Volume 5, Issue 5, May 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Detection and Protection of Application-Layer DDoS Attacks for Websites

Sirisha N[1], B. Durga Sri[2], P.Divya [3], K.B.K.S.Durga[4]

Dept of Computer Science & Engineering in MLR Institute of Technology, R.R Dist, Telengana, India
Dept of Computer Science & Engineering in St.Martin's Engineering College, R.R Dist, Telengana, India

Abstract: Distributed denial of service (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 popularWebsite. Focusing on the detection for such new DDoS attacks, a scheme based on document popularity is introduced. An Access Matrix is defined to capture the spatial-temporal patterns of a normal flash crowd. Principal component analysis and independent component analysis are applied to abstract the multidimensional Access Matrix. A novel anomaly detector based on hidden semi-Markov model is proposed to describe the dynamics of Access Matrix and to detect the attacks. The entropy of document popularity fitting to the model is used to detect the potential application-layer DDoS attacks. Numerical results based on real Web traffic data are presented to demonstrate the effectiveness of the proposed method.

Key Terms: Application-layer, distributed denial of service, HsMM , popular Website.

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.

II. RELATED WORK

Existing System:

Few existing papers focus on the detection of App-DDoS attacks during the flash crowd event.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. The intent of these attacks is to consume the network bandwidth and deny service to legitimate users of the victim systems.

Proposed System:

A novel anomaly detector based on hidden semi-Markov model is proposed to describe the dynamics of Access Matrix and to detect the attacks. Numerical results based on real Web traffic data are presented to demonstrate the effectiveness of the proposed method. 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. Different algorithms have been proposed to achieve this objective. This paper applies the FastICA which has been widely used for its good performance and fast convergence during estimation of the parameters. 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.

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.

1) Multidimensional Data Processing:

Multidimensional detection may become a mainstream method in anomaly detection. However, it is very difficult to deal with the multidimensional observation vector sequence without mass computation or assuming a special distribution for the observed data. Thus, PCA and ICA are used before the HsMM-based detector. Because the elements of each vector obtained through ICA are independent, the joint output probability distribution function of HsMM can be simplified as, where is the element of vector, which enables the detector to implement the multidimensional monitoring with less computation and without special assumption for the distribution of the original data.

2) HsMM:

HsMM (Hidden semi-Markov model) can describe most practical stochastic signals, including non-stationary and the non-Markova. It has been widely applied in many areas such as mobility tracking in wireless networks, This document play a vital role in the development of life cycle (SDLC)as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, "A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models. As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.The steps for Spiral Model can be generalized as follows: The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system. A preliminary design is created for the new system. A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product The final system is constructed, based on the refined prototype. The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

III. IMPLIMENTATION

Detection Principle:

Web user behavior is mainly influenced by the structure of Website (e.g., the Web documents and hyperlink) and the way users access web pages. In this paper, our monitoring scheme considers the App-DDoS attack as anomaly browsing behavior.We investigate the characteristic of Web access behavior.

Fig. 2 plots the HTTP request number and the user number per 5 s during the burst Web workload of a semifinal collected from the logs of the 1998 World Cup. From the maximum correlation coefficient 0.9986, between the series of request numbers and that of the user numbers, we can see that the normal flash crowd is mainly caused by the sudden increment of user amount. Fig. 6 plotted in the following experiment section shows that the entropy of the aggregate access behavior against our model does not change much during the flash crowd event, which implies that both the main access behavior profileof normal users and the structure of Website do not have obvious varieties during the flash crowd event and its vicinity area. This conclusion is the same as [5] and is similar to those of other HTTP traces, e.g.,Calgary-HTTP, ClarkNet-HTTP, and NASA-HTTP, which can be downloaded freely from.

Detection Architecture:

The overall procedure of our detection architecture is illustrated in Fig. 3. The scheme is divided into three phases: data preparation, training, and monitoring. The main purpose of data preparation is to compute the AM by the logs of the Web server. The training phase includes the three parts, given here.

1. PCA transition:

The steps are as follows.

a) Compute the average matrix and difference matrix, respectively.
b) Compute the eigenvectors and Eigen values of the covariance matrix.
c) Sort the eigenvalues and select the first eigenvectors,
where is given in this paper.
d) Construct the eigenmatrix by the first eigenvectors.
e) Transform the AM into -dimensional uncorrelated principal component dataset.

