Detection and Deflection of DoS Attacks Against the Multicast Source Discovery Protocol

Prashant Rajvaidya, Krishna Ramachandran and Kevin C. Almeroth

***Abstract*—Studying and solving security problems in the multicast infrastructure has been largely overlooked. Efforts have been more focused on developing, then deploying, and then debugging the basic protocols. Although a *number* of security vulnerabilities exist, the most troubling are a set of Denial-of-Service (DoS) attacks. The main reason for this concern is that the one-to-many nature of multicast can significantly magnify the effects of these attacks. Among the possible multicast DoS attacks, those that target the vulnerabilities of the Multicast Source Discovery Protocol (MSDP) can be most damaging. MSDP vulnerabilities are unusually easy to exploit and can lead to infrastructure-wide damage. In this paper, we study the vulnerability of MSDP to DoS attacks. In reality, attacks on MSDP have already occurred. Using these attacks as a basis, we develop and describe several new strains of even more virulent MSDP-based DoS attacks. We also propose a family of solutions to detect and "deflect" the effects of each strain of attack. Our techniques are evaluated by simulating their effectiveness against several attack workloads.**

I. INTRODUCTION

Network security attacks pose a serious threat to the stability of the Internet. These attacks include gaining unauthorized access by exploiting vulnerabilities in end hosts, disrupting protocol operations by injecting malicious control packets, consuming resources of network degrading its services, and many others. Among these attacks, Denial-of-Service (DoS) attacks are quite common and damaging to the network infrastructure[1]. DoS attacks aim to incapacitate the network by flooding it with unwanted traffic. Depending on the nature of the malicious traffic, DoS attacks can either consume the computing resources of network devices, e.g. routers, and cause them to malfunction or saturate network links to render them useless or both. DoS attacks usually affect the entire networks and deny legitimate users access to services. Although several mathematical models have been proposed to detect such anomalies[2], detecting DoS attacks is a challenging task. Difficulties arise because it is hard to differentiate unexpected but legitimate usage patterns from those due to malicious activity.

Authors are with the Department of Computer Science, University of California Santa Barbara, CA 93106-5110, E-mail:{prash, krishna, almeroth}@cs.ucsb.edu

Emerging Internet technologies like multicast are typically more vulnerable to security breaches because their protocols are in the early developmental stage. These protocols often neither well-designed to handle attacks nor have they been tested in real attack situations. Furthermore, there is also a lack of operational know-how that makes secured deployment difficult. In the specific case of multicast, security problems can be even more problematic than other services. The first reason is that multicast networks operate on top of the existing Internet infrastructure. Therefore, they inherit most of the security problems of the current Internet. Second, because the multicast service model is fundamentally different from that of the unicast, multicast networks introduce a whole new set of vulnerabilities. In spite of increased security problems, developments in securing the multicast infrastructure have not been keeping pace with other developments in field. Although a growing number of network administrators are deploying multicast, operational experience in how to detect and control attacks is insufficient. In the current Internet, efforts to make multicast protocols more secure have been superseded by the basic need to debug existing protocol, assure widespread deployment, and make the infrastructure stable. As a result, many vulnerabilities have neither been discovered nor rectified.

Among the known multicast security problems, the vulnerability to DoS attacks is of prime concern and the main focus of this paper. There are two main reasons for this concern. First, the broadcast nature of multicast magnifies the effects of DoS attacks by an orders of magnitude equal to the number of receivers. Incidences have already occurred where simple attacks were magnified such that their effect was infrastructure-wide and crippling. Second, much of the multicast infrastructure uses links and devices that provide both unicast and multicast connectivity. Therefore, a DoS attack that exploits the vulnerabilities of multicast can potentially hamper the routine operation of other services in the Internet.

In this paper, we investigate some of the vulnerabilities of the existing multicast infrastructure. Specifically, we focus on the vulnerabilities of Multicast Source Discovery Protocol (MSDP)[3]. MSDP is a good place to start because: (1) it has already been attacked, (2) it is key to the operation of the infrastructure, and (3) it provides a specific example whose results we hope to extend to other protocols. This paper starts by describing our investigation into the effects of MSDP DoS attacks that severely hampered multicast operation in January 2001. Because these attacks were so simple in nature, we also devised more virulent attacks that accomplish the same damage but are harder to detect. This makes the next task of developing solutions to these attacks more challenging. In the next step, we devise and describe some of the mechanisms for detection and *deflection*; i.e. countering the effects of these attacks. Finally, we model workloads for a variety of attacks, simulate the response of our proposed solutions, and evaluate their effectiveness.

The rest of the paper is organized as follows. In Section 2 we give a brief overview of the vulnerabilities of the multicast infrastructure and work in securing it. Section 3 investigates possible DoS attacks that exploit the vulnerabilities of MSDP. In Section 4 we describe the mechanisms for detecting and deflecting these attacks. In Section 5 we describe how simulations were conducted, describe the attack workloads that we have developed, devise metrics for evaluating these mechanisms, and present the evaluation results. The paper is concluded in Section 6.

II. Inter-Domain Multicast and its Vulnerabilities to Attacks

Since its creation, the multicast infrastructure has undergone several fundamental changes: from a dense mode overlay network–called the MBone–to native sparse mode deployment, and finally to the latest efforts in the direction of Source Specific Multicast (SSM)[4], [5]. During the early years, the focus was on protocol design issues like the efficient creation of distribution trees, congestion control, reliability and robustness. More recently, efforts are concentrated on developing new group paradigms, debugging existing protocols, and ensuring widespread deployment. As a result of these efforts, multicast protocols have matured, widespread acceptance is growing, and deployment is increasing[6]. However, amidst these developments, work in securing the multicast infrastructure has been largely overlooked.

In the current infrastructure, most of the multicast protocols have at least some security vulnerabilities. And furthermore, security problems exist in every part of the topology, from end-hosts to core routers. In this section, we discuss some of these security problems. We present vulnerabilities of various multicast protocols and describe how they can be exploited. We also describe how some of these attacks can effect even the basic operation of multicast. We begin by presenting an overview of the operation of the current infrastructure.

A. Operation of Current Multicast Infrastructure

Multicast is a network model for one-to-many and many-to-many delivery of high-bandwidth streams in a scalable and bandwidth-efficient manner. This model uses the notion of a logical group of participants, called a *multicast group*, that are interested in content being sent to them. Each such group is identified using a Class-D IP address that is usually chosen at the sender host. The sender simply sends data packets to a group and the network replicates these packets at appropriate nodes in the topology and forwards them to the participants of that group.

