Steve Bellovin
Internet Draft AT&T Labs Research
Document: draft-ietf-itrace-04.txt Marcus Leech
Tom Taylor
Nortel Networks
Expires: August 2003 February 2003

ICMP Traceback Messages

Status of this Memo

Abstract

It is often useful to learn the path that packets take through the Internet, especially when dealing with certain denial-of-service attacks. We propose a new ICMP message, emitted randomly by routers along the path and sent randomly to the destination (to provide useful information to the attacked party) or to the origin (to provide information to decipher reflector attacks).

Table of Contents

1. Introduction

It is often useful to learn the path that packets take through the Internet. This is especially important for dealing with certain denial-of-service attacks, where the source IP is forged. There are other uses as well, including path characterization and detection of asymmetric routes. There are existing tools, such as traceroute, but these generally provide the forward path, not the reverse.

We propose an ICMP [RFC792] Traceback message to help solve this problem. When forwarding packets, routers can, with a low probability, generate a Traceback message that is sent along to the destination or back to the source. With enough Traceback messages from enough routers along the path, the traffic source and path of forged packets can be determined.

1.1 Requirements Keywords

The keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", and "MAY" that appear in this document are to be interpreted as described in [RFC2119].

1.2 Definitions

Element: a component of the proposed message which is explicitly identified by a type code, and is encoded using a Type-Length-Value (TLV) format. Some elements will contain other elements as described below.

Field: a component of the proposed message which is identified through its relative position within the header or within a particular element.

Generator: the router which itself generates the ICMP Traceback message or on behalf of which this message is generated by some other entity.

Link: a logical connection between the Generator and another entity, along which the traced packet has passed.

Peer: the entity at the other end of the link, which either sent the traced packet to the Generator or received it from the Generator.

Traced Packet: the packet which is the subject of an ICMP TRACEBACK message.

2. Message Definition

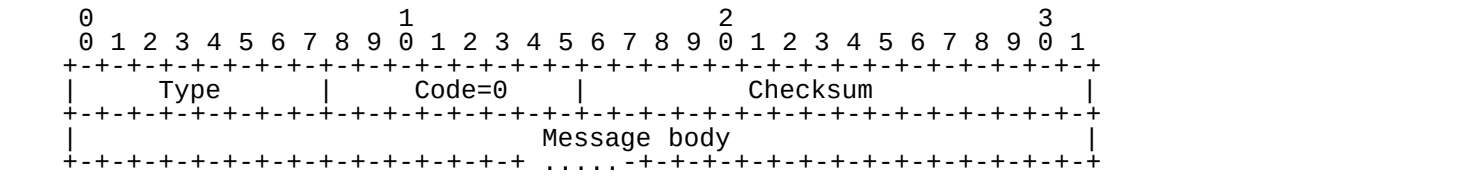
2.1 Conventions For Presentation

As indicated below, aside from the initial octet, the elements of the ICMP TRACEBACK message are concatenated without any padding to create word boundary alignment. The fields within each element are similarly concatenated without intervening padding. The diagrams presenting the individual elements therefore show the length and relative order of the fields making them up, but do NOT indicate alignment on any specific boundary. Each field beyond the initial type code and length is shown beginning on a separate line, although in fact fields are contiguous in the actual message.

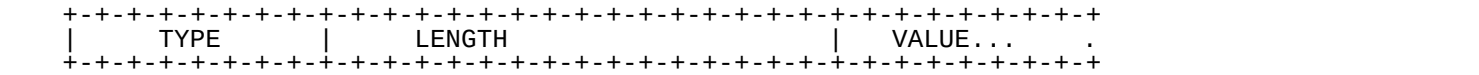
2.2 Overall Message Format

The proposed message is carried in an ICMP packet, with ICMP TYPE of TRACEBACK. (The numeric values for this field will be assigned by IANA. For IPv6, the TRACEBACK should be classified as Informational.) The CODE field MUST always be set to 0 (no code), and MUST be ignored by the receiver.

Traceback Message



The body of any ICMP TRACEBACK message consists of a series of individual elements that are self-identifying, using a TYPE-LENGTH-VALUE scheme as follows:



This structure is recursive, in that for certain element types the VALUE field will contain one or more components which are also in TYPE-LENGTH-VALUE (TLV) format. Top-level elements may appear in any order, and a receiver MUST be capable of processing them in any order. Elements contained within the VALUE field of a parent element may also appear in any order within that field and present a similar requirement to the receiver. Elements are placed consecutively within the message body without intervening padding; hence elements in general are not aligned to word boundaries.

