An Adaptable Inter-Domain Infrastructure Against DoS Attacks

Georgios Koutepas, Fotis Stamatelopoulos, Vasilis Hatzigiannakis, Basil Maglaris

Abstract— **Denial of Service Attacks have evolved to be one serious threat for Internet activities. Their massive, distributed, and hard to trace nature makes them impossible to be countered by the efforts of a single site. This paper presents an inter-domain infrastructure that aims to coordinate detection and response to such attacks. The main building block of the design is a lightweight software platform installed at each participating domain that provides messaging and alert services and the point of coordinated response control. We describe the operation of this *Cooperative IDS Entity* and focus on its policy control features. The response capability that enables an effective cooperation is adaptable to suit the security policies and needs at each site.**

Index Terms—**DoS, Response, Inter-Domain**

I. Introduction

Since their first appearances, Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks have evolved in complexity and sophistication and pose a serious threat to network connected systems. Their distinguishing characteristic is that they do not attempt to break into the target computer systems, like other more conventional attacks, but rather aim in the disruption of normal operations down to their complete halt. Possible targets are either high profile machines (web, transaction servers etc.), which are overwhelmed by legitimate, but resource consuming requests, or network connections congested by many incoming packets. The sources of the attacks are usually trojaned computer systems, which when instructed will initiate the flow of malicious traffic. Although small in scale and difficult to detect near the sources, the flows have a cumulative devastating effect when they reach their target.

Management-wise, DoS (and especially Distributed DoS) attacks present an interesting challenge since their nature makes them difficult to stop by the efforts of a single site. Factors that contribute to this are (a) the practice of attackers to spoof packet source IPs, (b) the possibility of the attack initiating from a wide range of networks worldwide, and (c) the inability of a domain to enforce incoming traffic shaping; detected malicious flows can be blocked locally but the assistance of the upstream network is needed in order to free the resources occupied by them on the incoming link. Consequently, any effective response procedure requires cooperation between sites. The attack characteristics have to be determined locally and communicated to the network(s) on the attack path. Currently, this is a manual, non-automatic and time-consuming procedure. It depends on the administrator's availability at the upstream network, his readiness to help, as well as the service policies in power.

According to the site's security policies the various actions that will be implemented usually consist of setting up tailor-made blocking or throttling filters on active network components. The implemented filters require monitoring for adjusting to shifting attack patterns and deactivation when the attack is over. Still, no matter how effective this response will be, the bandwidth penalty is still present throughout all the affected domains. To alleviate the resulting congestion extra steps must be taken and contacts be made between the sites on the attack path, to further resolve the situation. Obviously the further we move from the victim, the more dispersed this procedure becomes and there is less immediate interest from the domains to help.

Though capable of attack identification and early warning, conventional Intrusion Detection (ID) Systems usually cannot offer domain traversing and scalable response capabilities. One extra limitation is that, although there are solutions for message exchange between them (most notably IETF's Intrusion Detection Working Group [9]), they lack the underlying cooperation framework to transmit notifications or response directives to another domain. Even if this would be possible, security concerns would deny any direct manipulation of the necessary networking equipment. Furthermore, the implementation would have to identify the remote devices' topology and interfaces to issue appropriate commands.

In summary, the requirements for an effective response to a DoS attack are: (a) Early detection both at the victim site and at upstream stages, (b) flow of incident information between domains, effective and timely domain cooperation but according to each domain's policies (c) quick, automatic, and effective response in as many domains on the attack path as possible, and (d) avoiding extra network overloading due to these communications [8]. In our approach we assume that

Manuscript received November 15, 2002.

G. Koutepas (phone: +30-210-772.1448; fax: +30-210-772.1448), F. Stamatelopoulos V. Hatzigiannakis, and B. Maglaris are with the Network Management & Optimal Design Laboratory, Electrical & Computer Engineering Department, National Technical University of Athens, Zografou, GR 157 80, Athens, Greece. (e-mail: {gkoutep, fotis, vhatzi, maglaris}@netmode.ntua.gr).