2. ICA transition:

The steps are as follows.

a) Use the outputs of the PCA module (i.e., -dimensional uncorrelated principal component dataset to estimate the immixing matrix by ICA algorithm.
b) Transform the dimensional dataset into independent signals.

3. HsMM training:

a) Use the outputs of ICA module as the model training data set to estimate the parameters of HsMM.
b) Compute the entropy of the training data set and the threshold The monitoring phase includes the following steps:
1) Compute the difference matrix between the testing AM and the average matrix obtained in the training phase by the PCA.
2) Using the eigenmatrix, compute the feature dataset of the testing AM.
3) Using the de-mixing matrix, compute the independent signals.
4) The independent signals are inputted to the HsMM; entropies of the testing dataset are computed.
5) Output the result based on the threshold of entropy
that was determined in the training phase based on the entropy distribution of the training data set.

Constant Rate Attack:

Constant rate attack, the simplest attack technique, is typical among known DDoS attacks. We do not arrange the attack sources to simultaneously launch constant rate App-DDoS attacks and to generate requests at full rate, so that they cannot be easily identified through attack intensity. We use to denote the parameters of the constant rate attack. The notation is listed. Three parameters (i.e.,) are set randomly by each attack node before it launches the attacks. It shows the entropy varying with the time, where curve represents the normal flash crowd's entropy and curve represents the entropy of flash crowd mixed with constant rate App-DDoS attacks in zone B. Therefore, it is easy to find out that there exist attacks in the period B.

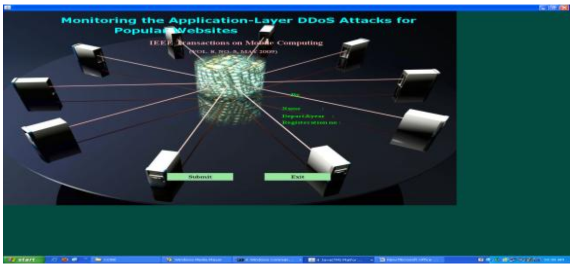
Increasing Rate Attack:

An abrupt change in traffic volume is an important signal to initiate anomaly detection. The attacker may use the gradually increasing rate. The state change in the victim network could be so gradual that services degrade slowly over a long period, delaying detection of the attack. We use to denote the parameters of the increasing rate attack. Five variables (i.e., are set randomly by each attack node before it launches the attacks. It shows the entropy changing with the time, where curve represents the normal flash crowd's entropy and curve represents entropy of the traffic that mixes normal flash crowd with increasing rate App-DDoS attacks, where the attacks start gradually in zone B and end gradually in zone D. As it shows, the entropy
can be used to discover the attacks in the early beginning.

Stochastic Pulsing Attacks:

Instead of constantly injecting traffic flows with huge rates into the network, pulsing attacks, which are also called shrew attacks, are much more difficult to be detected.

IV. SCREEN SHOTS

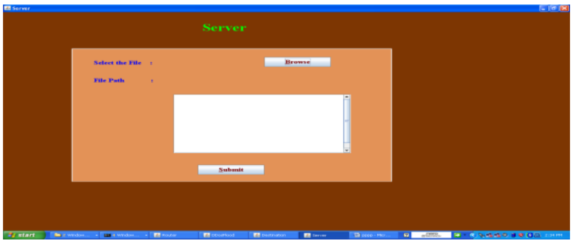


Monitoring the Application-Layer DDoS Attacks for Popular Websites

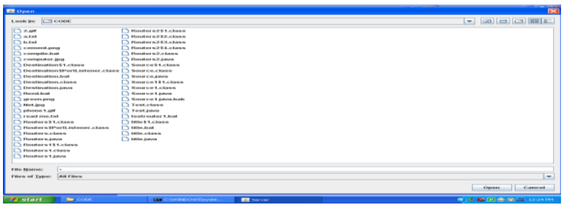
IEEE Transactions on Mobile Computing

(VOL. 8, NO. 5, MAY 2009)

A.1 Legitamate User Login Page.