The TYPE field is a single octet, with values in the range 0x01 to 0x7f for top-level elements and 0x81 to 0xff for sub-elements. The top-level type codes assigned in this document are as follows:

Type	Element Name
0x01	Back Link
0x02	Forward Link
0x03	Timestamp
0x04	Traced Packet Contents

0x05	Probability
0x06	RouterId
0x07	HMAC Authentication Data
0x08	Key Disclosure List

The sub-element type codess assigned in this document are as follows:

Type	Element Name	Notes
0x81	Interface Name	1
0x82	IPv4 Address Pair	1,2
0x83	IPv6 Address Pair	1,2
0x84	MAC Address Pair	1,3
0x85	Operator-Defined Link Identifier	1,3
0x86	Key Disclosure	4
0x87	Disclosure Signature	4

Note 1: this item is a sub-element within Back or Forward Link elements.

Note 2: at least one of these elements MUST be present within a Link element.

Note 3: either the MAC Address Pair or the Operator-Defined Link Identifier element but not both MUST be present within a Link element.

Note 4: the Key Disclosure List MUST contain one or more Key Disclosure elements and exactly one Disclosure Signature element.

LENGTH is always set to the length of the VALUE field in octets, and always occupies two octets, even when the length of the VALUE field is less than 256 octets.

2.3 Forward and Backward Link Elements

An ICMP TRACEBACK message MUST contain one Forward Link element or one Back Link element; it MAY contain one instance of each. A Link element specifies a link along which the traced packet travelled to or from the Generator. The purpose of the Forward and Back Link

elements is to permit easy construction of a chain of Traceback messages. They are further designed for examination by network operations personnel, and thus contain human-useful information such as interface names.

The Value field of a link element consists of three components:

* the interface name at the Generator only. (It is assumed that the Generator does not know its neighbors' interface names.) This is encoded in an Interface Name element.

* the source and destination IP addresses of the Generator and its peer. These are encoded in an IPv4 or IPv6 Address Pair element.

* the link-level association string. The association string is an opaque blob which is used to tie together Traceback messages emitted by adjacent routers. Thus all Link elements referring to the same link MUST use the same value for the association string, regardless of which entity generates them.

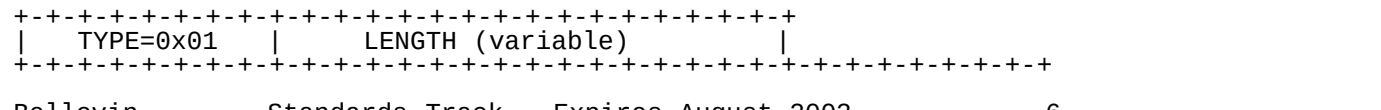
  On LANs, the association string is constructed by concatenating the source and destination MAC addresses of the two interfaces to the link, and is encoded in a MAC Address Pair element. If there are no such addresses (say, for a point-to-point link), a suitable string MUST be provisioned in both routers; this is encoded in an Operator-Defined Link Identifier element.

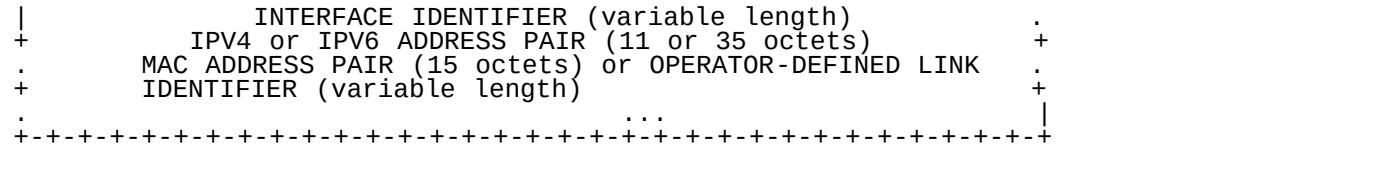
The fields of the Address Pair elements are always arranged in "forward order" from the point of view of the traced packet. That is, the "destination" field is always the address of the entity

closer to the ultimate recipient of the traceback packet. Thus, in Back Link elements, the generator's own address is placed in the destination field of the IP and MAC Address Pair subelements; in Forward Link elements, the generator's address is placed in the source field.

