# Design and Operational Characteristics of a Distributed Cooperative Infrastructure against DDoS Attacks

Georgios Koutepas, Fotis Stamatelopoulos, Vasilios Hatziyannakis, and Basil Maglaris
Network Management & Optimal Design Laboratory
Electrical & Computer Engineering Department
National Technical University of Athens
9 Iroon Polytechneiou str., Zografou, GR 157 80, Athens, Greece
{gkoutep, fotis, vhatzi, maglaris}@netmode.ntua.gr

**Abstract**:

In this paper we present an inter-domain cooperative infrastructure against Distributed Denial of Service (DDoS) Attacks. The infrastructure is established between the networks that choose to participate. A software system, the *Cooperative IDS Entity*, is deployed at each participating domain. The main operational characteristics of this Entity and of the infrastructure as a whole are presented and a number of parameters that influence DDoS discovery and reaction efficiency are discussed. We also examine an operation scenario on actual topologies and attempt to demonstrate the validity of the concept.

Keywords: Anti-DDoS, distributed infrastructure, cooperative reaction.

## 1. Introduction

Distributed Denial of Service (DDoS) attacks have evolved in complexity, sophistication and intensity to the degree that they pose a serious and continuous threat to the modern IT infrastructure. Denial of Service has started with small scale attacks (sometimes even using a single packet), aiming to disrupt the normal operation of critical systems through exploitation of software or protocol vulnerabilities. The target is not the system itself but its ability to offer useful services. The next step was the utilization of the world-wide Internet connectivity to launch massive, resource consumption (especially available bandwidth) attacks against the victim sites. The method described in this paper is designed to counter such bandwidth consuming DDoS attacks. A series of events on high profile commercial targets in February of 2000 (NIPC 2000) marked the issue as a serious and threatening problem, able to influence even very powerful systems or high bandwidth networks. More recently, some companies had to completely suspend operations due to continuous interruption to their Internet connectivity (The Register 2002). DDoS attacks have made the headlines as recently as March of 2003 (InfoWorld 2003).

In a typical DDoS attack the malicious users are using hijacked computers of any size, capabilities and geographical distribution. These attack sources, when instructed, will initiate the flow of malicious traffic against the victim. Although small in scale and difficult to detect near the sources, the flows have a cumulative devastating effect when they reach their target. To conceal the malicious traffic origins address spoofing is used on the attack packets. The attacks may also be staged in a number of control and "amplification" levels to further intensify their effects.

Management-wise, DDoS attacks present an interesting challenge because of 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 still needed in order to free the bandwidth occupied on the incoming link. Consequently, any effective response procedure requires cooperation between sites. When attempting to counter a DDoS, the event must be

communicated to networks on the attack path in order to take appropriate measures. Currently, this is a manual, non-automatic and time consuming procedure. It heavily depends on the administrator's availability and good will, as well as the service policies at the upstream networks. According each 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. Still, no matter how effective the response of the upstream provider will be for freeing the victim lines the bandwidth penalty is present all along the attack path. Extra steps must be taken and contacts be made between these networks to alleviate the problem. The further we move from the victim, the more dispersed this procedure becomes and there is less immediate interest from the domains to help. The findings of a drill on network attack preparedness in May of 2003 (GCN 2003), show that cooperation between networks is the only solution for stopping such attacks.

In summary, the requirements for an effective response to a DDoS attack are: (a) early detection at the victim site and at as many upstream domains as possible, (b) the ability to exchange incident information between domains without inflicting significant network load (Koutepas et al. 2002), (c) timely and effective response in as many domains on the attack path as possible.

Our work aims to satisfy the above requirements by introducing a framework for countering DDoS attacks through the cooperation across different networks. The proposed architecture is built around the concept of the *Cooperative Counter-DDoS Entity*. This is a modular software platform that is to be deployed within each participating domain and offers communication and response coordination services. Our approach allows different administrative realms to cooperate during a DDoS attack as a group of trusted partners, without loosing any authority within their own domain or compromising their security. Within the trusted community, each Entity will try to detect its position on the attack path (possible source, transient node, target, or not on path) without having to perform traceback procedures (a major problem in DDoS analysis). It can then react accordingly and in parallel to the other Entities community-wide. The Entity is viewed as the gateway to a trusted overlay network, where all members share security notices (IDS alerts) and automated assistance (active filtering) against DDoS attacks. The Entity behaviour within this trusted overlay network is guided by the local policy (simple rule-based system) which defines what information to share and how to react to shared security information.

The rest of the paper is organized as follows: Section 2 presents the architecture of the Cooperative Framework, outlines the cooperative operation and discusses a number of organizational aspects. Section 3 discusses the internal operation of the Entity. Section 4 shows Entity and Framework operation during an attack scenario. Our progress with testing and validating the prototype is also presented. Section 5 compares our work with other efforts in the field. Finally, Section 6 presents our conclusions on our approach and its benefits.

