# Ensemble-based DDoS Detection and Mitigation Model

Sajal Bhatia Desmond Schmidt George Mohay

Information Security Institute
Queensland University of Technology
GPO Box 2434, Brisbane 4001, Queensland, Australia
{s.bhatia, desmond.schmidt, g.mohay}@qut.edu.au

## ABSTRACT

This work-in-progress paper presents an ensemble-based model for detecting and mitigating Distributed Denial-of-Service (DDoS) attacks, and its partial implementation. The model utilises network traffic analysis and MIB (Management Information Base) server load analysis features for detecting a wide range of network and application layer DDoS attacks and distinguishing them from Flash Events. The proposed model will be evaluated against realistic synthetic network traffic generated using a software-based traffic generator that we have developed as part of this research. In this paper, we summarise our previous work, highlight the current work being undertaken along with preliminary results obtained and outline the future directions of our work.

## Categories and Subject Descriptors

C.2.0 [**General**]: Security and protection (e.g., firewalls); D.4.6 [**Security and Protection**]: Invasive software (e.g., viruses, worms, Trojan horses); K.6.5 [**Security and Protection**]: Unauthorized access (e.g., hacking, phreaking)

## General Terms

Security

## Keywords

DDoS attacks, Network Traffic Analysis, Flash Events, Modelling, Synthetic Traffic Generation, MIB Data Analysis

## 1. INTRODUCTION

A *Denial-of-Service* (DoS) attack is an explicit attempt by an attacker to disrupt an on-line service or make it unavailable to its legitimate users by overwhelming the service provider (or server) with a large number of requests. Some of these attacks ('flooding attacks') intend to saturate the target server's network bandwidth while others aim at consuming the available computing resources like CPU and memory. One of the earliest known DoS attacks occurred in 1974[1] at the Computer-based Education Research Laboratory (CERL), at the University of Illinois Urbana-Champaign. A novice programmer forced 31 computers in the laboratory to power-off by exploiting the default configuration of PLATO terminals to accept remote 'exts[2]' commands while running TUTOR as the programming language. The first reported large-scale DoS attack using the public Internet occurred in 1999 at the University of Minnesota [9]. In February 2000, popular websites Yahoo, eBay, and CNN were attacked and flooded with a large number of requests, thereby forcing them off-line and causing huge financial losses [10].

The focus of our research is the distributed form of DoS attack known as Distributed Denial of Service (DDoS). In this form of attack an aggregation of geographically scattered compromised machines (or 'bots') are controlled by a bot-master to attack a single victim. This distributed array of compromised systems is also known as a *Botnet*. Some recent targets of DDoS attacks have been in Estonia (2007) [15], Georgia (2008) [8] and against PayPal (2010-2011)[3]. More than a decade since the first attack was reported [9], and notwithstanding the amount of research done in this area, DDoS attacks in various guises still exist and continue to constitute a pernicious threat to the Internet community. Their attack vectors are continuously evolving and hence the problem is still far from being completely resolved. Thus, developing techniques for efficient DDoS attack detection and mitigation continues to be an active and important area of research.

Some of the key challenges in developing a practical solution to the problem are:

- The detection mechanism should be capable of identifying a variety of network-level and application-level flooding attacks.
- Any detection mechanism should be capable of initiating real-time or near real-time mitigation strategies to alleviate the impact of a DDoS attack.
- The evaluation of any proposed solution should be conducted on realistic datasets, either synthetically generated or publicly available, rather than on commonly

*SIN'12,* October 25–27, 2012, Jaipur, India

[1] http://www.platohistory.org/blog/2010/02/perhaps-the-first-denial-of-service-attack.html

[2] an 'ext' command would make the PLATO terminal, connected to an external peripheral device to lock itself.

[3] http://www.pcworld.com/businesscenter/article/212701/

used decade old datasets like KDD [1] or using DDoS attack tools like Trinoo, many of which do not work now without considerable modifications.

- The proposed system should be able to efficiently differentiate DDoS attacks from Flash Events which share a number of characteristics with DDoS attacks [12].

The term *Flash Event* (FE) is used to refer to a situation when a large number of legitimate clients concurrently accesses a web-server, either following a newsworthy event or when redirected from popular web-sites like Slashdot or other social media. Both DDoS attacks and FEs are accompanied with high levels of incoming traffic, often leading to Quality-of-Service (QoS) degradation. Therefore, it is important for any viable DDoS attack detection solution also to accurately detect FEs and distinguish them from DDoS attacks as different actions need to be undertaken by the network administrator upon identification of either.

In this paper we propose a model, the DDoS Detection and Mitigation Model (henceforth D2M2) to address these challenges, and present a partial implementation of D2M2 together with a description of what remains to be done. D2M2 is intended to detect a variety of DDoS attacks and to distinguish them from FEs. In order to test and evaluate our model and the techniques we have implemented, we need also to be able to generate realistic test data because public datasets are too limited and have their own limitations like age, pseudonymized IP addresses, and zero payloads. Our work on DDoS attack detection is taking place in two phases: detecting DDoS attacks through the analysis of simple network traffic properties like source IP's, and using an ensemble of network traffic analysis and MIB (Management Information Base) server load data analysis. Using these two strategies, we expect not only to be able to detect different types of network and application-layer based DDoS attacks, which can go undetected by individual detection techniques, but also to differentiate DDoS attacks from FEs which share some similar characteristics.