DoS attacks exhibit method uniformity. This is one of the factors that allow reaction to the attack in the form of filters in its path(s).

In the work presented in this paper we attempt to fulfill all the above requirements by introducing a framework for effective counter-DDoS cooperation between domains[1]. We propose the *Cooperative IDS Entity*, a software system (installed on each domain) that constitutes a trusted peering point within the proposed distributed framework.

The IDS Entities are deployed on top of the local IDS hierarchy[2] receiving messages from it and exchanging messages with their peers on other domains. Special measures are taken to ensure the integrity and security of these exchanges. Furthermore each Entity has a limited local response capability. The inference engine of the IDS Entity combines local and remote information about on-going security events and responds to them locally according to administrator-defined policies.

In this work we focus on the communication and procedural part for the effective operation of the proposed framework. Section 2 presents briefly the setup of the framework and the main components of the infrastructure. Section 3 describes the usage of multicast as the main transport method and the messages exchanged between the IDS Entities. In Section 4 we put the infrastructure in perspective by describing how it reacts to a DoS attack. Section 6 discusses other approaches similar to ours. Finally in Section 6 we make a synopsis of the framework and present some next steps of testing and extending the infrastructure.

II. The Architecture and Components

A. Overview

Intercepting and controlling a DDoS attack requires actions, along the attack path, on as many networks as possible. The combined macroscopic action against the attack comprises of the individual network reactions. Each of these has different security and access policies due to the different administration. We define each network under the same security authority as the basic operational block participating in our infrastructure. That is the individual *Domain* in our design. In each participating domain one (or more) IDS Entities are deployed. Within the proposed framework these Entities play the role of the trusted point of presence and communication peer.

Locally the same Entities are viewed as part of the trusted internal network, under the direct control of the administrator. They provide the *enabling* medium for participating in the trans-domain cooperation. Each Entity aggregates security notifications from its peers and from the local IDS hierarchy, combines the data to track down possible on-going events, and concludes on their flow characteristics, source and destination within the local topology. The output and discovery process parameters of the IDS Entity are configured by the local administrator. Thus, he receives information suited to his own awareness needs, rather than having to screen arbitrary notifications. Additionally, should it be verified that the domain is in the path of the attack, the Entity has limited response capabilities, namely the automatic configuration of security access lists on strategically positioned active network components. A sufficient number of reacting Entities along the path of the attack will lessen its effects and minimize the overall lost bandwidth.

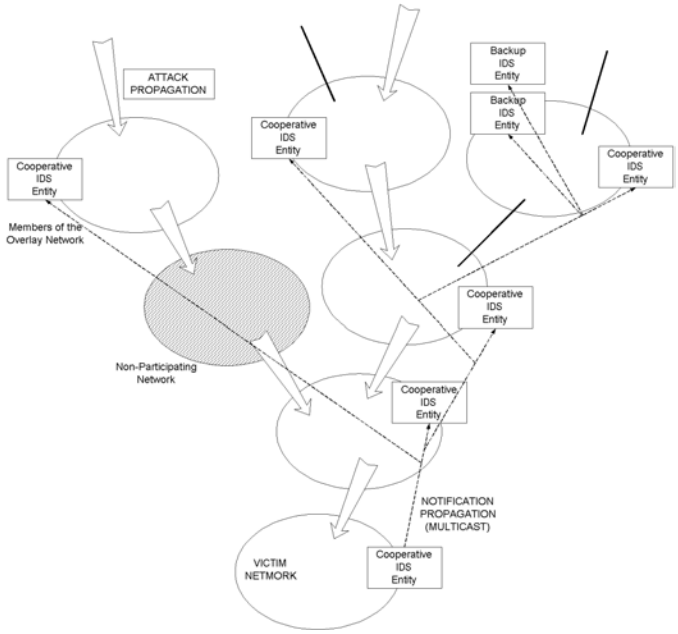

Fig. 1: Example of the infrastructure layout and activation under a DDos attack

The proposed Cooperating IDS Entities constitute an *overlay* early warning network that allows the quick propagation of attack notifications as well as the automatic and simultaneous response all along the attack path. This overlay network unifies the different IDS technologies deployed within each domain; the resulting IDS infrastructure operates as a dispersed network of cooperating detection sensors. An overview of the Entity deployment across various networks can be seen in Figure 1.

