MAFIA: A Multicast Management Solution for Access Control and Traffic Filtering

Krishna N. Ramachandran and Kevin C. Almeroth

Department of Computer Science
University of California
Santa Barbara, CA 93106, USA
{krishna, almeroth}@cs.ucsb.edu

Abstract. Recently, multicast has seen only marginal wide-scale deployment. One of the main reasons is the lack of support for security and traffic management. Although there has been some recent work, these efforts have not emphasized the critical need to deploy security features side-by-side with management solutions. In this paper, we propose MAFIA, a multicast management solution with the specific aim of strengthening multicast security through multicast access control, multicast traffic filtering, and the prevention of DoS attacks. MAFIA achieves these tasks by making use of information about multicast group memberships available at different locations in a network. We have also designed various deployment solutions for MAFIA. We have implemented one such solution using the GNU/Linux operating system.

1 Introduction

Although IP multicast is an efficient technology for the delivery of multimedia, only a small percentage of end users receive multimedia through multicast streams. This has mainly to do with the lack of global deployment of multicast. Several reasons have been cited for slowing deployment, chief among them being concerns with multicast security and multicast management [1].

The need for security solutions is only now receiving attention from the engineering and research communities [2]. Unfortunately, the need for management solutions has received little attention [1],[3]. This is in spite of the importance of management solutions for purposes of multicast access control and traffic management. Access control is needed so that multicast access can be controlled on per host and per multicast source/group basis. For example, in an enterprise it may make sense to allow only some privileged hosts to send and receive traffic from a multicast group, whereas all other hosts are restricted to only receiving traffic from that group. Traffic management is important for reasons of efficient bandwidth utilization, quality of service, pricing, and security.

Some solutions have been proposed for multicast access control and multicast traffic filtering. For example, research has been done to control multicast access through encryption and the selective distribution of the keys used for encryption [4],[5]. However, such solutions rely on complex cooperation between the participants of a multicast group and function at layers above the network layer. In the case of multicast traffic filtering, we are aware of one solution [6] that relies on

a proxy-based approach to unicast multicast datagrams to end receivers. Their approach does not scale well for two reasons. First, it relies on unicast delivery of packets to end receivers. Second, it requires changes to host software. In fact, we are not aware of any firewall — commercial or experimental — that natively supports multicast traffic filtering. In reality, most firewalls are configured to drop UDP packets and therefore, also block multicast traffic. To allow multicast traffic through firewalls, tunneling techniques have been proposed [7],[8]. However, these techniques do not address the real issue, which is to ensure that harmful multicast traffic does not enter or leave the protected enclave.

For the quicker adoption of multicast, multicast management solutions are needed that can 1) control multicast group membership, 2) filter multicast traffic flowing in and out of the enterprise, and 3) prevent multicast denial of service attacks using multicast access control as a prevention technique. In this paper, we propose such a management solution, called MAFIA. There are two main challenges in the design of MAFIA. First, MAFIA needs to accommodate the "open" IP service model that multicast is based on. In such a model, the identity of an end host is not maintained by routing entities. This makes group membership control difficult because the identity of a host is needed to control its membership to a group. Second, MAFIA needs to be deployed such that it intrudes as little as possible on the normal operation of the network. By this we mean that end hosts should not be aware of the existence of such a solution. Consequently, they need not cooperate with MAFIA for it to function. Also, routers should undergo little change to support MAFIA. Accommodating these goals comes with some tradeoffs. We discuss the associated tradeoffs when we evaluate the deployment options for MAFIA against a number of factors such as ease of deployment, flexibility in terms of functionality, and routing state overhead.

The remainder of this paper is organized as follows. In Section 2, we reason why multicast management should complement security specific solutions to improve multicast security. Section 3 presents the requirements for MAFIA. In Section 4, we propose the MAFIA architecture and also discuss various options for its deployment. In Section 5, we evaluate the deployment options and in addition, discuss our implementation of MAFIA. Finally, we conclude in Section 6.