Most of the current infrastructure only supports a service model called Any Source Multicast (ASM). ASM primarily rely on four protocols[7]: the Internet Group Management Protocol (IGMP)[8], the Protocol Independent Multicast (PIM) protocol[9], the Multicast Border Gateway Protocol (MBGP)[10] and the Multicast Source Discovery Protocol (MSDP)[3]. MBGP acts as an inter-domain route exchange protocol that allows the propagation of reachability and path information across domains. PIM uses this route information to create and manage inter-domain distribution trees through which transfer data from source(s) of a group to its participants. Hosts initiate group join and leave operations using IGMP. Finally, MSDP acts as a source announcement protocol and is responsible for propagating information about active sources across the entire infrastructure.

Figure 1 presents a simple illustration of the steps involved in the process of multicast data delivery in ASM. A source (S) starts transmitting data to a group. The First Hop Router (FHR) for S generates a PIM register message and sends it to the Rendezvous Point (RP)–and also an MSDP peer–for the domain. From here, through exchange of MSDP Source Active (SA) messages, information about the source is propagated to the rest of the infrastructure. When a host (R) wishes to receive the content being transmitted to a particular group, it sends an IGMP join request to its own FHR. The FHR, in turn, sends a PIM join message to the domain's RP. Finally, the PIM distribution tree is extended to the receiver and R starts receiving the data that S is sending.

One of the biggest drawbacks of ASM is that it requires networks to have knowledge about, and maintain state for, all multicast sources that are actively sending traffic to any

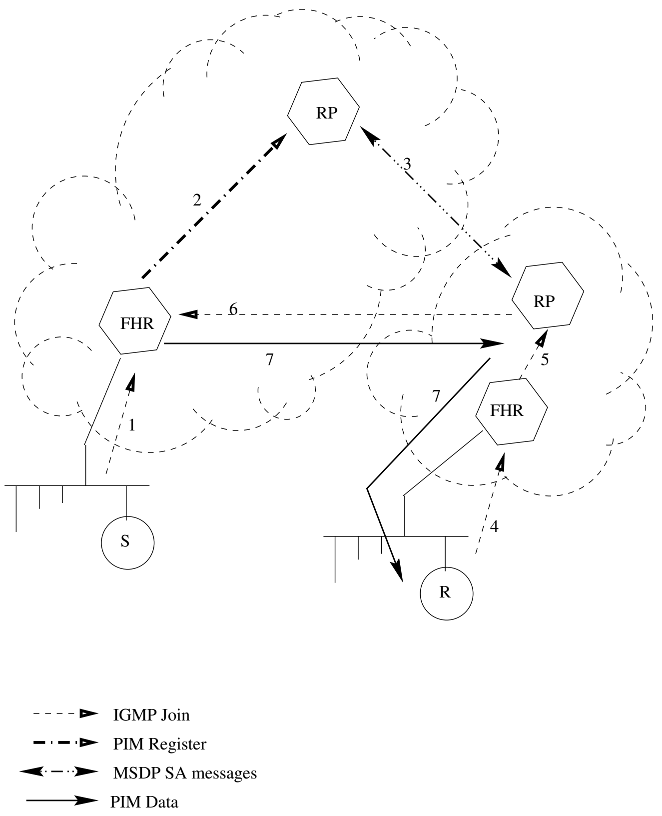

Fig. 1. Multicast data delivery in ASM.

group. This knowledge is critical, because in order to support many-to-many data delivery, ASM must assume the responsibility of knowing the IP addresses of a group's source(s). This responsibility is satisfied via the functionality provided by MSDP. MSDP propagates and maintains information about *all* active sources within the network. Not only does this result in added overhead for network devices (routers), but, as we will soon describe, is one of the main multicast security vulnerabilities.

Although ASM is currently the primary service model for multicast data delivery, serious efforts are underway to simplify it such that it is not necessary for the network to have knowledge about all multicast sources. The one effort that has received the most attention and recognition is the development of a new service model called Source Specific Multicast (SSM)[4], [5]. SSM better supports one-to-many delivery of data, but does not provide a many-to-many delivery service. One of the distinctive characteristics of SSM operation is that it allows receivers to directly provide information about multicast source(s) to the network without requiring the network to maintain any state for it. Therefore, SSM eliminates the need to run MSDP. Receivers relay information pertaining to a multicast source directly to their first hop router with the help of a new version of IGMP[11]. A modified version of PIM, called PIM-SSM[5], is the routing protocol that enables data delivery via SSM. As a result of not needing to run MSDP and not needing PIM RPs, SSM has fewer security vulnerabilities than ASM.

B. Security Vulnerabilities in Multicast

Current deployment of multicast has several security problems, primarily, because the multicast service models have not been designed to provide much security. Although, just like in the case of unicast, multicast allows some security mechanisms to be built on top of existing protocols. However, progress in developing and deploying such mechanisms has been quite slow. Most of the deployment has been reactive and not pro-active–solutions are deployed only after weaknesses have been exploited. The multicast security problems pertain to three area: (1) securing data from eves droppers; (2) providing admission control; and (3) securing protocol control traffic. Each of these problem areas are described in further detail below.

Securing data from eves droppers: One of the elementary security problems that arises in multicast networks pertains to the security of data from eves droppers. Although this problem is the same for unicast networks as well, it becomes harder to solve for multicast because conventional solutions like IPsec and other encryption mechanisms can not be easily applied to multicast data delivery. This is the case because solutions such as IPsec rely on key management and distribution related tasks. These tasks are relatively easy to perform for end-to-end data delivery in unicast. However, in multicast, these tasks suffer scalability challenges due to large groups and dynamic group membership.

Providing admission control: Admission control to multicast groups would allow a group manager, possibly the primary source, to decide which hosts could participate in a group. However, no such mechanisms exist, and as a results, several types of attacks can be launched simply through end hosts being able to join groups arbitrarily. For example, a host can start sending bogus high bandwidth data to a popular multicast group as a form of denial-of-service. In addition, bogus members can join multicast groups with high bandwidth content and cause the distribution tree to cause congestion on key paths in the network. Other kinds of attacks are possible when groups implement additional services on top of data delivery like congestion control and reliability. A single group member, incorrectly claiming high loss or congestion, can cause a source to significantly reduce its transmission rate.

Securing protocol control traffic: All of the multicast protocols mentioned so far, including IGMP, PIM, MBGP, and MSDP, have security vulnerabilities regardless of whether they are being used for ASM or SSM.

These vulnerabilities stem from the weaknesses of the control and reporting mechanisms that the protocols use. Malicious control traffic can be injected into the network to cause applications at the participant hosts to malfunction, disrupt data delivery for the entire group, or to flood the network with erroneous control traffic. IGMP's security problems exist because the query messages and membership reports that it uses can be spoofed to create inaccurate state about group membership. For example, forged leave messages can cause the generation of numerous bogus query messages from the router. Another example is forged join messages or membership reports that cause a router to create and maintain bogus state. For PIM, forged control messages can be used to alter the structure of multicast distribution trees and also to hamper the data distribution across them. Vulnerabilities in MBGP are mainly due to the susceptibility of the underlying BGP protocol[12]. Finally, vulnerabilities in MSDP exist because control traffic, e.g. Source Active (SA) messages, can easily be spoofed and sent throughout the infrastructure. Because MSDP is responsible for propagating information about all active sources, a bogus MSDP peer can generate a large number of bogus SAs and send these to every single other MSDP peer. This is a very basic DoS attack that can have widespread negative repercussions.