2.3.1 Back Link (TYPE=0x01)

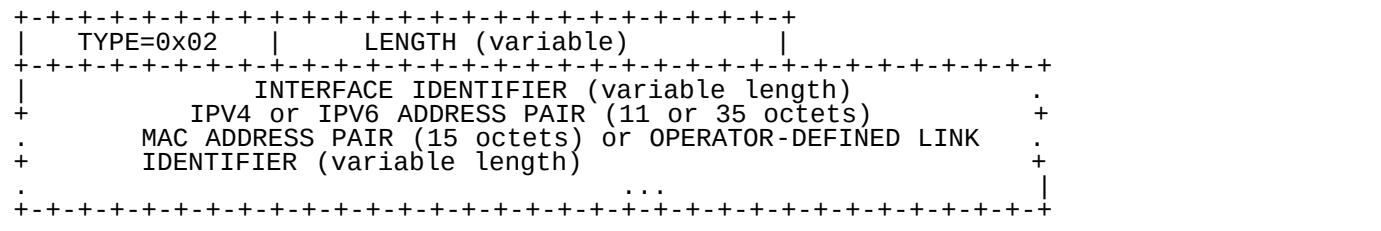
The Back Link element provides identifying information, from the perspective of the Generator, about the link that the traced packet arrived from. The VALUE field of this element consists of three TLV subelements, one each for the Interface Identifier, the IP Address Pair, and the association string. Element lengths shown include the type and length fields. Elements may appear in a different order from that shown.





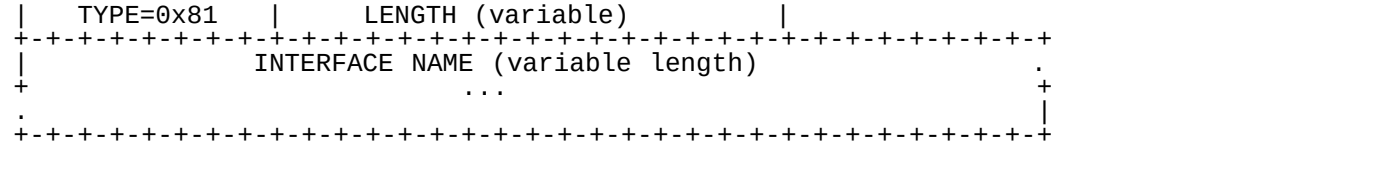
2.3.2 Forward link (TYPE=0x02)

The Forward Link element provides identifying information, from the perspective of the Generator, about the link that the traced packet was forwarded on. Its structure is the same as that of the Back Link element.



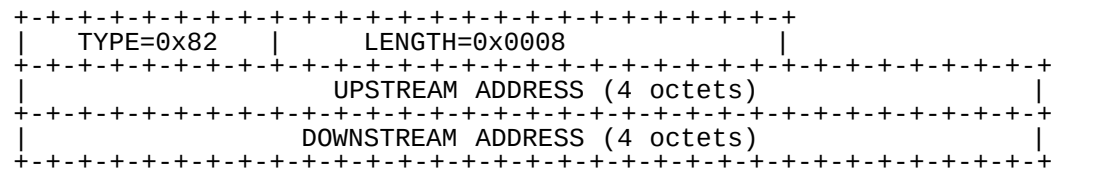
2.3.3 Interface Identifier (TYPE=0x81)

This element contains the name of the interface to the link at the generating router. The length is variable. The VALUE field typically contains a human-readable character string.



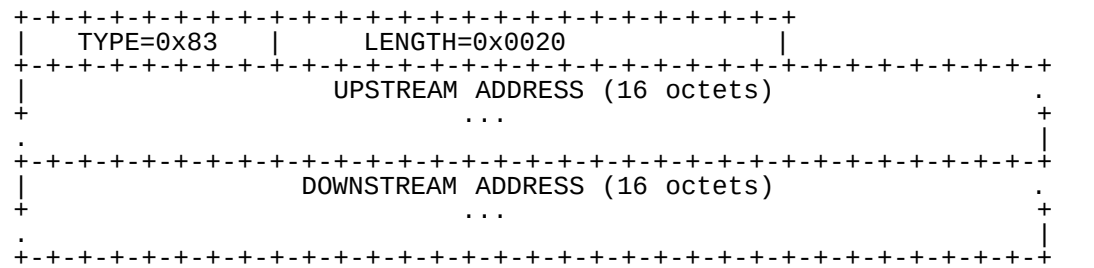
2.3.4 IPV4 address pair (TYPE=0x82)

This element contains two 4-octet IPV4 addresses of the ends of the corresponding link; hence the LENGTH field is always 0x0008. As noted above, the addresses MUST always be presented in the order of their traversal by the traced packet.



