Alliance Formation for DDoS Defense

Jelena Mirkovic
University of Delaware
449 Smith Hall
Newark, Delaware 19716
sunshine@eecis.udel.edu

Max Robinson*
UCLA
3564 Boelter Hall
Los Angeles, California
90095-1596
max@cs.ucla.edu

Peter Reiher
UCLA
3564 Boelter Hall
Los Angeles, California
90095-1596
reiher@cs.ucla.edu

Additional authors given at end of paper.

ABSTRACT

Currently, there is no effective defense against large-scale distributed denial-of-service (DDoS) attacks. While numerous DDoS defense systems exist that offer excellent protection from specific attack types and scenarios, they can frequently be defeated by an attacker aware of their weaknesses. A necessary requirement for successful DDoS defense is wide deployment, but none of these systems can guarantee wide deployment simply because deployment depends more on market and social aspects than on the technical performance of the system.

To successfully handle the DDoS threat we must abandon the current paradigm—the design of defense systems that operate in isolation—and shift toward a new paradigm, a distributed framework of heterogeneous systems that cooperate to achieve an effective defense. Heterogeneity is dictated by two major factors. First, the necessary requirements for a successful defense are detection, response and traffic differentiation. These requirements must be met at disjoint points in the Internet and require a disjoint set of functionalities from the defense systems. Second, heterogeneity is dictated by the current state of the DDoS defense field in which numerous systems exist that can offer similar performance and compete for market share. In this paper we show how the paradigm shift can be accomplished quickly and painlessly through the design of DefCOM, a distributed framework that enables the exchange of information and services between existing defense nodes.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; C.2.3 [**Computer-Communication Networks**]: Network Operations—*network management*

General Terms

Security, Design

1. INTRODUCTION

Distributed denial-of-service attacks are a major threat that remains unmitigated in spite of many commercial and research efforts. Existing defense mechanisms yield excellent performance for certain attack types, scenarios and legitimate traffic patterns, but they fail to offer performance guarantees in a generic case. The reason lies in homogeneous system design and incorrect deployment assumptions. Current defense systems attempt to collocate three vital functionalities necessary for a successful defense: accurate attack detection, effective response, and preferential treatment of legitimate traffic. Each of these three functions performs most effectively at disjoint points in the network, so collocation leads to at least one functionality performing suboptimally and becoming a weak point, allowing the attacker to craft his attack to disable the system at a given deployment point. Furthermore, wide deployment is the necessary condition for effective DDoS defense. In sparse deployment some attack scenarios will be successful simply because they are diffuse enough to bypass the system, or large enough to overwhelm it. However, no single defense mechanism can guarantee wide deployment, as deployment depends on market conditions and social aspects. The current DDoS defense market is populated by numerous systems that have comparable performance, and a fair number of networks deploy some of these. A solution that leverages the capabilities of existing systems is preferable to one requiring their replacement.

To successfully handle DDoS attacks, we need a paradigm shift. Instead of building defense systems that operate in isolation, we need to build a distributed framework of defense nodes where heterogeneous systems can plug in and cooperate to achieve a better overall defense. Each system would autonomously perform those functions that it is best at, and compensate for its weaker traits through cooperation with other systems. Division of work would enable nodes to become more specialized, leading to better overall performance. The wide deployment necessary to handle diffuse attacks would be achieved naturally by incorporating existing defense nodes in the framework. As the attacks

*Max Robinson was partially supported by The Aerospace Corporation.

New Security Paradigms Workshop 2003 Ascona Switzerland

evolve, new systems could join and either replace or enhance the functionality of the old ones. The paradigm shift could be achieved quickly and painlessly if the framework has a modular architecture that only requires existing nodes to implement a small set of functions to join the framework and communicate with others, and the nodes can greatly enhance their performance by doing so. Diverse implementations will need a consistent way to plug in, even as they evolve. The trustworthiness of the framework code can be established with a standard set of interfaces and protocols, either commercially produced or available via (inspectable) open source. We illustrate the design of such a distributed framework by proposing DefCOM, a peer-to-peer network of heterogeneous defense nodes that perform cooperative defense.

In Section 2 we offer a brief overview of the DDoS threat and discuss goals of a successful defense. Section 3 discusses differences between the old and the new paradigm, and Section 4 describes a migration strategy. In Section 5 we describe DefCOM, a sample design of a distributed cooperative framework for DDoS defense. We conclude the paper in Section 6.

2. DISTRIBUTED DENIAL-OF-SERVICE

Distributed denial-of-service attacks occur when numerous subverted machines (*agents*) generate a large traffic volume toward the victim, overwhelming its resources. DDoS attacks are an incarnation of the perfect crime in the Internet realm. Perpetrating a DDoS attack requires hardly any knowledge or skills and, without effective defenses, the victim suffers damage for the entire duration of the attack. Furthermore, attackers need not fear punishment, as it is extremely difficult to trace back the attack and locate even the agent machines, let alone the culprits who infected them.

