

Efficacy of Coordinated Distributed Multiple Attacks (A Proactive Approach to Cyber Defense)

P. Defibaugh-Chavez, S. Mukkamala, A. H. Sung

Department of Computer Science

Institute for Complex Additive Systems Analysis

New Mexico Tech

(*pchavez|srinivas|sung@cs.nmt.edu*)

Abstract

In the network-centric approach to information operations, users share information robustly by means of a secure infrastructure that enables self-synchronization and, ultimately, more effective information operations which makes information availability a key component. Since information operations will only become more important in future network operations, a well-developed attack tool that is able to strike the adversary's information networks is a valuable asset and should not be ignored. The goal of the proposed coordinated distributed multiple attack (CDMA) is to strike a selected adversarial network and render its services useless. In this attack, a number of compromised systems are used as facilitators in a coordinated manner to launch an attack on a victim's host or network.

- *By distributing the attack and varying the type of attack the source attacker exhibits a decreased intensity of activity; therefore, the attack becomes harder to detect. Meanwhile, the concentrated effect on the victim is sufficient to overload network peripherals and systems, resulting in denial of service.*
- *In CDMA an adversary can target a wide range of vulnerabilities present in victims' operating systems, protocol stacks, applications, network infrastructure, etc.*

1. Introduction

Distributed Denial of Service (DDoS) attacks have been around for many years and will probably still be present for many to come. Over the years the black hat community (the "hackers") has increased the sophistication and potency of their DDoS attacks [1]. We have seen an increase in the number of nodes and a decrease in the time it takes to install a DDoS agent once a computer has been compromised [2]. Pre-scanned targets are now becoming more common.

Adversaries' perspective of information availability has changed in recent years that is "If I can't have it, nobody can". By their nature networks are connected,

distributed, open, and dynamic. Phenomenal growth of computing devices, connectivity speed, and the number of applications running on networked systems engender a risk to the interconnected systems. Malicious usage, attacks, and sabotage have been on the rise as more and more computing devices are put into use. Connecting information systems to networks such as the Internet and public telephone systems further magnifies the potential for exposure through a variety of attack channels. These attacks take advantage of the flaws or omissions that exist within the various information systems and software that run on many hosts in the network.

Efforts on how to define and characterize denial of service (DoS) attacks through a collection of different perspectives such as bandwidth, process information, system information, user information, and IP address is being proposed by several researchers [1-6].

This paper attempts to explore some more damaging attacks, ones that are highly synchronized (something we haven't seen in abundance in the wild) and that use a wide attack portfolio. Most DDoS attacks target only a handful of vulnerabilities but we will be attacking many vulnerabilities simultaneously. Once we have shown that a highly blended attack can be successful, meaning it consumed the victim's resources; we will explore how blended the attacks can be. One concern is that the overhead of orchestrating the attacks may become too large or that the attacks are too blended to be effective. These are the areas we wish to explore.

In the next section we discuss the theoretical aspects of DoS and DDoS attacks. Section three covers theoretical aspects of coordinated distributed multiple attacks. The methodology is given in section four. The results of our experiments are given in section five which is followed by our conclusions with a brief discussion of the work. Our grateful acknowledgements are just before the references which are in the last section.

2. Denial of Service Attack Schemes

Attacks designed to make a host or network incapable of providing normal services are known as denial of service attacks. A "denial-of-service" attack is

characterized by an explicit attempt by attackers to prevent legitimate users of a service from using it that service [7-10]. Examples include:

- Attempts to "flood" a network, thereby preventing legitimate network traffic
- Attempts to disrupt connections between two machines, thereby preventing access to a service
- Attempts to prevent a particular individual from accessing a service
- Attempts to disrupt service to a specific system or person

There are different types of DoS attacks: a few of them abuse the computer's legitimate features; a few target the implementations bugs; and a few exploit the mis-configurations. A DoS attack can be viewed as a process that has a beginning, an end goal, and which may or may not utilize resources acquired during the process of the attack. Three different networks are involved in a DoS attacks can be divided into three networks; these are not a separation based on physical location, but rather a logical description of their roles in the attack [2].