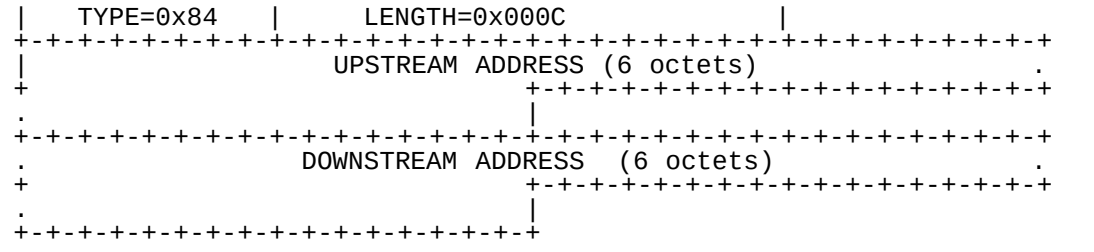
2.3.5 IPV6 address pair (TYPE=0x83)

This element contains two 16-octet IPV6 addresses of the ends of the corresponding link; hence the LENGTH field is always 0x0020. As noted above, the addresses MUST always be presented in the order of their traversal by the traced packet.



2.3.6 MAC address pair (TYPE=0x84)

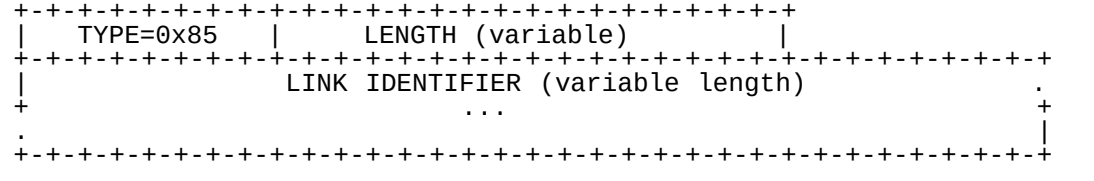
This element contains two 6-octet IEEE MAC addresses of the ends of the corresponding link; hence the LENGTH field is always 0x000C. As noted above, the addresses MUST always be presented in the order of their traversal by the traced packet.



2.3.7 Operator-defined link identifier (TYPE=0x85)

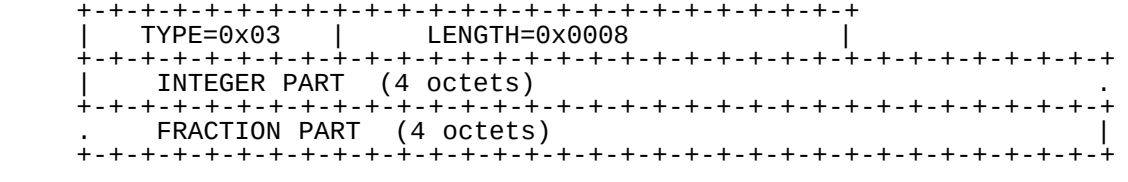
The value of this element is an opaque field of varying length. If the peer also emits ICMP TRACEBACK messages for the same link, it

MUST use the same value. Further definition will emerge in a later document.



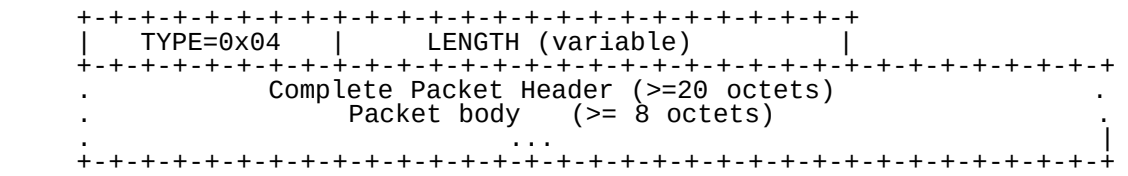
2.4 Timestamp (TYPE=0x03)

This element contains the time, in NTP timestamp format (eight octets) [RFC1305], at which the ICMP Traceback packet was generated. This element MUST be present at the top level within the TRACEBACK message. The timestamp MUST be consistent with (i.e. lie between) the starting and ending time of validity of the applicable hash key as reported in Key Disclosures in subsequent ICMP Traceback packets.



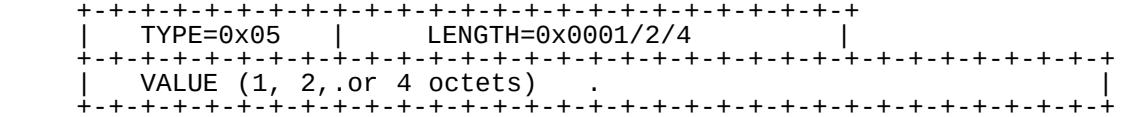
2.5 Traced packet (TYPE=0x04)

This element provides the contents of the traced packet, as much as can reasonably fit, subject to link and router resource constraints. This element MUST be present at the top level within the TRACEBACK message, and MUST contain at least the IP header and as much of the remainder of the traced packet as will fit without making the ICMP Traceback packet exceed the minimum path MTU (576 octets for IPv4, as specified in [RFC2460] for IPv6). [ISSUE: should this wording stand or should we go with the more restrictive value?]



