ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

IJIRSET

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)

On 21st&22ndMarch, Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

# A Robust Defense Scheme for the Detection of Distributed Denial of Service Attack through Web Proxy System

A.Aafreen, Kannan Balasubramanian,M.Tech, Ph.D.

PG Scholar, Dept. of Computer Science, Mepco Schlenk Engineering College, Sivakasi, Tamilnadu, India.

Professor, Dept. of Computer Science, Mepco Schlenk Engineering College, Sivakasi, Tamilnadu, India.

**Abstract**— Indirect attack has been a serious threat to server security due to their covert nature. Web proxy Distributed Denial of Service Attack is an increasingly common internet phenomenon and is capable of making the internet services unavailable. Such type of attack cannot be easily discovered by most existing defense systems since malicious traffic is hidden in the aggregated traffic. Also the source of the attack traffic and normal traffic cannot be distinguished, because both of them share the same IP of the proxy server.

To overcome this problem, a new improved Hidden semi-markov model is proposed. Therefore applying this proposed method protects the origin server from the web proxy based HTTP attacks. Web proxy's access behavior can be regarded as the combination of the externally observable behavior and the internal driving mechanism. The internal driving mechanism can be estimated by the observable features of proxy-to-server traffic through the Hidden semi-markov model. Hidden semi-markov model describes the dynamic behavior process of the aggregated traffic. The false positive rate is also detected with respect to the incoming traffic.

**Keywords—** Traffic analysis, traffic modelling, distributed denial of service attack, attack detection, attack response.

## I.INTRODUCTION

A Distributed Denial-of-Service (DDoS) attack is an attempt to make a computer resource unavailable to its intended user. One common method of attack involves saturating the target machine with external communication requests, so much so that it cannot respond to legitimate traffic or responds so slowly as to be rendered essentially unavailable.

There is other type of covert DDoS attacks (i.e) web proxy based distributed denial of service attacks. The key point of WPDDoS (Web Proxy Distributed Denial of Service Attack) of attack is that an attacker may exploit the communication mechanism of proxy servers to attack the victim via the ready-made hierarchical web proxy network [1][2][3].This means that any publicly accessible internet proxy server may be passively involved in the WPDDoS attack events and may unconsciously act as an attacker. In the actual environment, an attacker can easily turn a proxy server into an attack tool by forcing the proxy to forward its attack HTTP requests to the victim server.

Thus a single attacker can simultaneously trigger a lot of proxy servers to attack a web server without the need of invading them. The thread of WPDDoS mainly comes from the following.

1. These attacks are based on the HTTP protocol. Thus the attack traffic can pass through most of current border firewalls. Moreover, most of the existing detection systems designed for TCP-or IP-layer DDoS attacks are vulnerable to effectively discover the attack signals raised by the WPDDoS attacks which work on the application layer.

2. In Fig.1 the attack traffic and the normal traffic are forwarded to the victim server by the hierarchical proxy network. Thus the victim server can judge which proxy's outgoing traffic includes the attack behaviour, but cannot accurately block the attack traffic via its source IP [4][5].This increases the false positive rate (FPR) for most of the existing detection system.

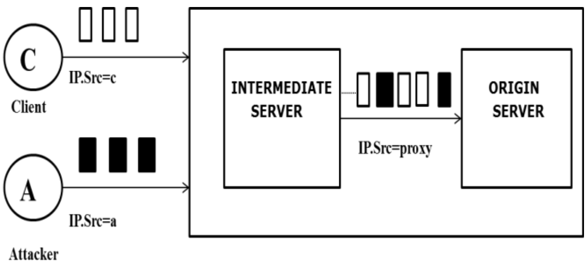

Fig. 1 A Simple Example of WPDDoS Attack

3. The attack traffic and the normal traffic continue to be aggregated by each web proxy that they pass through. Hence it is not easy for the detection system to discover the real attack pattern. Combined with the botnets, WPDDoS may be more aggressive.

These issues not only make the DDoS attacks easier but also increase the attack performance. The proposed scheme achieves the goal via behavior analysis. It assumes that the time varying aggregated traffic sent by a particular web proxy and observed by the victim server is controlled by a series of underlying behavioral patterns of the web proxy[11][12].Transition between two consecutive behavior patterns represents that the driving mechanism of the aggregated traffic is changing. Then a mathematical model namely HMM is applied which describes the behavior characteristics of the web proxies.

## II.PROPOSED SYSTEM

### A. Model Definition

Hidden Markov Model is used for the detection of traffic in the Distributed Denial Service of service attacks. To present the effectiveness of the proposed method the work is confined with the following conditions.

1. It deals with the flooding of the HTTP request from the attacker to the web proxy.

2. The web proxy forwards all the requests to the web server. The web server forwards or rejects the request based on the recessive attributed namely normal or abnormal behavior.System design of the proposed technique is depicted in Fig: 2The diagram shows how the proxy to server traffic is generated and also about the decision of the web server whether to accept or reject the incoming request.

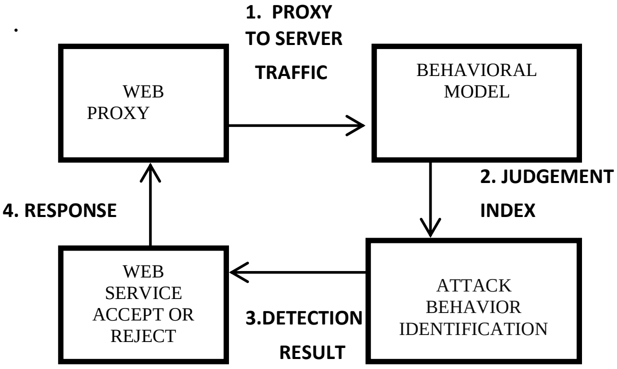
WEB PROXY
1. PROXY TO SERVER TRAFFIC
BEHAVIORAL MODEL
2. JUDGEMENT INDEX
ATTACK BEHAVIOR IDENTIFICATION
3.DETECTION RESULT
WEB SERVICE ACCEPT OR REJECT
4. RESPONSE

Fig. 2. The Framework of the Proposed Scheme

The above diagram illustrates the following steps.