Security sensitive administrators will be quick to point out the risk associated with an automated unit controlling network components. To address these concerns there are several safeguards in the design. Firstly, the configuration changes are time restricted and generate notifications to the management console. Secondly, although the entity interacts with its peers, it is solely the administrator that defines the exact details of the local response, i.e. which network component to reconfigure, in what way, for how long, and under what conditions. Entity configuration is performed through editing the response policy database and the communications filtering control file.

[1] By the term "domain" we refer to a network or group of networks being under the same administrative authority. Our work thus refers to cooperation between different managerial realms.

[2] This refers either to single monolithic or to fully distributed and/or hierarchical ID Systems

B. The Cooperative IDS Entities

The building block of the proposed framework, the Cooperative IDS Entity, is a flexible and portable software system. Each Entity is installed in a central point within the domain, possibly near the management console, where the IDS notifications from various points are collected. Its software architecture is lightweight and modular. In the prototype we have based the implementation on the Java framework for achieving portability and platform independence. The internal structure of the Entity and its software architecture are shown in Figure 2.

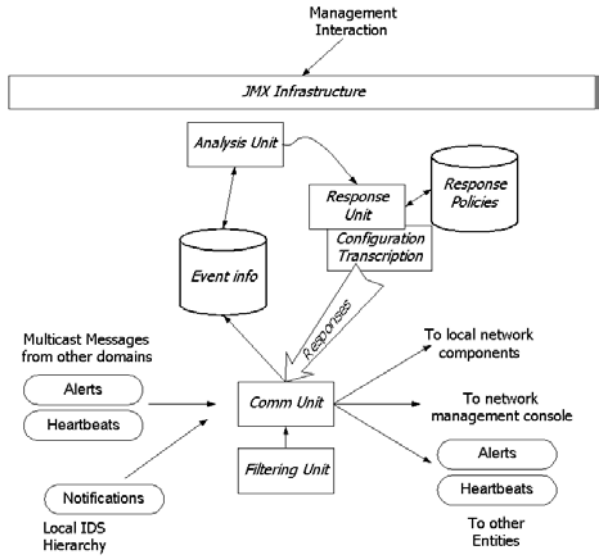

Fig. 2: The software architecture of the IDS Entity

Currently the main components are the following:

1) The Communications Unit attends to all the communication needs, receiving messages either from the local IDS or from remote peers and sending notifications when instructed. It is managed by an access control list to prevent incoming over-flooding or to stop specific outgoing messages when the administrator does not desire this.

2) The Communication Filtering Unit arranges the access control mentioned above and offers an extra regulating layer in the design should it be decided to be administered by a different individual than the main manager that controls the overall Entity's operation.

3) The Analysis Unit is the inference engine for correlating an ongoing event with signs received from the local sensor network. Its operation and its sensitivity can be configured using manager set thresholds, like the number of messages from a source, the search time window etc. The administrator can define combined discovery queries that correlate local and remote events and lead to discoveries.

4) The Response Unit is where discovered events from the previous stage are associated with policy actions. The actions defined may be notifications to the management console, messages to other domains, escalating to direct response upon specified (critical for effecting pass-through attack traffic) networking equipment. The Entity does not choose by itself the components to act upon and does not even need to have a mapping of the domain's network topology. The actions are pre-configured in the policy file. The same policy file determines which pieces of information will be transmitter outside the network to the peers. The process of configuring these policies can be unified with the presentation of the network on the management console.

5) The Configuration Transcription component is the only part that may differ between domains since its purpose is to translate the generalized configuration directives produced from the Response Unit to commands suitable for the local networking equipment in operation.