There are several features of DDoS that hinder successful defense:

- **Large volume.** Aggregated attack streams form a large-volume flow that is likely to overwhelm any defense system. This greatly hinders an autonomous (single-point) defense that can be performed close to the victim, as the system must resort to inexpensive per-packet processing to keep up with the flow.

- **Seemingly legitimate packets.** Attack packets can be identical to legitimate packets, since the attacker only needs volume, not content, to inflict damage. Thus, the defense system cannot reach a decision based on individual packets, but must keep a fair amount of statistics to correlate packets and detect irregularities.

- **IP spoofing.** Attackers commonly place a fake address in the IP source field of attack packets. This is done mostly to disguise agent machines, but it can also be used as means to perpetrate *reflector attacks.*[1] Spoofing greatly hinders attempts to locate agent machines or to perform a fair allocation of resources to each client.

The goal of DDoS defense is to alleviate the effect of an attack on a victim and to provide good service to the victim's legitimate clients during the attack. To meet this goal, DDoS defense endeavors concentrate on three approaches: (i) preventing the attack, (ii) enabling the victim to survive the attack, or (iii) detecting the attack and responding to it.

Prevention approaches address the vulnerabilities that can be misused for the attack and take steps to amend them. While these approaches are invaluable for improving Internet security, it will take a long time until they percolate to enough machines to take the wind out of the sails of DDoS.

Survival approaches dynamically enlarge the victim's resources during the attack, enabling it to serve both attack and legitimate traffic efficiently. These approaches work for a limited number of services (e.g., static Web pages). They are, in fact, an arms race between attacker and victim, where the more resourceful party wins. Because the attacker can easily gain more resources by recruiting more agents, he can usually win the battle.

Response approaches detect the attack and respond by cutting off attack streams. To be successful in meeting DDoS defense goals, response approaches must meet the following requirements:

- **Accurate detection.** The system must be able to detect most or all attacks that inflict damage on the victim.

- **Effective response.** The system must reduce the attack flows to manageable levels, regardless of their volume or distribution.

- **Selective response.** The system must differentiate between legitimate and attack packets, and ensure good service to legitimate traffic during the attack. Collateral damage caused by the response must be lower than the damage suffered by legitimate clients in the absence of the response.

3. PARADIGM SHIFT

The current DDoS defense field features a multitude of defense systems that either deploy single-point solutions (the system is installed at one point in the Internet) or distributed networks of homogeneous nodes that perform cooperative defense. In the following sections we examine how this approach leads to poor performance, and then describe the needed paradigm shift and transition methodology. We also give an overview of related work in the DDoS defense field.

3.1 Current Paradigm

Current DDoS defense systems can be divided into autonomous (single-point) and distributed systems. Autonomous

[1] During a reflector attack, the attacker spoofs the address of the victim, requesting a service from several distributed servers. Those servers send replies to the victim, overwhelming its resources.

systems consist of a single defense node that observes the attack and applies the response. Distributed systems consist of multiple defense nodes (frequently with the same functionality) that are deployed at various locations and organized into a network. Nodes communicate through the network and coordinate their actions to achieve better overall defense.

3.1.1 Autonomous Defense

DDoS attack streams originate from distributed attack machines, are forwarded by core routers and converge at the victim network or some nearby core router. We observe this process as an interaction of three types of networks: *source networks* that unwittingly host attack machines, several *intermediate networks* that forward attack traffic to the victim, and the *victim network* that hosts the target. Each of the involved networks (source, intermediate, or victim) can carry DDoS defense systems. We now observe the level of defense that can be provided by an autonomous system deployed at only one of these points.

It is relatively easy to design and implement accurate attack detection for a DDoS defense system positioned at the **victim network** because it can closely observe the victim and notice anomalous behavior.[2] However, the range of response is limited: large-volume attacks can overwhelm the defense, and attacks deploying seemingly legitimate packets will force the defense system to inflict collateral damage during the response. The differentiation of legitimate streams from attack streams is complex at this point, since they have been heavily aggregated by the time they reach the victim network. Examples of victim-end systems are firewalls [17], intrusion detection systems [2], access lists for critical resources [4], capability-based systems [26], client-legitimacy-based systems [20], resource accounting [6, 9, 12, 27, 28] and protocol security mechanisms [1, 13, 18, 25].