## 2. Architecture and Deployment

### 2.1. Infrastructure deployment

The solution offered in this paper suggests a "community" of trusted partners. The Cooperative Counter-DDoS Entity is assigned to each participating network domain. It comprises the medium of communication and the instrument of response functionality, for the Framework of cooperative domains. At each domain it is the receiving point for messages

about the security of the community members as well as the operational status of neighbouring Entities. It also transmits info about local security events and assessments. Locally the Entity acts as the highest-level node of any operating IDS hierarchy. Using info from peer nodes and the local ID Systems the Entity evaluates the security status and infers about the presence of an on-going attack.

The messages exchanged by the Entities are composed according to the, XML based, Intrusion Detection Message Exchange Format (IDMEF). The IETF is developing this protocol, currently in draft state, to enable cooperation between Intrusion Detection Systems (Curry et al. 2003). Following the IDMEF paradigm messages exchanged between Entities are divided in two categories: Heartbeats and Alerts. Heartbeats are periodic notifications that a node is in operation, with connectivity to the Framework. Alerts are sent by Entities when they detect and identify security events.

The second important function of the Entity is to offer limited response capability. This departs from the usual IDS informational-only functions. Our design goal is to automate DDoS response, even to a minimum extent. The Entity is fitted with the capability, after concluding on the security event evaluation, to interact with local network components in a limited and tightly controlled manner. The intended action is the temporary configuration of DDoS attack suppressing filters that will fit the characteristics of the specific security event. Security sensitive administrators will be quick to point the risk associated with an automated unit configuring network components. To answer these concerns there are several safeguards in the design. Firstly, the changes are made temporarily, with notifications sent to the management console. Secondly, through the Entity's policy configuration the administrator can exactly define the effects of these changes to the network.

### 2.2. Framework Operation

Typical Entity deployment and the Cooperative Framework operation are shown in Figure 1. During a developing attack the flows of malicious traffic target a network, passing through other domains. DDoS discovery may occur at any of the domains that the attack traverses depending on the detection methods employed and the sensitivity of the ID systems. If the attack is not discovered before, the target network will experience immediate problems with its available bandwidth. This will prompt the local Entity to transmit an Alert or, if loosing connectivity, will result in the stop of Heartbeat messages.

Each Entity that will receive this Alert (or fail to receive the Heartbeat) will combine this information with its local IDS readings to establish whether an attack flow is going through its domain. The combination of arriving Alert notifications about a discovered on-going event with the stop of Heartbeats from parts of the network will cause the Framework Entities to pass to Alerted states. The closer a network is to the victim (if it's on the attack path) the strongest irregularity it will sense in its IDS readings that will produce more local IDS Alerts. Consequently the Framework Entities will become activated with higher probability as we come closer to the victim. This also ensures that measures will be taken mainly along the attack path where the Alerts will be more persistent. If the network is indeed a passage way for the attack then, according to the response policies, filters may automatically be set at the border routers and reduce the malicious traffic effects.