2 Multicast Security and Multicast Management

This section describes in detail why solutions to manage multicast need to be deployed side-by-side with security-specific solutions. We start first by broadly classifying multicast security as follows:

1. prevention of *Data Attacks*: the protection of data exchanged between hosts. Data attacks compromise the confidentiality and integrity of data.
2. prevention of *Control Attacks*: the protection of control information exchanged between multicast routing entities. Control attacks compromise the multicast routing state stored in routers.

Data can be protected using encryption. However, in the absence of access control, encryption alone cannot prevent an *edge-receiver attack* [9], an attack in which a multicast receiver joins an encrypted transmission to simply waste bandwidth or to record the encrypted traffic. If the traffic is in fact recorded, it could be decrypted in non-realtime by leveraging easily available computing power. This compromises data confidentiality. Furthermore, in the absence of access control, encryption cannot prevent an *edge-sender attack* [9], an attack in which a malicious host sends bogus packets to interfere with the successful delivery of group traffic to other receivers. For example, suppose some participants in a multicast group are in a video conference. A malicious participant or outsider can transmit bogus traffic to this group and garble the legitimate traffic in the group. Therefore, host access control needs to be used along with techniques such as encryption to protect data.

Prevention of control attacks can be partially achieved using encryption technologies such as IP-Sec [10]. With encryption, routing entities exchange control information over a secure channel this making it almost impossible for a malicious routing entity to inject bogus routing state. However, just the encryption of control traffic does not prevent denial of service attacks against routing entities. For example, a receiver, by subscribing to a large number of multicast groups, can waste bandwidth and overload routers with excess Protocol Independent Multicast (PIM) forwarding state. As another example, consider attacks launched by the RAMEN or Sapphire worms. These attacks, launched from end hosts, resulted in routers becoming overloaded with large amounts of bogus Multicast Source Discovery Protocol (MSDP) state. Therefore, just the encryption of control traffic is not sufficient to prevent such attacks, as these attacks were launched from end hosts whose access to multicast could not be controlled.

Even if multicast access control can be achieved, a sometimes overlooked problem is that UDP is the cause of some security breaches and is often blocked [6]. Blocking of UDP traffic may be too stringent a requirement for enterprises where the potential savings with the use of multicast far outweigh the threat, if any, with its use. Two solutions to minimize the threat with UDP are as follows:

- limit the use of multicast to only trusted hosts and groups. This can be done by controlling access with the use of *multicast security policies*. A multicast security policy defines which groups and hosts are considered safe. These policies can be enforced at the above mentioned control points.
- filter traffic flowing into the enterprise using state gathered from multicast routing protocols in accordance with the multicast security policies.

Clearly, it follows from the above discussion that management of multicast in an enterprise — through access control and the multicast traffic filtering — is needed along with security specific solutions to improve the overall security of a multicast deployment.

3 MAFIA Requirements

In this section, we discuss the requirements for MAFIA in detail. Briefly, the requirements are: (1) Multicast Access Control, (2) Multicast Packet Filtering, and (3) Prevention of DoS Attacks.

3.1 Multicast Access Control

Multicast access control can be broadly classified as *host access control* and *designated router access control*. Host access control controls which host can be a member of a certain multicast group. Controlling the membership behavior of a group of hosts on a subnet to subnet basis is achieved through designated router access control. For host access control, exact host-to-group associations are needed. For designated router access control, designated-router-to-group associations are needed. These two associations i.e. host-to-group associations and designated-router-to-group associations, are available at two distinct locations in the network, we call the *Last Hop Control Point* and the *Central Control Point*. The two points are shown in Fig. 1. The two access control functions are further defined below:

- **Host Access Control**. Since a host can be either a sender or receiver in a multicast group, host access control can be of the following two types:
  1. *Receiver Access Control*: The reception of multicast traffic on per (S,G), and per host basis can be controlled. Receiver access control can be very useful in bandwidth control and the prevention of edge-receiver attacks.
  2. *Source Access Control*: Source behavior can be controlled on per group and per host basis. Source access control can be very useful in the prevention of edge-sender attacks.
- **Designated-Router Access Control**. Having designated router access control is useful in the following cases:
  1. the last hop control points lie in a different administrative domain.
  2. host access control is not implemented or not necessary on the last hop.
  3. to prevent denial of service attacks launched from different subnets. At the granularity of each last hop control point, a distributed attack would not be detectable. To detect such attacks, a global view of the entire network is needed, which can be obtained by looking at designated router membership behavior.

  Since designated routers act on behalf of receivers and sources, the two resulting access control types are *Proxy-Receiver Access Control* and *Proxy-Source Access Control*. With proxy-receiver access control, the reception of multicast traffic on per (S,G), and per subnet basis can be controlled. With *proxy-source access control*, notifications of new sources sent by designated routers to the local rendezvous point (RP) can be controlled.

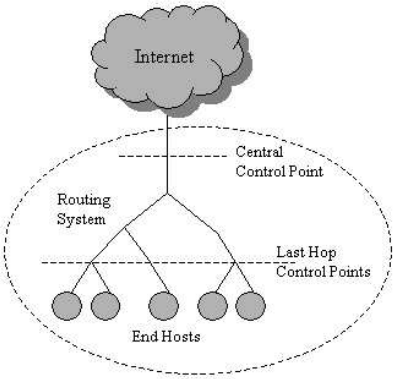

Fig. 1. Protocol regulation points in a network-hierarchy.

3.2 Packet Filtering

UDP traffic is generally blocked by network administrators. One of the reasons is that UDP's connectionless nature can be exploited for performing "port-spoofing" attacks. This problem is not specific to multicast communication. It applies to unicast as well. An effective mechanism to prevent UDP based attacks is to filter malicious packets at the firewall using the multicast policy.

3.3 Prevention of Multicast DoS Attacks

Multicast protocols are vulnerable to DoS attacks. Some attacks result from flawed protocol implementations [11]. However, most of the easily exploitable problems are due to poor protocol specifications. For example, MSDP, a protocol used to advertise the actively transmitting multicast sources, exchanges MSDP Source Active (SA) messages that carry advertisements using a *flooding* mechanism. Flooding of SAs makes MSDP inherently unscalable by design. Attacks by the RAMEN and Sapphire worms are examples of how MSDP's flooding mechanism can be exploited [2]. Using IGMP, it is extremely easy for an end-host to launch edge-sender and edge-receiver attacks. Another consequence of such attacks is the overloading of PIM routers because of the large amount of PIM state created for delivering unwanted traffic.

The problem with DoS attacks can be most effectively solved using a dual approach. One is to limit the use of multicast to only trusted hosts and groups. This can prevent internally launched IGMP, PIM, and MSDP DoS attacks. The second approach is filter bogus packets that result from DoS attacks launched from external networks. For instance, MSDP attacks launched from external networks can be prevented by filtering bogus MSDP SAs.

4 MAFIA Architecture

From our discussion of the MAFIA requirements in Section 3, it follows that two separate functional modules are needed: one to filter UDP packets (requirement

1) and the other to control multicast access (requirement 2). We call the two modules the *MAFIA Packet Filter* and the *MAFIA Access Controller* respectively. The Packet Filter is co-located with the protected enclave's firewall at a central control point (see Fig. 1). The access controller, on the other hand, is situated in the interior of the protected enclave. Requirement 3 from MAFIA is necessary to prevent IGMP, MSDP, and PIM DoS. IGMP edge-sender and edge-receiver attacks are prevented by the MAFIA access controller. Its detailed operation is explained later in this section. PIM DoS attacks are prevented as a consequence of preventing the IGMP attacks. In addition, preventing an IGMP edge-sender attack will also prevent the launch of MSDP DoS attacks from the inside of the protected enclave. This is because containing the number of groups a sender can transmit traffic to will automatically limit the number of MSDP SA messages generated by the local RP. However, externally launched MSDP attacks can still affect the MSDP peer in the protected enclave if the influx of bogus SAs from the outside is not prevented. Since, the MAFIA Packet Filter is co-located with a firewall, the filtering of SAs is also done by the MAFIA packet filter.