III. Vulnerabilities of MSDP to DoS Attacks

Study of the MSDP DoS vulnerabilities is a critical need in securing the multicast infrastructure. MSDP vulnerabilities stand out because they are very easy to exploit and because the scope of their effects is usually infrastructure-wide. With only a little bit of effort, sources announcements can be blocked, and even worse, MSDP peers and the whole MSDP topology can be made to collapse.

In this section, first, we describe a set of MSDP-based DoS attacks and present the factors that can be varied to generate different, more virulent "strains". Then, we describe an inadvertent attack that was caused by an Internet worm, called the *Ramen Worm*. This attack triggered a large SA storm in the infrastructure. We describe what happened and analyze the effects that it had on the MSDP infrastructure. An understanding of the Ramen Worm sets the stage for the next section where we develop more virulent attacks, and then solutions to these attacks.

A. Operation of MSDP-Based DoS Attacks

All the MSDP-based DoS attacks aim to flood the infrastructure with bogus source announcement messages, called Source Active (SA) messages. A unique vulnerability of MSDP is that any host on an MSDP-enabled network can start an attack with minimal effort and yet with wide-ranging effects in a short period of time. MSDP-based attacks are easy to initiate because of two basic operating behaviors in MSDP: (1) every time a source in a multicast network first sends an IP packet to a new Class D address, an MSDP peer in the domain sends a corresponding SA announcement to all of its peers; and (2) every MSDP peer broadcasts all SAs it receivers to every other connected MSDP peer. While the first operation makes generation of bogus or malicious SAs easy, the second operation enables these SAs to be propagated across the entire infrastructure. Thus, a single host can make its local MSDP peer generate a large number of SA announcements, flood the multicast infrastructure with these announcements, and cause every MSDP peer to endlessly process these messages.

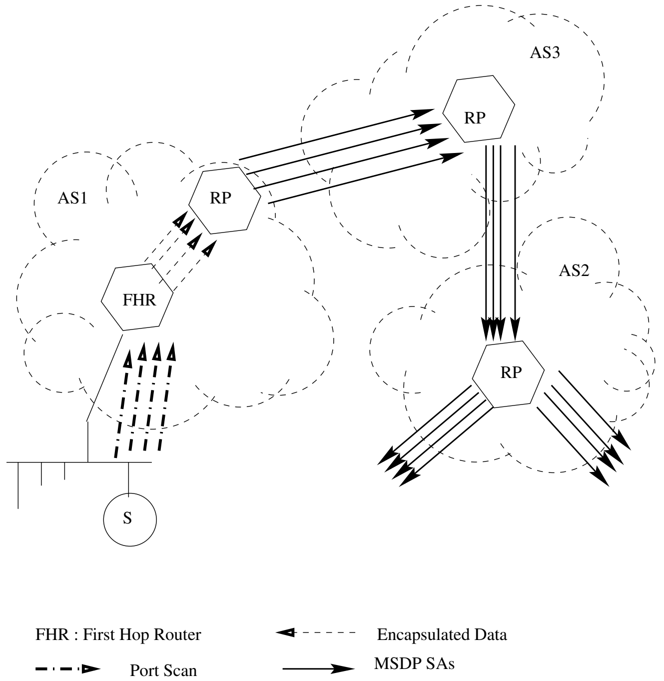

Fig. 2. A Simple MSDP-based DoS attack.

Figure 2 illustrates an instance of this kind of attack. A host can send one packet to each of a large set of Class D addresses in quick succession. Regardless of the size and the contents of these packets, the First Hop Router (FHR) for this host will assume it to be valid multicast traffic, and will forward it to the local RP/MSDP peer. This will cause the MSDP peer to generate one SA for each of the packets (one for each new multicast group). The SA will then be sent throughout the multicast infrastructure.

Several variants of this basic attack are possible. There are three basic characteristics that can be varied to create different kinds of attacks. These include: (1) characteristics of SAs including source address and group address, (2) location where SA message originated, and (3) the rate at which SAs are generated. Different strains of attacks can be created by varying these characteristics. Using different combinations of these characteristics, widely varying SA floods can be created. The kind of SA flood created will affect what happens in the network in terms of where, what, and how severe the result is. The kind of SA flood also affects how the attack can and should be detected and deflected. More details on each characteristics include:

SA Characteristics: The most important requirement for creating a successful SA flood is that each SA is unique. Three fields in an SA message can be varied to create unique announcements. These include:

1) *Source Address*: IP address of the source that sent the original multicast data packet.
2) *Group Address*: Class D multicast group address that this packet was sent to.
3) *RP address*: IP address of the RP/MSDP peer that generated the SA message for the data packet.

Of these three fields, the group address is the easiest for an end host to manipulate. The next easiest is the source address. Through simple IP address spoofing, a malicious host can also vary the contents of this field. Changes in the RP address are also easy, but after often caught by the reverse path forwarding rules used by MSDP peers to prevent unnecessary flooding of valid SAs. The Ramen Worm only exploited the first field, and simply iterated over the entire range of multicast group addresses.

SA Origination Point: Changing the source address of an SA message requires changing the originating host for the original data packet. This can be done by either spoofing the address of another host or involving multiple hosts in the attack. Because an MSDP peers have a rudimentary protection scheme that checks to make sure the host is inside its domain, the range of source addresses that can be spoofed is limited to IP addresses in the RP's domain. Changing the RP address is quite difficult. As mentioned above, a host cannot easily generate a bogus SA with a spoofed RP address. However, a distributed attack can be initiated and many hosts in many different domains can each create bogus SA messages that have different RP addresses.

SA Generation Rate: The most damaging MSDP DoS attack would be the one where a large of number of SAs are generated in a very short period of time. Such attacks cause an instant surge in the number of SA messages. As a result, they are much easier to detect and deflect. A more virulent attack could counter a detection scheme that looks for sudden surges in the number of SA by flooding a slowly increasing number of SAs. Over time the rate of new SAs introduced into the infrastructure can be increased to the point where widespread problems are created. The ultimate goal of such an attack is to make the attack signature look similar to a non-malicious load surge, a surge that might happen because of an external event like September 11, 2001.

B. Ramen Worm

In January 2001 a series of DoS attacks overwhelmed the multicast infrastructure with an unusually large number of SA messages. An Internet worm, called the *Ramen Worm*, triggered these attacks with the simple attack mechanism described above and shown in Figure2. Interestingly, attacks caused by the Ramen Worm were purely accidental. They were not targeted at multicast and, in fact, did not intend to cause a DoS attacks at all. The Ramen Worm was, actually, intending to just port-scan a random set of IP addresses by sending ICMP messages. It triggered the flood of bogus SA messages because part of the address range it scanned belonged to the Class D multicast address range. The activity of the Ramen Worm exposed just how fragile the multicast infrastructure is.