This work-in-progress paper presents the conceptual design of the proposed model and our work to date in implementing and evaluating a partial prototype. In addition we describe the current work being undertaken on synthetic traffic (DDoS attack and FE) generation using a software-based traffic generator tool and an experimental test-bed, being developed as a part of this research. The paper also gives an overview of the work to be done in future towards the completion of our prototype.

The remainder of the paper is structured as follows. Section 2 gives an overview of related work relating to the two DDoS attack detection strategies used in our work viz. network traffic data analysis, and MIB data analysis. Section 3 presents a conceptual design of the proposed model and summarises our previous work on network traffic analysis for DDoS detection and FE classification. Section 4 details the current work we are undertaking and Section 5 outlines the future directions of the research work. Finally, Section 6 provides a summary of the work presented in the paper.

## 2. BACKGROUND AND RELATED WORK

The literature suggests that the DDoS attack detection problem has been viewed broadly from two different directions: incoming network traffic analysis and MIB data analysis. While the former encompasses the analysis of different network traffic parameters to detect the occurrence of DDoS attacks, the latter focuses on the statistical analysis of MIB data gathered from SNMP (Simple Network Management Protocol) agents. This section of the paper summarises some of the recent pertinent work done in each of these directions along with an overview of the research done in differentiating DDoS attacks from similar looking FEs.

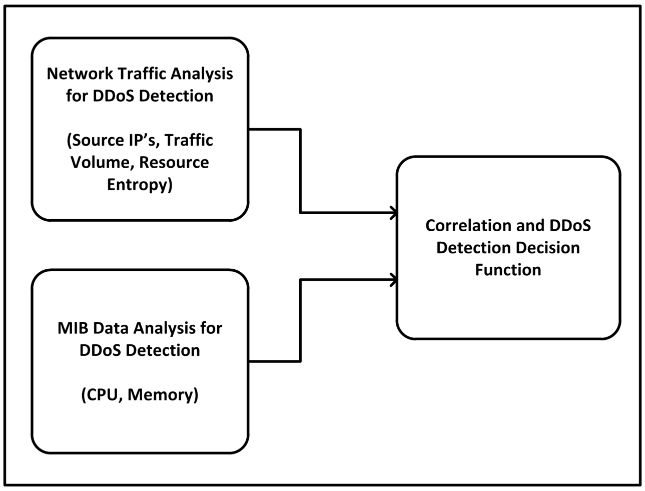
Network Traffic Analysis for DDoS Detection
(Source IP's, Traffic Volume, Resource Entropy)
MIB Data Analysis for DDoS Detection
(CPU, Memory)
Correlation and DDoS Detection Decision Function

**Figure 1: Ensemble-based DDoS Detection and Mitigation Model**

DDoS detection based on network traffic analysis focuses on either signature-based or anomaly-based analysis of incoming and outgoing network traffic. Research presented in [14] detects DDoS attack traffic by analysing the TCP/IP packet headers against some pre-defined rules. Jin et. al [11] present a covariance model using the flags in control fields of the TCP header to detect SYN flooding attacks. Research in [20] used statistical analysis of four macro level IP flow based features: average number of packets per flow, percentage of correlative flow, 'one direction generating speed' and 'ports generating speeds' for filtering DDoS attack traffic. Peng et.al [18] proposed a historical source IP address based technique to filter out the attack traffic at the edge router. The proposed technique maintained a database of the source IP addresses which completed a three-way TCP handshake. The IP Address Database (IAD) was used to decide whether to accept the incoming packets. Only the traffic originating from source IP addresses present in the IAD were allowed access. However, the IAD was potentially at-risk of being corrupted by those source IP's which had first completed the TCP handshake but later on participated in the attack.

The other approach for DDoS attack detection relating to our work is based on analysing MIB data collected via SNMP agents. Research presented in [21] and [5] used SNMP MIB statistical data, instead of raw packet data, and proposed a flooding attack detection mechanism. Their mechanism used SNMP MIB variables from IP, TCP, UDP, and ICMP groups and a Support Vector Machine (SVM) for classifying attack traffic into TCP-SYN, UDP and ICMP based flooding attacks. [17] used 16 MIB variables from 6 groups (System, Interface, IP, TCP, UDP and ICMP) and proposed a threshold-based flooding attack detection mechanism. One of the characteristics observed in literature relating to SNMP MIB data analysis based DDoS attack de-