Figure 2 illustrates the conceptual view of the MAFIA architecture. It shows a third module, the MAFIA policy server. The policy server maintains the multicast policy. Updates to the policy are always done at the policy server. The updates are then mirrored at the access controller and the packet filter to reduce the latency involved in serving an access request.

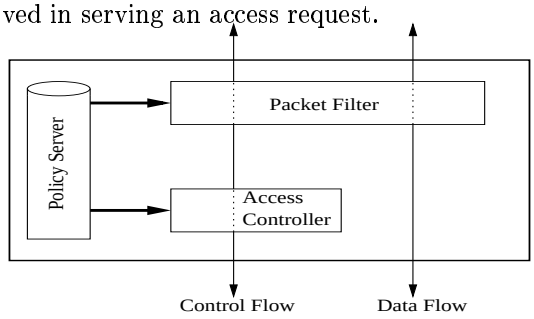

Fig. 2. MAFIA architecture.

This rest of this section describes the architecture of the MAFIA access controller and the MAFIA packet filter.

4.1 MAFIA Access Controller

The MAFIA access controller implements the four types of access control discussed in Section 3.1. For host access control, host-to-group associations are needed. These are available only at the last hop control points. On the other hand, for designated router access control, designated-router-to-group associations are needed. This information is available at the centralized control point. As these two types of associations are available at two distinct locations in the network hierarchy, the MAFIA access controller is composed of two separate modules present at each of these two locations. We call these modules the

MAFIA Last Hop Control Point (MLHCP) and the MAFIA Centralized Control Point (MCCP). Figure 3 illustrates the placement of the MLHCP and MCCP in a network. Although the MAFIA Access Controller is made up of the MLHCP and the MCCP, their architecture details and deployment considerations are discussed separately. This is because both modules function independent of each other and as a result do not affect each other's operation in any way.

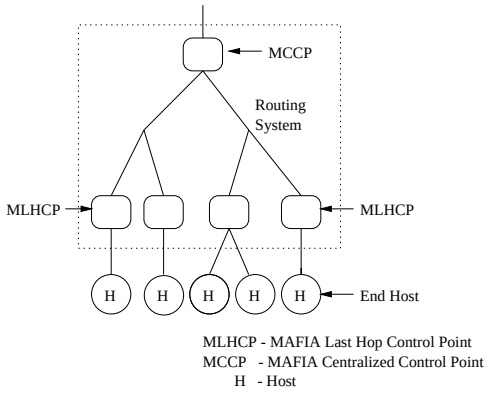

Fig. 3. MAFIA access controller.

MAFIA Last Hop Control Point (MLHCP). The MLHCP implements receiver and sender access control. IGMP membership reports are used to implement receiver access control. When the MLHCP receives an IGMP report, it uses the locally cached multicast policy to decide if the requested access is permitted. For sender access control, the arrival of a multicast datagram is the trigger to authorize a transmission. This is because a multicast sender can send traffic to a group without ever joining the group.

The MLHCP can be deployed either actively or passively, depending on whether the MLHCP is also a routing entity. The MLHCP deployed with the designated router is an active MLHCP, as it performs routing functions. Router vendors provide support for host access control at the designated router through static ACLs (access control lists). However, the flexibility offered by this is limited as additional tasks such as inspecting packet payloads or maintaining state between multiple packets cannot be easily done on routers.