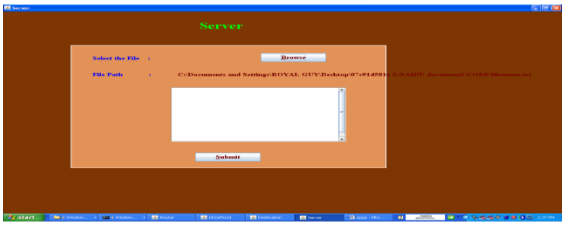
A.2 Browse by legitimate user



```
2.gif                                  Routers2$1.class
a.txt                                  Routers2$2.class
b.txt                                  Routers2$3.class
comment.png                            Routers2$4.class
compile.bat                            Routers2.class
computer.jpg                           Routers2.java
Destination$1.class                    Source$1.class
Destination$PortListener.class         Source.class
Destination.bat                        Source.java
Destination.class                      Source1$1.class
Destination.java                       Source1.class
flood.bat                              Source1.java
green.png                              Source1.java.bak
Net.jpg                                Test.class
phone1.gif                             Test.java
read me.txt                            testrouter1.bat
Routers$1.class                        title$1.class
Routers$PortListener.class             title.bat
Routers.class                          title.class
Routers.java                           title.java
Routers1$1.class
Routers1.class
Routers1.java
```

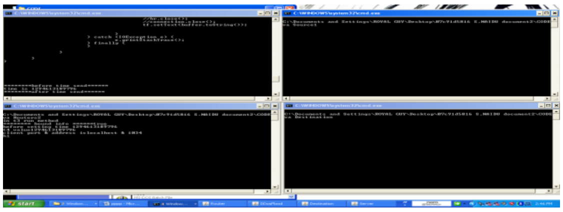
File Name:
Files of Type: All Files

A.3 Select a desired file from the list

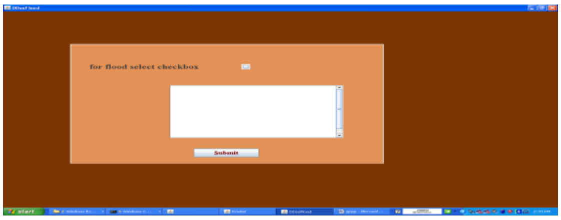


File Path : C:\Documents and Settings\ROYAL GUY\Desktop\07c91d5818 S.NAIDU document2\CODE\filename.txt

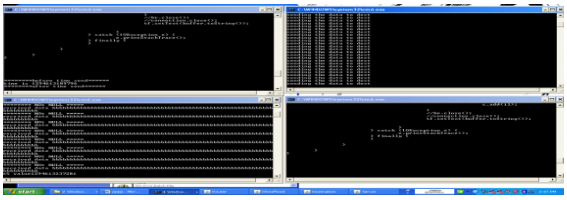
A.4 Submit the desired file.



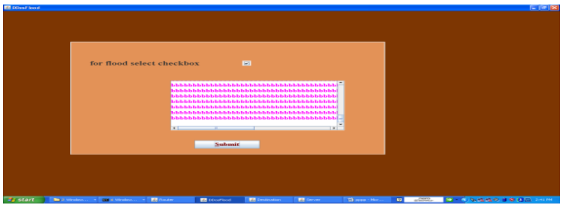
A.5 Legitimate user,Attacker,Router,Destination Windows.Submit After the desired file



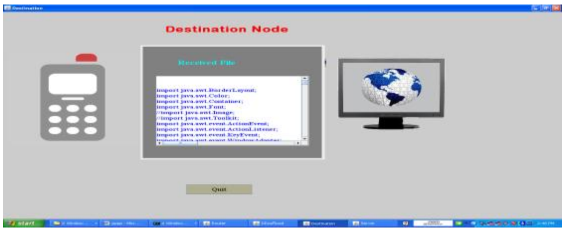
A.6 Submit the flood by Attacker.



A.7 Legitimate user, Attacker, Router, Destination Windows after completion of the flood by Attacker

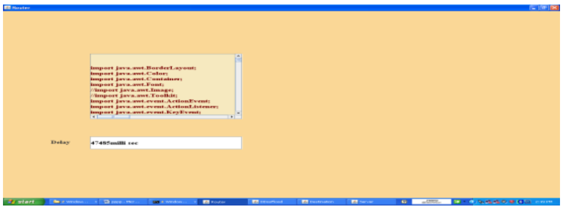


A.8 Attacker's Node after the flood.



```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Container;
import java.awt.Font;
//import java.awt.Image;
//import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
```

A.9 Destination Node.



```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Container;
import java.awt.Font;
//import java.awt.Image;
//import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
```

Delay 47495milli sec

A.10 Delay time for user's desired service.