We have been able to gauge the impact of Ramen Worm attacks on the global infrastructure with the help of data collected from multiple routers using our global monitoring infrastructure, Mantra[13], [14], [15]. Furthermore, we have also been able to substantiate our claims that MSDP DoS attacks can rapidly span the entire infrastructure and have serious negative repercussions. Figure3 plots the number of SA messages seen in the MSDP infrastructure in a one month period starting on January 12, 2001. These results are based on the aggregate view of MSDP tables collected from multiple routers at 15-minute intervals. These results show that there were 40 distinct attacks during the data collection period. The number of bogus SAs generated because of these attacks ranged from 10000 to close to 45000. Further investigation reveals that each of these attacks was launched from a different domain.

An interesting analysis can be made by measuring the average length of an attack. Using data logs of Ramen worm attacks, we know that attacking sources do the port scan within two minutes. However, our results show that the average length that an attack lasts is 35 minutes. The reason is that MSDP takes time to propagate SA messages and will continue to do so even long after the attack stops. In essence, the MSDP topology is processing and forwarding the backlog of SA messages until all messages

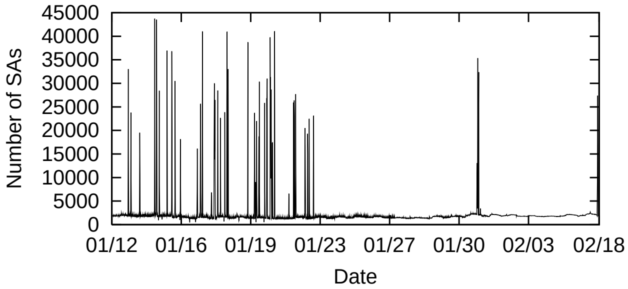

Fig. 3. Effects of Attack Because of Ramen Worm.

have been processed. As a result, even short lived attacks can have a lasting adverse impact on the infrastructure.

IV. Detection and Deflection of MSDP-Based DoS Attacks

A detection and deflection mechanism can be defined as a provision to detect the occurrence of and deter the effects of an attack. The important goals are to detect an attack as soon as possible, to maximize the portion of the attack traffic deflected, and to minimize effects on legitimate traffic. Detection and deflection of MSDP-based attacks work in tandem because an efficient detection mechanism also detects the characteristics of SAs that are the cause of the attack. The knowledge about which SAs are bad and which are good is critical to maximizing the number of bogus SAs deflected, while having a minimal effect on the legitimate SAs.

Two approaches are possible to develop and apply detection and deflection techniques: (1) use provisions in the existing protocol implementation; and (2) develop additional mechanisms that can be deployed either on a link adjacent to an MSDP peer or in the router itself. The first approach involves using existing MSDP router configuration options to limit or filter received SA messages. On the other hand, the second approach provides flexibility to develop new and more powerful detection and deflection techniques and to deploy them in the network. Solutions based on these two approaches are described in the next two sections.

A. Solutions Based on Existing Protocol Capabilities

Existing implementations of MSDP provide configuration options which can be used to alter how MSDP routers accept, store, and propagate SAs. Although the available configuration options may vary slightly by vendor, three general categories exist: (1) SA arrival rate limiting, (2) SA cache size limiting, and (3) SA filtering. These are described in more detail below.

Rate Limiting: Rate limiting is a technique employed by a router to limit the number of messages arriving per unit of time on any of its interfaces. Although rate limiting was originally used as a congestion control technique, it has gained popularity as a counter-measure against MSDP DoS attacks. Rate limiting to prevent DoS attacks can be of the following types: (1) register message rate limiting; and (2) SA rate limiting. *Register message rate limiting* allows an administrator to set a limit on the number of register messages that an RP processes. RP register messages are received from a sender's first hop router when an initial data packet is sent. This technique prevents a DoS attack within the RP's domain. However, an RP employing this mechanism, cannot prevent a DoS attack that originates in some other domain and is propagated to it. This is where the second solution is effective. *SA rate limiting* can be used to prevent DoS attacks originating in a domain other than the RP's domain. The SA rate limiting mechanism sets a threshold on the rate at which an RP can receive SA messages from its MSDP peers. SA messages transmitted to an RP by its peers in excess of the threshold are dropped. The threshold is static and is set by an administrator who uses some knowledge about the normal traffic load.

SA Cache Size Limiting: MSDP peers use an SA cache to store the most recent SAs that they receive. In the event of a DoS attack, these caches can overflow because of bogus SAs. An increase in the cache size will not only result in an increase in the memory requirements of the router, but will also increase the processing overhead to perform cache lookups. Furthermore, most attacks generate SAs far in excess of even a very large cache. To deflect the effects of an attack, an SA cache can be disabled or its size made very small. The biggest drawback of such a techniques is that it negatively affects normal operation. Without a cache, new receivers joining a group only find out about sources when the source's MSDP peer re-advertises the SA. Therefore, if the cache is disabled or if its size is limited, join latency for new receivers hearing about sources can potentially be very large.

Filtering: Filtering is a mechanism that uses Access Control Lists (ACLs) to implement admission control at the RP. It is possible to restrict sources and groups advertised by an MSDP peer by filtering the messages based on IP addresses and prefixes. However, increased administration is needed to keep the filters up to date. Also, filtering is not effective against attacks that use spoofing of either source addresses or group addresses. In reality, filtering is really only effective against the most obvious of attacks.

Although existing solutions are easy to use and deploy, their effectiveness in countering MSDP DoS attacks is limited. The biggest drawback of all of these approaches are that the limits and filters have to be configured statically. Therefore, configurations are tightly bound to "normal" traffic conditions. If traffic conditions change because of some significant external event, static configurations can actually do damage to normal, correct operation. Another drawback is that existing solutions give equal weight to all SAs and do not consider SA characteristics. This eliminates any ability to detect good SAs from bad ones. Because attack deflection involves dropping some SAs, without the ability to know good from bad, an MSDP peer cannot guarantee that only bogus SAs will be dropped. As a result, most existing solutions can only blunt the effects of an attack. They can neither completely stop it nor can they keep good traffic from being lost.

B. Proposed Solutions

The development of new attack detection and deflection solutions provides flexibility and functionality not available in existing techniques. There is added flexibility in terms of where the solutions can be deployed. For example, a new solution could be deployed either on the links connecting MSDP peers or at a peer itself. Deployment on the link will require the installation of hardware for snooping and processing, e.g. a PC. On the other hand, deployment in a router eliminates the need for additional, separate hardware, but requires the router, and possibly the routing protocol, to be modified. Deployment in routers also requires additional processing power.