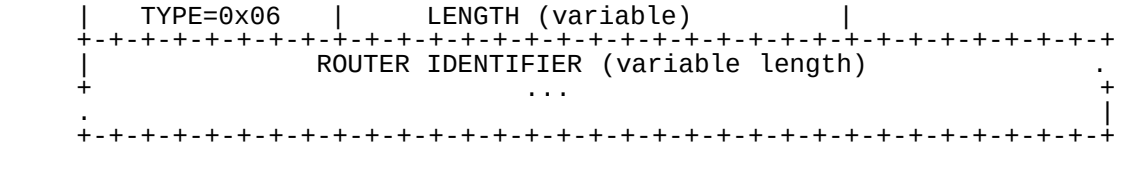
2.6 Probability (TYPE=0x05)

This element contains the inverse of the probability used to select the traced packet. It appears as an unsigned integer, of one, two, or four octets. This element SHOULD be present at the top level within the TRACEBACK message.



2.7 RouterId (TYPE=0x06)

This element contains opaque identifying information, useful to the organization that operates the router emitting the ITRACE message. This element MUST be present at the top level within the TRACEBACK message.



2.8 Authentication data

An attacker may try to generate fake Traceback messages, primarily to conceal the source of the real attack traffic, but also to act as another form of attack. We thus need authentication techniques that are robust but quite cheap to verify.

The ideal form of authentication would be a digital signature. It is unlikely, though, that routers will be able to afford such signatures on all Traceback packets. Thus, although we leave hooks for such a variant, we do not further define it at this time.

What is provided instead is a hash code (the HMAC Authentication Data element), supported by signed disclosure of the keys most recently used (the Key Disclosure and Public Key Information elements). The current key MUST NOT be included in this disclosure.

2.8.1 HMAC Authentication data (TYPE=0x07)

This element MUST be present. It contains four subfields:

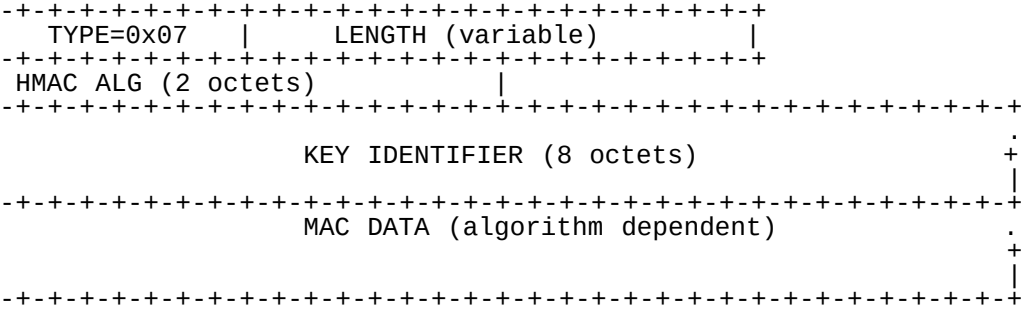
* algorithm, two octets. This identifies the hash algorithm used. It must be one of the hash algorithm codepoint values defined in http://www.iana.org/assignments/ipsec-registry.

* keyid: eight octet key identifier

* MAC data: variable

  The MAC data field covers the entire IP datagram, including header information. Where header information is mutable during transport, such information is set to zero (0x00) for purposes of calculating the HMAC. Mutable fields for IPv4 and IPv6 are identified in [RFC2402] section 3.3.3.1.

  This field is as long as is appropriate for the given MAC algorithm.

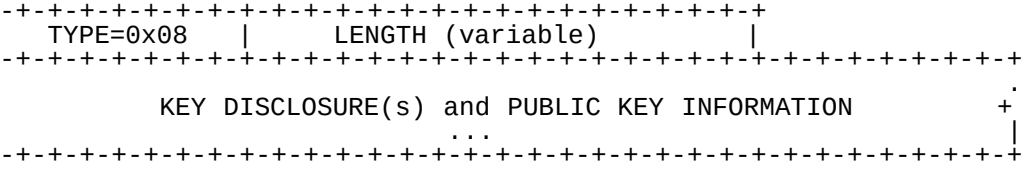


2.8.2 Key Disclosure List (TYPE=0x08)

A packet SHOULD contain a list of the keys most recently used to create HMAC authentication values for ICMP Traceback packets. The key currently in use MUST NOT be included in this disclosure. The disclosure need not appear if there are no keys to be disclosed according to the criteria [TBD].

