Network Working Group	M. Bhatia
Internet-Draft	Alcatel-Lucent
Intended status: Standards Track	S. Hartman
Expires: September 08, 2011	Painless Security
	D. Zhang
	Huawei
	March 07, 2011

A Generic Mechanism to solve Inter-Session Replay Attacks for Routing and Signaling Protocols

draft-bhz-karp-inter-session-replay-00

Abstract

This draft proposes a common solution for routing protocols to enhance their capability in tolerating inter-session replay attacks when using manual keys for securing their protocol packets.

Requirements Language

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

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

A replay attack is a network attack where an adversary intercepts a valid message transmission and retransmits it sometime later. In certain types of replay attacks, the retransmitted message may also be carefully tampered with. [RFC6039] demonstrates that nearly all the routing protocols and their security mechanisms are vulnerable to replay attacks to some extent. These attacks permit attackers multiple capabilities. Often, by replaying packets, attackers can create a disruption, causing routing information to be removed or signaling to fail because of the attack. Other replays permit an attacker to mask network failures. For example an attacker can maintain an adjacency even when a link or router has failed, allowing the attacker to observe traffic or forcing traffic to be blackholed. Another class of replay attacks permits an attacker to inject old routing information, possibly in place of routing information from a router that is currently down. Successful replay attacks on routing protocols can introduce incorrect routing information into the victims' routing tables, can break their adjacencies, and can eventually disrupt network communication.
Replays may be effective even with very little effort on the part of an attacker. For instance, replaying an OSPF Hello packet with an empty neighbor list can cause all the neighbor adjacencies with the router which originally sent the packet to be reset. All the existing security

mechanisms for routing protocols use a non-decreasing cryptographic sequence number to deal with replay attacks. However, this leaves the routers still vulnerable to inter-connection replay attacks where the packets from one session are re-sent and accepted during a later session. None of the existing authentication mechanisms in the routing protocols can prevent this without the assistance of automatic key management mechanisms.

Providing routing protocols with an inter-session replay protection is one of the threats that has been recognized in scope for the work being done in the KARP WG and has been documented in [I-D.ietf-karp-threats-reqs]. This document proposes to provide a generic solution that can be implemented as part of the KARP framework that can be used by all routing and signaling protocols to prevent inter-session replay attacks.

This document proposes introducing a boot count, denoted as the KARP Boot Count (KBC), to enhance the capability of routing protocols in tolerating inter-session replay attacks. KBC is used to record the number of times a router has cold-booted. As a part of the KARP infrastructure, the value of this count must be maintained by all the implementations compliant to this standard in their non-volatile memory.

The following sections explain why the existing security and authentication mechanisms cannot protect the routing and the signaling protocols against inter-session replay attacks. The proposed solution is then introduced and we explain how unlike the existing anti-replay mechanisms, this solution will also work well with automated key management techniques.

2. Existing Mechanisms

Most routing protocols (e.g., OSPF, BFD, and RIP) and signaling protocols (LDP, RSVP, etc) include a non-decreasing cryptographic sequence number within the authentication data of each new packet that a router originates. The receivers keep track of this sequence number and only accept a protocol packet if it carries a cryptographic sequence number that is greater than or equal to the cryptographic sequence number carried in the last valid protocol packet. Using this mechanism, receivers can trivially protect the router against simple replay attacks.

[RFC2328] uses a 32-bit non-decreasing crypto sequence number for every OSPFv2 packet. Once a router has increased its sequence number, an attacker cannot replay an old packet to a neighbor that has an active adjacency without being detected. Note that the sequence numbers are not required to increase for each packet. Additionally, OSPFv2 provides a per-LSA sequence number to prevent an old LSA from being installed.

OSPFv3 [RFC5340] relies on the IP Authentication Header (AH) [RFC4302] and the IP Encapsulating Security Payload (ESP) [RFC4303] to cryptographically sign routing information passed between routers.

[RFC4552] describes the authentication mechanism that OSPFv3 uses. It discusses, at length, the reasoning behind using manually configured keys, rather than some automated key management protocol such as IKEv2 [RFC4306]. The primary problem is the lack of a suitable key management mechanism, as OSPFv3 adjacencies are formed on a one-to-many basis and most key management mechanisms are designed for a one-to-one communication model. Since [RFC4552] uses manual keying it clearly states that it provides no protection against replay attacks. This can be exploited in several ways as described in [RFC6039].