- **Origination Network:** The origination network is the collection of one or more nodes that the adversary has access to and uses to initiate the DoS attack. The adversary is not limited to a single individual, and in the cyber world, physical control of a node is not a requirement for access to the node.
- **Facilitation Network:** The facilitation network is the collection of zero or more nodes used by the adversary after an attack's initiation to help accomplish the attack on the victim network.
- **Victim Network:** The victim network is the collection of one or more nodes to which the adversary wishes to deny service.

A brief attack summary and a signature of the most common DoS attacks are given below [7,8].

- **SYN Flood (Neptune):** DoS attack performed against every TCP/IP implementations where an adversary utilizes the half open TCP connections to flood the data structure of half open connections on the innocent server causing to deny resulting in access denial to legitimate requests. This attack in some cases can cause permanent failure. The service can be regained automatically. Looking for a number of simultaneous SYN packets coming from the same host or unreachable host in a given short period of time can prevent this attack.
- **Smurf:** DoS attack performed against all the systems connected to the Internet in the victim network in which where an adversary convinces uses a separate network to send Internet Control Message Protocol (ICMP) echo reply messages

from remote locations to the request packets IP broadcast addresses of the victim network to deny services. This attack causes temporary denial of services and can be automatically recovered. Looking for a large number of echo replies to the innocent machine from different places without any echo request made by the innocent machine helps in detecting this attack.

- **UDPstorm:** DoS attack performed against networks where an adversary utilizes the UDP service feature to cause congestion and slowdown. This attack denies the services permanently and can be resumed with system administrator intervention. This attack can be identified by looking for spoofed packets and inside network traffic.
- **Ping of Death (PoD):** DoS attack performed against older versions of operating systems where an adversary tries to send an oversized IP packet, and the system reacts in an unpredictable manner, causing crashing, rebooting, and even freezing in some cases. This attack causes temporary failure of services. Looking for Internet Control Message Protocol (ICMP) packets that are longer than 64000 bytes and blocking them is the way to will prevent this attack.

Distributed Denial of Service Attacks

The DoS attacks can take an advantage form from the distributed nature of the Internet by launching a multiplicative effect, resulting in a distributed DoS. A Distributed Denial of Service (DDoS) attack takes these basic DoS building blocks and goes another step, replicating the attacking host hundreds of times, distributed around the Internet [4].

These distributed attack "zombies", are controlled remotely, by one or more "bot nets"; therefore, so that the attacks are can be centrally controlled. Even if one attacking machine is traced and shut down, the others can continue the attack. This makes the attack much more difficult to eliminate completely. On the other hand, the defenses against the component attacks can still help. The zombies must be placed on the hosts having gained access to the host through some remote system vulnerability. Most of the compromised agent machines appear to have been attacked through known vulnerabilities in Remote Procedure Call (RPC) services. Due to the use of dynamic protocols and address spoofing, detecting distributed and automated attacks still remains a challenge. A brief attack summary and a signature of the most common DDoS attacks are given below [4].

- **Trinoo:** DDoS attack that uses UDP flood to attack the systems on the Internet. An adversary uses TCP and UDP packets from remote locations to deny services. The trinoo daemons were originally believed to be UDP based, access-restricted remote command shells,

possibly used in conjunction with sniffers to automate recovering sniffer logs. An adversary does not require root or administrative privileges to launch the attack. The original attack script uses the following ports: 1524 TCP, 27665 TCP, 27444 UDP, and 31335 UDP. These are used as default ports for communication between victims, bolt bot nets, and zombies. Please note the port numbers can easily be changed.

