Comparison of DDOS Attacks and Fast ICA Algorithms on The Basis of Time Complexity

D. Raghu[1], M. Arani[2], Ch. Raja Jacob[3]

[1,2,3] *Dept of Computer Science and Engineering, Nova College of Engineering & Technology,*
JNTU-K, Andhra Pradesh.

[1]raghuau@gmail.com, [2]aarani.mantena@gmail.com, [3]rchidipi@gmail.com

***Abstract*— In Distributed denial of service (DDOS) attack, an attacker may use your computer to attack another computer by taking security weakness an attacker could take control of your computer. He could then force your computer to send huge amounts of data to a website. Or send spam particular email address. The "Attack" is distributed because the attacker is using multiple computers including yours to enter the denial of service attack. DDoS attack is a continuous critical threat to the Internet. Derived from the low layers, new application-layer-based DDoS attacks utilizing legitimate HTTP requests to overwhelm victim resources are more undetectable. The case may be more serious when such attacks mimic or occur during the flash crowd event of a popular Website. Distributed denial of service attacks on root name servers are several significant Internet events in which distributed denial-of-service attacks have targeted one or more of the thirteen Domain Name System root name servers. The root name servers are critical infrastructure components of the Internet, mapping domain names to Internet Protocol (IP) addresses and other information. Attacks against the root name servers can impact operation of the entire Internet, rather than specific websites.**

***Keywords*— Application-layer, Distributed denial of service (DDoS), Flash Crowd, Name Servers, ICA algorithm.**

I. INTRODUCTION

Distributed denial of service (DDoS) attack has caused severe damage to servers and will cause even greater intimidation to the development of new Internet services. Traditionally, DDoS attacks are carried out at the network layer, such as ICMP flooding, SYN flooding, and UDP flooding, which are called Net-DDoS attacks in this paper. The intent of these attacks is to consume the network bandwidth and deny service to legitimate users of the victim systems. Since many studies have noticed this type of attack and have proposed different schemes (e.g., network measure or anomaly detection) to protect the network and equipment from bandwidth attacks, it is not as easy as in the past for attackers to launch the DDoS attacks based on network layer.

When the simple Net-DDoS attacks fail, attackers shift their offensive strategies to application-layer attacks and establish a more sophisticated type of DDoS attacks. To circumvent detection, they attack the victim Web servers by HTTP GET requests (e.g., HTTP Flooding) and pulling large image files from the victim server in overwhelming numbers. In another instance, attackers run a massive number of queries through the victim's search engine or database query to bring the server down .We call such attacks application-layer DDoS (App-DDoS) attacks. The MyDoom worm and the CyberSlam are all instances of this type attack. On the other hand, a new special phenomenon of network traffic called flash crowd, has been noticed by researchers during the past several years.

On the Web, "flash crowd" refers to the situation when a very large number of users simultaneously access a popular Website, which produces a surge in traffic to the Website and might cause the site to be virtually unreachable. Because burst traffic and high volume are the common characteristics of App-DDoS attacks and flash crowds, it is not easy for current techniques to distinguish them merely by statistical characteristics of traffic. Therefore, App-DDoS attacks may be stealthier and more dangerous for the popular Websites than the general Net- DDoS attacks when they mimic (or hide in) the normal flash crowd.

In this paper, we meet this challenge by a novel monitoring scheme. To the best of our knowledge, few existing papers focus on the detection of App-DDoS attacks during the flash crowd event. This paper introduces a scheme to capture the spatial-temporal patterns of a normal flash crowd event and to implement the App-DDoS attacks detection. Since the traffic characteristics of low layers are not enough to distinguish the App-DDoS attacks from the normal flash crowd event, the objective of this paper is to find an effective method to identify whether the surge in traffic is caused by App-DDoS attackers or by normal Web surfers.

Our contributions in this paper are fourfold: 1) we define the Access Matrix (AM) to capture spatial-temporal patterns of normal flash crowd and to monitor App-DDoS attacks during flash crowd event; 2) based on our previous work , we use hidden semi-Markov model (HsMM) to describe the dynamics of AM and to achieve a numerical and automatic detection; 3) we apply principal component analysis (PCA) and independent component analysis (ICA) to deal with the multidimensional data for HsMM; and 4) we design the monitoring architecture and validate it by a real flash crowd traffic and three emulated App-DDoS attacks.

This paper is organized as follows: Section II Existing system. Section III describes Proposed workSection IV presents experimental results and performance analysis. Section V presents conclusion and future scope.