We have developed three schemes for detecting and deflecting attacks. One goal is to be able to efficiently and quickly detect an attack and then deflect the attack so there are minimal effects on the infrastructure. A second goal is to be able to detect and deflect as many strains of MSDP DoS attacks as possible.

Our schemes evolve to counter almost all types of attacks, and certainly include those described in Section 3. These schemes rely on the dynamic prediction of normal traffic patterns. Predictions act act as dynamic thresholds such that when received announcements exceed the threshold, an attack is assumed. We use a single exponential smoothing equation, a time series equation, for this purpose. This equation relies on a damping factor α which will decide the weight of recent historical values. We further use δ to provide latitude in the predicted value. This permits some variance in the threshold as long as the variance is less than the δ.

The three schemes for detection and deflection of attacks are describe below.

- *Scheme 1 - Prediction of Total Number of SAs Received*: This is the simplest of the three schemes. In this scheme, predictions are made for the total number of SAs that an RP is expected to receive from all of its MSDP-peers. This scheme is based on the same principle as that of *SA Rate limiting*, described above. The difference is that the threshold for our scheme is dynamic instead of static. In fact, once our proposed scheme has detected an attack, it can be deflected using an existing scheme like SA rate limiting. This scheme is ideal for detecting attacks that generate a burst of bogus SAs. However, if the attack is more clever, this scheme does not work well.
- *Scheme 2 - Prediction of Source Specific SA Rates*: This scheme relies on predicting the number of SAs corresponding to *each* multicast source address. An attack is confirmed when the SA rate for one or more sources is more than what is predicted by the threshold. The goal is to immediately identify malicious hosts that are sending bogus SAs such that an attack can be deflected simply by blocking the SAs from that source. These scheme works better than Scheme 1 because filtering is not indiscriminate. If a single SA has initiated an attack, this scheme should filter all the bad SAs and almost no good SAs. The weakness of this scheme though is that it is susceptible to a spoofed source IP address. However, spoofed IP addresses can be easily detected using a new variant of this scheme by instead of building predications for a *single* address, a prediction is developed for a range of addresses. The new variant works because, as discussed previously, an attacker can spoof addresses only belonging to the domain from where that attack is being launched. Once an attack is detected, the bogus SAs can be deflected by using filtering provisions available at the routers. Depending on the variant of the detection scheme used, filters can be either specify a single source address or a prefix range. A weakness of this scheme is that sometimes it is difficult to determine exactly what range of addresses constitutes a single domain.
- *Scheme 3 - Prediction of RP Specific SA Rates*: This scheme has the same working principle as that of Scheme 2. The only difference is that this scheme relies on predicting the number of SAs advertised by *each* RP. Thus, an attack is believed to occur when the number of SAs advertised by an RP is more than the threshold. Using individual RP SA rates is useful when the address space for the originating domain is either large or has diverse sets of prefixes and, hence, Scheme 2 is not effective. SAs corresponding to the attacks detected by this scheme can be deflected again by applying RP-specific SA filters

in MSDP peers. We expect Scheme 3 to be effective even against the most virulent attacks. One such attack is a distributed MSDP DoS attacks where an attacker uses hosts in multiple domains. Even if widely spread, our dynamic prediction algorithms should be able to detect an attack.

The schemes we have developed offer increasing levels of protection against increasingly virulent schemes. The tradeoff is that the stronger the scheme, the more processing and state overhead required. Such is the nature of attacks and defenses. We now turn our attention to evaluating the effectiveness of our schemes.

V. Evaluation

In this section, we evaluate the effectiveness of the attack detection and deflection schemes proposed in Section 4. Our methodology is to generate SA traffic that is representative of various types of attacks and to assess the effectiveness of our various schemes in detecting and deflecting these attacks. We use three metrics to assess the effectiveness of our schemes are: (1) number of good SAs dropped; (2) number of bad SAs dropped; and (3) time to detect an attack. First, we describe the workloads that we have generated for this paper. Next, we evaluate the effective of our detection and deflection against these workloads.

A. SA Workloads

An SA workload is a chronological list of all the SA messages that an MSDP peer receives within a specific time period. We use three types of workloads, each of which represents a different type of attack. Our goal is to generate workloads that are representative of SA traffic generated by an attack. Furthermore, our goal is to develop workloads that represent attacks that are generic to the distance from the MSDP peer in question and the attacker. This is important because workloads for the same attack can vary when seen from the point-of-view of different routers.

The workloads used in this paper are based on data collected over a 24 hour period from a core MSDP-router in the Internet2 backbone. This data set was collected on January 12, 2002, and consists of *real* SA traffic received by the router. Essentially, it is a chronological log of all the SAs that the router had received within the 24-hour time period. We generate our workloads by first generating SA-traffic traces specific to the attacks and from the point-of-view of the router that is being attacked. Then we alter these traces to make them represent the point-of-view of a general RP in the network. We do this by applying an exponential function to the inter-arrival time of SAs in the original trace. Finally, we superimpose these altered traces on the base workload to generate the attack workload.

In the three workloads that we have used in this paper, during normal operation, the number of legitimate SAs is in the range of 6500 to 7500. The number of bad SAs and their distribution over time varies with workload. The variations among workloads originate from the changes in the SA traffic traces for different attacks that we superimpose on them. Two traces vary either in terms of the rate of increase of SAs or in terms of SA characteristics or both. Three workloads and the characteristics of the corresponding attacks are listed below:

Workload 1 - Fixed Source and Varying Group: This workload represents an attack where a burst of SAs are injected into the network in a very short period of time. The source address and the RP address remain the same for all the bogus SAs in the workload. This attack is similar to the *Ramen Worm* described in section III and illustrated in Figure 2. The only difference is that the group addresses of SAs generated by this attack are not incremented by one for each message. Instead, this attack chooses random IP addresses in the Class D range to avoid generating SA traffic where group addresses in the consecutive SAs have the same IP-address prefixes. Figure 4 shows the characteristics of this workload. The attack lasts for three minutes and the number of unique bogus SAs in the workload throughout the interval is about 25000.

Workload 2 - Fixed Source and Varying Group at Slow Rate: The SA characteristics of this workload are the same as that of the SAs in Workload 1, i.e. the addresses for the source and RP remain the same while the group address varies. Even the peak number of unique bogus SAs is the same in both the workloads, i.e. about 25000. The differences are that the attack represented by this workload lasts for about one hour and the rate of SA generation is only 600 SAs per minutes. The peak count of 25000 is achieved by re-advertising all the SAs from one time period again in the next one along with about 600 new bogus messages. Re-advertising the SAs is critical because, otherwise, SAs will time-out. Only the recently received SAs will be maintained in the SA table and the peak count of 25000 will not be reached. Figure 5 shows the characteristics of this workload.