Each unit is an MBean and the Java Management Extensions (JMX) [13] provides the communication facilities between them and the management console. Through a secure HTTP/SSL interface the administrator may change the operational parameters of each module. The JMX infrastructure allows the modular management and control of the various units: they can be installed, activated, or deactivated, at run time. The design also permits the easy addition of new modules for potential extended functionality.

III. Operation and Communications

With the proposed architecture and its entities we have established an operational infrastructure, the conceptual *Overlay Network*. The effectiveness of this network and its added value is derived by the combined and coordinated action of multiple domains when dealing with an attack. To this ends the right, enabling communication methods and operational procedures have to be defined. We have chosen to base our communications on the low footprint multicast methods with additional (unacknowledged) unicast UDP messages for special purposes.

A. The protocols

One of the main problems encountered in every cooperative IDS scheme is how to prevent the alert notifications from escalating and causing extra traffic in a network already overloaded by an attack. The usage of multicast as the transport method solves this drawback and offers multiple other advantages like the opportunity to group domains following their physical or administrative separation. Additionally, we believe that the nature of multicast communication makes it less susceptible to attacks (lack of fixed target).

Among other communication specifics the participating domains agree on a particular multicast group of which they will all be members, use for exchanging their messages and therefore will be routing through their network (or arranging as tunnels through third non-participating and non-multicast supporting domains). The main characteristics of multicast communications we utilize are:

1) Independence from a specific installation host. The IDS Entity can be deployed on any host reachable by multicast

within the domain. Its functionality can also be (automatically or manually) transferred to a backup host if the main system becomes disabled for any reason.

2) Stealthy presence. Since the Entity may operate and communicate from any host it does not have to reveal its exact location, preventing some attacks. We view the exchanges between domains as potential weak points, for exposing the nodes of the infrastructure to attackers. Many times these exchanges may be traversing non-trusted networks. We make a note here that the Intrusion Detection systems operating within the domain also require to transmit to the IDS Entity, but this (a) is a risk associated with the danger of the local ID systems becoming compromised and (b) can be solved in the same way by putting these to transmit on a local multicast channel, if this is feasible by their firmware.

3) One-to-many flexibility. Except from being an obvious bandwidth saving feature, this offers the opportunity to deploy in parallel many *backup* IDS Entities on he receiving end, even in different subnets of the domain, where they will be less vulnerable to a single attack threat. These will be keeping the same operational state with the active one and take its role if it gets disabled. Thus, we avoid a single point of failure.

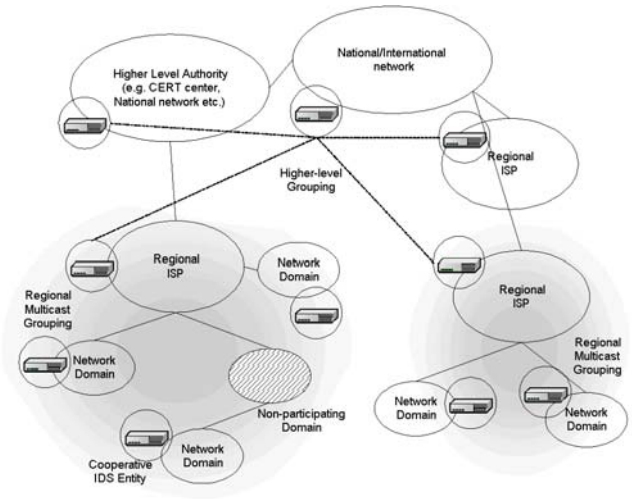

Fig. 3: Multicast grouping of two levels

4) Clustering through usage of multicast groups. In a widely deployed infrastructure, it is possible, instead of keeping a single grouping, where everybody is talking with everyone, to arrange smaller *clusters* of domains. This may be done using various multicast transmission addresses, a different one for each cluster, and/or specific (low) TTL on the messages to avoid their propagation beyond a given radius. Clustering of the domains can be arranged according to physical network layout, geographical, national or other factors and localize the response actions and exchange of messages. This is not unlike the paradigm of "areas of responsibility" between CERT teams. Second level clusters can propagate security notices of non-local events further and escalate the response procedure when needing to do so. An example of such grouping is shown in Figure 3.