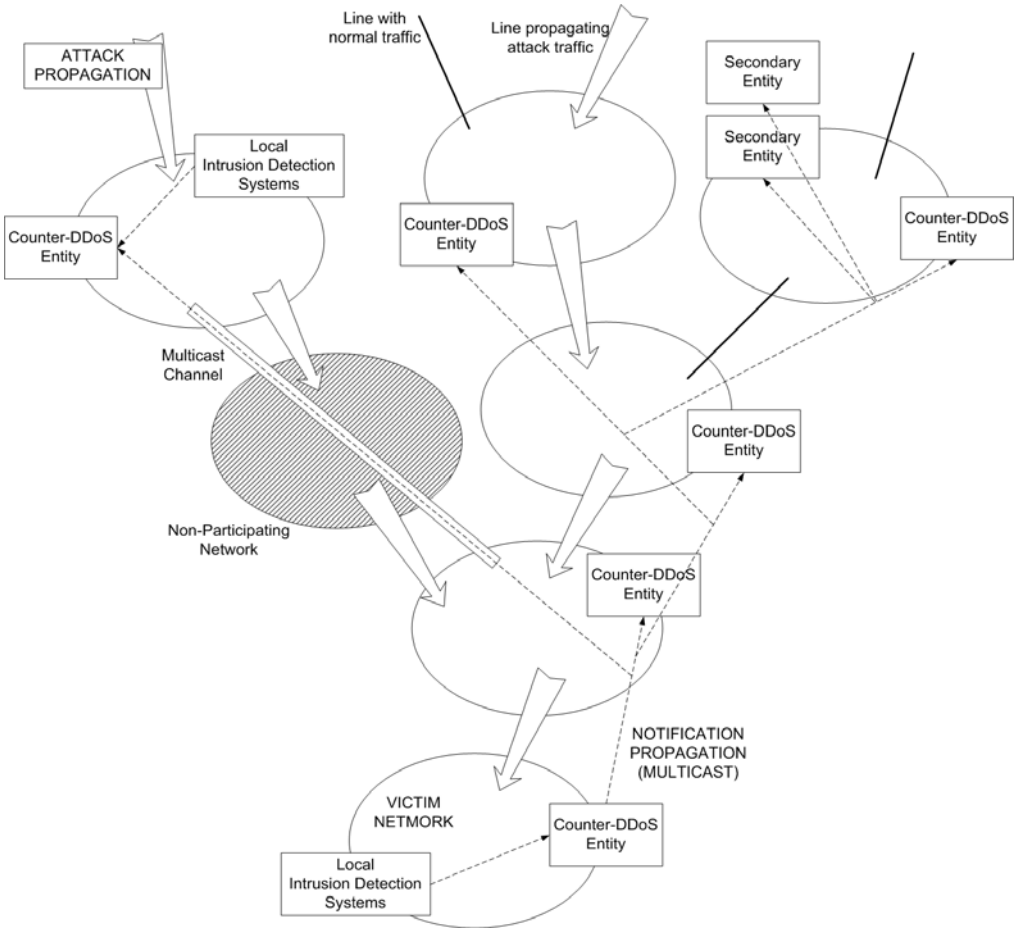


**Figure 1: Typical Entity deployment and reaction process during an attack**

### 2.3. Community Organization

Enrolling domains in an agreement for cooperation and the propagation of response actions mimics Computer Emergency Response Teams' (CERT) style of operation: each CERT has an area of responsibility ("constituency") and cooperates with other such teams, considered trusted partners. Similarly, in our approach, the community of the cooperating domains is set up off-line and all managerial details (encryption keys, notification frequency, etc.) are arranged before the proposed architecture is deployed. Many domains are already discussing security matters between them and make efforts to cooperate. Building a cooperating community consisting of limited number of domains does not require global consensus and Internet steering bodies' approval like in the case of Internet-Firewalls (Wan et al. 2002). The established credibility of CERT teams can play an auxiliary role in the arrangements as a trusted party for key exchanges, dispute resolving etc.

### 2.4. Communications within the Framework

One of the main problems encountered in every cooperative scheme is how to prevent communications between nodes from escalating and causing extra traffic in a network already overloaded by an attack. In our approach we have chosen Multicast as the transport method because it solves this drawback and offers some extra organizational advantages. The Entities

depend on Multicast connectivity existing between the sites. At each domain there is a limited and well specified number of subscribers to the service, so Multicast Tree maintenance (inactive branch pruning etc.) is de-facto simpler than the standard on-off client registration. Domains that are not participating in the Framework and not supporting this type of communications can be bypassed by tunnels. The multicast inter-domain connectivity of the Cooperative Framework can be built using any of the currently available methods (Almeroth 2000) according to what is more effective and efficient for the partner networks.

Having a Multicast communication base makes it possible to organize and scale the architecture by the use of protocol groups and variable TTL values. Messages exchanged can be distributed to neighboring nodes within a certain radius using different TTL values. The domains participating in the anti-DDoS Framework agree preliminary on the communication specifics (or arranging tunnels through other, non-participating and non-multicast supporting domains).

An important aspect of multicast is that the Entities can maintain a "stealthy" presence. After registering with the local multicast router they can passively listen to notifications and transmit "to the group" without revealing their real address. Likewise, many similar systems, even on different subnets of the domain, can work in parallel, monitor the same multicast groups, with all of them maintaining the same state and being prepared for fail-over procedures. Dispersing of the Entities reduces the possibility of a single attack impairing all of them simultaneously.

### 2.5. Security Considerations

An attacker orchestrating a DDoS attack could "tune into" the right Multicast group and listen for signs of detection and response communications. It is possible then to direct the hostile machines to a new pattern of malicious traffic eluding any newly installed filters. Another concern is that fake alert messages describing non-existent events could initiate Entity responses to hinder legitimate traffic. These security concerns indicate that Entity communications need to be secured.

Communication security in our system uses symmetric (single key) encryption, message signing and time-stamping. The Cooperative Framework requires exchanges between limited numbers of participants so symmetrical cryptography is realizable with periodical off-line key exchanges. To avoid message spoofing or duplication, each entrusted Entity also digitally signs its messages and includes a time stamp (the Entities are time-synchronized).
The Entities support the operation of the whole cooperative scheme, so they can be considered critical points of failure. We have tried to address this concern by making the design lightweight, modular and portable. In the current solution the Entity is easily transferable to another machine in the event of a malicious or accidental failure. As mentioned above, it is also possible for a secondary Entity to operate in parallel with automatic fail-over when the main one stops performing. This weakness, however, benefits the detection procedure. Failure to transmit "Heartbeat" messages at regular time intervals indicates a possible security event. Heartbeats not received are one of the parameters considered in the assessing algorithm of the Entity inference engine.
Finally we have to consider the situations that the Multicast backbone itself may come under attack. This could certainly affect the connectivity of the whole overlay network. Although this is a real weakness of the networking infrastructure it presumes existing router vulnerabilities and well informed attackers that will specifically target them. We also

investigate alternative methods of communication that offer the same functionality. Peer-to-peer exchanges and application level multicast are two promising candidates.

