Hop Integrity and the Security of Routing Protocols

C.-T. Huang E. N. Elnozahy† M. G. Gouda
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712-1188
{chuang, gouda}@cs.utexas.edu

† *IBM Austin Research Lab*
11400 Burnet Rd, M/S 9460
Austin, TX 78758
mootaz@us.ibm.com

Abstract

A computer network is said to provide hop integrity iff when any router p in the network receives a message m supposedly from an adjacent router q, then p can check that m was indeed sent by q, was not modified after it was sent, and was not a replay of an old message sent from q to p. In this paper, we describe how to achieve hop integrity in networks that support Internet Protocol (IP). Then, we use three famous protocols used in IP networks, namely RIP, OSPF, and RSVP, to illustrate how hop integrity can secure the communications between adjacent routers.

Keywords: *authentication, Internet, network protocol, router, security, denial-of-service attack, smurf attack, message insertion, message modification, message replay, RIP, OSPF, RSVP.*

1. Introduction

Most computer networks suffer from the following security problem: in a typical network, an adversary, that has an access to the network, can insert new messages, modify current messages, or replay old messages in the network. In many cases, the inserted, modified, or replayed messages can go undetected for some time until they cause severe damage to the network. More importantly, the physical location in the network where the adversary inserts new messages, modifies current messages, or replays old messages may never be determined.

A well-known example of such an attack in networks, that supports the Internet Protocol (IP), is called the smurf attack. In smurf attack, an adversary utilizes the "ping" message in Internet Control Message Protocol ICMP [13]. According to ICMP, every host that receives a ping message is required to return a "pong" message to the source of the ping message. The adversary makes use of this requirement by inserting a ping message whose original source is the address of a victim host and whose ultimate destination is a multicast address. Therefore, the victim host is flooded by pong messages returned from all hosts that belong to the multicast address.

In this (and other [3, 9]) type of attacks that are usually known as denial-of-service attacks, an adversary inserts into the network messages with wrong original source. These messages are accepted by unsuspecting routers and may cause severe damage to the network. To counter these attacks, each router p in the network should route a received m only after it checks that the original source in m is a computer adjacent to p or that m is forwarded to p by an adjacent router q. The first check can be achieved by ingress filtering [5]. To achieve the second check, we designed hop integrity protocols [6]. On one hand, messages whose recorded sources are modified at the network boundary will be detected and discarded by ingress filtering. On the other hand, messages whose recorded sources are modified between adjacent routers in the middle of the network will not be detected by ingress filtering, but will be detected and discarded by hop integrity.

On our way of developing a working prototype of hop integrity, we note that the messages used by routing protocols, along with other types of messages, are also secured by hop integrity against message insertion, modification, and replay. The protection of routing messages is important because the routing messages carry routing information that is vital to the correctness of routing protocols [8]. If these routing messages are messed up by an adversary, the operation of routing

protocols will be disrupted and normal data messages will not be routed to their ultimate destination. There have been a number of works on how to extend specific routing protocols to make them more secure [4, 12, 15]. However, if hop integrity is deployed in the network, then without any other security mechanism added, any routing protocol that is used in the network gets secured, along with other security features provided by hop integrity.

In this paper, we first discuss briefly how to achieve hop integrity in IP networks. Then, we show how hop integrity can be used to protect routing protocol messages against message insertion, modification, and replay. We take three famous protocols, namely RIP, OSPF, and RSVP, for the purpose of illustration.

2. Hop Integrity

A *network* consists of computers connected to subnetworks. (Examples of subnetworks are local area networks, telephone lines, and satellite links.) Two computers in a network are called *adjacent* iff both computers are connected to the same subnetwork. Two adjacent computers in a network can *exchange* messages over any common subnetwork to which they are both connected.

The computers in a network are classified into *hosts* and *routers*. A message m is transmitted from a computer s to a faraway computer d in the same network as follows. First, message m is transmitted in one hop from computer s to a router r.1 adjacent to s. Second, message m is transmitted in one hop from router r.1 to router r.2 adjacent to r.1, and so on. Finally, message m is transmitted in one hop from a router r.n that is adjacent to computer d to computer d.

A network is said to provide *hop integrity* iff the following two conditions hold for every pair of adjacent routers p and q in the network.

i. *Detection of Message Modification*:
Whenever router p receives a message m over the subnetwork connecting routers p and q, p can determine correctly whether message m was modified by an adversary after it was sent by q and before it was received by p.

ii. *Detection of Message Replay*:
Whenever router p receives a message m over the subnetwork connecting routers p and q, and determines that message m was not modified, then p can determine correctly whether message m is another copy of a message that is received earlier by p.

To achieve hop integrity, the IP header of each IP message is extended to include two new fields: a sequence number (of length four bytes) and a message digest (of length sixteen bytes). Thus, each IP message becomes of the following form:

(hd, sq, md, tx)

where
hd is the original IP header of the message
sq is the added sequence number of the message
md is the added message digest
tx is the original text of the message

Before a router p forwards an IP message (hd, sq, md, tx) to an adjacent router q, p computes sq as the sequence of the previous IP message from p to q plus one. Also, p computes md as follows.

md := MD5(S|hd|sq|tx|S)

where
MD5 is the standard message digest function [14]
S is a secret shared between routers p and q
| is the concatenation operators on bit strings.