II. Existing System

Few existing papers focus on the detection of App-DDoS attacks during the flash crowdevent.Net-DDoS attacks versus stable background traffic, Net-DDoS attacks versus flash crowd (i.e., burst background traffic) are dealt the existing system. Some simple App-DDoS attacks (e.g., Flood) still can be monitored by improving existing methods designed for Net-DDoS attacks, e.g., we can apply the HTTP request rate, HTTP session rate, and duration of user's access for detecting. Most existing methods used on document popularity for modeling user behavior merely focus on the average characteristics (e.g., mean and variance), we use a stochastic process to model the variety of the document popularity, in which a random vector is used to represent the spatial distribution of document popularity and is assumed to be changing with time Existing algorithms of HsMM will be very complex when the observation is a high-dimension vector with dependent elements in the spatial-temporal matrix of AM. In the practical implementation, the model is first trained by the stable and low-volume Web workload whose normality can be ensured by most existing anomaly detection systems, and then it is used to monitor the following Web workload for a period of 10 min.

Stochastic pulses are very difficult to be detected by the existing methods that are based on traffic volume analysis, because the average rate of the attacks is not remarkably higher than that of a normal user. In contrast to existing anomaly detection methods developed in biosurveillance, the non stationary and the non-Markovian properties of HsMM can best describe the self-similarity or long-range dependence of network traffic that has been proved by vast observations on the Internet.

Each layer can be developed independently of the other provided that it adheres to the standards and communicates with the other layers as per the specifications.

- Presentation Layer
- Business Rules Layer
- Data Access Layer
- Database/Data Store

The intent of these attacks is to consume the network bandwidth and deny service to legitimate users of the victim systems.

A. DDoS Attacks

Denial of Service (DDoS) attacks has proved to be a serious and permanent threat to users, organizations, and infrastructures of the Internet. The primary goals of these attacks are to prevent access to a particular resource like a web server. A large number of defenses against DoS attacks have been proposed in the literature, but none of them gives reliable protection. There will always be vulnerable hosts in the Internet to be used for DoS purposes. In addition, it is very difficult to reliably recognize and filter only attack traffic without causing any collateral damage to legitimate traffic. This paper describes how DoS attacks can be carried out and how a victim can mitigate them in ordinary IP networks. Especially wireless ad hoc networks have their additional vulnerabilities, but these kind of wireless networks are not the subject of this paper. A DoS attack can be carried out either as a flooding or a logic attack. A flooding DoS attack is based on brute force. Real-looking but unnecessary data is sent as much as possible to a victim. As a result, network bandwidth is wasted, disk space is filled with unnecessary data (e.g., spam E-mail, junk ftp data, intentional error messages), fixed size data structures inside host software are filled with bogus information, or processing power is spent for unusual purposes. To amplify the effects, DoS attacks can be run in a coordinated fashion from several sources at the same time (DDoS). A logic DoS attack is based on an intelligent exploitation of vulnerabilities in the target. For example, a skillfully constructed fragmented IP datagram may crash a system due to a serious fault in the operating system (OS) software. Another example of a logic attack is to exploit missing authentication requirements by injecting bogus routing information to prevent traffic from reaching the victim's network. There are two major reasons making DoS attacks attractive for attackers. The first reason is that there are effective automatic tools available for attacking any victim, i.e., expertise is not necessarily required. The second reason is that it is usually impossible to locate an attacker without extensive human interaction or without new features in most routers of the Internet [1].This paper gives a short tutorial on DoS attack mechanisms in IP networks and some important defenses proposed in the literature. The emphasis of this paper is on DoS attacks in general, and DDoS attacks are treated as a subset of DoS attacks. DDoS attacks are based on the same mechanisms as basic DoS attacks, but there is one exception during the deployment phase. A DDoS tool needs to be installed on many vulnerable hosts. The spreading mechanisms for DDoS tools are described in a separate section. The installation of DoS software on a single vulnerable host is, however, a common prerequisite for most DoS attacks. Thus defenses described in this paper are applicable to both DoS and DDoS attacks. The set of defenses described in this paper is definitely not exhaustive, but it gives a good overview of the different possibilities in combating DoS attacks. It is claimed in this paper that a comprehensive set of defenses are needed to get defense in depth against DoS attacks. It is important to have defenses for both the deployment and the attack phase. The earlier the preparation or actual use of a Do Stool is detected, the better the chances are for mitigating an attack. The selection of a cost-effective set of defenses must, however, include many business aspects. The most important assets of an organization must be protected with a finite amount of money. The selection and implementation of different defenses should be guided by a risk management process. This paper is organized as follows. First the basic terminology is explained.