Figure 4 illustrates the passive solution. Here, the designated router ignores (using ACLs) IGMP reports received from all hosts except ones received from the *designated host* — a dual network interface host that acts as a proxy for all other hosts in the subnet. Interface a of the designated host listens to all IGMP reports generated on the last hop subnet. Interface b receives all PIM Register messages generated by the designated router. When a host sends an IGMP membership report expressing interest in receiving traffic from a group, the report is received by the designated router and the designated host. As the designated router is configured to ignore all reports except ones from the designated host, it ignores the report. When the designated host receives the membership report, it checks

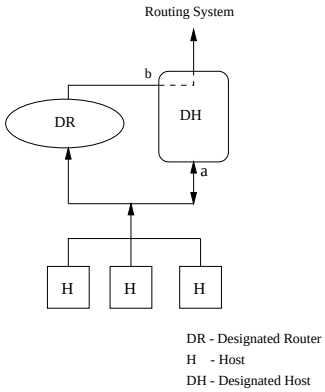

Fig. 4. Designated host as MLHCP.

if the host is permitted to perform the requested operation. If the requested operation is permitted, the designated host in turn generates a membership report with the same information as contained in the original report. Since this report is generated by the designated host, the designated router now accepts it and initiates the creation of the distribution tree.

For sender access control, as the MLHCP receives all PIM register messages, it checks whether the operation is permitted by the policy. If the PIM register message is not permitted, the MLHCP simply drops the message. An alternate configuration for this interface is to operate in snooping mode. In snooping mode, the interface listens only to the PIM register messages. When the MLHCP functions in this mode, it cannot prevent messages from reaching the upstream RP. Therefore, to counteract the effect of the PIM register message, the MLHCP masquerades as the RP and originates a PIM unregister message towards the designated router. When the designated router receives the PIM unregister message, it ignores any PIM register messages that the RP may generate. Consequently, it does not forward any data towards the RP. The disadvantage with the snooping configuration is that the MLHCP cannot prevent unauthorized PIM register messages from reaching the RP. Therefore, if a host randomly sends data to a large number of multicast groups — like in a RAMEN worm attack — a large number of PIM register messages will reach the RP. This may launch a MSDP SA flood.

MAFIA Centralized Control Point (MCCP). The MCCP performs designated router access control. It implements proxy-receiver access control by filtering PIM Join messages. Proxy-sender access control is implemented by filtering PIM Register messages. In considering MCCP's deployment, it helps to classify the MCCP based on the role it plays in multicast routing as follows:

- **Active MCCP**: An active MCCP is also a multicast routing entity and therefore takes part in the creation and maintenance of distribution trees. Figure 5 shows one possible deployment solution where the MCCP is im-

plemented in a multicast router. Router vendors already provide some support for controlling designated router behavior by way of access control lists (ACLs). However, the flexibility offered by such a solution is limited. For instance, certain signature-based attack detection techniques [2] require that state be maintained between packets. Such flexibility is not offered by ACLs.

– **Passive MCCP**: A passive MCCP is not a multicast routing entity. However, it receives every protocol message destined for upstream routers. It can be a dedicated system that performs complex tasks such as inspecting packet payloads, maintaining state between multiple packets, detecting attack signatures, and packet monitoring. Figure 6 illustrates a passive MCCP deployment. In this deployment, packets are filtered before they reach upstream routers. The MCCP can also be deployed in snooping mode. In snooping mode, the MCCP cannot filter packets. Therefore, it reacts to an unauthorized request by sending a protocol message that counteracts the request. So, for PIM Join messages, the snooping MCCP will send a PIM Prune message to the upstream PIM router. For a PIM Register message, the snooping MCCP will send a PIM Unregister message to the designated router that sent the register message. A snooping MCCP, however, is not completely effective in preventing PIM flooding attacks. This is because if a large number of unauthorized requests are sent in a short period of time, PIM routers will get overloaded, albeit temporarily, with large amounts of state.