When router q receives an IP message (hd, sq, md, tx), it checks whether sq is less than the next sequence number expected by q. If so, q discards the message. Otherwise, q compares the two values md and MD5(S|hd|sq|tx|S). If these two values are different, then q discards the message. Otherwise, q accepts the message and proceeds to forward it over the next hop.

To realize hop integrity, two "thin" protocol layers need to be added to the protocol stack in each router. The first protocol is a multithread application protocol that allows the router to periodically change the secrets that this router shares with adjacent routers. (Because the secret shared between two adjacent routers is used to compute the message digests of a large number of IP messages between the two routers, this secret needs to be changed frequently, say every four hours.)

The second protocol is within the Internet Protocol IP itself. It computes the sequence number and message digest of every sent IP message and verifies the sequence number and message digest of every received IP message. Formal specification and verification of these two protocols are presented in [6].

In the next three sections, we use three widely used protocols in the Internet, namely RIP, OSPF, and RSVP, to show how hop integrity can secure routing protocols.

3. Security of RIP

RIP, which is a shorthand for Routing Information Protocol, is a widely used routing protocol for IP-based networks [7]. RIP allows a router to exchange routing information with its adjacent routers. It is a distance-vector protocol, which means that the routing information a router receives from an adjacent router is a vector of distances (measured in the number of hops) from the adjacent router to all possible destinations in the network (usually the number of hops). Each router then

independently uses the routing information it receives from its adjacent routers to compute its best routes to all possible destinations in the network. (At the beginning of the execution of RIP, the routes computed by one router may not conform to those computed by another router, because a router does not have much routing information about the network initially. However, with the periodical update, routing information of each router will spread over the network and eventually the routes computed by different routers will converge to the same.)

There are two types of messages used in RIP, namely request and response messages. A router can send a request message to its adjacent routers to ask these routers to send back their current routing tables. A router that receives a request message returns a response message that contains its own routing table. Moreover, a router sends a response message to all its adjacent routers every 30 seconds.

There is a security need for protecting the response messages that contain routing information in RIP. In the absence of any protection for response messages, an adversary sitting between two routers in the network can disrupt the network in several ways. First, the adversary can either insert a fake response message with incorrect routing information that it fabricates. Second, the adversary can modify a correct response message and make its routing information. Third, the adversary can also replay a previous response message whose routing information is no longer correct. When a router that receives a response message with incorrect routing information (from the adversary), it will unsuspiciously accept the message and use its incorrect routing information to update its own routing table. Even worse, the router will send its now incorrect routing table to the adjacent routers. Consequently, the router may compute bad routes for destinations because of the false routing information it receives, and routing loops may be formed because of the spread of the incorrect routing information.

The original RIP version 1 does not have any mechanism for authenticating the response messages. In RIP version 2, a simple authentication mechanism is added to every response message in the protocol [10]: a 16-byte clear text password is inserted in every response message. This authentication mechanism is easy to implement, but it cannot provide enough protection against the insertion, modification, or replay of response messages. For example, the adversary can copy the password and use it in the fake response messages it inserts, or copy a response message and replay it later.

When hop integrity is implemented in a network, the RIP response messages communicated between adjacent routers are protected against message insertion, modification, and replay. Therefore, RIP messages are secured by hop integrity and no other mechanism is needed to make RIP more secure.

4. Security of OSPF

OSPF, which is a shorthand for Open Shortest Path First, is another widely used routing protocol in the Internet [11]. Unlike RIP, OSPF is a link-state protocol, which means that each router gathers information on the state of its links with adjacent routers and advertises this link state information to all other routers in the network. The process by which a router advertises its link state information to all other routers in the network is called flooding. By periodical flooding, each OSPF router in the same network shares a synchronized database of link state records. This database represents the current topology of the network, and is used by a router to compute its best routes to all possible destinations in the network.

OSPF protocol is composed of three subprotocols: Hello, Exchange, and Flooding protocols. The Hello protocol uses hello messages to check whether an adjacent router and the link connected to that router are up or not. A link between two routers is considered up if messages can go in both directions. After establishing their two-way connectivity, two routers can use the Exchange protocol to achieve the initial synchronization of their link state database by exchanging database description messages. A link might change its state as time goes by. Therefore, the router that is responsible for a link that changes its state needs to advertise the new state of the link to all other routers in the network. This is done by using Flooding protocol to send a link state update message to all other routers, and other routers who receive this update message should send back an acknowledgment message so as to keep every router's link state database synchronized.

The possible security threats faced by OSPF can be listed as follows. First, an adversary may insert a fake message that incorrectly advertises some link as the best route to other networks, so as to congest that link. Second, an adversary may modify a message that contains the state information of an important link, so that an area in the network becomes unreachable. Third, an adversary may pretend to be some router and may insert a fake update message that requests all other routers to purge all link state records of the impersonated router. By repeating this trick, the adversary can slash the link state database in every router.