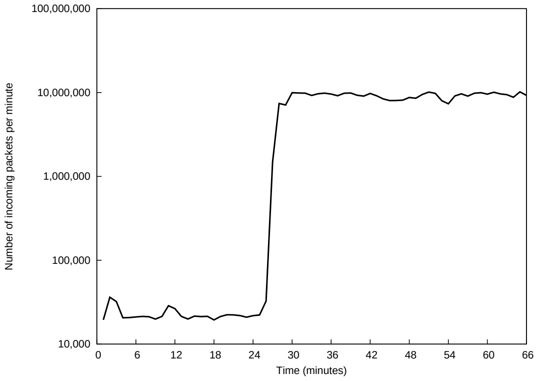

**Figure 2: Incoming traffic profile for CAIDA DDoS attack dataset**

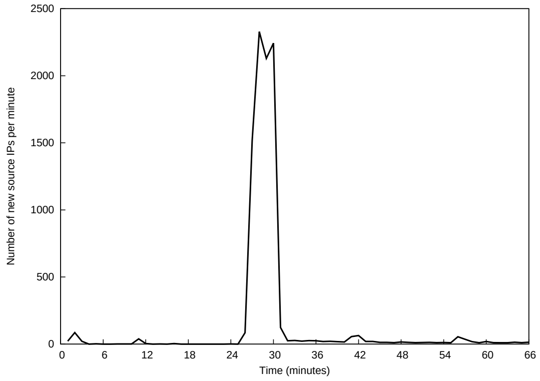

**Figure 3: New Source IP addresses for CAIDA DDoS attack dataset**

tection was the use of a rather constant set of network-traffic related MIB variables like ipInReceive, tcpAttemptFails, icmpInDestUnreachs etc. Therefore, the techniques using this set of MIB variables focused only on detecting network layer flooding attacks like TCP-SYN flood. However, the proposed D2M2 aims to detect application layer flooding attacks, in addition to the network layer flooding attacks, for which it uses server-load based MIB variables i.e. CPU and memory utilisation.

A majority of the attack detection techniques focus on DDoS attacks without taking FEs into account, although FEs share some similar characteristics with DDoS attacks. Some of the approaches which do consider FEs, tag them as an abnormal activity without differentiating them from DDoS attacks. FEs originate from legitimate clients as opposed to DDoS attacks which usually come from compromised machines, so one current research challenge is to differentiate activity generated by humans from that generated by (compromised) machines.

Amongst some of the techniques available to address this challenge are graphical puzzles or CAPTCHA's which have been rather heavily used. Research in [13] used these puzzles to propose a system to protect web-servers based on probabilistic authentication. Use of CAPTCHA's however introduces additional delays for legitimate clients when accessing websites. Research presented in [12, 16] proposed methods to distinguish DDoS attacks from FEs. However, their research was based on datasets not available in the public domain thus making it difficult to validate their results. In our previous work, we analysed some publicly available datasets and present three parameters which could be used to differentiate DDoS attacks from FEs [7], and subsequently proposed a server-side model of FEs [6] with a view to generating realistic FE traffic.

# 3. WORK TO DATE

In this section we present our model and summarise the work completed so far towards its implementation. In addition to developing the model, the work to date has mainly focused on network traffic analysis based DDoS detection, its differentiation from similar looking FEs and modelling FEs to facilitate their synthetic traffic generation.

## 3.1 DDoS Detection and Mitigation Model

In order to accurately identify a wide variety of network and application layer DDoS attacks and mitigate their effect, we propose a DDoS Detection and Mitigation Model (D2M2). The proposed model uses an an ensemble of two different DDoS attack detection strategies: network traffic analysis based DDoS detection and MIB (server-load) data analysis based DDoS detection. The combination of these two strategies is expected not only to detect a wide range of DDoS attacks which can go undetected by individual strategy used in isolation, but also to be able to differentiate DDoS attacks from similar looking FEs. Figure 1 presents a conceptual design of our proposed model.

## 3.2 DDoS Attack Detection using IP Addresses

A popular dataset utilized by DDoS researchers has been the CAIDA "DDoS Attack 2007" Dataset [2] which contains approximately one hour of pseudonymized traffic traces from a DDoS attack on August 4, 2007 (20:50:08 UTC to 21:56:16 UTC). The characteristics of this attack in terms of traffic and previously unseen or new source IP addresses are presented in Figures 2 and 3 respectively. Figure 3 shows a dramatic increase in the number of previously unseen IP addresses at the commencement of the attack. This has motivated one of the key questions identified for building the proposed D2M2: *How accurately can the onset of a DDoS attack be detected using simple incoming network traffic properties like source IP's?* In order to address this research question, we proposed a DDoS detection algorithm using 'source IP address' as the analysis parameter [3]. The proposed attack detection algorithm comprised of two main functions:

- *ipac* function for classifying source IP addresses, and
- *ddos* function for identifying high-rate flooding attacks

The *ipac* or IP address classification function extracts the source IP address information from the incoming network traffic and determines whether the source IP address has been seen previously or is new (Not Seen Previously - NSP). In order to perform this classification, *ipac* maintains two lists of source IP addresses: W (White-list) and R (Recent). The list W is initialised with the source IP's of known attack-free or normal traffic and the list R is used to temporarily hold the NSP source IP's and copy them to W once it has been determined that the system is no longer under attack. R is used to avoid polluting the white-list.