- Tribal Flood Network (TFN): DDoS tool capable of waging ICMP flood, SYN flood, UDP flood, and Smurf style attacks, as well as providing an "on demand" root shell bound to a TCP port. TFN daemons were identified as having been compromised by exploitation of buffer overrun bugs in the RPC services "statd", "cmsd" and "ttdbserverd". The attacker(s) control one or more clients, each of which can control many daemons. The daemons are all instructed to coordinate a packet-based attack against one or more victim systems by the zombie. Remote control of a TFN network is accomplished via command line execution of the client program, which can be accomplished using any of a number of connection methods (e.g., remote shell bound to a TCP port, UDP based client/server remote shells, ICMP based client/server shells such as LOKI, SSH terminal sessions, or normal "telnet" TCP terminal sessions.). No password is required to run the client, although it is necessary to have the list of bolt bot nets at hand available in an "iplist" file. The original attack script uses the following ports: 16660 TCP, 65000 TCP, ICMP ECHO, and ICMP ECHO REPLY. These are used as default ports for communication between victims, bolt nets and zombies.
- Stacheldraht (German for "barbed wire"): Stacheldraht (German for "barbed wire") This attack combines features of the "trinoo" distributed denial of service tool, tool with those of the original TFN, and adds encryption of communication between the attacker and stacheldraht masters and automated update of the agents. Stacheldraht deals with the weaknesses of TFN by adding an encrypting "telnet alike" (stacheldraht term) client. The original attack script uses the following ports: 16660 TCP, 65000 TCP, ICMP ECHO, and ICMP ECHO REPLY. These are used as default ports for communication between victims, bolt bot nets and zombies.
- Tribe Flood Network 2000 (TFN2K): TFN2K allows masters to exploit the resources of a number of bolt bot nets in order to coordinate an attack against one or more designated targets. The original script effects, affects UNIX, Solaris, and Windows NT platforms that are connected to the Internet, directly or indirectly. However, the tool could easily be ported to additional platforms. TFN2K is a two-component

system: a command driven client on the master and a daemon process operating on an agent. The master instructs its agents to attack a list of designated targets. The agents respond by flooding the targets with a barrage of packets. Multiple agents, coordinated by the master, can work in tandem during this attack to disrupt access to the target. Master-to-zombie communications are encrypted, encrypted and may be intermixed with any number of decoy packets. Both master-to-zombie communications between clients, handlers, and agents does not use any specific port (it may be supplied on run time or it will be chosen randomly by a program) but is a combination of UDP, ICMP, and TCP packets. The master can spoof its IP address.

3. CDMA Attack Schemes

Traditional DDoS attacks have been around for quite some time and have been well studied. A traditional attack consists of a single network attack in which multiple nodes are launched against a single target with the intention of denying the victim's services to its legitimate users. Although there have been significant efforts to curb DDoS attacks many researchers believe there is no way to stop them as long as the attackers can gain access to large groups of zombie nodes.

CDMAs are taking traditional DDoS attacks a step further. Instead of a single attack we target multiple vulnerabilities using a diverse selection of protocols as well as varying the attacks over time. The attack portfolio allows us to strike a larger target area (i.e., more vulnerabilities) than a target with just one vulnerability. This coupled with the constantly changing source makes CDMAs hard to detect and block. Even if most of the attacks are blocked, some may still bypass the target's defenses [10].

In this attack (a schematic diagram is given below as figure 1), a number of compromised systems are used as facilitators in a coordinated manner to launch an attack on a victim's host or network.

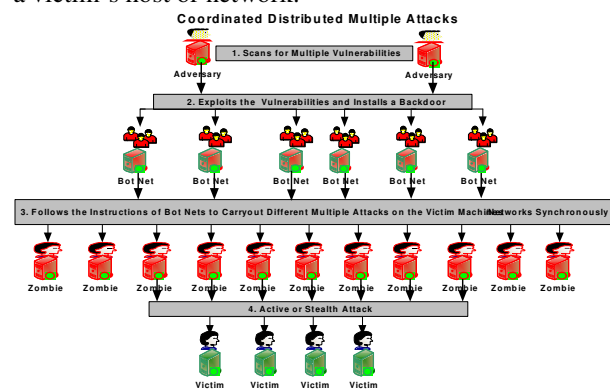


Figure 1: CDMA schematic

A very simple example of a CDMA would be a multi-vector attack. Each attacking node has a different type of attack (whether it is a different vulnerability or a different protocol) that is used throughout the duration of the overall attack.