When hop integrity is implemented in a network, the hello messages, database description messages, update messages, and acknowledgment messages that are communicated between adjacent routers are protected against message insertion, modification, and replay. Therefore, OSPF messages are secured by hop integrity and no other mechanism is needed to make OSPF more secure.

5. Security of RSVP

RSVP, which is a shorthand of ReSerVation Protocol, is a resource reservation protocol designed for providing integrated services in the Internet [2]. RSVP allows a host that wants to receive application data flows from a sending host to request from the network a specific degree of service in advance (although there is no guarantee that the requested service will be provided by the network). RSVP also allows a router to exchange service requests with other routers to establish and maintain state of the service it provides. Once the requested service is established, the host that requested the service is guaranteed that each router along the data path (between this host and the sending host) has reserved needed resources for the service, and that the provided service will last till the end of the transmission of the data flow.

There are two main types of messages used in RSVP, namely Resv and Path messages. Each sending host periodically sends a Path message to all receiving hosts of a data flow that this sending host generates. The Path message is designed to mark the path that is traveled by data messages. Each router along the data path maintains a state that remembers the previous router corresponding to this particular data flow. With the path information marked by Path messages, each receiving host is able to send Resv messages, which contain the reservation requests, toward the sending host. When receiving a Resv message, each router on the path determines how many resources it can grant to this reservation request, and relays the Resv message toward the sending host.

The security issues concerned with RSVP are the integrity and authentication of service request messages. If an adversary spoofs the source address of a service request message, and the service request message is accepted by unsuspecting routers along the data path, the adversary can steal the established service. If an adversary modifies the parameter of service specified in a service request message, or replays several service request messages and inserts them into the network, the normal service provided by the network may be severely reduced or totally denied.

An extension to RSVP provides a mechanism to protect RSVP messages against message modification, message spoofing, and message replay [1]. The proposed scheme uses a secret shared between a pair of adjacent RSVP routers to compute a keyed cryptographic digest of a RSVP message, and includes the digest as part of the RSVP message. However, a working key management protocol is missing in that proposal and manual key management may be necessary at its current stage. By contrast, if hop integrity along with ingress filtering is deployed in the network, RSVP messages, along with all other types of messages, will be protected against message modification, message spoofing, and message replay. Hop integrity is also easier to manage because it updates shared secrets in a distributed way (by a pair of adjacent router themselves).

6. Concluding Remarks

In this paper, we introduced the concept of hop integrity in computer networks, and outlined how to achieve it in IP networks. Then, we discussed how hop integrity can be used to secure routing protocols, and illustrated it with three widely used protocols, namely RIP, OSPF, and RSVP. In fact, we argue that every protocol that involves communications exchanged between adjacent routers can be secured by the deployment of hop integrity in the network.

In the next step of this research, we hope to use experimental results to justify our argument that hop integrity can secure routing protocols, when we finish the development of a prototype of hop integrity.

Acknowledgment

This work is supported by an IBM Faculty Partnership Award for the third author for the academic years 2000-2001 and 2001-2002.

References

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

[2] Braden, R., L. Zhang, S. Berson, S. Herzog, S. Jamin, "*Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification*", RFC 2205, September 1997.

[3] "*TCP SYN Flooding and IP Spoofing Attacks*", CERT Advisory CA-96.21, available at http://www.cert.org/.

[4] Cheung, S., "An Efficient Message Authentication Scheme for Link State Routing", *Proceedings of the 13th Annual Computer Security Applications Conference*, San Diego, California, December 1997, pp. 90-98.

[5] Ferguson, P., D. Senie, "*Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*", RFC 2827, May 2000.

[6] Gouda, M. G., E. N. Elnozahy, C.-T. Huang, T. M. McGuire, "Hop Integrity in Computer Networks", *Proceedings of the IEEE International Conference on Network Protocols*, Osaka, Japan, November 2000.

[7] Hedrick, C., "*Routing Information Protocol*", RFC 1058, June 1988.

[8] Huitema, C., *Routing in the Internet*, Second Edition, Prentice-Hall, Upper Saddle River, NJ, 2000.

[9] Joncheray, L., "A Simple Active Attack Against TCP", *Proceedings of the 5th USENIX UNIX Security Symposium*, 1995, pp. 7-19.

[10] Malkin, G., "*RIP Version 2: Carrying Additional Information*", RFC 1723, November 1994.

[11] Moy, J., "*OSPF Version 2*", RFC 1583, March 1994.

[12] Murphy, S., and M. Badger, "Digital Signature Protection of the OSPF Routing Protocol", *Proceedings of the 1996 Internet Society Symposium on Network and Distributed Systems Security*, San Diego, California, February 1996.

[13] Postel, J., "*Internet Control Message Protocol*", RFC 792, September 1981.

[14] Rivest, R. L., "*The MD5 Message-Digest Algorithm*", RFC 1321, 1992.

[15] Smith, B., S. Murthy, and J. J. Garcia-Luna-Aceves, "Securing Distance Vector Routing Protocols", *Proceedings of the 1997 Internet Society Symposium on Network and Distributed Systems Security*, San Diego, California, February 1997.