Workload 3 - Varying Source and Varying Group at Slow Rate: Unlike the SA characteristics for Workloads 1 and 2, both the source as well as the group addresses change in Workload 3. Source addresses are randomly chosen from a consecutive address space of (2^{16}) IP addresses. The idea is to restrict the range of IP addresses to

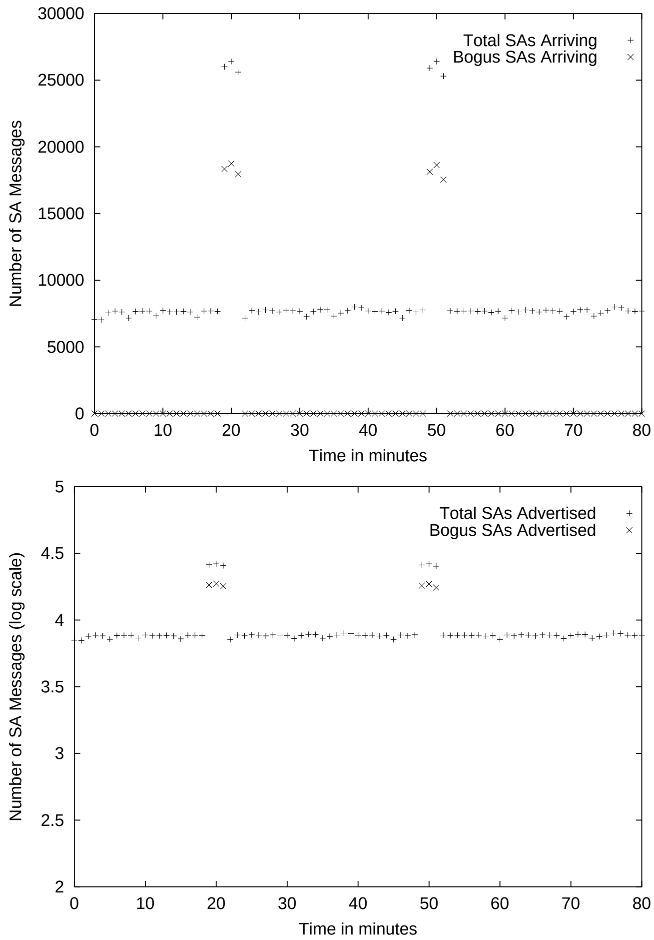

Fig. 4. Workload 1

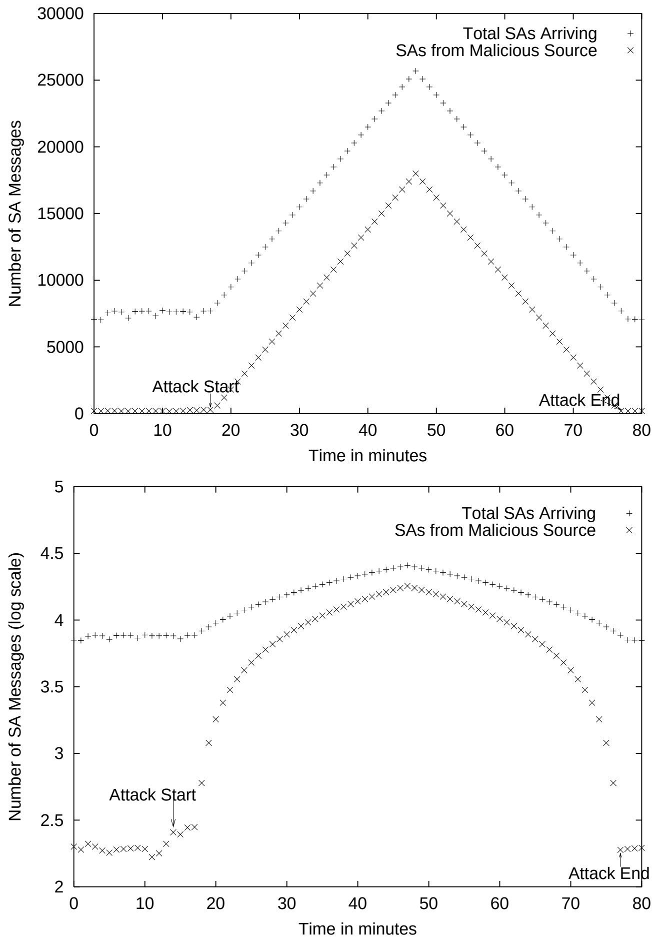

Fig. 5. Workload 2

that of the typical size of a multicast domain. This is important because, as explained before, an RP will advertise only the SAs from its own domain. Figure 6 shows the characteristics of this workload.

B. Results

We assess the effectiveness of schemes, described in Section IV, in detecting and deflecting MSDP-based DoS attacks here. We apply our schemes against the three workloads. Then, we assess its effectiveness on the basis of three metrics, described above. We measure latency in detecting the attack and the number of bogus and legitimate SAs dropped.

Scheme 1 Evaluation: Figure 7 shows scheme 1 detecting the attack represented by workload 1. On the other hand, Figure 8 illustrates how our scheme performs against the metrics discussed above. As evident from these results in these figures, when the attack occurs, the number of SAs arriving at the RP crosses the predicted number of SAs. Therefore, the attack is detected in the first interval in which any bogus SAs are received. As soon as the deflection starts; i.e. the next interval; more than 80% of the bogus SAs are dropped. The most alarming observation is that about 75% of legitimate SAs are dropped as well. This is because, Scheme 1 does not differentiate between legitimate and bogus SAs and drops SAs indiscriminately. This, also, further substantiates our claim that prompt detection is not the only requirement. Intelligent deflection is also very important.

To evaluate the effectiveness of Scheme 1 against other types of attacks we apply it to Workload 2. Figure 9 shows the results and illustrates the failure of Scheme 1 in detecting the attack. The main reason for this failure is that the rate of SAs generated in Workload 2 is very low and, as a result, the predicted value always follows the actual number of SAs received. Therefore, the scheme never detects the attack.

Scheme 2 Evaluation: Figure 10 shows Scheme 2 detecting the attack represented by Workload 2. The results in this figure shows that although Scheme 2 detects the attack, the latency in detection is 4 minutes. Once the number of SAs coming from the malicious source crosses the predicted maximum from the source, all the SAs coming from that source are dropped. As a result, the number