III. Proposed Work

A.1FAST ICA

ICA is a statistical signal processing technique. In contrast to the PCA which is sensitive to high-order relationships, the basic idea of ICA is to represent a set of random variables using basis function, where the components are statistically independent and as non-Gaussian as possible:

The ICA task is briefly described as follows. Given the set of input samples **X**=[$\overrightarrow{x_1} \dots \overrightarrow{x_T}$] , where T is the number of samples, and $\overrightarrow{x_t}$ = [$\overrightarrow{x_{1t}} \dots \overrightarrow{x_{Nt}}]^T$ is the N-dimensional observed vector at the t^{th} time unit. The observed vector $\overrightarrow{x_t}$ is assumed to be generated by a linear combination of statistically independent and stationary components (sources), i.e.,

$$\overrightarrow{x_t} = \mathbf{R}\overrightarrow{\mathrm{y}_t}, \qquad t=1,\dots,T \qquad (1)$$

Where $\overrightarrow{y_t}$ = [$\overrightarrow{y_{1t}} \dots \overrightarrow{y_{Nt}}]^T$ is the N-dimensional statistically independent signal vector at t^{th} time unit and **R** is the N × N mixing matrix. Corresponding to the sample dataset **X**, the source data set can be denoted as **Y**=[$\overrightarrow{y_1} \dots \overrightarrow{y_T}$]=[$\mathrm{y}_1\mathrm{y}_2 \dots \mathrm{y}_\mathrm{N}]^T$. The issue is how to determine the N× N invertible de-mixing matrix **W** = $\boldsymbol{R}^{-1}$ so as to recover the components of $\overrightarrow{y_t}$ by exploiting information hidden in $\overrightarrow{x_t}$, i.e., to determine **W** such that components y_{1t}...y_{Nt} of the transformed vector

$$\overrightarrow{y_t} = \mathbf{W}\overrightarrow{\mathrm{x}_t} \qquad (2)$$

are mutually independent. We denote **W** by its components or vectors as **W**=$(\overrightarrow{w_1}, \dots, \overrightarrow{w_n})^T$, where $\overrightarrow{w_\iota}$ is the transpose vector of the i^{th} row of **W** with constraint E$\left\{ \left(\overrightarrow{w_\iota}^T \overrightarrow{x_t} \right)^2 \right\}$=1.

If some abnormities hiding in the incoming Web traffic are found, the "defense" system will be implemented. For example, we can cluster the Web surfers and evaluate their contributions to the anomalies in the aggregate Web traffic. Then, different priorities are given to the clusters according to their abnormalities and serve them in different priority queues. The most abnormal traffic may be filtered when the network is heavy loaded. We implement the algorithm in the NS2 simulator. The network topology is generated by GT-ITM Topology Generator provide by NS2. The simulation includes 1000 client nodes each of which replays one user's trace collected from one of the semifinals of FIFAWorldCup98. The ratio of randomly selected attack nodes to whole nodes is 10%. Furthermore, we assume the attackers can intercept some of the request segment of normal surfers and replay this segment or "hot" pages to launch the App-DDoS attacks to the victim Web server. Thus, when the attack begins, each potential attack node replays a snippet of another historical flash crowd trace. The interval between two consecutive attack requests is decided by three patterns including constant rate attacks, increasing rate attacks and random pulsing attacks. We use the size of requested document to estimate the victim node's processing time (delay) of each request, i.e., if the Requested document is larger, the corresponding processing time will be longer. By this way, we simulate the victim's resource (e.g., CPU) cost by client's requests. Fig. 1 shows our simulation scenario. The whole process lasts about 6 h. As shown in Fig. 2, the first 2 h data are used to train the model, and the remaining 4 h of data including a flash crowd event are used for test. The emulated App-DDoS attacks are mixed with the trace chose from the period of [3.5 h, 5.5 h]. Fig. 3 shows our method on how to collect the observed sequences for detection system. The time unit of this experiment is 5 s. We group 12consecutive observations into one sequence, the "moving" step is one observation unit and a new sequence is formed using the current observation and the preceding 11 observations. Thus, two consecutive Sequences will have 11 overlapped observations. We used 25 consecutive sequences, which last for 36 observations Units or 3 min, to detect anomaly accesses.