An example of a multi-vector attack is given in figure 2 below.

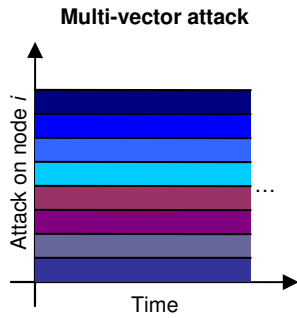


Figure 2: multi-vector attacks

Multiple attack vectors allow for a large target area on the victim which potentially gives the attack a greater chance of succeeding. The attack vectors are statically assigned and do not change over the duration of the attack.

If the attack vectors are distributed over time, the attack constantly changes but only targets one vulnerability at a time. Figure 3 below shows such an attack.

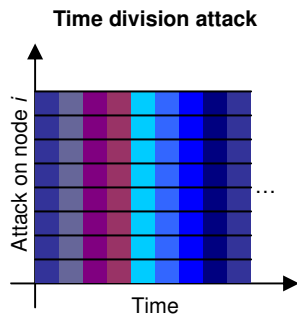


Figure 3: Time division attacks

Time division attacks are potentially harder to detect because each individual attack type (syn flood, udp flood, etc) is relatively short and could be mistaken for a network anomaly. Such an attack requires a higher level of synchronization than traditional DDoS attacks or the multi-vector attacks.

By combining both the multi-vector attack and the time division attack we get an attack that is highly blended with respect to the attack vector and time. We call this type of attack the “checkerboard” attack because of the grid-like pattern it depicts (Figure 4).

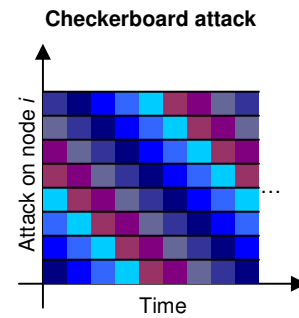


Figure 4: checkerboard attacks

Checkerboard attacks are constantly targeting a wide selection of vulnerabilities and each individual node is continually changing the type of attack it sends. This allows for a very robust attack as well as capitalizing on the stealthy nature of the time division attacks. In this paper we primarily study this type of attack.

In table 1 below are listed the attributes of both traditional DDoS attacks and CDMA. This illustrates the differences between the two attack schemes.

Table 1: Traditional and CDMA attributes

Traditional DDoS	CDMA
Single attack vector	Attack portfolio
Single protocol	Multi-protocol
Single use of bandwidth	Multiple bandwidth uses muxed together
Single target, many-to-one	Many-to-one or many-to-many targeting

Highly blended attacks such as the checkerboard attack pose a few problems for the instigator. Will such a blended attack be effective? The attack vectors may be so diverse and split into such small time segments that they fail to affect the victim node sufficiently. Synchronization between nodes can also be an area of concern if the attack relies on precise timing. It is very unlikely that the network delay between each attacking node and the victim will be constant or consistent and the attacker should expect even more erratic behavior during an attack as various network buffers become exhausted. These are some of the problems we hope to explore in this paper.

4. Methodology

To evaluate the practicality of CDMA a testing lab was built in which several experiments were run. Our goal was to determine whether or not highly blended attacks were possible and if so, just how effective were they upon delivery. We accomplished this by subjecting a victim node to several attacks (both traditional DDoS and

CDMA) and measured various system parameters such as network performance and memory usage.

An idealistic testing environment consisted of several hundred attacking nodes and a victim node connected to the Internet. From here we could study how attack traffic was affected by the random nature of the Internet. Unfortunately, such a setup was beyond our means. Our solution was to approximate the idealistic layout by using two separate computer laboratories on New Mexico Tech’s campus and use the Internet between them as the connecting “cloud”. This allowed us to have a controlled and manageable environment but still introduced random traffic and unknown network configurations. The layout of our testing environment is given in figure 5 below.