The *ddos* function maintains two states: A (under attack) and NA (not under attack). The *ddos* function is invoked at regular time-intervals to analyse the rate of incoming packets from NSP source IP's. It uses Cumulative Sum Algorithm (CUSUM) to detect the abrupt changes in the rate of incoming packets, based on which it determines the change of state from NA to A or vice-versa. Bit vectors are used to implement lists W and R used in the algorithm. Currently, the NSP algorithm has been implemented for IPv4 ($2^{32}$) address space and requires 0.5 GB of storage space. Figure 4 gives a diagrammatic representation of the NSP algorithm.

Our previous work [3] proposes a technique for detecting high-rate flooding attacks using source IP's and presents a proof of concept implementation using bit vectors. The paper shows how a simple network traffic parameter can be used to effectively detect high-rate flooding attacks. One of the limitations identified in our approach is the fact that when the system is under attack all the new source IP's are treated as malicious. This can potentially give rise to false positives. Our future work will attempt to address this limitation by extending the rejection criterion beyond NSP source IP's. The further work will also attempt to use more features such as the distribution of IP's for detecting DDoS attacks.

## 3.3 Parametric Differences Between DDoS Attacks and FEs

DDoS attacks can often be mistaken for FEs since both of them share various similar characteristic features. Therefore, it is important to accurately detect DDoS events and to distinguish them from FEs. Preliminary work done in this direction attempts to address the related research question: *In what circumstances and how precisely can we distinguish between a DDoS attack and a FE?*

Research presented in [7] addresses this question by analysing two publicly available datasets, each representing a real-world DDoS attack (CAIDA 'DDoS Attack 2007' Dataset [2]) and an FE ('1998 FIFA World Cup' Dataset [4]), and proposes a set of parameters which could potentially be used to separate the two network activities i.e. DDoS attacks and FEs. The proposed parameters, though not entirely orthogonal, are capable of capturing different aspects of network traffic, which can then help in distinguishing DDoS attacks from FEs. The three proposed parameters are:

1. Change in rate of incoming traffic
2. Change in rate of new source IP addresses
3. Distribution of requests among source IP addresses

The rationale behind the first parameter is that in case of a DDoS attack the participating machines or bots are programmed to send packets at pre-defined rates. In order to maximise the intended damage the bots are concurrently triggered by the bot-master and instructed to send traffic in huge volumes. This characteristic traffic pattern differs from a FE where it takes a finite amount of time for the news to spread across the web community. Hence, the rate of incoming traffic, as observed by the target server, is not as dramatic as in case of a DDoS attack and thus could be could be used as one of the differentiating features.

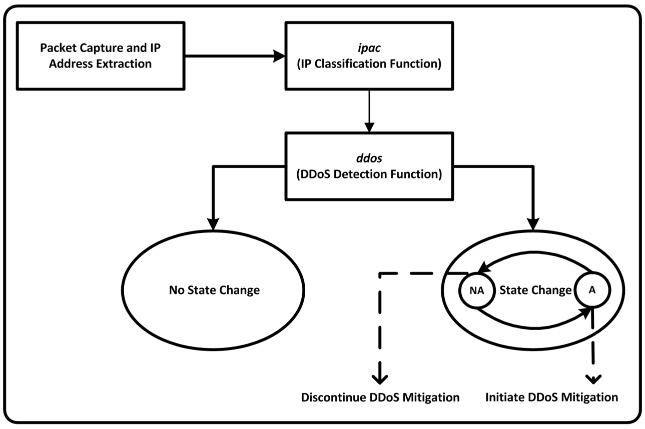
Packet Capture and IP Address Extraction
ipac
(IP Classification Function)
ddos
(DDoS Detection Function)
No State Change
NA
State Change
A
Discontinue DDoS Mitigation
Initiate DDoS Mitigation

**Figure 4: DDoS Detection using NSP source IP's**

The second proposed parameter (change in rate of new source IP's) is based on a recent research which shows that a typical size of a Botnet's *live population* (number of active bots) is often limited to a few thousand machines [19]. As each bot has limited capabilities, in order to create the desired impact they are simultaneously activated by the bot-master. This leads to an abrupt change in the previously unseen source IP's by the victim with the onset of the attack. This relatively small and finite set of available bots (as compared to the number of legitimate clients in a FE) forces their re-use during the course of the attack resulting in a minimal or no change in the rate of new source IP's as observed by the victim. This characteristic differs from a FE scenario where the spread of news brings 'new' clients on-line to access the information and hence the target server continuously experiences new source IP's.

The final parameter is based on the argument that during a DDoS attack, as the attacker attempts to maximise the utilisation of the finite available resources, it often forces each bot to send large amounts of traffic. Thus, the number of packets per bot is high and the entire outgoing traffic is more or less evenly distributed amongst participating bots. Whereas during a FE, barring some enthusiastic clients, most clients are only usually interested in information specific to the event. Therefore, comparatively fewer requests originate from each client during a FE as opposed to a DDoS attack. Thus, a distribution of requests amongst clients could be another distinguishing parameter between DDoS attacks and FEs.

In our previous work [7] we analysed two publicly available datasets and proposed a set of parameters (change in rate of incoming traffic, change in rate of new source IP addresses, and distribution of requests among source IP addresses) which, in conjunction, could be used to efficiently differentiate between DDoS attacks and FEs.

