RSVP Node in the path

However, this creates an additional challenge for RSVP authentication. In the presence of a non-RSVP hop, with some RSVP messages such as a Path message, an RSVP router does not know the RSVP next hop for that message at the time of forwarding it. In fact, part of the role of a Path message is precisely to discover the RSVP next hop (and to dynamically re-discover it when it changes, say because of a routing change). For example, in Figure 4, R1 knows that the next IP hop for a Path message addresed to the receiver is R2, but it does necessarily not know if the RSVP next hop is R3 or R5.

This means that per interface and per neighbor keys cannot easily be used in the presence of non-RSVP routers on the path between senders and receivers.

By contrast, group keying will naturally work in the presence of non-RSVP routers. Referring back to Figure 4, with group keying, R1 would use the group key to sign a Path message addressed to the receiver and forwards it to R2. Being a non-RSVP node, R2 and will ignore and forward the Path message to R3 or R5 depending on the current shortest path as determined by routing. Whether it is R3 or

R5, the RSVP router that receives the Path message will be able to authenticate it successfully with the group key.

5.4. Subverted RSVP Nodes

A subverted node is defined here as an untrusted node, for example because an intruder has gained control over it. Since RSVP authentication is hop-by-hop and not end-to-end, a subverted node in the path breaks the chain of trust. This is to a large extent independent of the type of keying used.

For interface or per-peer keying, the subverted node can now introduce fake messages to its neighbors. This can be used in a variety of ways, for example by changing the receiver address in the Path message, or by generating fake Path messages. This allows path states to be created on every RSVP router along any arbitrary path through the RSVP domain. That in itself could result in a form of Denial of Service by allowing exhaustion of some router resources (e.g. memory). The subverted node could also generate fake Resv messages upstream corresponding to valid Path states. In doing so, the subverted node can reserve excessive amounts of bandwidth thereby possibly performing a denial of service attack.

It has to be noted specifically that even though the per interface or per neighbor keys have only local significance, the messages themselves can be created arbitrarily so that they are then authenticated and forwarded by the RSVP neighbor of the subverted node, eventually potentially affecting the entire RSVP domain.

For group keying the impact of subverted nodes on the path is comparable. Group keying allows the additional abuse of sending fake messages to any node in the RSVP domain, however, in practice this can be achieved to a large extend also with per neighbor keys, as discussed above.

5.5. RSVP Encryption

The keying material can also be used to encrypt the RSVP messages, instead of, or in addition to authenticating them. The same considerations apply for this case as discussed above for the authentication case. Group keys are applicable only within a trusted domain, but they have the potential of passing a non-RSVP speaker without further configuration. Per interface or per nighbor keys work also inter-domain, but do not operate in the presence of a non-RSVP router.

6. Security Considerations

This entire document discusses security of the RSVP authentication and encryption mechanisms, depending on the key scheme used.

7. Informative References

[I-D.weis-gdoi-for-rsvp] Weis, B., "Group Domain of Interpretation (GDOI) support for RSVP", July 2007.

[RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

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

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

[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.

Authors' Addresses

Michael H. Behringer
Cisco Systems Inc
Village d'Entreprises Green Side
400, Avenue Roumanille, Batiment T 3
Biot - Sophia Antipolis 06410
France

Email: mbehring@cisco.com
URI: http://www.cisco.com

Francois Le Faucheur
Cisco Systems Inc
Village d'Entreprises Green Side
400, Avenue Roumanille, Batiment T 3
Biot - Sophia Antipolis 06410
France

Email: flefauch@cisco.com
URI: http://www.cisco.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).