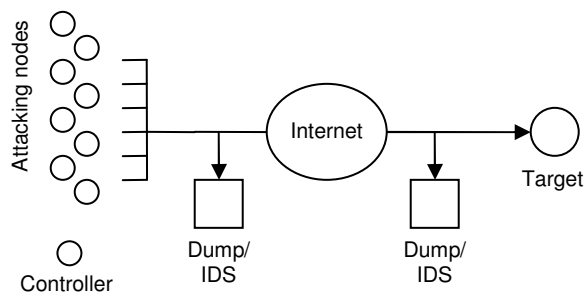


Figure 5: Layout of testing network

As shown in figure 5, Intrusion Detection Systems (IDS) and packet capture programs were installed in between the attacking subnet and the victim subnet. In this paper the IDS is used mainly for validation of the attacks but in future work we hope to use it to explore how stealthy an attack can be. By capturing the packets we can replay them for future tests.

The majority of our experiments attempt to see how well a particular attack affects the victim node, meaning how many resources the attack can consume. To answer this question the victim monitors various performance counters.

Legitimate network packets consume various kinds of shared resources such as bandwidth, memory, processing, and operating system structures. Most of the network peripherals require system and network resources to process the information passing by through the network. An adversary identifies a few activities that are resource intensive and targets the devices with such activity making rendering it nonfunctional. A few possible scenarios of resource exhaustion involve the following: are buffers, file descriptors, address space, disk space, CPU cycle, and bandwidth.

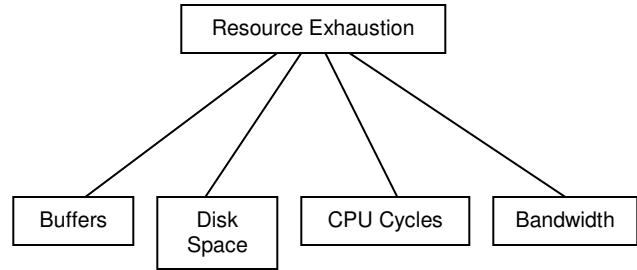


Figure 6: Measuring the Efficacy of an Attack Scheme

IPv4: The number of datagrams discarded (sent and received) as well as the number of datagrams sent or received are recorded.

Main memory: Page faults, page reads, cache bytes, cache limit, and the number of committed bytes are all recorded.

Network interface: The number of packets sent and received, bytes sent or received per second, the number of packets discarded, and the network interface’s output queue length are all recorded.

Physical disk: The current disk queue length, bytes written or read per second, and the number of reads or writes per second are recorded.

Processor: The victim’s CPUs are monitored for interrupts and the distribution of processor time (user time, privileged time, etc).

These performance counters present a breadth view of the node’s health. Some attacks affect different resources and by monitoring the system as a whole we can see how combined attacks affect critical system resources.

4.1 Attack programs

Our initial experiments used eight attack programs. These programs were found “in the wild”, meaning we collected them from several black hat websites. These attacks were chosen because they contained the original source code and were easy to use. All of these programs use a command line interface, which made it easy to write scripts to control them.

The list of attacks used in this paper is given below in table 2.

Table 2: attack programs used

Program	Protocol	Spoofed	Attack type
Ath	ICMP	Yes	Bad data
Bomba	IGMP	No	Oversized packet
Bonk	UDP	Yes	Bad offset
jolt2	ICMP, UDP	No	Fragmentation
Kod	IGMP	No	Fragmentation

Smurf	ICMP, UDP	Yes	Ping flood
Suf	UDP	Yes	UDP flood
Syn	TCP	Yes	Syn flood

To validate the attack scripts several calibration attacks were performed on the victim. Snort was used as the IDS and was placed before the sending nodes and before the victim. If an attack was valid it must (1) be detected by the IDS and (2) consume some resource on the victim. All eight scripts were launched from all of the attacking nodes in a traditional DDoS attack.

4.2 Control code

The control code acted as the glue that held all the attack programs together. Because we did not have a complex routing scheme amongst the attacking nodes a simple client-server relationship was sufficient.