# 3. Entity Operation

## 3.1. Entity States

The various Entity operational states and transitions between them are presented in Figure 2 and are the following:

Down: Not operating or the administrator has suspended the communications module. It is the only state that the Entity is not transmitting Heartbeats.

Normal: stays in this state while it is normally receiving Heartbeats and has not received any Alert messages.

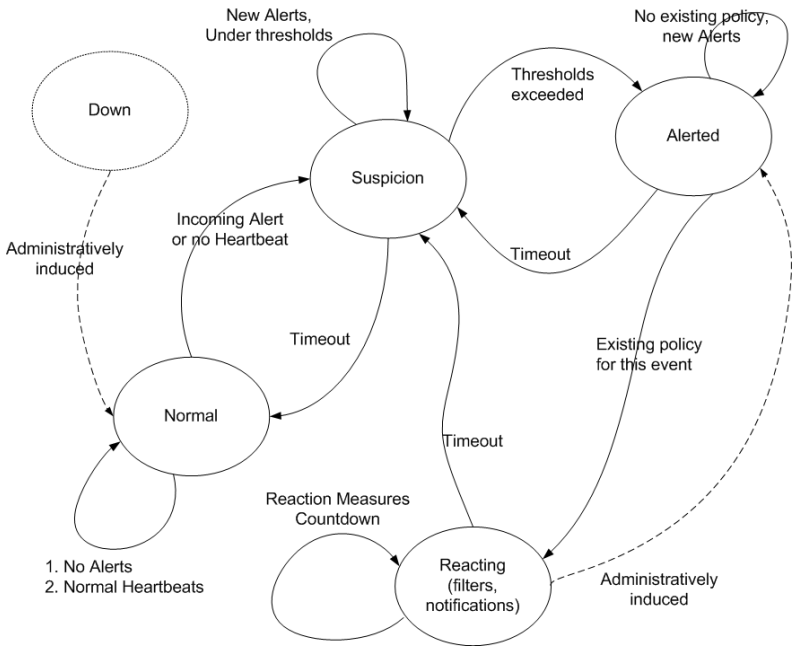


**Figure 2: State Transitions of the Counter-DDoS Entity**

Suspicion: The Entity comes to this state because of a number of occurrences; either it fails to receive a Heartbeat, or has received an Alert. The latter may be from another Entity or from local IDS sources. Each distinct attack registered at the Entity is receiving an exclusive Counter. All incoming Alerts are examined for matching any on-going event. A positive match will increase the corresponding Counter.

Local IDS messages can (if the administrator decides it) have a stronger effect on the counter increase. This way they can be treated as higher-value, more trusted than remote Alerts. This feature is useful in networks that become targets of DDoS attacks, to accelerate state changes and broadcast Alerts for community reaction. Additionally, we expect a higher probability of local IDS notifications in cases that an attack is traversing the domain.

Continuing failure to receive Heartbeats does not increase any of the counters. If new Alerts have not been received for a certain period the Entity returns to Normal operation.

Alerted: the Entity passes to this state if any attack Counter exceeds a set threshold. Upon reaching this state a notification is sent to the management console and the policy rules are searched for entries matching the current attack characteristics. If a corresponding policy entry is not found then this state is retained for a period of time. The countdown is renewed if during this period another similar Alert arrives. After timeout the Entity returns to the Suspicion state and the corresponding Counter set to the value of Alert threshold-1.

Reacting: we pass to this state if there is a policy entry matching the detected event. This state is similar to the Alerted state but signifies that reaction measures are being taken. If the manager monitors the state of the Entity he can know immediately whether an action is currently being taken. A reaction timeout in this state is specified in the policy entry and is equivalent to the time that any actions will be applied to network equipment. Once the timeout is reached we return to the Suspicion state, with the counter set to a value lower that the Alert threshold. New Alerts coming while the entity is in the Reacting state don't affect the countdown to reaction timeout. This way of operation aims to prevent the Entity from staying in this "active" state indefinitely in the case that Alerts keep coming. The Entity may also pass manually back to the Alerted state by the manager which would prompt re-reading the policy and applying any newer rules.

The administrator sets a number of parameters that configure the described operation of the Entity through the state transition process.

### 3.2. Software Architecture

The software architecture of the Cooperative Counter-DDoS Entity is lightweight and modular. In the prototype we have based the implementation on the Java framework for portability and platform independence. The software architecture of the Entity with message flows is shown in Figure 3.

The design is comprised of a number of independent modules joined under the administrative features of the Java Management Extensions (JMX) API (Sun Microsystems 2002). Each unit is implemented as an MBean. JMX provides the communication facilities between them and the management infrastructure. A number of different interfaces are available for access to the modules, e.g. a secure HTTP/SSL connection and SNMP. The administrator can change the configuration parameters of whole Entity or control each module individually. The JMX infrastructure allows a number of management actions upon the modules: they can be installed, activated, or deactivated, at run time without affecting the whole Entity.