We have also chosen to implement unicast UDP messaging on occasions that we have to traverse non-multicast supporting networks or when it is necessary to have point-to-point contacts. Our design proposes unicast communications only as an extra feature, not necessary in the general case.

B. Message Structure

Our approach in message definition is based upon the IDMEF specification. The main purpose is the easy interoperation of our system with any message exchange system that would be integrated now or in the future within IDS products. This obviously makes the deployment of the framework within an already operating Intrusion Detection environment easier.

Under the IDMEF specification, messages carry their information in XML and are of two types: heartbeats and alerts. The UML description of the messages is the direct derivative of the corresponding IDMEF specification [10].

We proceed now to describe the main characteristics of the messages in our design, especially the ones that differ from IDMEF. The reader should compare the message structure with that of the original IDMEF specification in [10].

1) Heartbeats

These are regularly timed periodic indications on the health and operational status of each Entity. Although the entities do not have to reveal their exact location these messages give a macroscopic view of the operation of the infrastructure with the participating domains. Possible stoppage of these heartbeats will be an indication of problems at the sending Entity and will be considered with all the rest of the security information.

The frequency of the outgoing heartbeats is defined by the domain's administrator, but in the general case it should be agreed upon together with all the rest of the cooperation specifics. On example heartbeat is presented in Figure 4.

```
<Heartbeat ident="Everything fine">
<Node ident="5555"> NTUA ids entity</Node>
<CreateTime ident="Internet">2002.04.01 AD at
03:44:56 PM EEST</CreateTime>
</Heartbeat>
```

Fig. 4: A typical heartbeat message

2) Alerts

These may be coming from the local IDS hierarchy (as defined in the specification) or from Entities of other domains. The extension made to IDMEF is the separation of Entity messages (where there is the requirement of minimum information disclosure) from the local IDS hierarchy messages where the maximum amount of information is required to specify the local security condition. A sample alert, coming from an IDS Entity is seen in Figure 5. The notification may include multiple indications on the source of the attack.

```
<Alert ident="Dos Attack">
<CreateTime>  2002.04.01 AD at 03:44:56 PM
EEST</CreateTime>
<Entity entityid="panoptis" domaincode:ntua>
</Entity>
<Source       ident="3434"       spoofed="yes"
interface="le0">
<Node ident="45656">
<Address ident="ip-v4"> 147.102.13.13</Address>
</Node>
</Source>
<Source       ident="5656"       spoofed="no"
interface="le0">
<Node ident="mama">
<Address ident="ip-v4"> 147.102.13.16</Address>
</Node>
</Source>
<Target ident="34" decoy="no" interface="le0"
port="80">
<Node ident="45656">
<Address ident="ip-v4">147.102.50.13</Address>
</Node>
</Target>
<Classification
origin="1">12345</Classification>
</Alert>
```

Fig. 5: A typical alert message, coming from a peer Entity

C. *Security*

The architecture poses various security questions, both in the infrastructure and locally.

We have to assume the possibility of eavesdropping on the multicast communications since they take place over the public network and in some occasions may cross non-cooperating networks; the attacker orchestrating a DoS attack could "tune into" the right group and listen for signs his attack has been detected and actions are taken against it. Then he may direct the hostile machines to alter the pattern of malicious flows hoping to elude any newly installed filters, or he may even initiate "whack-a-mole[3]" patterns. Although secure multicast solutions that have been proposed are not completely mature yet, we overcome their limitations organizationally. The high-level exchanges between the (limited in numbers) framework participants, allow for symmetrical (single key) cryptography as an acceptable encryption method. The domains' administrators can arrange periodical off-line key exchanges, or entrust the duty of key generation and distribution to one of them, or even to a trusted third party like a CERT team.