The OSPF WG is currently working on an alternate mechanism [I-D.ietf-ospf-auth-trailer-ospfv3] to protect OSPFv3 protocol packets that does not depend upon IPsec for authentication. This draft proposes a new mechanism that works similar to OSPFv2 [RFC5709] for providing authentication to the OSPFv3 packets and as a side effect also solves the replay protection problems that exists in OSPFv3.

As part of the solution OSPFv3 routers append a special data block, referred to as, the authentication trailer to the end of the OSPFv3 packets. It contains a 32-bit non decreasing cryptographic sequence number that is used to protect against the replay attacks.

Bidirectional Forwarding Detection (BFD) is specified in [RFC5880]. There is a 32-bit cryptographic sequence number associated with every BFD packet that is used to protect against replay attacks. Note that the sequence number is incremented for each successive packet transmitted within a session for Meticulous Keyed (MD5 or SHA-1) Authentication. When using Keyed (MD5 or SHA-1) Authentication (the non-meticulous variant), the receiver of a packet only requires the sequence number of the packet to be greater than or equal to the last sequence number received.

In order to improve the anti-replay capability of RSVP, a 64-bit monotonically increasing sequence number is associated with every RSVP packet[RFC2747].

3. Inter-Session Replay Attacks

In the security mechanisms where the per-packet sequence numbers only need to be updated occasionally, replay attacks can be quite intuitive. For instance, an attacker can replay the last OSPFv2 packet without being detected since a router executing OSPFv2 accepts packets with sequence number greater than or equal to what they had last received. Of course, this issue can be easily addressed by mandating that protocols must only accept protocol packets if they come with a sequence number that is greater than what they have received till now. However, even if the sequence numbers are monotonically increased, the security mechanisms for routing protocols are still vulnerable to "inter-session" replay attacks if automatic key management mechanisms are unavailable. In normal conditions, it will take a very long period for a sequence number to reach its maximum. However, on many occasions (e.g., reboot), a router may re-initialize its sequence number. In this case, the sequence number of new packets is less than the sequence number of packets previously sent on the link. If an adversary replays the packets intercepted before the re-initialization, it is difficult for the victims to distinguish a replayed packet from the valid ones.

4. Proposal

The basic idea of the proposed solution is to guarantee that the sequence number of a router will always monotonically increase even after a cold reboot. The first part of the solution requires that the sequence numbers increase for every packet, updating the requirement of protocols such as OSPFv2 that only require non-decreasing behavior. This also means that BFD should use the meticulous version of the authentication mechanism as against the regular, since the former requires the cryptographic sequence number to increase for each successive packet that is transmitted for a session. It is insufficient to update the behavior of senders in this regard: receivers MUST check that sequence numbers increase for every packet.

The second part of the solution requires routing protocol implementations to maintain a KARP boot count (KBC) that records the number of times the router has cold booted in a non-volatile storage, similar to how it is done in the SNMPv3 security architecture. In fact, the same boot count MAY also be shared by SNMPv3 and the KARP infrastructure. Before sending out a packet, the routing protocols can request for this count value and can append it before the sequence space that it maintains. How each routing protocol achieve this is an implementation specific issue and beyond the scope of this document.

If the sequence number of a routing protocol (e.g., RSVP) is 64 bits, the sequence space is then broken down to two halves. The most significant 32-bits would indicate the KARP boot count. The least significant 32-bits is a counter that increases for every packet sent.

If the cryptographic sequence number of a routing protocol is 32 bits, it is recommended to extend the sequence number space to 64 bits. The most significant 32-bits would indicate the KARP boot count. The least significant 32-bits would carry the current sequence number that protocols maintain, which increases with each successive packet transmitted within a session. Upon receiving a packet, the receiver MUST verify that the sequence number in the packet is strictly greater than the sequence number of the previous packets received.

In the later case, if an implementation does not intend to expand the length of the sequence number, it could divide this 32-bit cryptographic sequence number space into a 7-bit and a 25-bit field. The most significant 7-bits could then indicate the KARP boot count. The least significant 25-bits is a counter that increases for every packet sent.