#### *3.2.1. The Communications Unit*

It handles Multicast messaging and incoming message parsing. Messages arriving may be Heartbeats or Alerts. They are unencrypted and then checked for validity by signature and timestamp. The IDMEF XML payload passes through a parser. The data, including date info, is stored in the Event Database where it is available to the Analysis Engine and to the manager for further examination. Heartbeats are renewing their time registration in the Database. The Communications Unit is also broadcasting local Heartbeat messages (with the Entity's characteristic identification) within the defined periodic intervals. It also transmits appropriately composed Alert messages using data sent by the Analysis Unit.

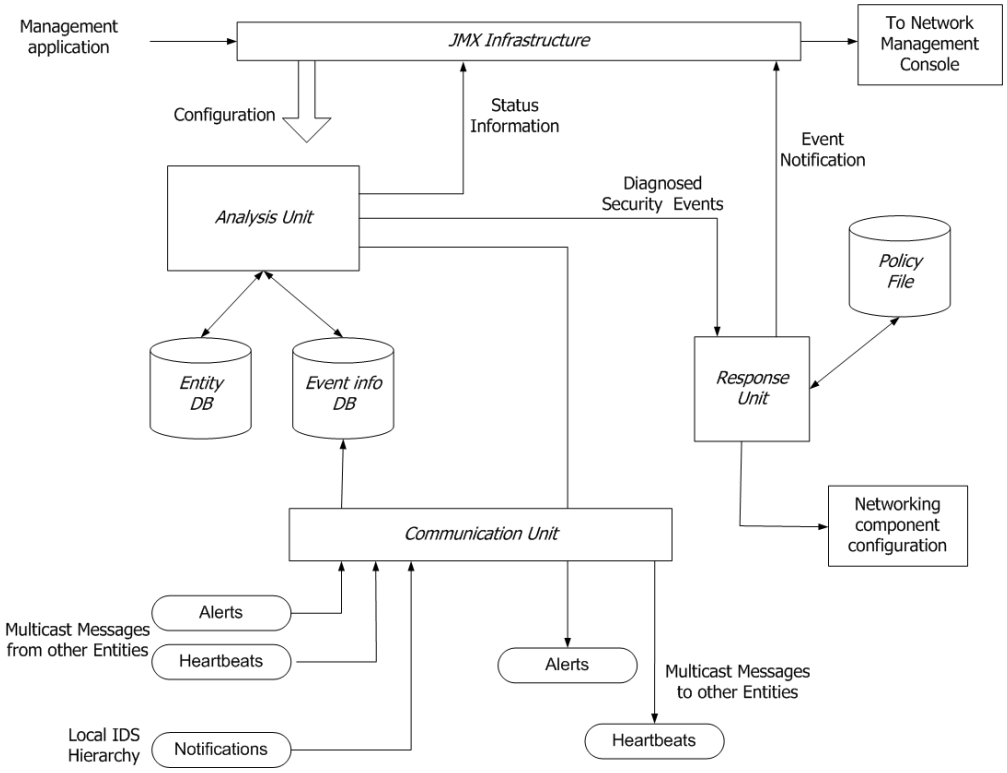


**Figure 3: The Software Architecture of the Entity**

*3.2.2. The Analysis Unit*

This is the inference engine for correlating event reports from local and Framework sensors. Its operation and its sensitivity are configured by a number of manager set parameters. The Analysis Unit is responsible for initiating outgoing Alerts. We have tried to avoid Alert replication that would result in traffic buildup. Alerts are produced from the local domain IDS notifications under the following conditions:
1. The Entity must be in the Alerted (or Reacting) state to send an Alert.
2. The notifications must be about a (D)DoS attack
3a. Either the Entity has come to the Alerted state exclusively by local IDS notifications, or
3b. Local notifications about an event must arrive at a rate higher than a set threshold even if the Entity has received external Alerts about it. The quick accumulation of local notifications indicates the higher possibility the domain to be on the attack path.

*3.2.3. The Response Unit*

In this Unit events are checked against existing entity Policies. The received Alerts are stored in the Events database All the Alerts are identified as belonging to the same attack by the combination of Event Type and final destination (derived from the Alert messages received). Examining this table the Entity inference engine can conclude on the positioning of the domain relative to the DDoS attack path as one of the following cases.
(1) It is either the source of the attack or on its path
(2) It is on the attack's path
(3) It is the target of the attack, or
(4) It is out of the attack path

Once a particular event has raised the Entity's state to Alerted, the Response Unit searches the database entries referring to it and establishes the positioning of the local domain on the attack path, as described before. The Case can be any of (1), (2), (3), or (4), Cases (1)&(2), or

Cases (2)&(3). This attribute, the attack type, and attack destination are used for matching the event with a configured policy entry.