An attacker may also attempt to generate spoofed or duplicated messages. Bogus alerts describing non-existent events could initiate filter configurations to hinder legitimate traffic, resulting to internal DoS. Of course, the Analysis Unit as described earlier does not have to depend on a single information source to reach a conclusion and each incoming message is assigned a trust factor. Still, to avoid such attacks, each entrusted IDS Entity digitally signs its messages and includes a time stamp[4]. The accompanying digital signature is verified against pre-exchanged public keys.

Another concern is that the IDS Entities may constitute a single point of failure for each domain to participate to the infrastructure. Their securing through obscurity is a single measure that cannot ensure their complete safety. So, this problem is addressed by making the design lightweight, modular and portable. The current solution we offer requires the transfer of the Entity to another machine in the event of a malicious or accidental failure. The multicast methods also allow the parallel operation of "backup" units, ready to take over. We are currently researching the method to achieve this transition transparently for the operating infrastructure.

Finally, it is expected that any DoS attack will have its effect upon normal network communications and consequently to the cooperative scheme. Usually, in such attacks leaf networks may be cut off completely. The rest if the cooperative infrastructure is capable to operate even when some of its members are off-line. The suspension of communications with a part of the infrastructure will only serve (through the stoppage of heartbeat messages) to raise awareness about that part of the network.

IV. Operation and Scalability of the Framework

A. *Response to an attack*

The start of the detection procedure may take place simultaneously at many different domains that the DoS attack traverses. Obviously the network that is the target of the attack will experience immediate problems with its available bandwidth. However, several intelligent network intrusion detection systems are also able to notice the process of the attack along its way, using anomaly detection or by simply monitoring the changes in the numbers of flows and packets. The closer a network is to the attack victim (if it's on the attack path) the strongest anomaly it will sense in its readings.

The immediate reaction of the victim will be to notify, by a multicast-one-to-many message, its local group of cooperative peers. Each domain that will receive this message will combine it with its local IDS readings indicating an attack flow going through and will conclude (with variable sensitivity – it's manager configured) on the presence on an ongoing attack. Even if the victim domain is completely disabled there is the possibility that some, unacknowledged UDP notifications will manage to pass on the outgoing channel. Even if this cannot happen the other domains will sooner or later diagnose the anomalous situation because of the stoppage of heartbeat messages from the victim. In either case they will have enough evidence to combine with their own IDS readings to reach the conclusion of an attack. The need for warning from two (or more) sources serves as a safety against false positives. It will also ensure that measures will be taken only along the attack path and not elsewhere. Of

[3] This term describes DoS attacks highly variable in their sources and activity, where essentially new sources start as current ones are stopped. The result is a difficult to counter fluctuating malicious flow.

[4] We presume that the Entities will be time-synchronized using ntp or other method.

course, in any case the whole list of notifications is fully available to the manager through the provided interface.

Following the deduction of an attack event (the procedure of decision implies that if such an event is diagnosed, then it concerns the domain) the instructions set up on the policy file are followed. These will certainly include notification to the network management console and possibly dispatching an alert message to the multicast group peers if one (about the same event) has not already been sent by another Entity. The other possible action specified on the policy file is to set up filters throttling the traffic having the particular attack characteristics. The filters will be set at the networking equipment closest to the victim and at the domain's incoming traffic points to minimize lost internal bandwidth. In this case of active response, another notification is sent to the management console and a time restriction, stated in the policy file, controls the time span of the intervention.

B. Notification Escalation

This procedure will result in all the members of the "regional" group of domains to be notified and the ones that are on the attack path to take whatever actions are dictated by their enterprise policies. The domain at the edge of the group will also be member of the higher-level group (see Figure 3). A combination (a) the attack sources and (b) the local reaction policies may have as a result the decision to further escalate the notification procedure and transmit the attack characteristics to the corresponding higher-level multicast group. There, the procedure of deducting the attack path will be repeated and possibly further escalated, or specialized to a specific lower-level group where action may also be taken against the attack.