A DDoS defense system in the **intermediate network**, usually installed at a core router, detects the attack through anomalies observed at this router. As core routers handle large-volume, highly aggregated traffic, they are likely to overlook all but large-scale attacks. Detected attacks can be quickly suppressed, thanks to abundant network resources. However, the response is likely to inflict collateral damage because core routers can only accommodate simple rate-limiting requests and cannot dedicate memory or processor cycles to traffic profiling. Local aggregate-based congestion control (ACC) [16] is one example of an intermediate network defense system. ACC augments routers in the intermediate network to detect congestion caused by a DDoS attack and respond to it by rate limiting high-bandwidth aggregates. One weakness of ACC is that it is not effective at selective response because it fails to differentiate good traffic that matches the aggregate signature, and therefore ends up dropping the good traffic along with the bad. Congestion control is not quite the same as selective response. It is not sufficient to only identify bad traffic—it is also necessary to identify good traffic, so that the good traffic can receive preferential treatment. Adjacent ACC routers can cooperate, turning the local ACC approach into a distributed solution: the congested router that cannot handle the aggregate itself issues a rate-limit request through a pushback mechanism [8] to its immediate upstream neighbors.

A DDoS defense system located in the **source network** faces hard detection challenges, because, due to distribution, it can observe only a small portion of the attack. On the other hand, the small attack volume enables effective response and complex profiling that minimizes collateral damage. Examples of source-end defense systems include Multops [7], Reverse Firewall [5] and D-WARD [19].

3.1.2 Distributed Defense

It is evident that no single deployment point can achieve successful defense in autonomous operation. Therefore, the DDoS problem requires a *distributed* solution in which defense nodes located throughout the Internet cooperate to achieve better overall defense.

Several distributed systems deploy their nodes in a contiguous manner, thus achieving better effectiveness. Adjacent nodes communicate and coordinate their actions, usually propagating the response toward the sources of the attack. Examples include Pushback [16] routers and IDIP [24] and MANAnet [5] cooperative neighborhoods. The requirement for contiguous deployment is a major drawback of these approaches, as a single legacy router will divide the system into two separate parts, thus greatly limiting the defense.

Other distributed systems organize remote nodes into a network, thus accommodating non-contiguous deployment. Examples of these approaches are the Secure Overlay Service (SOS) [10, 11], the Active Security System (ASSYST) [3] and COSSACK [21].

All existing distributed systems (contiguous and networked) provide communication and cooperation only between their nodes, and operate in isolation from other defense systems.

3.2 New Paradigm

Current research techniques have resulted in excellent defense systems, but the generic DDoS threat still remains unmitigated. The reason lies in failure to extend and benefit from node specialization and to provide for wide deployment of defense nodes.

3.2.1 Node Specialization

Since various deployment points bring different benefits (as discussed in Section 3.1.1), a defense node should be specialized to perform those functions that its resources and location facilitate, and delegate more challenging functions to other nodes. Specialization is crucial because it enables nodes to deploy more sophisticated techniques and thus yield better performance.

Attack detection is best done at the victim, while response is most effective and collateral damage is minimal at the source. We can thus envision a defense system in which the attack is detected near the victim, and then the detection signal, along with the attack signature, is propagated to the source networks. Source networks then act to constrain malicious flows, and to profile and preferentially serve legitimate flows. Since no solution can be deployed globally, a

[2] While perhaps not all end nodes will deploy detector capability, surely those most concerned about the threat of DDoS will.

mechanism is needed to constrain attack flows from legacy or uncooperative source networks. Intermediate network nodes are ideally suited to perform this function, because a wide coverage of attack paths can be achieved with few deployment points due to the highly interconnected core topology. Route-based filtering [22] provides a ready example—the deployment of route-based filters at only 18% of autonomous systems provides almost perfect prevention of IP spoofing. A further benefit of engaging the intermediate network in DDoS defense is a reduction in communication overhead for coordination, as the addition of core defense nodes into the overlay network can more efficiently propagate attack detection signals.

3.2.2 Heterogeneous Cooperative Defense

Since DDoS attacks involve a large number of distributed machines, wide deployment of defense nodes is a necessary requirement for a successful defense. A system must ensure that its defense nodes are spread throughout the Internet to capture the majority of attack streams. Lacking that, the most sophisticated system can be defeated simply by bypassing its nodes.

On the other hand, no single system can guarantee wide deployment, since that would require dominance of the market —an unattainable goal for many technologies. Even if the new technology offered superior performance, the customers that have already purchased some DDoS defense system would be unwilling to forfeit their investment and readily switch to a new system. Sparse deployment leads to low performance guarantees, thus discouraging deployment even further.

To break from this vicious circle, we must switch from isolated to cooperative defense, enabling heterogeneous defense systems to exchange information and service. Wide deployment is then achieved naturally by accommodating legacy nodes, and various policies can be devised between nodes to negotiate service.