[EDITOR's NOTE: it has been suggested that we need to state an upper limit for the number of keys to be disclosed. The Editor suggests that the basic principle governing the number of keys that should be disclosed is that there be a reasonable probability (e.g. 80%) that a host receiving an ITrace packet will also receive a key disclosure for that packet. This makes the number of keys a function of the rate of generation of ITrace packets and the rate at which keys are changed. Further analysis may give more concrete results.]

The key disclosure is provided in the Key Disclosure List element. This element MUST contain at least one Key Disclosure sub-element, and MUST also contain a Public Key Information sub-element pointing to the keys used to sign the Key Disclosures. In accordance with the general rule for construction of the TRACEBACK message, the sub-elements may be presented in any order and the receiver MUST be able to process them regardless of the order in which they are presented.



2.8.3 Key Disclosure (TYPE=0x86)

The primary content of the Key Disclosure element consists of a key used to authenticate previous TRACEBACK messages and the starting and ending times between which that key was used. The algorithm is assumed to be the same as that used to authenticate the current message (and shown in the HMAC Authentication Data element). The element MUST also contain a digital signature covering the Key Disclosure element.

The structure of the Key Disclosure element is as follows:

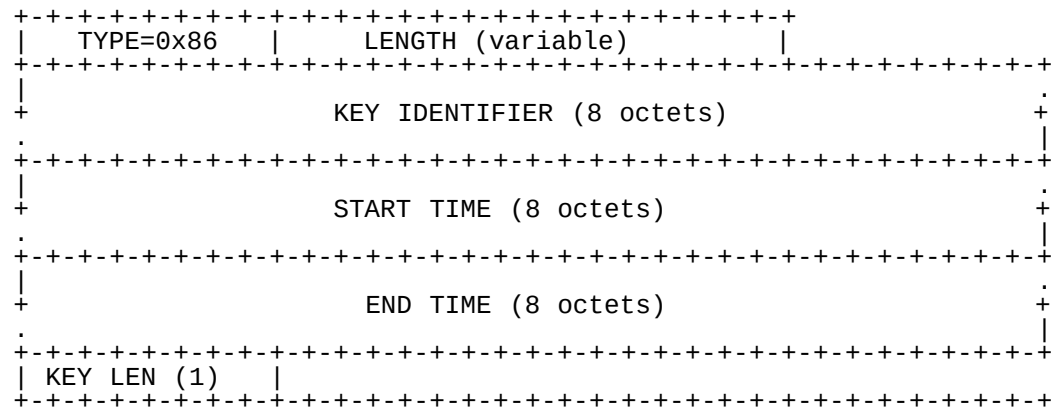
* keyid for the key being disclosed: eight octets

* validity: two NTP timestamps giving validity period (start, end)

* key length: one octet

* key material: variable [key length] octets

  Keying material for the chosen HMAC function MUST conform to the requirements for keys outlined in [RFC2104].



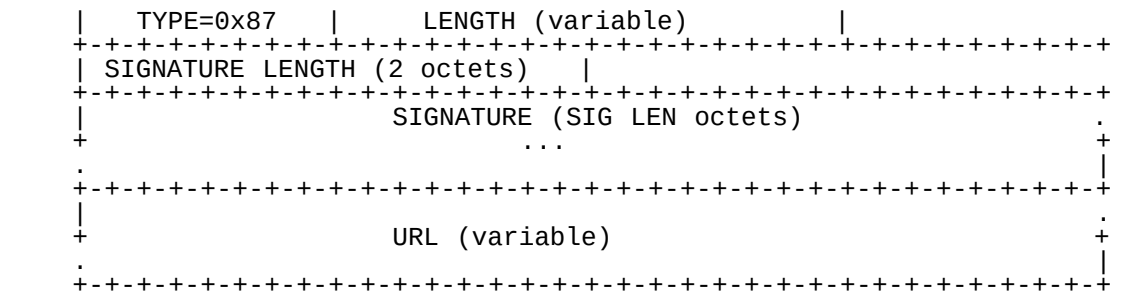
2.8.4 Disclosure Signature (TYPE=0x87)

The Disclosure Signature sub-element contains a digital signature for the the Key Disclosure List. The signature covers the entire Key Disclosure List element, including the Disclosure Signature element, but excluding the signature length and actual signature within that element. This element also contains a URL, suitable for retrieving an X.509 certificate that contains the public key used to sign key-disclosure elements. See [HTTP-CERTSTOR] for a possible implementation.