C. Organizational Matters

It is obvious that this response algorithm closely matches the procedure currently followed manually when dealing with a DoS attack (or some times even conventional attacks). In our design we have tried to overcome the obstacles of (a) slow reaction, (b) during-the-event cooperation efforts, (c) unavailability or inappropriate reaction policies and procedures, and (d) administrative, language, or other barriers preventing effective action. Our design attempts to automate this process and arrange beforehand for all matters that may come up during counter-attack actions. Organizationally the grouping of domains and the propagation of notifications on a need-to basis mimics current CERT or other security teams' style of operation that use areas of responsibility or "constituencies".

On the part of the actual inter-domain agreements, we believe that peering agreements could offer a somewhat analogue of such a situation. Many domains are already discussing security matters between them and certainly make efforts to cooperate during on-going events. The next step could be to decide their participation on the proposed framework that will enhance their preparedness without requiring considerable effort or resources, that could be adjusted to fit their enterprise policies, and where all participants have equal position. Again CERT teams could play a central role in the coordination of the cooperation and as a trusted party for key exchanges, dispute resolving etc.

V. Related Work

The Cooperative Intrusion Traceback and Response (CITRA) framework [1], [2], is work close to ours. It uses the concept of community (administrative domains) and all of them organized in neighborhoods. It detects intrusions at the low level at each community, focusing at the boundaries. The detectors distribute attack reports to their neighbors who can then trace the attack path and initiate responses to the intrusion. The communications employ device independent response directives and use centralized reporting and coordination. The communications take place with the Intruder Detection and Isolation Protocol (IDIP). This approach also employs some features similar to ours like response policies and the possibility of using multicast. The main differences of our proposal are that (a) we operate on a higher level and focus in automation and acceleration of enterprise cooperation, (b) in the implementation part we use the functionality of the multicast method as the prime element of our architecture, (c) we have based our implementation in the IDMEF protocol for a more standard and easy integration with existing components, and (d) we take a liberal approach to domain participation, not depending in the seamless integration of every one to achieve an effective solution.

In [3], [4] and [5], J. Ioannidis, et al. present their solution of controlling the high bandwidth aggregates that comprise a DDoS attack. Once they have established the aggregates of the DDoS attack they move on to block them at the routers using tailored filters. They communicate their findings between cooperating routers using the special Pushback protocol. As a result they trace malicious aggregates closer to their source and control their bandwidth allocation. The difference in our work is the higher level approach to the problem. We focus our effort in providing a cooperation-enabling infrastructure for domains and we are independent from the Intrusion Detection part, although we are coupling our prototype with a router auditing tool to provide DoS identification. ID is a matter open to the choices of the individual domains. Another significant difference of the Pushback approach is that it requires administrative access to each individual router, which could cause problems when expanded across domains. Furthermore, tracing back on a low-level requires the collaboration of every intermediate device in the path from the victim closer to the source.

The Indra system [12], also utilizes multicast communications for distributed Intrusion Detection and Response operations but this approach within a single domain. Specifically, IDS hosts that have identified an attack attempt, using secure multicast messages, communicate the characteristics to other nodes in the network notifying them in

advance. Having faced similar problems for the securing and authenticating multicast messages they have developed a secure messaging API offering cryptography.

Finally many concepts discussed here are also present in [6], where the prerequisites for forming a cooperative Intrusion Detection framework are discussed. Although that work does not deal with the problems of countering Denial of Service Attacks or usage of multicast communications for cooperation purposes, there are many similarities with our approach in the elements that form a secure and effective basis for cooperation, like the trust concerns between domains etc.

VI. Conlusions and Future Work

We consider that the Cooperative IDS Entity and the cooperation infrastructure it provides can offer a new way for responding to security events in general and specifically to DoS attacks. Our system performs in the fields of message dispatch, alert correlation between domains and active coordinated response. The main points of the design are:

1) High-speed and automated response to DoS attacks, performed in parallel along the attack path.