## 3.4 Modelling Flash Events

Even though our previous work [7] identified a set of parameters which could possibly differentiate DDoS attacks and FEs, the problem of having real FE datasets for testing and evaluating the D2M2 was yet to be resolved. A very limited number of FE datasets are available in the public domain. Moreover, a majority of these datasets are web-server logs in Common Log Format (CLF) which makes it difficult to replay them over the network for experimentation purposes. Our approach to address this problem i.e. lack of FE datasets, was to model FEs and use that model to generate synthetic FE traffic closely approximating the real-world scenarios.

We present our work in this regard in [6] in which we present a detailed study of FEs and their classification into three broad categories: *predictable*, *unpredictable* and *secondary.* In that work, we describe a FE according to three key components: the volume of incoming traffic, the related source IP's and the accessed resources, and use them to propose a server-side model for FEs.

We consider a FE to comprise of two major phases: a *flash-phase* and a *decay-phase* and argue the near absence of any *sustained* or *plateau* traffic phase. We use the incoming traffic volume component, its increase and subsequent decrease, to propose a simple exponential model. Our analysis also shows that during a FE, the variation in source IP's closely resembles the variations in incoming traffic volume. This resemblance is based on our observation that the overall increase in incoming traffic during a FE is mainly due to a substantial increase in the interested clients rather than to an increase in the number of requests per client.

Another key observation presented in our paper [6] that could potentially be used to detect FEs and distinguish them from DDoS attacks was the *randomness of the accessed resources.* This characteristic was based on the speculation that during a FE, most clients are interested in specific information related to that event. Thus, the number of distinct web-resources being accessed during a FE should be lower as compared to the non-flash-event times. One way to measure this characteristic is using Shannon's entropy, a measure of uncertainty associated with a random variable. In our analysis, a *unique web-resource* represented the random variable for calculating the resource-entropy. The analysis of a public domain dataset, presented in the paper, confirmed our speculation.

Our previous work [6] presents a FE model using a small set of configurable parameters and validates the proposed model using some publicly available datasets representing different types of FEs. The proposed mathematical model, including the resource-entropy characteristic is currently being used to generate realistic FE traffic.

# 4. CURRENT WORK

Realistic network traffic, for both DDoS attacks and FEs, is essential for testing and evaluating our proposed model and its implementation. As with FEs, very few datasets representing real DDoS attacks are available in the public domain, mainly due to the associated legal and privacy issues. Our proposed solution to this problem is synthetic traffic generation. The current work being undertaken is aimed at addressing this essential requirement and in the process finding a suitable answer to the research question: *How can the network traffic be parameterised and synthetically generated to closely approximate the real-world traffic?*

## 4.1 Synthetic Traffic Generation

Some of the key network traffic characteristics identified for testing and evaluating the proposed D2M2 are: firstly, since the source IP address has been identified as one of the key parameters which would be used to detect DDoS attacks and separate them from FEs, the packets in the network traffic should cover a *wide spectrum of source IP's.* Secondly, the proposed D2M2 would correlate the network traffic analysis with the MIB server load data analysis for detecting DDoS attacks. Therefore, it is important that the packets have a *valid source and destination IP address* to ensure that a valid TCP-level connection is established between the host and the target. And finally, for emulating real-world application-level attacks and FEs, it is required that the packets contain *valid data.* An additional requirement identified as a part of synthetic traffic generation is that the transmission pattern of the synthetic data should be a controllable mix of 'normal and attack' traffic for an efficient evaluation of D2M2.

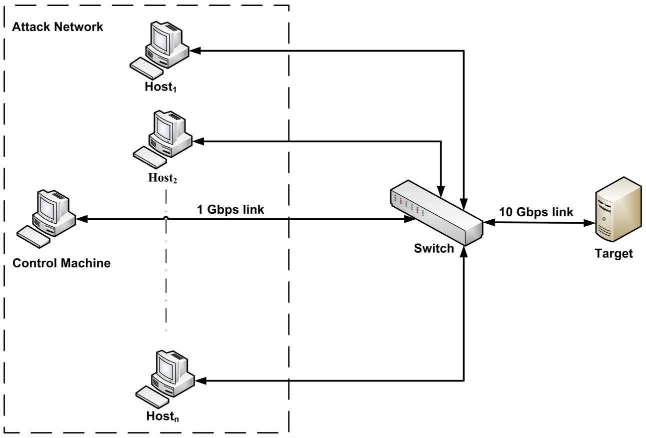

**Figure 5: Test-bed architecture**

To generate network traffic with aforementioned characteristics, the experimentation began with some widely referenced open source traffic generation tools like D-ITG[4], hping[5]. However, it was discovered that most of these tools were not designed for generating DDoS attack and FE traffic with the characteristics required for our research. Moreover, most of these tools were written some years ago and thus required modifications.

Another option explored for generating synthetic network traffic with the desired characteristics was to use IP-aliasing combined with the GNU Wget tool[6]. IP aliasing is a well-known technique, available on most computing platforms, for assigning a large number of distinct IP's to a single hardware (Network Interface Card or NIC) address. Wget is a command line utility for retrieving files using HTTP, HTTPS and FTP protocol. Thus, using IP-aliasing with Wget could potentially create an appearance of a set of 'bots' or 'legitimate clients', each one with a valid source IP and hardware address, to be used for emulating realistic DDoS attacks and FEs respectively. This approach however lacked the required scalability and was also heavy on host-machines' available resources.