The URL is present because digital signatures are useless without some way of authenticating the public key of the signer. The ideal form of authentication would be a certificate-based scheme rooted in the address registries. That is, the registries are the authoritative source of information on who owns which addresses; they are thus the only party that can easily issue such certificates.

Until such a PKI is in existence, we suggest that each ISP publish its own root public key. Current registry-based databases can be used to verify the owner of an address block; this information can in turn be used to locate the appropriate root key.

The Disclosure Signature element has the following structure:



3. Procedures

3.1 Generation Of Traceback Packets

A router implementing this scheme SHOULD generate and emit an ICMP Traceback packet with probability of about 1/20,000, although local site policy MAY adjust this to better suit local link utilization

metrics. It MUST then randomly select with equal probability to send this packet to the origin or the destination of the sampled packet.

Some requirements are imposed on the IP header of the Traceback packet. In particular, the source address SHOULD be that associated with the interface on which the packet arrived. If that interface has multiple addresses, the address chosen SHOULD, if possible, be the one by which this router is known to the previous hop. If the interface has no IP address, the "primary" IP address associated with the router MAY be used. ("Primary" is discussed below.)

The initial TTL field MUST be set to 255. If the Traceback packet follows the same path as the data packets, this provides an unambiguous indication of the distance from this router to the destination. More importantly, by comparing the distances with the link elements, a chain can be constructed and partially verified even without examining the authentication fields.

The TOS field SHOULD be copied from the TOS field of the traced packet.

3.1.1 Implementation Requirements -- Message Generation

The probability of Traceback generation SHOULD be adjustable by the operator of the router. A default value of about 1/20000 is suggested. If the average maximum diameter of the Internet is 20 hops, that translates to a net increase in traffic at the origina and destination of about .1%; this should not be an undue burden on the recipient. The probability SHOULD NOT be greater than 1/1000.

Packet selection SHOULD be based on a pseudo-random number, rather than a simple counter. This will help block attempts to time attack bursts. There does not appear to be any requirement for cryptographically strong pseudo-random numbers.

A suggested scheme involves examination of the low-order bits of a linear congruential pseudo-random number generator (LCPRNG). If they are all set to 1, the packet should be emitted. This permits easy selection of probabilities 1/8191, 1/16383, etc. N.B. While the low-order bits of LCPRNGs are not very random, that does not matter here. As long as the period of the generator is maximal, all values, including all 1s in the low-order bits, will occur with the proper probability.

Although this document describes a router-based implementation of Traceback messages, most of the functionality can be implemented via outboard devices. For example, suitable laptop computers can be used to monitor LANs, and emit the traceback messages as appropriate, on behalf of all of the routers on that LAN.

3.1.2 Implementation Requirements -- Message Reception

Hosts SHOULD be designed so that the operator can enable and disable the collection and storage of ICMP TRACEBACK messages as required. Hosts SHOULD also be designed so that the operator can limit the rate at which the host accepts ICMP TRACEBACK messages. Messages exceeding this rate would be silently dropped. If such functionality is implemented, the host SHOULD provide a counter displaying how many messages have been dropped.

3.2 Configuration

The association string used in the Forward and Back Link elements can be built up from the MAC addresses of the link endpoints. If there are no such addresses (say, for a point-to-point link), a suitable string MUST be provisioned in both routers, to be used as the Operator-Defined Link Identifier.

3.3 Processing Of Received Messages

To circumvent attacks in the course of which false ICMP TRACEBACK messages are emitted, these messages SHOULD be validated before use. Malformed messages SHOULD be silently discarded. Some further validation can be done before the HMAC keying information is disclosed. In particular, when messages appearing to relate to adjacent segments of a chain have been identified, recipients SHOULD use the TTL field differences in conjunction with the link elements to verify the chain.

Because HMAC key disclosure is done only after the end of the period of validity for the key, authentication of a given set of ICMP TRACEBACK messages requires that further messages be collected and examined beyond the period of interest, until the required key appears. The processing entity SHOULD then verify the signature on the key before applying the key itself to validation of the message.

4. Related Work

Another scheme proposed for packet Traceback is by Savage et al [SWKA00]. It relies on a very clever encoding of the path in the IP header's ID field. That is, in-flight packets may have their ID field changed to provide information about the path. The recipient can decode this information.

There are a number of advantages of this compared to ICMP Traceback. No extra traffic is generated. More importantly, the trace information is bound to the packets, and hence doesn't follow a different path and isn't differentially blocked by firewalls or policy routing mechanisms. However, there are disadvantages as well. For one thing, the ID field cannot be changed if

fragmentation is necessary (though they propose some schemes to ameliorate this). Moreover, AH [RFC2402] provides cryptographic protection for the ID field; if it is modified, the packet will be discarded by the receiving system. And IPv6 has no ID field at all. A number of other packet-marking schemes have been proposed.