IV. Experimental Results

In order to implement the required comparison of DDOS attacks and Fast ICA algorithms comparisons

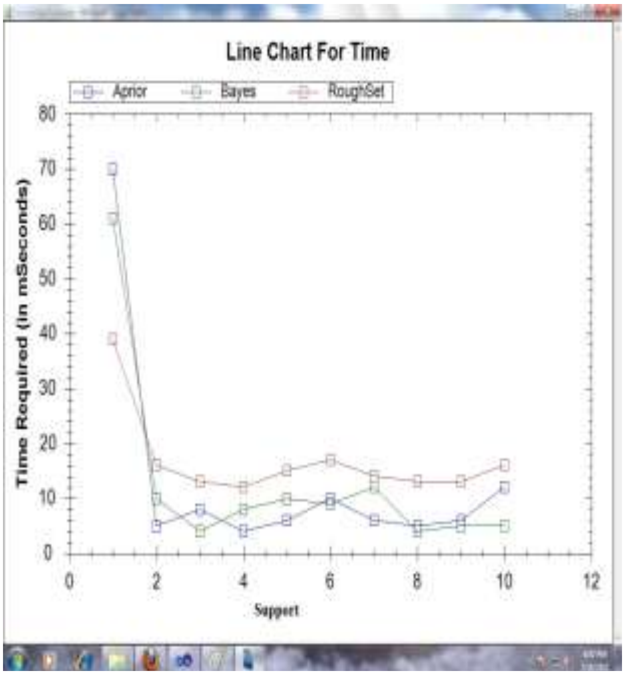

Fig. 1 Line chart shows comparison of time complexity

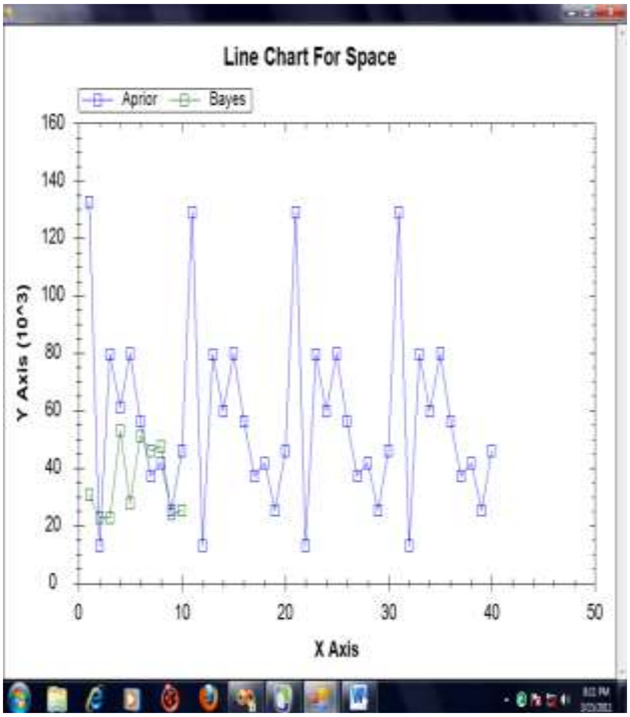

Fig.3 Line chart shows comparison of space complexity

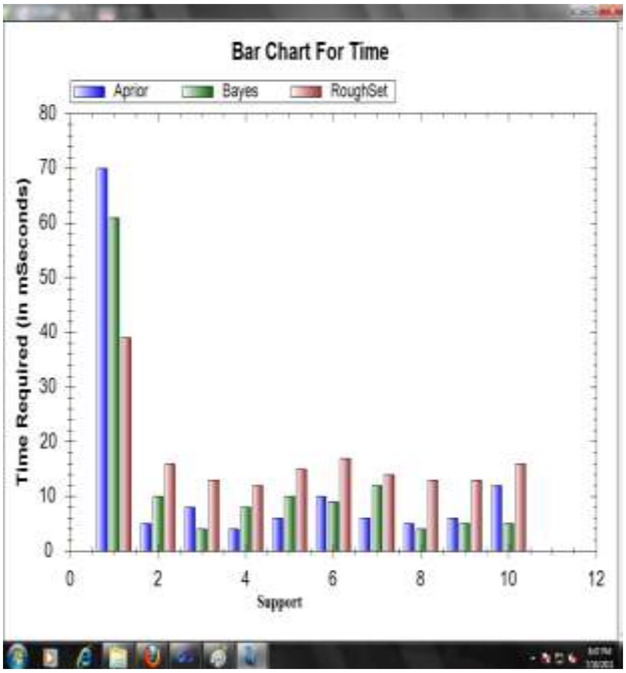

Fig.2 Bar chart shows comparison of time complexity

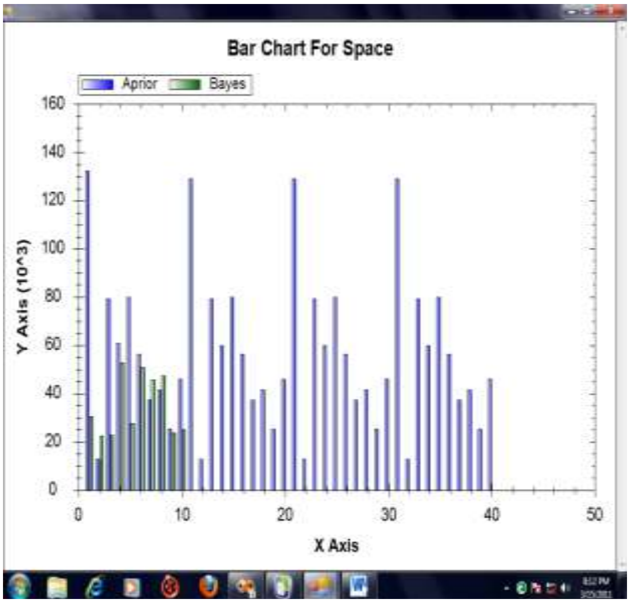

Fig.4 Bar chart shows comparison of space complexity

V. CONCLUSION AND FUTURE WORK

Creating defenses for attacks requires monitoring dynamic network activities in order to obtain timely and signification information. While most current effort focuses on detecting Net-DDoS attacks with stable background traffic, we proposed a detection architecture in this paper aiming at monitoring Web traffic in order to reveal dynamic shifts in normal burst traffic, which might signal onset of App-DDoS attacks during the flash crowd event. Our method reveals early attacks merely depending on the document popularity obtained from the server log.

The proposed method is based on PCA, ICA, and HsMM. We conducted the experiment with different App-DDoS attack modes (i.e., constant rate attacks, increasing rate attacks and stochastic pulsing attack) during a flash crowd event collected from a real trace. Our simulation results show that the system could capture the shift of Web traffic caused by attacks under the flash crowd and the entropy of the observed data fitting to the HsMM can be used as the measure of abnormality. In our experiments, when the detection threshold of entropy is set 5.3, the DR is 90% and the FPR is 1%. It also demonstrates that the proposed architecture is expected to be practical in monitoring App-DDoS attacks and in triggering more dedicated detection on victim network.

