A. Ballardie
Consultant

October 1997.

''shared-mtrace'': A Multicast 'traceroute' facility for Shared Trees

Status of this Memo

Abstract

''mtrace'' [1] is a very useful tool for diagnosing IP multicast routing problems, such as multicast routing loops and misconfigured multicast routers, associated with source-rooted RPF-based distribution trees.

For ''mtrace'' to be useful in a shared tree environment (e.g. PIM [2], CBT [3], GUM [4]) its behaviour must be modified. This draft specifies that behaviour, which is sufficiently general to be applicable to all shared tree types and operating modes. A new ''wildcard'' mode of behaviour is also described, which allows a trace of a complete shared tree. Authentication is recommended in this mode because of its potential as a vehicle for denial of service attacks.

It is intended that this draft become a document of the Mbone Deployment (mboned) working group of the IETF. Therefore, comments are solicited and should be sent to mboned's mailing list <mboned@network-services.uoregon.edu> and/or the author.

1. Introduction

"mtrace" [1] evolved some years ago from the need to have tools that provide fault detection and diagnosis of multicast paths - "mtrace" traces the path between a source and receiver via the specified group distribution tree, which must be source-rooted.

Since then, multicast protocols that build shared multicast distribution trees have become more widely deployed, such as PIM and CBT. The latest IDMR offering, GUM [4], is in its early stages of development, and builds an inter-domain shared tree of domains. To accommodate the newer shared tree trend, it is necessary to specify the behaviour of "mtrace" when used with shared trees - we call this "shared-mtrace"; "shared-mtrace" does not alter the underlying means by which "mtrace" gathers information from multicast routers on a path.

2. Overview

The design of "shared-mtrace" has been aimed at it being generally applicable to all shared tree types, and their modes of operation. This avoids requiring one such tool per shared tree {algorithm, mode} pair. Besides the capability to trace a path between a particular source and receiver over a shared tree, we have also included the capability to trace the path between a shared tree's core/RP/root-domain and all the leaves of the tree. This is called a wildcard trace. This mode of operation may be useful for mapping (graphically, or otherwise) a shared tree's current topology. It is recommended that a wildcard trace request be subject to authentication at the the core/RP/root-domain. Security is discussed further in the "Security Considerations" section.

For a wildcard trace, "shared-mtrace" traces the path from the group's core/RP/root-domain to each of the leaves of the tree. The path between the source and the core/RP/root-domain, which may or may not form a branch of the shared tree, must be traced separately.

For a receiver-specific trace, "shared-mtrace" traces the path from the receiver to the core/RP/root-domain, or in some cases (e.g. some instances of PIM), the path from the receiver all the way to the specified trace source. The ability to trace from the receiver all the way to the source IN ONE STEP depends on the presence of (S, G) forwarding state in the routers along the path from the receiver. The presence of (*, G) rather than (S, G) in the routers along the path

results in a trace only being able to progress as far as the core/RP/root-domain.

In some cases (e.g. some instances of CBT), the path between the core and specified source may form part of the distribution tree. Therefore, the trace does not terminate at the core; the trace request is encapsulated and sent to the specified source address, from where it progresses in the direction source -> core, provided group forwarding state exists. When this second trace phase is complete (or if no group forwarding state exists) the the collection of completed trace response segments can be returned to the designated trace response address.

Since there are potentially two phases to "shared-mtrace", the correct interpretation of a trace response is more complicated than with "mtrace". It is necessary for the receiver of a trace response to be able to identify the group's core/RP/root-domain, and also be able to identify whether the trace has progressed over a source tree (i.e. via (S, G) state). This information allows the response to be arranged/ordered correctly, and therefore interpreted correctly. For example, for two different traces (wildcard or receiver-specific), one may be completed in a single phase (e.g. PIM, where (S, G) state exists between the receiver and source), whilst the other may be completed in two phases - the first between receiver and core, the second between source and core (e.g. CBT, where source belongs to a subnet with at least one group member).

In order to be able to make this information available, we propose the following:

+o a unique response code is used on the core/RP/root-domain's response segment. All other response segments use the same response code.

+o if the trace continues beyond the core/RP/root-domain (without being encapsulated to the source), or the trace completes without the inclusion of the core/RP/root-domain's response segment, then the trace must have utilized (S, G) state. That is, the utilization of source specific state is implicit.

3. "shared-trace" Algorithm

A "shared-mtrace" request is unicast from the network manager's

host/workstation to:

+o the core/RP/root-domain for wilcard traces

+o the specified receiver for receiver-specific traces

from where the trace begins.

The algorithm is as follows:

```
if (wildcard dst) /* wildcard trace */
      while (out_intf_list != NULL)
            insert response segment;
            forward trace over each out_intf;
else
      while (!first hop rtr for src))
            insert response segment;
            if ((S, G) for src)
                  forward trace over (S, G) parent_intf;
            else
                  if (RP)
                        encapsulate trace to src;
                        if (multicast_proto == bidirectional)
                              while ((!RP) && (*, G))
                                    insert response segment;
                                    forward trace over (*, G) par-
                                    ent_intf;
                              break;
                  else
                        insert response segment;
                        forward trace over (*, G) parent_intf;

unicast/multicast trace response to response address
end
```

4. Security Considerations

In wildcard mode, "shared-mtrace" could be used as a vehicle to mount a denial of service attack. We therefore recommend that wildcard mode be subject to the following constraints:

+o all wildcard traces begin at the group's core/RP/root-domain, which must authenticate the source of the trace request, using a

proven, cryptographically strong, authentication technique such as Keyed MD-5 [5]. This technique has been adopted by many other Internet protocols, e.g. [ripv2, snmpv2]

+o the core/RP/root-domain may optionally maintain access control lists (inclusion or exclusion lists) as an initial check point (before authentication).