4. A DISTRIBUTED FRAMEWORK FOR DDOS DEFENSE

To move from isolated to cooperative defense we must build a framework through which heterogeneous nodes can join and communicate with each other. The framework must scale to a large number of participants and must offer high security guarantees, against both external threats and malicious participants. To protect against false positives, the framework must use strong authentication for all its participants, coupled with strictly limiting the power to raise alarms to victims or their authorized, authenticated representative alert generator node(s). An extremely important requirement is for the framework to provide guaranteed good service to legitimate traffic. If a defense node can differentiate between legitimate and attack packets, it must be able to communicate this difference to downstream nodes, ensuring preferential service to legitimate traffic. In the rest of this section we describe framework architecture and transition strategy, and in Section 5 we present a sample framework design.

4.1 Framework Architecture

We envision the peer-to-peer paradigm as a logical choice for a framework architecture. A peer-to-peer network can easily scale to a large number of nodes. Further, leaving the hierarchy out of the architecture makes the system more resilient to attacks—there are no nodes that are more important than other nodes whose failure would cripple the system.

The nodes in the framework can roughly be classified into three categories, based on the functionality they provide: *alert generator* nodes that detect the attack and deliver its signature to the rest of the peer network, *core* nodes that rate limit high-volume transit traffic matching the signature, and *classifier nodes* that perform selective rate limiting, differentiate between legitimate flows and attack flows, and cooperate with other defense nodes to ensure preferential service for legitimate traffic.

Edge networks have sufficient resources to provide alert generator and classifier functionalities, since they relay small traffic volumes. Core node functionality is redundant in this case because it is superseded by classifier functionality. Core networks relay high volumes of traffic and are thus likely to provide only core node functionality and limited alert generator functionality (by detecting congestion and selecting traffic aggregates that contribute to it).

Defense nodes must be able to communicate and must support at least the following messages:

- Attack alerts—authenticated alerts should be propagated from alert generator nodes to the rest of the peer network securely.
- Rate-limit requests—each node should be able to communicate rate-limit requests to its upstream core or classifier neighbors, thus controlling its incoming traffic. Upstream nodes can grant or deny the request and the requesting node can then police the incoming traffic from these peers according to their behavior.
- Resource requests—each node should be able to issue a resource request to its downstream neighbors, thus bargaining for a larger share of the victim's resources in favor of its clients. Various policies between peers can regulate granting of these requests.
- Traffic classification—classifier nodes must be able to communicate enough information about legitimate traffic to their downstream peers to ensure that the bulk of the legitimate traffic will not be dropped. In this manner classifier nodes can honor service guarantees to legitimate clients. Additionally, the framework must ensure that previously policed (rate-limited) traffic has a higher priority when competing for downstream resources than non-policed traffic.

4.2 Transition Strategy

Many networks today deploy nodes that provide alert generator functionality, such as firewalls, intrusion detection and monitoring systems. Those networks could support the transition by simply deploying a communication layer enabling the defense system to deliver authenticated alerts to the peer network. Core routers already provide rate limiting and service differentiation, and support remote installation of services through SNMP messages. For transition, this functionality should be extended so that SNMP messages can be delivered between peers in a secure and authenticated fashion. Classifier functionality is not currently provided by a large number of networks, although it is offered by source-end defense systems such as [5, 19]. The motivation for deployment of classifier nodes will be increased once the framework is deployed, since networks with classifier nodes will be able to offer service guarantees to their clients in spite of DDoS attacks.

Nodes in this framework can work well in partial deployment. Minimally, a single source-side classifier node deployment provides that network's clients with good service to a victim protected by a core node. The motivation for deployment at the victim network is obvious. The initial motivation for deployment in the source network would come from a desire to guarantee good service from the protected server to its clients in the source network. From there, the defense system's effectiveness will increase steadily with increasing deployment. The transition to a larger cooperative defense could start when a small number of edge and core networks join the framework and provide alert generator and core node functionalities. Because of the core network's topology and resources, even a small number of core nodes will impact the majority of attack flows [22]. The cooperation between alert generator and core nodes will allow accurate detection and efficient response, alleviating the effect of the attack on the victim. As the deployment of alert generator and core nodes increases, there will be increasing benefit and more motivation to deploy classifier nodes, since the possible service guarantees will improve and will apply to more potential victims.

5. DEFCOM

The Defensive Cooperative Overlay Mesh (DefCOM) is an example design of a distributed framework for DDoS defense. DefCOM consists of heterogeneous defense nodes organized into a peer-to-peer network, communicating to achieve a dynamic cooperative defense. A high-level overview of DefCOM's operation is given in Figure 1. It shows the presence of legacy routers (white circles), core defense nodes (black circles), alert generators (light grey circles), classifier nodes (dark grey circles) and the two types of traffic sources: legitimate clients and attackers. Some traffic sources are behind classifier defense nodes, while others are not.