A different approach is hash-based traceback, by Snoeren et al [SPSSJTK01]. In this scheme, routers along the path are queried about whether or not they have seen a certain packet; a very compact representation is used to store recent history. The problem is that queries must be done very soon after the attack, unless the routers have some way of offloading historical data to bulk storage.

[SDS00] descibes a scheme for coupling IDS systems. A sensor that detects an attack tells its neighbors; they in turn look for the same signature, and notify their neighbors. The current prototype only works within an administrative domain; work is currently under way to produce an inter-domain version.

5. Security Considerations

It is quite clear that this scheme cannot cope with all conceivable denial of service attacks. It is limited to those where a significant amount of traffic is coming from a relatively small number of sources. Furthermore, those sources must themselves be in some sense evil or corrupted. An attack based on inducing innocent and uncorrupted machines to send traffic to the victim would be traceable only to these machines, and not to the real attackers.

A lengthy discussion of the possibility of flooding attacks using fake ITrace packets to fill host buffers and render the tool useless took place after the previous version of this document was issued. (Thread " Problems with implementation - DoS attacks possible" on the ITrace E-mail list, initiated 1/20/2003 by Tomasz Grabowski). A major issue is how quickly hash keys should be rotated so disclosure can take place. Further work is needed to resolve this.

6. IANA Considerations

TBD

7.Acknowledgements

The ICMP Traceback message is the product of an informal research group; members include (in alphabetical order) Steven M. Bellovin, Matt Blaze, Bill Cheswick, Cory Cohen, Jon David, Jim Duncan, Jim Ellis, Paul Ferguson, John Ioannidis, Marcus Leech, Perry Metzger, Robert Stone, Vern Paxson, Ed Vielmetti, Wietse Venema.

8. References

[RFC792] : J. Postel, "Internet Control Message Protocol", RFC 792, Internet Engineering Task Force, September 1981.

[RFC1305]: David L. Mills, "Network Time Protocol (Version 3): Specification, Implementation and Analysis", RFC 1305, Internet Engineering Task Force, March 1992.

[RFC2104]: H. Krawczyk, M. Bellare, R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, Internet Engineering Task Force, February 1997.

[RFC2119]: S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, Internet Engineering Task Force, March 1997.

[RFC2402]: S. Kent and R. Atkinson, "IP Authentication Header", RFC 2402, Internet Engineering Task Force, November 1998.

[RFC2460]: Deering, S. and R. Hinden, "Internet Protocol, Version 6, (IPv6) Specification", RFC 2460, December 1998.

[RFC2463]: Conta, A. and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 2463, December 1998.

[RFC3279]: L. Bassham, R. Housley, W. Polk, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and CRL Profile", Internet Engineering Task Force, April 2002.

[HTTP-CERTSTOR]: Peter Gutmann, "Internet X.509 Public Key Infrastructure: Operational Protocols: Certificate Store Access via HTTP", Work in progress (draft-ietf-pkix-certstore-http-04.txt), February 2003.

[SWKA00] : Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson, "Practical Network Support for IP Traceback", Technical Report UW-CSE-2000-02-01, Department of Computer Science and Engineering, University of Washington, http://www.cs.washington.edu/homes/savage/traceback.html.

[SDS00] : D. Schnackenberg, K. Djahandari, and D. Sterne, "Infrastructure for Intrusion Detection and Response", Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX), Hilton Head Island, SC, January 25-27, 2000.

[SPSSJTK01]: A.C. Snoeren, C. Partridge, L.A. Sanchez, W.T. Strayer, C.E. Jones, F. Tchakountio, and S.T. Kent, "Hash-Based IP Traceback", BBN Technical Memorandum No. 1284, http://www.ir.bbn.com/documents/techmemos/TM1284.ps, February 7, 2001.

9. Author Information

Steven M. Bellovin,
AT&T Labs Research
Shannon Laboratory
180 Park Avenue
Florham Park, NJ 07974
USA
Phone: +1 973-360-8656
Email: smb@research.att.com

Marcus D. Leech
Nortel Networks
P.O. Box 3511, Station C
Ottawa, ON
Canada, K1Y 4H7
Phone: +1 613-763-9145
Email: mleech@nortelnetworks.com

Tom Taylor [Editor]
Nortel Networks
P.O. Box 3511, Station C
Ottawa, ON
Canada, K1Y 4H7
Phone: +1 613-736-0961
Email: taylor@nortelnetworks.com