REFERENCES

[1] K. Poulsen, "FBI Busts Alleged DDoS Mafia," 2004. [Online]. Available:http://www.securityfocus.com/news/9411

[2] "Incident Note IN-2004-01 W32/Novarg. A Virus," CERT, 2004.[Online].

[3] Available: http://www.cert.org/incident_notes/ IN-2004-01.html

[4] S. Kandula, D. Katabi, M. Jacob, and A. W. Berger, "Botz-4-Sale: Surviving Organized DDoS Attacks that Mimic Flash Crowds,"MIT, Tech. Rep. TR-969, 2004 [Online]. Available: http://www.usenix.org/events/nsdi05/tech/ kandula/kandula.pdf

I. Ari, B. Hong, E. L. Miller, S. A. Brandt, and D. D. E. Long,

[5] "Modeling, Analysis and Simulation of Flash Crowds on the Internet,"Storage Systems Research Center Jack Baskin School of Engineering University of California, Santa Cruz Santa Cruz, CA,

[6] Tech. Rep. UCSC-CRL-03-15, Feb. 28, 2004 [Online]. Available: http://ssrc.cse.ucsc.edu/, 95064

[7] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites," in *Proc. 11th IEEE Int. World Wide Web Conf.*, May 2002,pp. 252–262.

[8] Y. Xie and S. Yu, "A detection approach of user behaviors based

[9] onHsMM," in *Proc. 19th Int. Teletraffic Congress (ITC19)*, Beijing, China, Aug. 29–Sep. 2 2005, pp. 451–460.

[10] Y. Xie and S. Yu, "A novel model for detecting application layer DDoS attacks," in *Proc. 1st IEEE Int. Multi-Symp. Comput.Computat.Sci.(IMSCCS|06)*, Hangzhou, China, Jun. 20–24, 2006, vol. 2, pp. 56–63.

[11] S.-Z. Yu and H. Kobayashi, "An efficient forward-backward algorithmfor an explicit duration hidden Markov model," *IEEE Signal Process.Lett.*, vol. 10, no. 1, pp. 11–14, Jan. 2003.

[12] L. I. Smith, A Tutorial on Principal Components Analysis [EB/OL],2003 [Online].