Defense nodes are organized in a peer-to-peer network whose topology construction allows approximation of the underlying routing topology.[3] During the attack they discover the *victim-rooted traffic tree,* thus identifying upstream-downstream relationships between peers. They then devise the appropriate rate limits to restrain the attack traffic, and place them as close to source networks as possible. At the same time, classifier nodes differentiate legitimate from attack streams. All nodes in the framework give preferential service to legitimate traffic.

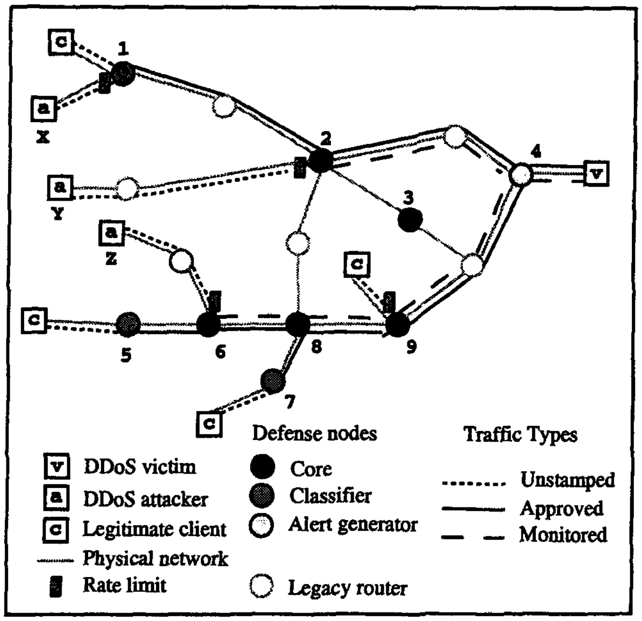

Figure 1: DefCOM overview.

5.1 Traffic Tree Discovery

When a DDoS attack occurs, the alert generator node closest to the victim detects it and propagates the *alert message* to all nodes in the peer network. In Figure 1, alert generator node 4 detects a DDoS attack on victim *V* and informs all other defense nodes through the DefCOM peer network. This process is optimized so that those nodes that do not forward any traffic to the victim also do not propagate the signal further; e.g., node 3 will receive the alert but will not send it to its neighbors. Those nodes that observe traffic to the victim (1, 2, 4, 5, 6, 7, 8, and 9) are called *active nodes.* They cooperate to trace out the topology of the victim-rooted traffic tree by deploying *secure traffic stamping.* Tracing of the tree structure enables each node to assign upstream or downstream classification to its peers, thus defining its policy and message types to be sent to these peers. Secure packet stamping actually serves four purposes: (1) discovery of the victim-rooted traffic tree topology, (2) differentiation of traffic types, (3) protection of legitimate traffic and (4) transparent operation through legacy routers. Each active defense node picks a *stamp* and communicates it securely to its neighbors. The node places this stamp in the header of packets it forwards to the victim. It also observes packets it receives from its neighbors, looking for their stamps. A node becomes a parent of a neighbor if it observes its neighbor's stamped traffic. A parent sends an explicit message to its children to inform them of their child status.

To protect the packet stamping mechanism from misuse, every pair of neighboring nodes uses stamps unique to them, and changes the packet stamps on a frequent basis, using

[3] Nodes can either start off by carefully constructing a peer network topology to resemble the routing topology, or converge to it through reconfiguration guided by traffic observation. The latter can be accomplished by detecting the presence of intermediate nodes between two nodes (that were assumed to be neighbors) using a tool like traceroute, or observing BGP updates.

encryption for privacy and authentication to establish a secured communication channel for this exchange.

Determining the best location in packets to place stamps will be addressed in future work. For instance, for IPv4, IP Traceback [23] suggested overloading the IP identifier field in the IP header, pointing out that less than 0.25% of IP packets are fragmented. The eventual adoption of IPv6 will offer better options for packet stamping.

A malicious outsider falsely reporting a DDoS attack will be a serious problem. If the false report is believed and DefCOM deploys distributed responses that rate limit traffic inbound to the supposed victim, potential damage can be done to legitimate traffic. This attack would effectively use DefCOM to degrade the victim's service (it couldn't quite *deny* service, but it could reduce it). We plan to investigate scalable methods to allow potential victims to delegate alert-generation responsibility and to enable DefCOM nodes to authenticate alert signals. DefCOM nodes will only recognize those alarms that are signed by the victim or its delegated alert generators. A scalable authentication through a public key infrastructure (PKI) will probably be necessary to verify the alarms, but the PKI infrastructure itself must be protected. That problem is likely to be manageable, because DefCOM's defenses can be activated to protect PKI servers, and also because the PKI service would not be a general public service; it is limited to the set of previously authenticated DefCOM nodes, so approaches like [10] are likely to be effective.