This solution assumes that boot counts never wrap within the lifetime of a particular encryption key. Also, the solution assumes that nonvolatile storage is always updated on a boot. Under these assumptions, a sequence number will not be re-used. This is sufficient to guarantee that while two routers are exchanging communications, packets from an old session cannot be replayed. However it does not demonstrate freshness. Many routing protocols discard replay state when an adjacency is dropped or when a router reboots. Once this state is discarded, an attacker can successfully replay packets from an old session. See the discussion in Section 5.

5. Security Considerations

This solution does not try to provide guarantees of freshness: it does not protect against the replay of an antique session while a router is down. For instance, if an OSPF router is taken out of service for some reason, an attacker can replay packets as soon as the adjacencies with the router time out. Actually, this issues is a common problem encountered by all existing anti-replay solutions for routing protocols. To address this issue, the liveliness of routers would need to be checked before the generation of any adjacency. The challenge/ response solution is proposed in[I-D.bhatia-karp-ospf-ip-layer-protection] to address this issue.

Updates to routing protocols that use this solution need to discuss residual attacks, particularly those resulting from the lack of freshness guarantees. For example this solution would likely be insufficient for RIPv2 because as soon as a router goes down, old packets from that router could be used to inject routing information. However attacks against a link-state protocol may be quite limited and this solution may be appropriate.

The security of this solution depends on the boot count always increasing for each new boot unless the key changes. This creates significant operational requirements. If equipment is replaced but its router identity (an IP address for several protocols) is re-used, then the key MUST be changed or the boot count preserved from the old equipment. Failure to take one of these steps permits attackers to replay packets from the old equipment until the boot count of the new equipment catches up with that of the old equipment. This will very likely permit an attacker to disrupt adjacencies between the new equipment and other routers. More serious attacks may be possible as well.

6. IANA Considerations

The implementations that decide to extend their sequence space from 32 bits to 64 bits need to require a new Auth Type from IANA as this will be incompatible with the earlier authentication mechanisms.

7. Acknowledgements

The funding for Sam Hartman's work on this draft is provided by Huawei.

8. References

8.1. Normative References

[RFC2082]	Baker, F., Atkinson, R. and G.S. Malkin, "RIP-2 MD5 Authentication", RFC 2082, January 1997.
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[RFC2328]	Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
[RFC2747]	Baker, F., Lindell, B. and M. Talwar, "RSVP Cryptographic Authentication", RFC 2747, January 2000.
[RFC4552]	Gupta, M. and N. Melam, "Authentication/Confidentiality for OSPFv3", RFC 4552, June 2006.
[RFC5340]	Coltun, R., Ferguson, D., Moy, J. and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.
[RFC5880]	Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.

8.2. Informative References

[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.
[RFC4306]	Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.
[RFC4302]	Kent, S., "IP Authentication Header", RFC 4302, December 2005.
[RFC4303]	Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
[RFC6039]	Manral, V., Bhatia, M., Jaeggli, J. and R. White, "Issues with Existing Cryptographic Protection Methods for Routing Protocols", RFC 6039, October 2010.
[I-D.ietf-ospf-auth-trailer-ospfv3]	Bhatia, M, Manral, V and A Lindem, "Supporting Authentication Trailer for OSPFv3", Internet-Draft draft-ietf-ospf-auth-trailer-ospfv3-11, November 2011.
[I-D.ietf-karp-threats-reqs]	Lebovitz, G, Bhatia, M and R White, "The Threat Analysis and Requirements for Cryptographic Authentication of Routing Protocols' Transports", Internet-Draft draft-ietf-karp-threats-reqs-03, June 2011.
[I-D.bhatia-karp-ospf-ip-layer-protection]	Bhatia, M, Hartman, S and D Zhang, "Security Extension for OSPFv2 when using Manual Key Management", Internet-Draft draft-bhatia-karp-ospf-ip-layer-protection-03, February 2011.

Authors' Addresses

Manav Bhatia Bhatia Alcatel-Lucent India EMail: manav.bhatia@alcatel-lucent.com

Sam Hartman Hartman Painless Security USA EMail: hartmans@painless-security.com

Dacheng Zhang Zhang Huawei China EMail: zhangdacheng@huawei.com