The policy entry refers to the type of action that will be taken. Currently this can be traffic blocking or throttling/shaping. These traffic restrictions referring to the type of attack traffic and its destination will be applied on domain border routers. Each policy entry that refers to an action also includes the period of time it will be in force. The Entity does not have to identify the components to act upon and does not even need to have the topology of the domain's internal network.

Additionally, the Response Unit undertakes messaging to the Management Console (through JMX) each time the entity reaches the Alerted state irrespective to any reaction measures.

## 4. Operation Scenario and Validation

### 4.1. A Typical Operation Example

In order to better illustrate the operation of the Entity and the Framework in general we present a hypothetical deployment on a typical network and the reaction to a DDoS attack scenario. The topology is shown in Figure 4.

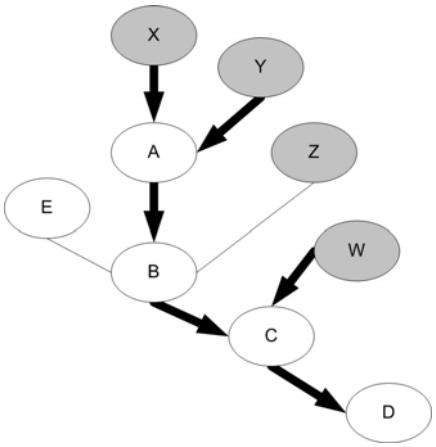


**Figure 4: The Example Network**

The network of the example is comprised of a number of domains (networks of distinct administrative authority) and the interconnections between them. We have based this network architecture on the actual topology of the Greek Academic Network (GRNET), interconnecting the country's educational and research institutions with the core European research network and the Internet. From the Trans-European interconnection network to the last leaf node there is only a small number of intermediary domains.

Each white oval (A, B, C, D, and E) represents a domain participating in the Cooperative Framework, with an Entity operating within. The gray ovals (W, X, Y, and Z) represent non participating domains. In the example we examine the scenario of an on-going DDoS attack targeting one of the leaf domains, D. The DDoS attack is highly dispersed, arriving simultaneously from three different domains (networks X, Y, and W). We will look into the operation and reaction of the Entity in the central Domain B. Domain B is similar in its connectivity with the core GRNET domain. As it is seen in the diagram, successful reaction measures on B would result in the control of 2 sources of the malicious traffic that the end

victim has now way of knowing. One last assumption we make is that there is no local IDS operating in B or it has been configured to low sensitivity in order to avoid false positives. This assumption is useful for showing that the effectiveness of the Framework is to a great extend independent of the individual IDSes. Our effort has been to compensate for the lower IDS sensitivity and accuracy by combining, through Framework interoperation, data from many sources.

The Entities on A, C and D will send, through multicast, their findings to the community. A typical message from A is shown in Box I. For an explanation of the XML elements the reader should refer to the current IDMEF draft (Curry et al. 2003). The only element used in our design that is an extension of the standard is *AdditionalData* that shows the "next-hop Domain". This is deduced from IDS attack traffic flow analysis.

```
<Alert ident="67890">
<Analyzer analyzerid="Domain_A_Entity-123"></Analyzer>

<CreateTime ntpstamp="0x12344321.0x87655678">2003-04-01T03:45:57,03+02:00
</CreateTime>

<Source ident="X">
<Service>
<Portlist>25, 80</Portlist>
</Service>
</Source>

<Source ident="Y">
<Service>
<Portlist>25, 80</Portlist>
</Service>
</Source>

<Target ident="D">
<Node>
<Address category="7">147.102.13.10</Address>
</Node>
</Target>

<Classification origin="1">SYN Attack</Classification>
<AdditionalData type="8", meaning="Description">DoS</AdditionalData>
<AdditionalData type="8", meaning="Next-Hop Domain">B</AdditionalData>

</Alert>
```

**Box I: A Typical Alert Message**

After receiving a number of Alerts the Entity at the domain B moves to the Alerted State as described in 3.1. The message information store that refers to the particular attack is then examined. This data is shown in Table I.

**Table I: Representation of Information Referring to a Particular Event in the Database**

| | Alert Sender | Source Domain | Target Domain | Next-hop Domain | Event Type |
|---|---|---|---|---|---|
| 1 | A | X | D | B | 1 (Using SYN packets) |
| 2 | A | Y | D | B | 1 |
| 3 | C | B | D | D | 1 |
| 4 | D | C | D | None | 1 |

Based on this data the Analysis Unit of the Entity deduces the following:
- The type of the Attack is a DDoS attack using SYN packets
- The final target is domain D
- Domain B is on the attack's path (table lines 1 and 2) – Path Case (2)
- Domain B may be the source or on the attack's path (line 3) – Path Case (1)
- The positioning of the domain relative to the attack path is a combination of Cases (1 & 2).

Corresponding policy entries should match a number or all of the specific attack's characteristics (Type, Destination, and Path Route) in order to yield results. A typical policy line is shown in Table II.