Alert-generator nodes might become compromised and issue false alarms. We also plan to investigate ways to allow victims to revoke alert generator responsibility from compromised nodes. Broadcast encryption [15] is a promising approach that has scalable revocation properties compared to traditional PKI.

5.2 Distributed Rate Limiting

Once the tree topology is determined, the nodes cooperate to deploy rate limits that limit attack traffic while protecting and granting preferred service to legitimate traffic. An optimal deployment puts the rate limits as close to the leaves of the tree as possible. This relieves congestion higher up the tree, near the victim-root. In Figure 1, core defense nodes 1, 2, and 6 deploy rate-limits to stop or reduce attack traffic from attack nodes X, Y and Z to V.

The rate limit is initially propagated from the root of the tree (a node that has no parents) downstream. Each node assigns an equal share of its rate limit to its children and communicates this through *rate-limit requests*. The resource division created through this initial rate-limit process can be modified through *resource requests* issued by a node to its parents.

The resource request functionality must be provided by the system, since some children of a node may have a greater number of legitimate clients upstream than other children. On the other hand, the resource request process is particularly vulnerable to manipulation by malicious participants (who could request a large amount of resources at the expense of others) and must be regulated. A good policy is that initially, every child gets an equal share. If one or more children are not using their full share, then the parent allocates a portion of the unused bandwidth to the child requesting more than its equal share. In this manner, a child can only get a larger share as long as it is not at the expense of another child.

Generally, the problem of having malicious participants exists in any distributed system (such as routing and the DNS infrastructure), and is still unsolved. However, we believe that by limiting the extent of trust between nodes we can limit the amount of unfairness that the malicious participant can introduce in the framework. We plan to devise monitoring and policing functions to ensure that rate requests are obeyed and resource requests are granted in accordance to negotiated policy. Unreasonable requests should be dropped and trust should be reduced for the requesting node. Subsequent requests from the untrusted node will have a lower probability of being granted.

Nodes should report traffic statistics (offered traffic, dropped traffic, and passed traffic destined for the victim) to their parent along with their resource requests. Parents should aggregate the statistics and report them to their own parent. In this manner, the root of the tree has all the necessary information to determine when the attack has abated. The root can report the aggregated statistics to the original alert generator node that signed and raised the attack alarm that started up the defense. The alert generator can then, perhaps with human supervision, turn off the defenses by issuing a signed *attack over* message using the same mechanisms used to distribute the original attack alarm. We will eventually address the problem of the particularly advanced opponent who waits for an attack-over message to resume an attack.

5.3 Differentiated Service for Legitimate Traffic

Defense nodes use secure packet stamping to provide different service levels, ensuring that the policed traffic has priority in resource allocation.

Each defense node maintains an *approved stamp* and a *monitored stamp*. Classifier defense nodes profile the traffic originating from their networks, and securely mark those packets that are deemed legitimate with an approved stamp. The remaining limited resources will be filled with suspicious traffic, carrying the monitored stamp. In Figure 1, classifier nodes 1, 5, and 7 vouch for traffic from legitimate clients C. Node 9 also rate limits traffic from legitimate client C, if necessary. Since this client does not have a classifier node to vouch for it, it is an unknown source and is thus subject to rate limiting. The traffic that passes a rate limit in the core node will bear a monitored stamp, informing downstream nodes that it has been policed.

The rate-limit algorithm in core nodes should provide preferential service to marked traffic by apportioning its allowed resources first to approved traffic (that has passed through a classifier node), then to monitored traffic (that has passed through a core node and has been policed) and lastly to unstamped traffic (containing an unknown mix of legitimate and attack traffic).

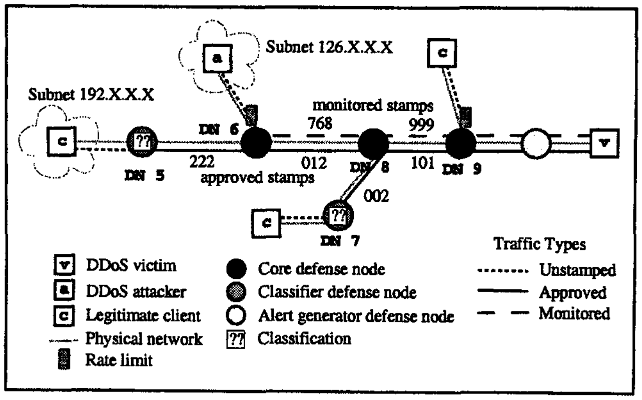

Figure 2: Packet stamping and flow classification.