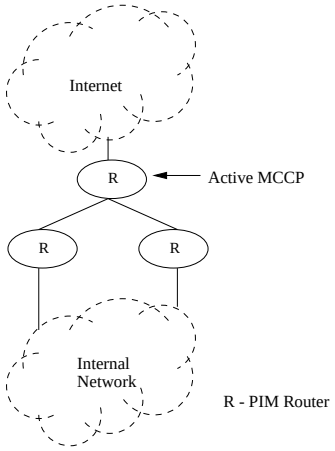

Fig. 5. Active MCCP.

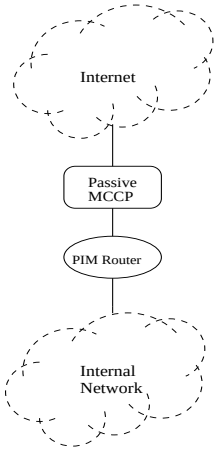

Fig. 6. Passive MCCP.

4.2 MAFIA Packet Filter

The MAFIA packet filter does two types of packet filtering:

UDP Filtering. UDP filtering is simple UDP flow filtering to ensure that UDP packets that flow through the firewall match certain criteria. If the UDP packets do not match the given criteria, they are dropped. The criteria for UDP packet filtering is specified as part of the multicast policy. The simplest criteria is to

ensure a multicast datagram in an incoming or an outgoing stream carries a destination multicast address that corresponds to some "live" multicast distribution tree. MAFIA keeps track of "live" trees by tracking PIM Join messages and corresponding PIM Prune messages. In the normal case, multicast routers generally never forward packets that do not belong in a distribution tree. However, experience tells us that malfunctioning routers erroneously forward such packets. This simple criteria will ensure that bogus packets are dropped. More complex criteria can also be used to filter packets. For example, one criteria would be to drop all multicast datagrams destined to well-known ports. Such a criteria can be effective in preventing UDP port spoofing attacks. Signature-based schemes can also be applied to filter malicious UDP packets with unique signatures.

MSDP-SA Filtering. MSDP SA filtering is done by looking up the multicast policy and determining which multicast groups are permitted by the policy. Only SAs for "joinable" groups are let through the firewall and the remaining are dropped. As with UDP filtering, signature based schemes can be used to filter MSDP SAs. For instance, the RAMEN worm has a unique signature pattern of a large number of SAs with increasing class D addresses originating from the same source. Other SA filtering schemes proposed in [2] can also be applied for more effective filtering.

5 Evaluation

In this section, we evaluate the MAFIA architecture discussed in Section 4. Our goal is to evaluate the architectures against various evaluation metrics. First, we present the methodology used in our evaluations.

5.1 Methodology

The performance of MAFIA depends on the following factors:

- **MAFIA System Configuration**: Performance of MAFIA ultimately depends on its hardware and software configuration. Factors include processor speed, amount of memory available, network card capabilities, and operating system used.
- **Multicast Group Characteristics**: Performance depends on characteristics such as number of groups, sources, and receivers.
- **Traffic Characteristics**: The traffic characteristics depend on the group characteristics and the rate at which each multicast source transmits.
- **Multicast Policy**: If the multicast policy is complex and restrictive in terms of allowing multicast access, the load on MAFIA is greater.
- **Link Bandwidth**: Link bandwidth may be low enough for MAFIA to operate effectively even under maximum utilization. On the other hand, MAFIA may not be able to handle high bandwidth traffic.

We considered evaluating MAFIA through simulations. However, trying to accommodate the above factors in a simulation environment would be difficult and

the results would not lead us to any particularly non-obvious conclusions. Therefore, instead of presenting empirical results, we instead focus on an evaluation of the various ways MAFIA can be deployed. To this end, we limit our evaluations only to the MAFIA access controller. This is because the access controller, which is composed of the MLHCP and the MCCP, can be deployed in more than one way and each deployment option offers some interesting tradeoffs. The MAFIA Packet Filter, on the other hand, can be deployed in only one way.