**Table II: Policy Entries Matching the Characteristics of the Example Attack**

| Matching Part | | | Reaction Part | |
|---|---|---|---|---|
| Destination | Attack Type | Path Case | Action | Duration |
| D | DDoS packet type (*) | 1&2 | a. Throttle traffic 25%<br>b. Coming from source domain that gives Path Case 1<br>c. Packet Type the one derived from messages, Dest. D | 600 sec |
| * | DDoS packet type (*) | 1&2 | a. Throttle traffic 50%<br>b. Outgoing to the direction of target domain<br>c. Packet Type the one derived from messages, Dest. the target domain | 200 sec |

The combination of information stored from Alert messages with the knowledge of the neighbour domain interconnection topology makes it possible to direct the reaction to the appropriate ingress or egress routers. The Path-Case can be used to decide if the actions should be taken on the incoming or outgoing traffic of the network. This gives us the privilege of two policy implementation points. According to the domain's positioning relative to the attack path the appropriate reaction point is chosen in an effort to optimize filtering and minimize effects on "innocent" traffic.

The end result of this reaction process would be for Domain B in the example to set up traffic throttling on the routers that connect it with domain A (point of incoming traffic) and with domain C (outgoing traffic in the direction of the target).

### 4.2. Prototype Tests

We have implemented a prototype Entity and we are evaluating its operation within a testing network. Rather that deploying it on an operational network we have a set up a test environment comprised of a number of interconnected machines representing (through address and route manipulation) a group of interconnected domains. We have verified the stable operation of Entities communicating between them and assembling the Cooperative Framework. We have also observed the correct propagation of multicast Alerts through the architecture and the changes in the Entity states. The IDS messages that are supposed to trigger Entity operation are so far provided by a scripting facility that allows us to simulate various rates of detection success in the attack scenarios tried. The Entities that reach the

Alerted state are reacting according to policy rules but their reaction results are only stored in a log file that allows us to trace the combined Entity action timeline. So far we have been able to verify the speed and accuracy of the Framework operation an indication that one of our design goals, a better reaction speed over manual procedures, has been achieved.

The next step of the tests will be to integrate a WAN emulation facility between the Entities (with "Dummynet" (Rizzo 2001) being the strongest candidate), for simulating actual attacks and parallel traffic load. IDS messages will again be provided by the scripting facility and experiments made with various detection factors. The Entity reactions will be tested directly on the effect they will have on the attack traffic. The two metrics used for our testing will be (a) accuracy in setting up filtering that matches the attack patterns and is implemented on the right network points and (b) effects on normal (non attack) traffic. This testing procedure will allow us to fine-tune Entity operation and arrive to a number of standard configurations and policy rules that will be effective for countering present attacks.

## 5. Related Work

The Global Defense Infrastructure (GDI) proposed by K. Wan and R. Chang in (Wan et al. 2002) and (Wan 2001) is an approach against DDoS attacks most close to ours. In their proposal Minimally and Fully configured Local Detection Systems (LDSes) are to be placed at various strategic locations in all the Internet, like Network Access Points, etc. constituting the Distributed Attack Detection System (DAD). The distributed architecture is supported by messaging between the LDSes. The LDSes have modular design, with each one of the modules (for traffic analysis, attack detection, attack response etc.) occupying a different computer system. The FLDS perform misuse and anomaly based DDoS detection based both on their own findings and indicative alerts coming from neighboring systems. Suspicious attack alerts are communicated within the GDI using a reliable flooding mechanism. The messages are formed in IDMEF. Once attack detection has been established they install rate-limiting filters at the Internet core router they are connected to. MLDSes are a lighter implementation of the system used only for installing rate-limiting anti-DDoS filters based on information they receive from FLDSes.

Although there are many similarities in the design and operation of GDI with our approach (IDMEF messages, modular systems, communicating findings in the community, and setting up rate-limiting filters) the two solutions are different in their management goals, scale of deployment, and detection methods. The Cooperative Framework constitutes a system for the management and coordination of anti-DDoS efforts within small and trusted groupings of domains. Another difference is that the Entity used in our design is part of the domain's administrative realm, permitting high level of trust and configuration according to local policies. We do not perform any detection on our own but rather rely on the findings of the local ID Systems at each site. It could be argueed that the different IDSes would result in non-uniform detection rates in the Framework. We believe that the different approaches used by the deployed IDSes compensate for this handicap by allowing a variety of approaches in the detection procedure. Alert exchanges make all the Infrastructure's members aware of their findings.