2) The creation of an overlay network. We attempt to achieve an effective solution without necessarily the full cooperation of all networking infrastructure. Attack lessening results may be acquired even with a limited number, of geographically distributed participants, if the attack spans their domains.

3) The IDS Entities offer policy controlled communications and response thus they can fit in variable enterprise environments. Although the domain participates in a cooperation infrastructure, the type of response and the amount of its "openness" is the local administrator's prerogative. The domain remains "independent" with its policy choices.

4) Flexibility of the multicast methods employed. There is no single point of failure since the ability to use easily interchangeable (and "hidden") nodes within a domain is provided. Additionally there is low message overload and easy escalation of the events when needed, mimicking security organizations operation.

We are currently integrating the IDS Entity with an open-source DoS detection tool called Panoptis [14]. The particular tool detects such attacks by measuring anomalies in the number of flows and packets in border routers and has been proven to offer acceptable detection results [7]. We plan to evaluate the design in a simulated trans-domain environment, by replaying older attacks recorded on our university's network and by scripting attacks described in the bibliography. The results will help us fine-tune the analysis process to an acceptable false positive to event discovery ratio. We examine also the possibility of a limited deployment in our academic network, once its operational safety has been established. Further enhancements in the design include the parallel operation of more than one IDS Entities and the transparent transition to a new one, should the primary become disabled.

Acknowledgment

G. Koutepas would like to thank P. Christias for his valuable help and G. Androulidakis for pointing out some mistakes in the original version of this paper.

References

[1] D. Sterne, K. Djahandari, B. Wilson, B. Babson, D. Schnackenberg, H. Holliday, and T. Reid, "Autonomic Response to Distributed Denial of Service Attacks," In Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection, RAID 2001, Davis, CA, USA, October 2001, pp.134-149, Springer-Verlag.

[2] D. Schnackenberg, K. Djahandari, and D. Sterne, "Infrastructure for Intrusion Detection and Responce," In Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX II), Hilton Head, CS, January 2000.

[3] J. Ioannidis and S. Bellovin, "Pushback: Router-Based Defense Against DDoS Attacks", NDSS, February 2002.

[4] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling High Bandwidth Aggregates in the Network", draft, February 2001.

[5] S. Floyd, S. Bellovin, J. Ioannidis, R. Mahajan, V. Paxson and S. Shenker, "Aggregate-Based Congestion Control and Pushback," ACIRI Annual Review, December 2000.

[6] J. McConnel, et al., "A Framework for Cooperative Intrusion Detection", 21st National Information Systems Security Conference, Arlington, VA, USA, Oct 5-8, 1998.

[7] C. Kotsokalis, D.Kalogeras, and B. Maglaris, "Router-Based Detection of DoS and DDoS Attacks", HP OpenView University association (HPOVUA) Conference '01, Berlin, Germany, June 2001.

[8] G. Koutepas, F. Stamatelopoulos, and B. Maglaris, "Efficiency and Performance Issues in Distributed Intrusion Detection Systems", Applied Telecommunication Symposium 2002 (ATS 02), San Diego, CA, USA, April 2002.

[9] Intrusion Detection Working Group (IDWG) of the IETF, http://www.ietf.org/html.charters/idwg-charter.html

[10] D. Curry and H. Debar, "Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition", Internet Draft, June 20, 2002.

[11] D. Moore, G. Voelker, and S Savage, "Inferring Internet Denial-of-Service Activity", 10th Usenix Security Symposium, August 2001.

[12] Q. Zhang and R. Janakiraman, "Indra: A Distributed Approach to Network Intrusion Detection and Prevention", Washington University Technical Report # WUCS-01-30, 2001.

[13] "Java Management Extensions Instrumentation and Agent Specification, v1.0", Sun Microsystems, July 2000.

[14] Panoptis home page. http://panoptis.sourceforge.net/