5.2 Evaluation Criteria

We use the following criteria for our evaluations:

- **Ease of deployment**: We evaluate the ease with which the various architectures can be deployed.
- **Flexibility**: We evaluate the flexibility offered by an architecture in terms of the range of features (functionality) an architecture can support.
- **Traffic Rates**: We evaluate the capability to handle high traffic rates.
- **Routing state**: We evaluate scalability in terms of how much routing state needs to be maintained by an architecture.

5.3 Evaluation Results

MLHCP. We evaluated the active and passive MLHCP deployment options discussed in Section 4.1. The passive MLHCP is easier to deploy as it is deployed on a dedicated system and therefore requires no changes to router software. Furthermore, deployment can be easily done using "off-the-shelf" commodity hardware and software. The active MLHCP, on the other hand, is deployed on the router, which means that the router software needs to change to support the MLHCP. The passive MLHCP also offers more flexibility than an active MLHCP. This is because the passive MLHCP does not perform any routing functions and hence can perform more complex tasks such as maintaining state between multiple packets, packet logging, and traffic analysis.

However, a passive MLHCP cannot easily handle very high traffic rates (of the order of gigabits per second). This is because commodity hardware and software cannot easily scale to higher traffic rates. To overcome this problem, in [12], the authors propose traffic splitting architectures to scale commodity hardware and software to handle high traffic rates. The problem with such architectures, as the authors themselves acknowledge, is that they are difficult to implement. The active MLHCP, on the other hand, can handle high traffic rates because the traffic rates it handles after all depends on the capability of the router itself.

With respect to routing state maintained at the designated router, the active MLHCP results in no state being maintained for unauthorized requests. This is because an active MLHCP filters an access request before it reaches the designated router. As the passive MLHCP cannot filter unauthorized requests, state is maintained at the designated router, albeit temporarily, for unauthorized requests.

In summary, the passive MLHCP is a more flexible architecture and is easier to deploy. However, the passive MLHCP cannot handle high traffic rates easily and also results in more state being maintained at the designated router as compared to the active MLHCP.

MCCP. As with the passive MLHCP, the passive MCCP is more flexible and is easier to deploy. However, unlike the passive MLHCP, the passive MCCP does not result in state being maintained at upstream routers for unauthorized requests. This is because all PIM messages are filtered before they reach upstream routers.

The problem with high traffic rates is more serious at the passive MCCP. This is because the passive MCCP now has to filter requests coming from several last hop subnets. Consequently, the passive MCCP can easily become overloaded even at moderate link usage. Traffic splitting [12] could be used to alleviate the problem with high traffic rates. However, this adds complexity to the system.

In summary, the passive MCCP is easier to deploy, more flexible and maintains no more state than what results from an active MCCP. On the other hand, the active MCCP performs better at higher traffic rates as it is deployed on a router.

5.4 Implementation

We have chosen to implement MAFIA as a combination of the passive MLHCP and the passive MCCP. Both architectures offer better flexibility than their active counterparts. Moreover, they can be deployed with no changes to router software. This means that the implementation of MAFIA is not tied to any one router vendor's product.

The MLHCP and the MCCP have been implemented on the GNU/Linux operating system using the netfilter and iptables frameworks[1]. Using these frameworks, GNU/Linux offers a comprehensive packet filtering capability that is open-source, well tested, and widely deployed. Netfilter uses a loadable kernel module (LKM) called a *match* to filter packets. For MAFIA, we implemented two match modules called *MSDP match* and *PIM match*. The PIM match is used to filter PIM Register and PIM Join messages. The MSDP match is used to filter MSDP Source Active messages. The MAFIA packet filter uses the MSDP match and the UDP match (already existing in netfilter) to do MSDP and UDP packet filtering. The PIM match is used by the MCCP and the MLHCP. The MCCP uses the PIM match to filter PIM messages before they reach upstream routers. The MLHCP uses the PIM match to filter PIM messages originated by the designated router. In addition to the PIM match, the MLHCP also uses the IP match module to filter IGMP reports on the last hop. For authorized requests the MLHCP uses the libnet[2] packet-generation tool to generate IGMP reports.