The Cooperative Intrusion Traceback and Response (CITRA) framework (Sterne et al. 2001), (Schnackenberg et al. 2000) is also work close to ours. It uses the concept of communities (administrative domains) and all of them organized in neighborhoods. At each community it performs low-level intrusion detection, focusing at the boundaries. The detectors distribute

attack reports to their neighbors who can then trace the attack path and initiate intrusion responses. CITRA employs device independent response directives and uses centralized reporting and coordination. The communications take place using the Intruder Detection and Isolation Protocol (IDIP). This approach also has a number of 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 communication 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 standardized and easy integration with existing and future IDS 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 (Ioannidis et al. 2002), J. Ioannidis, et al. present their solution of controlling the high bandwidth aggregates that comprise a DDoS attack. The approach utilizes routers both for attack detection and response. DDoS attack traffic aggregates are established by monitoring discarded (overflow) traffic at router level that includes the attack packets. The system then moves on to block these traffic aggregates at the routers using tailored filters. The findings are communicated between cooperating routers using the special Pushback protocol. As a result the malicious aggregates are traced step-by-step closer to their sources and their bandwidth allocation controlled. 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 components. Another significant difference of the Pushback approach is that it requires administrative access to each individual router, which causes problems when expanded further that the border of a domain. Furthermore, tracing back on a low-level requires the collaboration of every intermediate device in the path from the victim on the way to the source. In our approach we have tried to minimize and overcome the problem of non-participating domains.

Finally many concepts discussed here are also present in (Frincke et al. 1998), 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 building an overlay network, 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. Finally, a number of future tests will be conducted through the large-scale experimental deployment within the Greek Research Network.

## 6. Conclusion

We presented a distributed framework that introduces a cooperative inter-domain approach in countering the DDoS problem. We also presented the implementation details of a proof-of-concept prototype (based on open and accepted standards) and discussed its operation. Finally, we provided an overview of related work and compared it to our approach.

The operation of our approach relies on building a “community” of trusted partners, each deploying a local software Entity. Entities exchange security information (Heartbeat and Alert messages) so that inclusion in the attack path is detected locally and without requiring traceback procedures. Reaction is activated in parallel, controlled in each domain by local policies. The proposed architecture is not an IDS, but rather a “message management system” independent of the underlying detection technologies.

Our experiments and analysis of the prototype identified the following strong points of the proposed framework: (a) the framework automates the manual cooperation activities between domains, thus minimizing reaction time, (b) effective reaction is possible without performing complicated and time-consuming traceback procedures, (c) reaction is local and it is based on locally-defined policy rules, thus achieving parallel operation without exposing network resources outside the domain, (d) the Entities communicate without knowing or exposing the topology of their “overlay network” by using multicast, (e) there is no single point of failure since the ability to use easily interchangeable (and "hidden") nodes within a domain is supported by the distributed Entity concept, (f) there is low message overload and easy escalation of the events when needed, mimicking security organizations operation.

## 7. References


National Infrastructure Protection Center (NIPC) (February 10, 2000) "ALERT 00-034" http://www.nipc.gov/warnings/alerts/2000/00-034.htm

The Register (January 22, 2002) "Cloud Nine blown away, blames hack attack", http://www.theregister.co.uk/content/6/23770.html

InfoWorld (March 26, 2003) "Al-Jazeera hobbled by DDOS attack", http://www.infoworld.com/article/03/03/26/HNjazeera_1.html?applications

Government Computer News (GCN) (2003) "Seattle cybergame preceded last week’s drill and simulated reality", http://www.gcn.com/22_11/homeland-security/22099-1.html

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

Curry, D., Debar, H. (2003) "Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition," IETF Internet Draft, draft-ietf-idwg-idmef-xml-10.txt, January 2003.

Almeroth, K. (2000) "The Evolution of Multicast: From the MBone to Inter-domain Multicast to Internet2 Deployment," IEEE Network, January /February 2000

Sun Microsystems (2002) "Java Management Extensions Instrumentation and Agent Specification, v1.2," February 2002, http://jcp.org/aboutJava/communityprocess/final/jsr003/index3.html

Rizzo, L. (2001) "IP_DUMMYNET", http://info.iet.unipi.it/~luigi/ip_dummynet/

Wan K. and Chang R. (2002) "Engineering of a Global Defence Infrastructure for DDoS Attacks," in Proc. of IEEE International Conference on Networking, Aug. 2002.

Wan K. (2001) "An Infrastructure to Defend Against Distributed Denial of Service Attack, MSc Thesis, The Hong Kong Polytechnic University," June 2001.

Sterne, D., Djahandari, K., Wilson, B., Babson, B., Schnackenberg, D., Holliday, H., and T. Reid (2001) "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, pp.134-149, October 2001

Schnackenberg, D., Djahandari, K., and Sterne, D. (2000) "Infrastructure for Intrusion Detection and Responce, "In Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX II), Anaheim, CA, USA, January 2000.

Ioannidis, J., Bellovin, S. (2002) " Implementing Pushback: Router-Based Defense Against DDoS Attacks," Network and Distributed System Security Symposium, NDSS '02, San Diego, CA, USA, February 2002.

Frincke, D., Tobin, D., McConnell, J., Marconi, J., and Polla, D. (1998) "A Framework for Cooperative Intrusion Detection," Proceedings of the 21st National Information Systems Security Conference, pp. 361-373, October 1998.