V. CONCLUSION AND FUTURE ENHANCEMENT

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 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. To handle the seriousness of ddos attacks mimic or occur during the flash crowd event of a popular Website a new approach proposed to be implemented.

REFERENCE

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

[2] "Incident Note IN-2004-01 W32/Novarg. A Virus," CERT, 2004.Available: http://www.cert.org/incident_notes/ IN-2004-01.html

[3] 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.

[4] I. Ari, B. Hong, E. L. Miller, S. A. Brandt, and D. D. E. Long,"Modeling, Analysis and Simulation of Flash Crowds on the Internet,"Storage Systems Research Center Jack Baskin School ofEngineering University of California, Santa Cruz Santa Cruz, CA, Tech. Rep. UCSC-CRL-03-15, Feb. 28, 2004 [Online]. Available:http://ssrc.cse.ucsc.edu/, 95064

[5] 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.

[6] Y. Xie and S. Yu, "A detection approach of user behaviors based on HsMM," in Proc. 19th Int. Teletraffic Congress (ITC19), Beijing,China, Aug. 29–Sep. 2 2005, pp. 451–460.

[7] 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.

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

[9] L. I. Smith, A Tutorial on Principal Components Analysis [EB/OL],2003 [Online]. Available: http://www.snl.salk.edu/~shlens/pub/ notes/pca.pdf.

[10] A. Hyvärinen, "Survey on independent component analysis," Neural Comput. Surveys, vol. 2, pp. 94–128, 1999.

[11] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," IEEE Trans. Neural Netw., vol. 10, no. 3, pp. 626–634, Jun. 1999.

[12] J. B. D. Cabrera, L. Lewis, X. Qin, W. Lee, R. K. Prasanth, B. Ravichandran, and R. K. Mehra, "Proactive detection of distributed denial of service attacks using MIB traffic variables a feasibility study," in Proc. IEEE/IFIP Int. Symp. Integr. Netw. Manag., May 2001, pp. 609–622.

AUTHOR'S PROFILE

Mrs. Sirisha N, Post Graduated in Software Engineering (M. Tech), JNTU Hyderabad, in 2012 and Graduated in Computer Science & Engineering (B.Tech) from JNTU Hyderabad in 2009. She is working as an Assistant Professor in Department of Computer Science & Engineering in **MLR Institute of Technology**, R.R Dist, Telangana, and India. She has 3+ years of Teaching Experience. Her Research Interests Include Computer Networks and Network Security.

Mrs. B. Durga Sri Post Graduated in Computer Science & Technology (M. Tech), Andhra University, Vishakapatnam in 2010 and Graduated in Information Technology (B. Tech) from JNTU Hyderabad in 2008. She is working as an Assistant Professor in Department of Computer Science & Engineering in **MLR Institute of Technology**, R.R Dist, Telangana, and India. She has 5+ years of Teaching Experience. Her Research Interests Include Computer Networks, Network Security, Data Warehousing and Data Mining.

Mrs.P.DIVYA, Post Graduated in Computer Science & Engineering (**M.Tech**) From **JNTU**, Hyderabad in 2013 and Graduated in Information Technology (B.Tech) from JNTU,Hyderabad in 2010. She is working as An Assistant Professor in the Department of Computer Science & Engineering in **St.Martin's Engineering College**, R.R Dist, Telengana and India.. He has 5+ years of Teaching Experience. Her Research Interests Include Network Security, Web Technologies.

Mrs.K.B.K.S.DURGA, Post Graduated in Information Technology (**M.Tech**) From **JNTU**, Hyderabad in 2009 and Graduated in Information Technolgy (B.E) from Osmania University, in 2005. She is working as An Associate Professor in the Department of Computer Science & Engineering in **St.Martin's Engineering College**, R.R Dist, Telengana and India.. She has 9+ years of Teaching Experience. Her research Interests Include Software Engineering, Network Security, Computer networks.