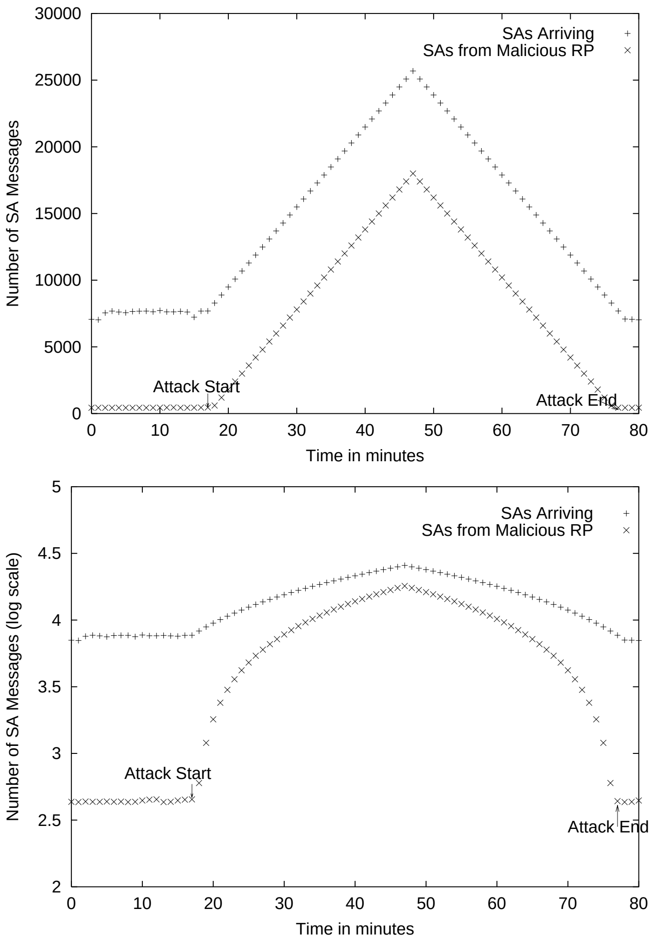

Fig. 6. Workload 3

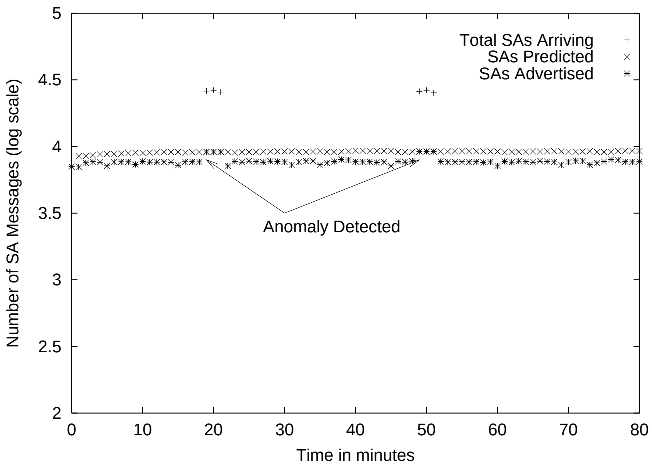

Fig. 7. Scheme 1 detects Workload 1 attack

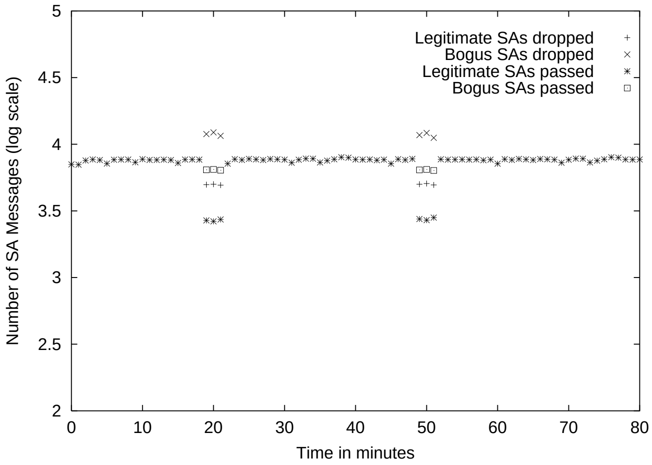

Fig. 8. Evaluation of Scheme 1

of bogus SAs goes down to 0. However, it is important to notice that all the legitimate SAs from this source are being dropped as well. This is one of the main disadvantages of this scheme because it might result in source black-holing; i.e. none of the SAs from this source will be processed or advertised. As Scheme 2 focuses on the source address SAs in detecting the attack, it fails to detect attacks where the source address changes like that in Workload 3. As described above, it is very easy to cause the source addresses in SA to change as long as the addresses are within the range of the domain.

Scheme 3 Evaluation: The operation of Scheme 3 is very similar to that of Scheme 2. The only difference is that while Scheme 2 detects attacks on the basis of per-source prediction, Scheme 3 uses per-RP prediction as the basis. However the number of good SAs to be dropped by Scheme 3 is likely to be more than that for Scheme 2. This is the case because average the number of SAs advertised by an RP is usually more than that for a source. Therefore, while Scheme 2 can result in source black-holing, for this scheme it will be magnified to several more sources. Figure 11 shows Scheme 3 detecting the Workload 3 attack.

C. *"September 11th" Case*

Although we have evaluated our schemes against several attacks, it is worth while to assess the operations of these schemes in scenarios when legitimate usage becomes exceptionally high. One such incident occurred on the September 11th when a popular video stream was available on the Internet only over multicast as unicast infrastructure was down. At this time, as all the receivers sent the feedback through their multicast video clients, several hosts were registered in the MSDP-infrastructure as sources. Figure 12 plots the increase in SAs for this incident as seen by Mantra, our global monitoring tool. [1] For this particular case, scheme 1 and scheme 3 will generated false positives because the usage patterns were higher than predicted. On the contrary, scheme 2 did not falter since the majority of the MSDP messages were generated do the receiver and not the actual source. This re-

[1] Figure shows results based on aggregate view of the MSDP infrastructure as observed after the 9/11 incident.

