Network Working Group
Request for Comments: 1853
Category: Informational

W. Simpson
Daydreamer

October 1995

IP in IP Tunneling

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard. Distribution of this memo is unlimited.

IESG Note:

Note that this memo is an individual effort of the author. This document reflects a current informal practice in the internet. There is an effort underway within the IETF Mobile-IP Working Group to provide an appropriate proposed standard to address this issue.

Abstract

This document discusses implementation techniques for using IP Protocol/Payload number 4 Encapsulation for tunneling with IP Security and other protocols.

Table of Contents

1. Introduction

The IP in IP encapsulation Protocol/Payload number 4 [RFC-1700] has long been used to bridge portions of the Internet which have disjoint capabilities or policies. This document describes implementation techniques used for many years by the Amateur Packet Radio network for joining a large mobile network, and also by early implementations of IP Security protocols.

Use of IP in IP encapsulation differs from later tunneling techniques (for example, protocol numbers 98 [RFC-1241], 94 [IDM91a], 53 [swIPe], and 47 [RFC-1701]) in that it does not insert its own special glue header between IP headers. Instead, the original unadorned IP Header is retained, and simply wrapped in another standard IP header.

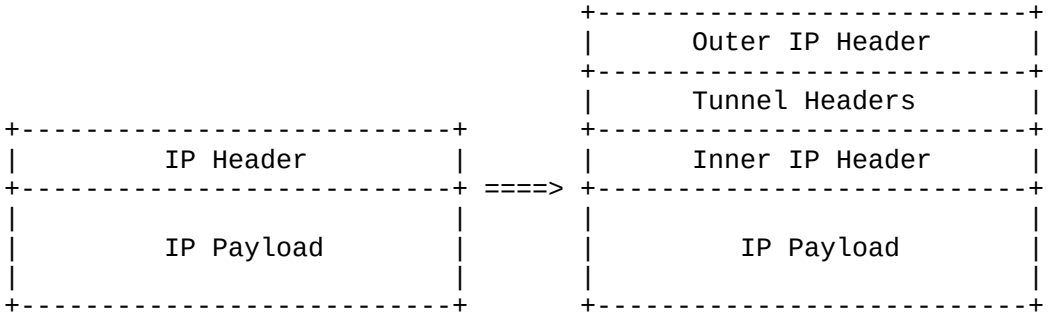
This information applies principally to encapsulation of IP version 4. Other IP versions will be described in separate documents.

2. Encapsulation

The encapsulation technique is fairly simple. An outer IP header is added before the original IP header. Between them are any other headers for the path, such as security headers specific to the tunnel configuration.

The outer IP header Source and Destination identify the "endpoints" of the tunnel. The inner IP header Source and Destination identify the original sender and recipient of the datagram.

Each header chains to the next using IP Protocol values [RFC-1700].



The format of IP headers is described in [RFC-791].

Type Of Service copied from the inner IP header. Optionally, another TOS may be used between cooperating peers.

This is in keeping with the transparency principle that if the user was expecting a given level of service, then the tunnel should provide the same service. However, some tunnels may be constructed specifically to provide a different level of service as a matter of policy.

Identification A new number is generated for each outer IP header.

The encapsulated datagram may have already been fragmented, and another level of fragmentation may occur due to the tunnel encapsulation. These tunnel fragments will be reassembled by the decapsulator, rather than the final destination.

Reserved

ignored (set to zero).

This unofficial flag has seen experimental use, and while it remains in the inner IP header, does not affect the tunnel.

Don't Fragment — copied from the inner IP header. This allows the originator to control the level of performance tradeoffs. See "Tunnel MTU Discovery".

More Fragments — set as required when fragmenting.

The flag is not copied for the same reason that a separate Identification is used.

Time To Live — the default value specified in the most recent "Assigned Numbers" [RFC-1700]. This ensures that long unanticipated tunnels do not interrupt the flow of datagrams between endpoints.

The inner TTL is decremented once before encapsulation, and is not affected by decapsulation.

Protocol — the next header; 4 for the inner IP header, when no intervening tunnel headers are in use.

Source — an IP address associated with the interface used to send the datagram.

Destination — an IP address of the tunnel decapsulator.

Options — not copied from the inner IP header. However, new options particular to the path MAY be added.

Timestamp, Loose Source Route, Strict Source Route, and Record Route are deliberately hidden within the tunnel. Often, tunnels are constructed to overcome the inadequacies of these options.

Any supported flavors of security options of the inner IP header MAY affect the choice of security options for the tunnel. It is not expected that there be a one-to-one mapping of such options to the options or security headers selected for the tunnel.

3. Tunnel Management

It is possible that one of the routers along the tunnel interior might encounter an error while processing the datagram, causing it to return an ICMP [RFC-792] error message to the encapsulator at the IP Source of the tunnel. Unfortunately, ICMP only requires IP routers to return 8 bytes (64 bits) of the datagram beyond the IP header. This is not enough to include the entire encapsulated header. Thus, it is not generally possible for an encapsulating router to immediately reflect an ICMP message from the interior of a tunnel back to the originating host.