The idea of using IP-aliasing with Wget however formed the basis of a software-based traffic generator, Botloader, developed as a part of this research. Botloader uses low-level system calls to Linux kernel for creating IP aliases and binds them to the NIC. Each 'bot' (aliased IP) has its own

[4] http://www.grid.unina.it/software/ITG/
[5] http://www.hping.org/
[6] http://www.gnu.org/software/wget/

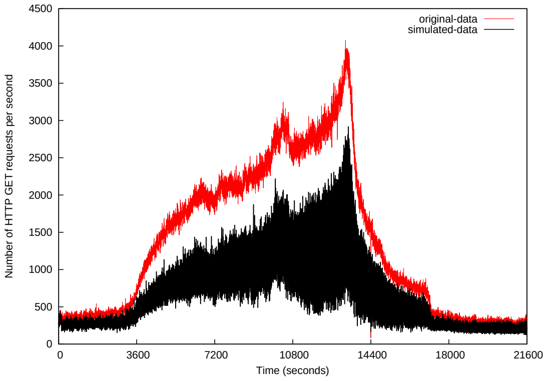

**Figure 6: Synthetic FE ($1^{st}$ Semi-final) data: incoming traffic volume**

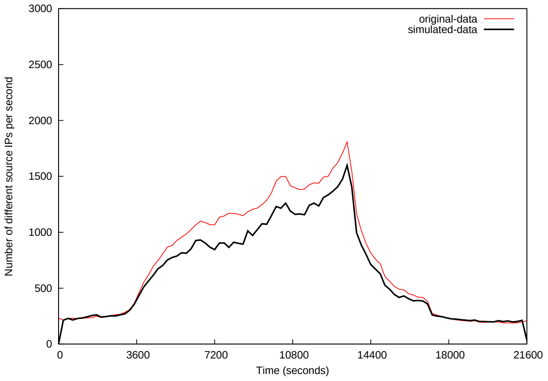

**Figure 7: Synthetic FE ($1^{st}$ Semi-final) data: different source IPs**

unique IP address and a network socket, and has access to a set of shared libraries, each representing a different attack type, and responds to commands specified in configuration file. At present, libraries for PING, TCP-SYN, UDP, and HTTP based attacks and FE traffic have been developed.

## *Experimental Set-up*

The experimental test-bed consists of 10 *host machines* hosting 'bots' (for DDoS attacks) or 'legitimate clients' (for FEs), controlled by a *control machine* and collectively sending traffic to a *target* server. The control machine acts like a 'master' and issues commands to initiate instances of Botloader in each of the participating hosts. All the machines in the attack network are connected via a 1 gigabit per second (Gbps) link to a layer 3 switch, which is further connected to the target machine via a 10 Gbps link. All the traffic coming from the attack network gets accumulated and is sent through the 10 Gbps link directly connected to the target machine. All the machines, except the target server, are standard PCs with 3.0 GHz Intel Core2 processors, 4 GB of memory and an integrated 1 Gigabit (Gb) network interface card. They are running Ubuntu 10.04 Desktop as the operating system (OS) and are connected via Dell PowerConnect 6224 switch. The target server is a Dell PowerEdge R710 with two Six-core Intel Xeon 2.27 GHz processors (hyperthreaded) and 32 GB of memory. It runs on Ubuntu 10.04 (Server) and uses Apache2 as the web-server.

In our initial configuration of the test-bed, all the ports of the attackers and the target were directly connected to the switch, and packets were redirected via layer 2 switching. Each of the 10 host machines was configured with 6,400 bots with aliased source IPs bound to the host hardware (MAC) address. Each of the 64,000 bots were assigned a unique source IP address within the class B IPv4 address space. Having all the host machines in a single VLAN in a class B IPv4 address space, enforced a theoretical maximum on the number of usable source IPs to close to 64,000.

Problems with this design soon surfaced when we tried to run the FE-emulation. The switch was designed to handle up to 896 IP-addresses, but we were flooding it with traffic from 64,000, because this closely mimicked the number of IP addresses in the original data (79,033). This also led to the target server having to maintain an ARP-table of 64,000, since without an entry to map the IP-address of outgoing packets to the required MAC-address, the destination of each packet would be unknown. This slowed down the server to the point where the overall traffic flow could not keep pace with the required emulation of nearly 4,000 requests per second at its peak. The experimental setup was changed and the network was divided into two separate VLANs, one to host the attack machines and one for the target, shown in Figure 5. In this set-up, the layer 3 switch was used as a router by enabling the inter VLAN routing.

## *Preliminary Results*

Using the set-up described above, experiments were conducted using Botloader in order to generate synthetic FE traffic. Traffic around the $1^{st}$ World Cup Semi-final match was used for synthetic traffic generation. Statistical analysis was performed on this data to compute parameters like number of packets, number of different source IPs, number of different resources accessed and normalised resource entropy per one second of sampling interval. This information was then used by Botloader to generate synthetic traffic.