Figure 2 illustrates in more detail how preferred service for legitimate traffic is achieved using the packet-stamping mechanism. This figure is a magnified lower branch of Figure 1.

A legitimate traffic flow originating in the 192 subnet is shown passing through defense node 5 in Figure 2. Originally unstamped, it leaves defense node 5 bearing stamp 222, classified as approved traffic. Upon reaching defense node 6, it is restamped with stamp 012. Upon reaching defense node 8, it is aggregated with another incoming approved traffic flow (bearing stamp 002), and both flows leave this node 8 bearing stamp 101 and continue toward the victim. In contrast, attack traffic from the 126 subnet is rate limited by defense node 6, and becomes a monitored flow leaving this node with stamp 768. This monitored flow is restamped at node 8 with stamp 999. Further aggregation of monitored flows is shown in defense node 9, where fresh unstamped traffic from a legitimate client enters, is rate limited, and aggregated with the 999 flow. This example shows how two stamps must be maintained between any two defense nodes. At this snapshot in time, for defense nodes 8 and 9, the 999 stamp defines monitored traffic, and the 101 stamp defines approved traffic.

5.4 Framework Security

Both DefCOM as a whole and individual defense nodes must themselves be defended against subversion and other attacks. We have some preliminary, promising approaches.

Unlike some peer-to-peer systems, it may be possible to design DefCOM to be resilient to DDoS attacks directed at the defense node peers themselves. Peers could act to protect other peers from DDoS attacks. One possibility is to give each defense node a limited ability to raise an attack alarm, only for traffic destined for itself. In this manner, DefCOM can be enlisted to protect its own defense nodes as well.

The distributed system as a whole should be able to constrain the damage that a subverted, malicious node can cause, as long as it has a good parent or ancestor node. Using the concept of the victim-rooted traffic tree, a malicious node is able to deny service to legitimate clients in the subtree rooted at itself, but cannot otherwise harm the service granted to other legitimate clients. This is because the total traffic that the malicious node can contribute (even if it falsely stamps the traffic as "approved") is strictly limited by the malicious node's parent, who will not allow the traffic to interfere with its other children. The total traffic the malicious node can contribute is also strictly limited to some maximum allocation by the propagation and subdivision of the distributed rate limits.

Malicious nodes may be able to hinder the spread of attack alarms. This could be mitigated by ensuring adequate redundant paths between nodes in the overlay network topology, as in [14]. DoS attacks that attempt to stop the initial broadcast of an attack alarm can be mitigated by having redundant, disjoint alert generators for a particular victim, and possibly by reserving a small amount of bandwidth with the highest priority service level for the encrypted, authenticated control message channel between defense nodes. DoS attacks that target the encryption or authentication channel are yet another area of future work, another step in the arms race between the defenders and the attackers, and will be universal problems for all security systems. They may perhaps be addressed with unequal work (client puzzles), filtering, or rate limiting message processing. Further issues of trust relationships between defense nodes are another area of future work.

Attacks against the stamping mechanism are also a possible problem. If an attacker can discover the "approved" stamp, he can get priority service for his attack traffic. However, the amount of that traffic he can pass is strictly limited to some share that won't harm other, real, legitimate clients outside the subtree rooted at the defense node where the attacker's falsely stamped traffic is inserted into the victim-rooted traffic tree. This threat can be mitigated by having enough bits in the stamp to ensure that discovery of a working stamp takes longer than the stamp rotation period. Obviously the rotation period can also be shortened (at the cost of increased overhead); these are all parameters to be explored.

6. CONCLUSION

Distributed denial-of-service is a major threat that cannot be addressed through isolated action of sparsely deployed defense nodes. Instead, various defense systems must organize into a framework and interoperate, exchanging information and service, and acting together against the threat. Through accommodation of existing defense solutions, the approach is very likely to achieve wide deployment, and the node specialization within the framework will lead to better overall performance. We have outlined the features and functionalities that the framework design should provide and illustrated this by proposing DefCOM—a peer-to-peer network of heterogeneous defense nodes. We have also proposed a viable transition strategy that embeds economic incentives for all involved parties. It is our firm belief that the Internet cannot be defended through isolated action, but rather through tight cooperation and joint enterprise of heterogeneous defense systems. Since attackers cooperate to perform successful attacks, defenders must also form alliances to defeat the DDoS threat.

7. ADDITIONAL AUTHORS

Geoff Kuenning (Harvey Mudd College, 1250 N. Dartmouth Ave., Claremont, CA 91711. Email: geoff@cs.hmc.edu).