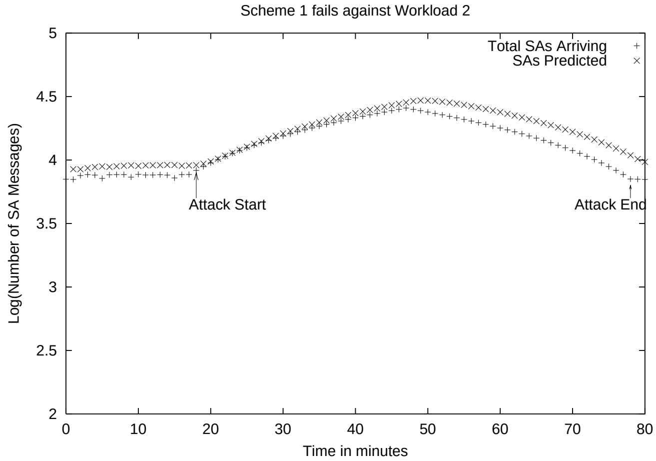

Fig. 9. Scheme 1 fails to detect Workload 2

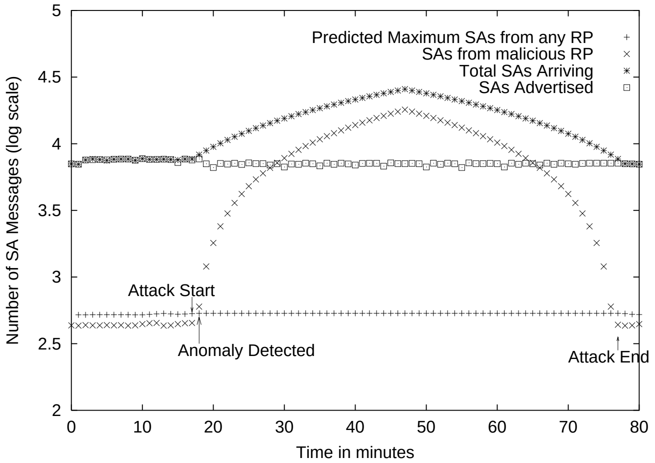

Fig. 11. Scheme 3 detects Workload 3

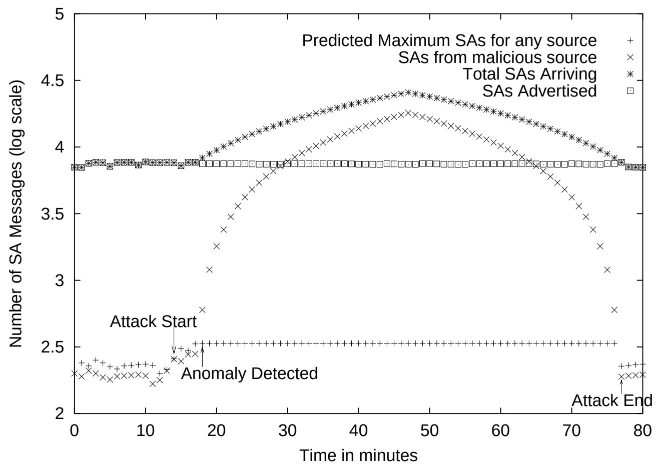

Fig. 10. Scheme 2 detects Workload 2

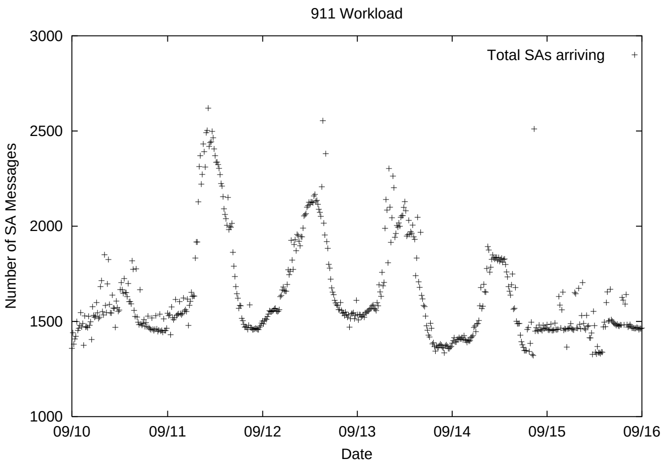

Fig. 12. 9/11/2001 Workload

sulted in the number of MSDP messages being distributed over a large number of MSDP sources there by reducing the number of messages per MSDP source. There could be other usage patterns under which scheme 2 might generate false positives while the other schemes might work.

VI. Conclusions

Denial-of-Service against multicast is a serious enough issue to warrant significant attention. The RAMEN worm attack was not even intended to harm the multicast infrastructure but still resulted in seriously crippling it. As multicast becomes more prevalent, it will be targeted by attacks much more sophisticated than the RAMEN worm.

It has been our focus in this paper to secure the Multicast Source Discovery Protocol (MSDP), a key component in the operation of inter-domain multicast. In doing so, we investigated the RAMEN worm attack, the first known attack against MSDP, to better understand what went wrong. We realized that MSDP is actually quite easy to exploit because of its simplicity and can be lethal because of its flood and prune behavior to exchange messages.

We then discussed existing solutions that have been deployed on the Internet to prevent the MSDP storm attacks. Although these solutions are effective against the simplest of the MSDP storm attacks, they will be ineffective against more virulent strains that have been specifically designed to attack the multicast infrastructure. To counter these virulent strains, we first finger-printed the different possible virulent strains of the RAMEN worm. We then proposed solutions to prevent such virulent strains from affecting the operation of inter-domain multicast. Finally, we simulated the effectiveness of our proposed solutions by creating various workloads and testing them against the proposed solutions.

References

[1] D. Moore, G. Voelker, and S. Savage, "Inferring internet denial-of-service activity.," in *Usenix Security Symposium*, Seattle,

Washington, USA, August 2001.
[2] D.E. Denning, "An intrusion-detection model.," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, February 1987.
[3] D. Meyer and B. Fenner, "Multicast source discovery protocol (MSDP)," Internet Engineering Task Force (IETF), draft-ietf-msdp-spec-*.txt, November 2001.
[4] H. Holbrook and B. Cain, "Source-specific multicast for IP," Internet Engineering Task Force (IETF), draft-ietf-ssm-arch-*.txt, November 2001.
[5] S. Bhattacharyya, C. Diot, L. Giuliano, R. Rockwell, J. Meylor, D. Meyer, G. Shepherd, and B. Haberman, "An overview of source-specific ip multicast(ssm) deployment," Internet Engineering Task Force (IETF), draft-ietf-ssm-overview-*.txt, March 2002.
[6] P. Rajvaidya and K. Almeroth, "Analysis of routing characteristics in the multicast infrastructure," Tech. Rep., University of California, Santa Barbara, July 2002.
[7] K. Almeroth, "The evolution of multicast: From the MBone to inter-domain multicast to Internet2 deployment," *IEEE Network*, January/February 2000.
[8] W. Fenner, "Internet group management protocol, version 2," Internet Engineering Task Force (IETF), RFC 2236, November 1997.
[9] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, G. Liu, and L. Wei, "PIM architecture for wide-area multicast routing," *IEEE/ACM Transactions on Networking*, pp. 153–162, Apr 1996.
[10] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, "Multiprotocol extensions for BGP-4," Internet Engineering Task Force (IETF), RFC 2283, February 1998.
[11] B. Cain, S. Deering, B. Fenner, I. Kouvelas, and A. Thyagarajan, "Internet group management protocol, version 3," Internet Engineering Task Force (IETF), draft-ietf-idmr-igmp-v3-*.txt, May 2002.
[12] H. Chang, S. Wu, and Y. Jou, "Real-time protocol analysis for detecting link-state routing protocol attacks," *ACM Transactions on Information and System Security*, February 2001.
[13] P. Rajvaidya and K. Almeroth, "A router-based technique for monitoring the next-generation of internet multicast protocols," in *International Conference on Parallel Processing (ICPP)*, Valencia, Spain, September 2001.
[14] P. Rajvaidya and K. Almeroth, "A scalable architecture for monitoring and visualizing multicast statistics," in *IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM)*, Austin, Texas, USA, June 2000.
[15] P. Rajvaidya and K. Almeroth, "Building the case for distributed global multicast monitoring," in *Multimedia Computing and Networking (MMCN)*, San Jose, California, USA, January 2002.