All control information was sent using UDP. Although this is an unreliable protocol and loss may occur (especially during heavily congested conditions such as during an attack), we were able to determine that losses were quite rare in our experiments. All control information was sent while the attacking nodes were idling. Since we used a lab environment, regular corruption problems were not a concern. UDP was chosen because it is quick and easy to implement

Node synchronization

One problem faced by the controller was synchronizing all the attacks. This was accomplished by first synchronizing all nodes to the controller and then instructing them to attack at a set time. The controller accomplished this by sending out its current time to all nodes from which the nodes computed a time offset by converting from local time to controller time.

Once the clocks were synchronized the controller informed the nodes of an absolute starting time which was a few seconds in the future. Each node received the same starting time (relative to the controller's clock) and began to count down until the attack. This allowed us to avoid most of the congestion and have a reasonably synchronized attack.

5. Results

The CDMA experiment used in these comparisons was a checkerboard attack with the attacks changing every second. This illustrates a heavily interleaved attack with a large number of attack vectors.

Our experiments were focused on exploring the feasibility of CDMA. We measured the performance of the victim machine during the calibration experiments (using traditional DDoS attacks) and compared this to the

performance during a full checkerboard attack. Figure 7 below shows the processor usage on the victim during a highly blended attack compared to the average processor usage for the traditional DDoS attacks.

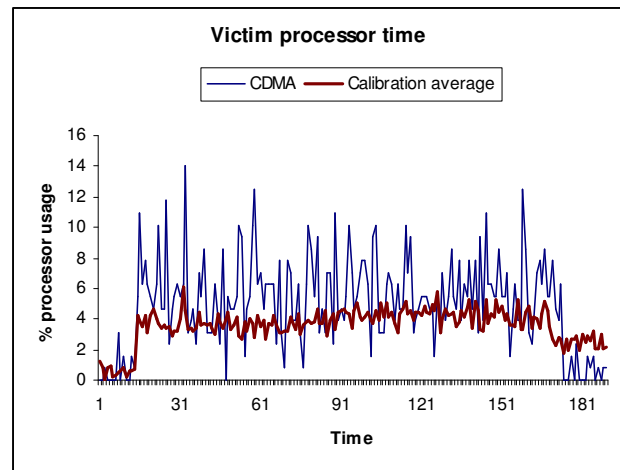


Figure 7: Victim processor time

The other performance counters followed the same trend as the processor usage: highly blended attacks are at least if not more effective than traditional DDoS attacks. One critical resource the attacker must take into consideration is bandwidth. A low bandwidth attack that is just as damaging as a high bandwidth attack is more desirable. Figure 8 below shows that CDMA's use (on average) slightly less bandwidth than their traditional counterparts, despite periodic spikes. Given the complex nature of the new attack this is a very desirable attribute.

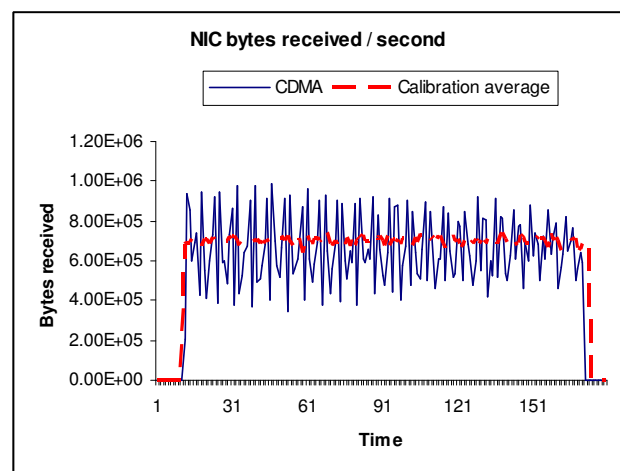


Figure 8: Victim bytes received

The attack portfolio used in this paper utilized only one TCP attack (syn flood) and relied on other protocols to do most of the work. In a real-world network one would expect to see a significant amount of TCP traffic

and using a larger proportion of TCP attacks could potentially cause more damage than other protocols. This deficiency in the experiments was particularly noticeable when the victim's output queue length (the number of datagrams waiting to be sent) during a syn flood was compared to a CDMA attack. During the syn flood the queue was severely affected but managed to recover during the CDMA.