8. REFERENCES

[1] T. Aura, P. Nikander, and J. Leiwo. DOS-resistant authentication with client puzzles. *Lecture Notes in Computer Science*, 2133, 2001.

[2] S. Axelsson. Intrusion detection systems: A survey and taxonomy. Technical Report 99-15, Department of Computer Engineering, Chalmers University, March 2000.

[3] R. Canonico, D. Cotroneo, L. Peluso, S. P. Romano, and G. Ventre. Programming routers to improve network security. In *Proceedings of the OPENSIG 2001 Workshop Next Generation Network Programming*, September 2001.

[4] Cisco. Strategies to protect against distributed denial of service attacks. http://www.cisco.com/warp/public/707/newsflash.html.

[5] Cs3, Inc. *MANAnet DDoS White Papers*. http://www.cs3-inc.com/mananet.html.

[6] A. Garg and A. L. N. Reddy. Mitigation of DoS attacks through QoS regulation. In *Proceedings of IWQOS workshop*, May 2002.

[7] T. M. Gil and M. Poletto. MULTOPS: a data-structure for bandwidth attack detection. In *Proceedings of 10th Usenix Security Symposium*, August 2001.

[8] J. Ioannidis and S. M. Bellovin. Pushback: Router-based defense against DDoS attacks. In *Proceedings of NDSS*, February 2002.

[9] A. Juels and J. Brainard. Client puzzles: A cryptographic countermeasure against connection depletion attacks. In *Proceedings of the 1999 Networks and distributed system security symposium*, March 1999.

[10] A. D. Keromytis, V. Misra, and D. Rubenstein. SOS: Secure overlay services. In *Proceedings of SIGCOMM 2002*, 2002.

[11] A. D. Keromytis, V. Misra, and D. Rubenstein. Using overlays to improve network security. In *Proceedings of SPIE ITCom Conference on Scalability and Traffic Control in IP Networks II*, July 2002.

[12] F. Lau, S. H. Rubin, M. H. Smith, and L. Trajkovic. Distributed denial of service attacks. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 2275–2280, Nashville, TN, USA, October 2000.

[13] J. Leiwo, P. Nikander, and T. Aura. Towards network denial of service resistant protocols. In *Proceedings of the 15th International Information Security Conference*, August 2000.

[14] J. Li, P. Reiher, and G. Popek. *Disseminating Security Updates at Internet Scale*. Kluwer Academic Publishers, 2003.

[15] J. Lotspiech, S. Nusser, and F. Pestoni. Broadcast encryption's bright future. *IEEE Computer*, August 2002.

[16] R. Mahajan, S. Bellovin, S. Floyd, V. Paxson, and S. Shenker. Controlling high bandwidth aggregates in the network. *ACM Computer Communications Review*, 32(3), July 2002.

[17] McAfee. *Personal Firewall*. http://www.mcafee.com/myapps/firewall/ov_firewall.asp.

[18] C. Meadows. A formal framework and evaluation method for network denial of service. In *Proceedings of the 12th IEEE Computer Security Foundations Workshop*, June 1999.

[19] J. Mirkovic, G. Prier, and P. Reiher. Attacking DDoS at the source. In *Proceedings of the ICNP 2002*, November 2002.

[20] E. O'Brien. *NetBouncer: A practical client-legitimacy-based DDoS defense via ingress filtering*. http://www.nai.com/research/nailabs/development-solutions/netbouncer.asp.

[21] C. Papadopoulos, R. Lindell, J. Mehringer, A. Hussain, and R. Govindan. COSSACK: Coordinated suppression of simultaneous attacks. In *Proceedings of DISCEX III*, April 2003. to appear.

[22] K. Park and H. Lee. On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets. In *Proceedings of ACM SIGCOMM 2001*, August 2001.

[23] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. In *Proceedings of ACM SIGCOMM 2000*, August 2000.

[24] D. Schnackenberg, K. Djahandari, and D. Sterne. Infrastructure for intrusion detection and response. *Advanced Security Research Journal*, 3(1), 2001.

[25] C. Schuba, I. Krsul, M. Kuhn, G. Spafford, A. Sundaram, and D. Zamboni. Analysis of a denial of service attack on TCP. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, May 1997.

[26] J. Shapiro and N. Hardy. EROS: A principle-driven operating system from the ground up. In *IEEE Software*, pages 26–33, January/February 2002.

[27] O. Spatscheck and L. L. Petersen. Defending against denial of service attacks in Scout. In *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation*, February 1999.

[28] Y. L. Zheng and J. Leiwo. A method to implement a denial of service protection base. In *Information Security and Privacy*, volume 1270 of LNCS, pages 90–101, 1997.