However, by carefully maintaining "soft state" about its tunnels, the encapsulator can return accurate ICMP messages in most cases. The router SHOULD maintain at least the following soft state information about each tunnel:

- Reachability of the end of the tunnel.
- Congestion of the tunnel.
- MTU of the tunnel.

The router uses the ICMP messages it receives from the interior of a tunnel to update the soft state information for that tunnel. When subsequent datagrams arrive that would transit the tunnel, the router checks the soft state for the tunnel. If the datagram would violate the state of the tunnel (such as the MTU is greater than the tunnel MTU when Don't Fragment is set), the router sends an appropriate ICMP error message back to the originator, but also forwards the datagram into the tunnel. Forwarding the datagram despite returning the error message ensures that changes in tunnel state will be learned.

Using this technique, the ICMP error messages from encapsulating routers will not always match one-to-one with errors encountered within the tunnel, but they will accurately reflect the state of the network.

3.1. Tunnel MTU Discovery

When the Don't Fragment bit is set by the originator and copied into the outer IP header, the proper MTU of the tunnel will be learned from ICMP (Type 3 Code 4) "Datagram Too Big" errors reported to the encapsulator. To support originating hosts which use this capability, all implementations MUST support Path MTU Discovery [RFC-1191, RFC-1435] within their tunnels.

As a benefit of Tunnel MTU Discovery, any fragmentation which occurs because of the size of the encapsulation header is done only once after encapsulation. This prevents more than one fragmentation of a single datagram, which improves processing efficiency of the path routers and tunnel decapsulator.

3.2. Congestion

Tunnel soft state will collect indications of congestion, such as an ICMP (Type 4) Source Quench in datagrams from the decapsulator (tunnel peer). When forwarding another datagram into the tunnel, it is appropriate to send Source Quench messages to the originator.

3.3. Routing Failures

Because the TTL is reset each time that a datagram is encapsulated, routing loops within a tunnel are particularly dangerous when they arrive again at the encapsulator. If the IP Source matches any of its interfaces, an implementation MUST NOT further encapsulate. Instead, the datagram is forwarded normally.

ICMP (Type 11) Time Exceeded messages report routing loops within the tunnel itself. ICMP (Type 3) Destination Unreachable messages report delivery failures to the decapsulator. This soft state MUST be reported to the originator as (Type 3 Code 0) Network Unreachable.

3.4. Other ICMP Messages

Most ICMP error messages are not relevant to the use of the tunnel. In particular, parameter problems are likely to be a result of misconfiguration of the encapsulator, and MUST NOT be reported to the originator.

Security Considerations

Security issues are briefly discussed in this memo. The use of tunneling may obviate some older IP security options (labelling), but will better support newer IP Security headers.

References

[IDM91a] Ioannidis, J., Duchamp, D., Maguire, G., "IP-based protocols for mobile internetworking", Proceedings of SIGCOMM '91, ACM, September 1991.

[RFC-791] Postel, J., "Internet Protocol", STD 5, RFC 791, USC/Information Sciences Institute, September 1981.

[RFC-792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, USC/Information Sciences Institute, September 1981.

[RFC-1191] Mogul, J., and S. Deering, "Path MTU Discovery", RFC 1191, DECWRL, Stanford University, November 1990.

[RFC-1241] Mills, D., and R. Woodburn, "A Scheme for an Internet Encapsulation Protocol: Version 1", UDEL, July 1991.

[RFC-1435] Knowles, S., "IESG Advice from Experience with Path MTU Discovery", RFC 1435, FTP Software, March 1993.

[RFC-1700] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, USC/Information Sciences Institute, October 1994.

[RFC-1701] Hanks, S., Li, T., Farinacci, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 1701, October 1994.

[swIPe] Ioannidis, J., and Blaze, M., "The Architecture and Implementation of Network-Layer Security Under Unix", Fourth Usenix Security Symposium Proceedings, October 1993.

Acknowledgements

These implementation details of IP Tunneling are derived in large part from independent work in 1990 by Phil Karn and the TCP-Group hams using KA9Q NOS.

Special thanks to John Ioannidis (then of Columbia University) for inspiration and experimentation which began this most recent round of IP Mobility and IP Security development. Some of this text was derived from [IDM91a] and [swIPe].

The chaining of headers was also described in "Simple Internet Protocol", by Steve Deering (Xerox PARC).

The overall organization and some of this text was derived from [RFC-1241], by David Mills (U Delaware) and Robert Woodburn (SAIC).

Some of the text on tunnel soft state was derived from "IP Address Encapsulation (IPAE)", by Robert E. Gilligan, Erik Nordmark, and Bob Hinden (all of Sun Microsystems).

Author's Address

Questions about this memo can also be directed to:

William Allen Simpson
Daydreamer
Computer Systems Consulting Services
1384 Fontaine
Madison Heights, Michigan 48071

Bill.Simpson@um.cc.umich.edu
bsimpson@MorningStar.com