Figure 6 compares the original and simulated data for the incoming traffic volume for the $1^{st}$ Semi-final match represented as the number of HTTP GET requests per second. The simulated FE traffic closely follows the original traffic however it fells somewhat short of the number of HTTP GET requests in the original data.

A comparison of the number of different source IPs per second registered on target server is shown in Figure 7. The number of different source IPs in the simulated traffic closely mimics the original FE data, although the total number of different source IPs seen by the target during the entire simulation is slightly less than expected. Out of the 64,000 different source IPs assigned to bots, 57,214 were used during the simulation. This deviation is an experimental artifact and is currently being researched. Using a larger address space (IPv4 class A) or configuring each host machine on a separate VLAN could potentially improve the simulation results.

During a FE, a majority of the user population is often interested in a specific set of information related to the event.

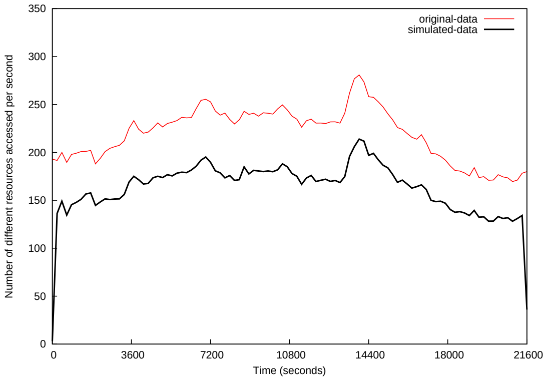

**Figure 8: Synthetic FE ($1^{st}$ Semi-final) data: different resources accessed**

This characteristic transforms to a rather constant request per client. Therefore, the number of different clients (source IPs) vary in a similar fashion as the incoming traffic volume, as seen in Figures 6 and 7

Figure 8 compares the different resources accessed for the original and synthetically generated FE data. The obtained results deviate from the original data and needs further investigation. Figures 7 and 8 have been smoothed using 'bezier' technique available in gnuplot.

The preliminary results show that the simulated FE data closely follows the original data particularly for different source IPs used for generating synthetic traffic. However, additional work is required to more closely mimic the incoming traffic volume and the resource access pattern of the original data and further improve the existing results. These issues are currently being investigated along with synthetic generation of different types of FEs and DDoS attacks.

# 5. FUTURE-WORK

Our overall objective for the project is to answer the following challenge: *How to use a minimal yet sufficient range of parameters to reliably detect DDoS attacks and at the same time mitigate its effects?* As a result our future remaining work in the project is to complete the prototype implementation of D2M2. This has been divided into two main phases: correlating network traffic and MIB data analysis for detecting DDoS attacks, and mitigating their impact upon detection. Both the phases are briefly discussed below.

## 5.1 Ensemble-based DDoS Detection

As the first part of our future work, we aim to address the research question: *How can the network traffic characteristics be correlated with server load (MIB) data to improve the detection of DDoS attack?*

To address this question we propose an ensemble DDoS detection technique that would combine the two different detection strategies i.e. network traffic analysis based DDoS detection and MIB data based DDoS attack detection. The proposed ensemble would correlate network traffic parameters (source IPs, incoming traffic volume, and entropy of resources accessed) with the server load MIB variables (CPU and memory utilisation) to detect a wide range of network and application level DDoS attacks. These parameters would be correlated using (logical) OR, AND and weighted mean, and their outcomes would be compared to find an optimal correlation technique for the proposed parameters. Figure 1 shows an abstract-level description of the ensemble-based DDoS attack detection.

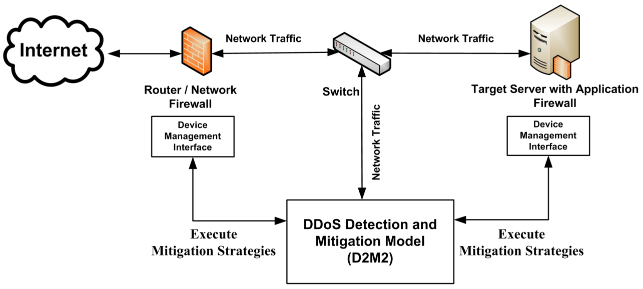
Internet
Network Traffic
Network Traffic
Router / Network Firewall
Switch
Target Server with Application Firewall
Device Management Interface
Network Traffic
Device Management Interface
Execute Mitigation Strategies
DDoS Detection and Mitigation Model (D2M2)
Execute Mitigation Strategies

**Figure 9: Deployment architecture of D2M2**

## 5.2 DDoS Mitigation

The second part of the future work would aim at incorporating some DDoS mitigation strategies into the proposed D2M2. This would address another key requirement of a workable DDoS attack detection system i.e. *How efficiently a response, in real-time or near real-time, can be activated to mitigate the impact of a DDoS attack?*

Figure 9 gives an off-line deployment architecture of the proposed D2M2. The incoming network traffic would be captured using a high-speed Endace DAG 7.5G2[7] NIC and fed into the D2M2. This high-speed NIC would ensure a minimal packet loss and thus enable the real-time or near real-time detection and mitigation of attacks. DDoS attacks would be detected using the proposed ensemble-based technique within the D2M2. Upon identification of a DDoS attack, as a part of the mitigation strategy, D2M2 will generate a list of 'white or legitimate' source IP addresses and will communicate it to the different network monitoring device (network and application-aware firewall) being used in the existing system.