As the time between attack changes decreased the desynchronization between nodes increased. This was due to normal network delays but future efforts will need to take this into consideration. A "rigid" attack is apt to fail and our future research will explore flexible attack schemes that capitalize on the benefits of CDMA's but are still practical in a real-world scenario.

6. Conclusion

One of the goals of this research was to explore the feasibility of launching a highly blended and distributed attack. Our results show that it is very easy to conduct such an attack using simple control code and open source black hat tools. However, other areas were not covered. Detecting CDMA's is one large area for future research that we will be spending considerable time on. We will also be looking into more sophisticated attacks and large network simulation.

The extent to which the attacks became blended was an area we also wanted to explore. As the time between attack changes decreases the attacks may lose some of their potency due to overhead and other network nuances. This may be true for attacks blended down to the packet level but we successfully launched blended attacks that changed every second, which has yet to be seen in the wild.

During the experiments, we observed an increase in the desynchronization among the attacking nodes and can predict that the desynchronization will continue to rise as the attacks become more and more blended. Future attack schemes will need to be resilient to variations among the nodes, perhaps only synchronizing the nodes belonging to the same subnet.

The deployment strategy of a CDMA network will also need to be carefully considered. Running only eight attack programs on eight different nodes forced us to spend several days working out the operating system and hardware configurations. Each protocol may be blocked at the node or at the router and a large number of mis-configured nodes can render the attack ineffective. This is also an area for future research.

Acknowledgments

Support for this research received from ICASA (Institute for Complex Additive Systems Analysis, a

division of New Mexico Tech), DoD IASP, and NSF SFS Capacity Building grants are gratefully acknowledged. We would also like to acknowledge several insightful discussions with Senthil Rajan that helped clarify our ideas.

7. References

- [1] W. J. Blackert, D. C. Furnage, and Y. A. Koukoulas, "Analysis of Denial of Service Attacks Using an Address Resolution Protocol Attack", Proc. of the 2002 IEEE Workshop on Information Assurance, US Military Academy, pp. 17-22, 2002.
- [2] T. Draelos, et. Al, "Distributed Denial of Service Characterization," *Technical Report*, Sandia National Laboratories, 2003.
- [3] J. Mirkovic and P. Reiher, A Taxonomy of DDoS Attacks and Defense Mechanisms, ACM SIGCOMM Computer Communications Review, Volume 34, Number 2, April 2004, pp. 39-54.
- [4] D. W. Gresty, Q. Shi, and M. Merabti, "Requirements for a general framework for response to distributed denial of service," Proc. Of Seventeenth Annual Computer Security Applications Conference, pp. 422-229, 2001.
- [5] K Houle, G. Weaver, "Trends in Denial of Service Attack Technology", CERT Coordination Center, 2001, http://www.cert.org/archive/pdf/DoS_trends.pdf
- [6] S. Staniford, V. Paxson, N. Weaver, "How to Own the Internet in Your Spare Time", Proceedings of the 11th USENIX Security Symposium, August 2002.
- [7] K. Kendall, "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems", *Master's Thesis, Massachusetts Institute of Technology*, 1998.
- [8] S. E. Webster, "The Development and Analysis of Intrusion Detection Algorithms", *S.M. Thesis, Massachusetts Institute of Technology*, 1998.
- [9] C. Shields, "What do we mean by network denial of service?", Proc. of the 2002 IEEE workshop on Information Assurance. US Military Academy, pp. 196-203, 2002.
- [10] S. Mukkamala, A. H. Sung, "Coordinated Distributed Multi Attacks (CDMA)", Proc. of IEEE International Conference on Advances in Intelligent Systems Theory and Applications, IEEE Computer Society Press, ISBN 2-9599776-8-8, PID 011-04.