We tested our implementation of MAFIA using a netfilter enabled GNU/Linux system. We used a packet generating tool written using libnet and libpcap[3] to

[1] http://www.netfilter.org

[2] Libnet packet assembly tool, http://www.packetfactory.net/libnet

[3] Libpcap packet capture tool, http://sourceforge.net/projects/libpcap

generate our test traffic. The packet generator uses information in its configuration file to randomly send a mixture of UDP, PIM Join, PIM Register, and MSDP SA packets to the Linux system. We used a restrictive multicast policy to create the appropriate netfilter rules on the GNU/Linux system. All packets sent by the packet generator that do not match the netfilter rules were dropped. This test confirms the correct operation of our MAFIA implementation.

6 Conclusions

The lack of multicast management adversely affects multicast security. A testament to this is the recent increase in the number of denial of service attacks against multicast. Moreover, multicast management enables network administrators to manage their multicast deployments for purposes of administrative control and efficient resource utilization.

In this paper, we have proposed MAFIA, a multicast management solution that addresses three requirements: multicast access control, multicast data and control packet filtering, and denial of service attack prevention. We have looked at various ways MAFIA can be deployed. In addition, we have evaluated each deployment option against various factors such as ease of deployment, flexibility, routing state overhead, and its capability to handle high traffic rates. As a result, network designers need to consider the tradeoffs associated with each deployment option before deploying MAFIA in the network. We have implemented MAFIA using the netfilter architecture in the GNU/Linux operating system. We plan to offer the netfilter extensions written for our implementation in an upcoming release of netfilter.

References

1. K. Sarac and K. Almeroth. Supporting Multicast Deployment Efforts: A Survey of Tools for Multicast Monitoring. *Journal of High Speed Networking–Special Issue on Management of Multimedia Networking*, March 2001.
2. P. Rajvaidya, K. Ramachandran, and K. Almeroth. Detection and Deflection of Denial of Service Attacks against the Multicast Source Discovery Protocol. *UCSB Technical Report*, May 2003.
3. E. Al-Shaer and Y. Tang. Toward integrating IP multicasting in internet network management protocols. *Computer Communications*, 24(5-6):473–485, 2001.
4. C.K. Wong, M. Gouda, and S. Lam. Secure group communications using key graphs. In *ACM SIGCOMM*, pages 68–79, 1998.
5. I. Chang, R. Engel, D. Kandlur, D. Pendarakis, and D. Saha. Key management for secure internet multicast using boolean function minimization techniques. In *IEEE Infocomm'99*, pages 689–698, 1999.
6. K. Djahandari and D. Sterne. An MBone proxy for an application gateway firewall. *IEEE Symposium on Security and Privacy*, 1997.
7. R. Finlayson. The UDP Multicast Tunneling Protocol. Internet Engineering Task Force (IETF), draft-finlayson-umtp-*.txt, September 2002.
8. D. Chouinard. SOCKS V5 UDP and Multicast Extensions to facilitate multicast firewall traversal. Internet Engineering Task Force (IETF), draft-ietf-aft-mcast-fw-traversal-*.txt, November 1997.
9. T. Hardjono and G. Tsudik. IP multicast security: Issues and directions. *Annales de Telecom*, 2000.
10. *IP Security Protocol (ipsec).* http://www.ietf.org/html.charters/ipsec-charter.html.
11. *Spoofed IGMP Report Denial of Service Vulnerability.* http://online.securityfocus.com/bid/5020/info.
12. C. Kruegel, F. Valeur, G. Vigna, and R. Kemmerer. Stateful intrusion detection for high-speed networks. *IEEE Symposium on Security and Privacy*, May 2002.