# 6. CONCLUSIONS

In this paper we presented a conceptual model, alongwith the preliminary work done, for detecting and mitigating DDoS attacks. In the proposed model, an ensemble of network traffic and MIB server load data analysis is used to detect DDoS attacks, to differentiate them from similar looking FEs, and to instigate source IP based mitigation strategies upon attack identification. The testing and performance evaluation of the proposed model is conducted using synthetic network traffic, closely representing real-world DDoS attacks and FE traffic, generated using a software-based traffic generator developed as a part of this research.

This research was partly supported by the Australia-India Strategic Research Fund.

# 7. REFERENCES

[1] KDD Cup 1999 Data. `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`.

[7] http://www.endace.com/dag-7.5g2-datasheet.html

[2] The CAIDA UCSD "DDoS Attack 2007" Dataset. http://www.caida.org/data/passive/ddos-20070804_dataset.xml.

[3] E. Ahmed, G. Mohay, A. Tickle, and S. Bhatia. Use of IP Addresses for High Rate Flooding Attack Detection. In *Proceedings of the International Information Security Conference (SEC 2010): Security and Privacy - Silver Lining in the Cloud, IFIP World Computer Congress, Brisbane, Australia, 20-23*, September 2010.

[4] M. Arlitt and T. Jin. 1998 World Cup Web Site Access Logs. http://www.acm.org/sigcomm/ITA/, 1998.

[5] C. Bao. Intrusion Detection Based on One-class SVM and SNMP MIB Data. In *2009 Fifth International Conference on Information Assurance and Security*, pages 346–349. IEEE, 2009.

[6] S. Bhatia, G. Mohay, D. Schmidt, and A. Tickle. Modelling web-server flash events. In *Proceedings of The 11th IEEE International Symposium on Network Computing and Applications (NCA) 2012*, pages 79–86. IEEE, 2012.

[7] S. Bhatia, G. Mohay, A. Tickle, and E. Ahmed. Parametric differences between a real-world distributed denial-of-service attack and a flash event. In *Proceedings of Sixth International Conference on Availability, Reliability and Security (ARES), 2011*, pages 210–217. IEEE, 2011.

[8] J. Bumgarner and S. Borg. The US-CCU report on the gerogian cyber campaign . Technical report, U.S. Cyber Consequences Unit, 2009.

[9] L. Garber. Denial-of-service attacks rip the Internet. *Computer*, 33(4):12–17, 2000.

[10] C. Grice. How a basic attack crippled Yahoo. Technical report, CNET News, 2000.

[11] S. Jin and D. Yeung. A covariance analysis model for ddos attack detection. In *Communications, 2004 IEEE International Conference on*, volume 4, pages 1882–1886. IEEE, 2004.

[12] J. Jung, B. Krishnamurthy, and M. Rabinovich. Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites. In *Proceedings of the 11th international conference on World Wide Web*, page 304. ACM, 2002.

[13] S. Kandula, D. Katabi, M. Jacob, and A. Berger. Botz-4-sale: Surviving organized DDoS attacks that mimic flash crowds. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, pages 287–300. USENIX Association, 2005.

[14] L. Limwiwatkul and A. Rungsawangr. Distributed denial of service detection using TCP/IP header and traffic measurement analysis. In *Proceedings of the 2004 International Symposium on Communications and Information Technologies (ISCIT 2004), Sapporo, Japan*, 2004.

[15] J. Nazario. Political DDoS: Estonia and Beyond . In *Invited Talk, in 17th USENIX Security Symposium, July 28-Aug 1, 2008, San Jose, CA, USA*, 2008.

[16] H. Park, P. Li, D. Gao, H. Lee, and R. Deng. Distinguishing between FE and DDoS Using Randomness Check. *Information Security*, pages 131–145, 2008.

[17] J. Park and M. Kim. Design and implementation of an snmp-based traffic flooding attack detection system. *Challenges for Next Generation Network Operations and Service Management*, pages 380–389, 2008.

[18] T. Peng, C. Leckie, and K. Ramamohanarao. Proactively detecting distributed denial of service attacks using source IP address monitoring. *NETWORKING 2004, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, pages 771–782, 2004.

[19] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. My botnet is bigger than yours (maybe, better than yours): Why size estimates remain challenging. In *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, page 5. USENIX Association, 2007.

[20] Yifu Feng, Rui Guo, Dongqi Wang, and Z. Bencheng. Research on the Active DDoS Filtering Algorithm Based on IP Flow. In *Proceedings of the 2009 Fifth International Conference on Natural Computation*, 2009.

[21] J. Yu, H. Lee, M. Kim, and D. Park. Traffic flooding attack detection with SNMP MIB using SVM. *Computer Communications*, 31(17):4212–4219, 2008.