+o it may be useful to allow more fine-grained levels of access to wildcard mode, for example, a source not included in an "include list" (or named in an "exclusion list") are restricted to wild-card traces whereby the trace source and response addresses must be the same. This would cause the target of an attack to be the attacker itself.

In general, we recommend that the header of all mtrace (shared-mtrace) packets should contain the necessary authentication fields. The additional fields indicated have been specified with numerous other Internet protocols, and allow authentication algorithm independence. The multicast traceroute header would then be as follows:

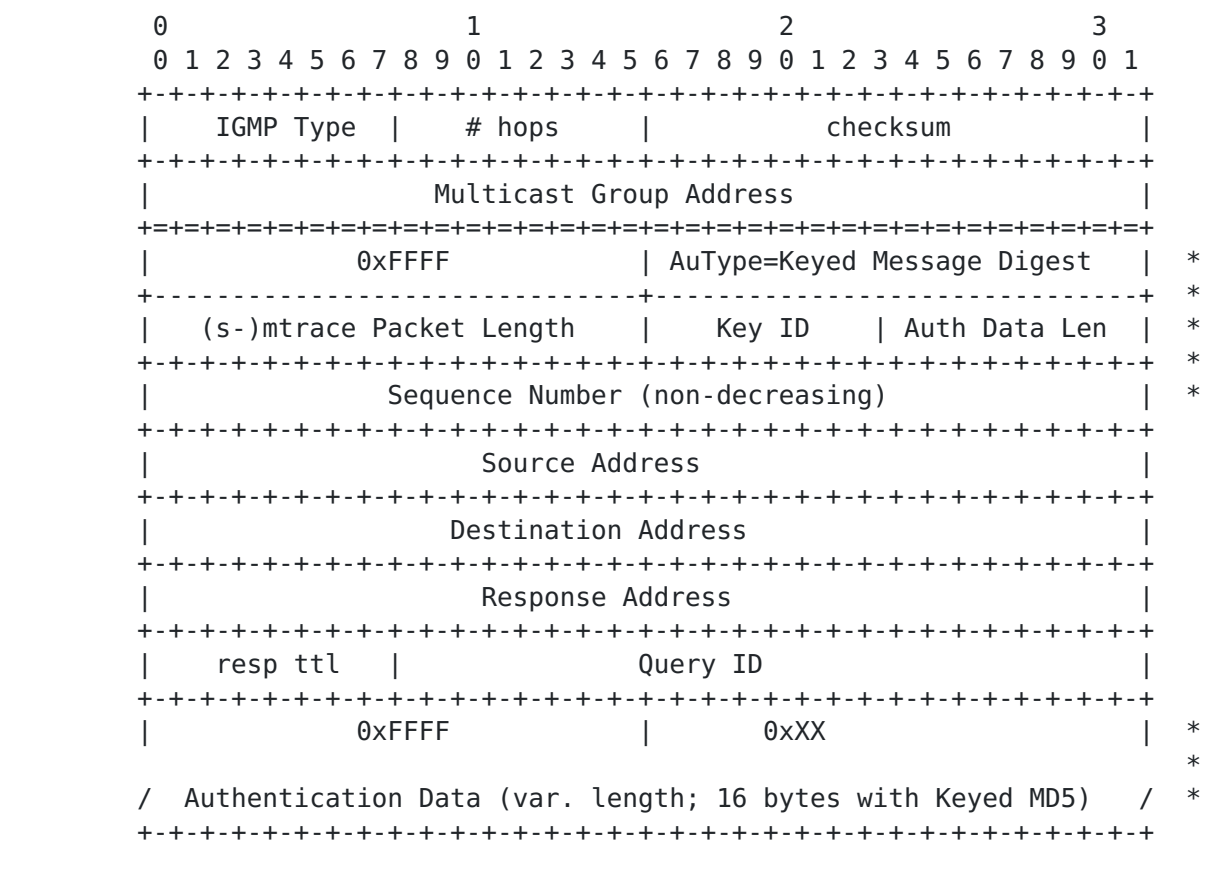

Figure 1. Proposed (shared-)mtrace Header for optional Authentication

The required additional fields for the purpose of authentication are indicated by asterisks (*).

Acknowledgements

Thanks to Bill Fenner (Xerox PARC) for providing useful comments and suggestions.

<to be completed>

References

[1] A traceroute facility for IP Multicast; W. Fenner and S. Casner; ftp://ds.internic.net/internet-drafts/draft-ietf-idmr-traceroute-ipm-00.txt; Working Draft, March 1995.

[2] RFC 2117, Protocol Independent Multicast (PIM) Sparse Mode Specification; D. Estrin et al; ftp://ds.internic.net/rfc/rfc2117.txt. August 1997.

[3] RFC 2189, Core Based Trees (CBT) Multicast Routing: Protocol Specification; A. Ballardie; ftp://ds.internic.net/rfc/rfc2189.txt. September 1997.

[4] Grand Unified Multicast (GUM) Protocol Specification; D. Thaler, D. Estrin, and D. Meyer, Editors; ftp://ds.internic.net/internet-drafts/draft-ietf-idmr-gum-00.txt; Working Draft, July 1997.

[5] IP Authentication using Keyed MD-5; P. Metzger and W. Simpson; RFC 1828; ftp://ds.internic.net/rfc/rfc1828.txt.

Author Information:

Tony Ballardie,
Research Consultant.
e-mail: ABallardie@acm.org