Step 1: A general mathematical model is developed to describe the traffic behavior of the proxy servers.
Step 2: An incoming aggregated traffic is evaluated by a given behavior model.
Step 3: The judgment index is the judgment criteria obtained from the HMM model based on the probability obtained from the accepted web pages.
Step 4: The web server returns the response to the web proxy based on the detection result.

### B. Behavior Model

Proxy's access behavior can be regarded as the combination of external manifestations and the intrinsic driving mechanism. The external manifestation includes the temporal and spatial locality. Here a temporal locality is utilized to build the behavior model.

Hence in order to obtain the statistics of the most recently accessed page, a least recently used (LRU) stack concept is used. It converts the reference stream into the stack distance stream and returns the stack distance from the top of the stack as the output.

In the proposed method the spoofed requests is identified from a number of genuine requests. This is done by generating a number of requests from the user to the web server. If the any consecutive two requests occur within the time interval of 10ns then the particular string is considered to be abnormal and it is not inserted into the stack. Similarly if the same string occurs for more than once within a small time interval then that particular string is also considered to be abnormal. The probabilities of the accepted and the rejected strings are noted. The probability of the False Positive Rate is also calculated.

### C. Stack Distance Model for Temporal Locality

Temporal locality refers to the property that referencing behavior in the recent past is a good predictor of the referencing behavior to be seen in the near future. The stack object references is a good model for characterizing the behavior of the proxies and cache. The main advantage of the stack distance model for describing the Web proxies access behavior.

Here a request string can be converted into a distance string that preserves the pattern of the activity. To define this notion, the files are assumed to be placed on a stack such that, whenever the file f is requested, it is either pulled from its position in the stack and placed on the top, or it is simply added to the stack if it is not in it.

Thus, starting with an empty stack, the reference stream is $F_i=\{f_1; f_2 ;\ldots\ldots.f_i\}$ where fi denotes the name of the i$^{th}$ requested file. Index i indicates that i requests have already arrived at a server. Thus, the unit of time is one request, an incoming request represents a new event occurring. The least recently used (LRU)stack is denoted by $L_i$, which is an ordering of all files of a server by recent usage. Thus, at index i, the LRU stack is given by $L_i=\{u_1; u_2; \ldots\ldots.u_N\}$, where $u_1; u_2;\ldots\ldots.u_N$are files of the serve rand $u_1$ is the most recently accessed file, $u_2$ the next most recently. In other words, $u_1$ is just accessed at index i, (i.e) $f_i = u_1$. Whenever a reference is made to a file, the stack must be updated. Considering that $f_{i+1} =u_j$then the stack becomes$L_{i+1} = \{ u_1; u_2; \ldots\ldots. u_N\}$. Suppose now that $L_{i-1}=\{ u_1; u_2;\ldots. u_N\}$and $f_i = u_j$, i.e., the request $r_i$ is at distance j in stack $L_{i-1}$.

Let $d_i$ denote the stack depth of the document referenced at index i. Then a new relation can be obtained by the following equation iffi $=u_j$ then $d_i = j$, where j denotes the stack depth of the requested document at index i. Thus, the reference trace{$f_1$; $f_2$; …..;$f_i$} defines a numerical sequence {$d_1$; $d_2$; …….; $d_i$} of trace distances.

Fig .3gives a possible state of the stack before and after the request for document ‘‘A’’ occurs. The stack distance for this access is 3. Repeating the above transformation for each access, the object reference steam can be translated to the stack distance stream. A reference symbol stream {A,D,C,A,B,D,E,A,B} is translated into the numerical stack distance stream {3,4,3,3,5,4,5,4,4}.Here popular files will tend to experience much shorter stack distances than rare files.

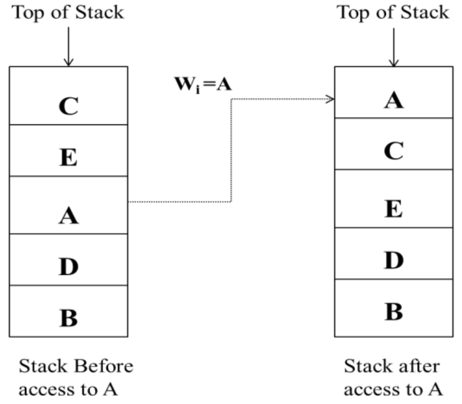

Fig. 3.Least Recently used Stack Model.

Let $d^f_t$denote the observed stack distance f the file f, when f obtains the$t^{th}$ visit. Then, the current mean$\mu_d$(f,t) and variance $\sigma_d^2(f,t)$ of the logarithmic stack distance value of f can be calculated by:

$$\boldsymbol{\mu_d(f,t)=\frac{1}{t}\sum_{t=1}^{t}\log(d_i^f)}$$

$$\sigma_d^2(f,t)=\frac{1}{t}\sum_{t=1}^{t}[\log(d_i^f)]^2-[\mu_d(f,t)]^2$$

Considering that both the mean and the variance will change their values as time goes by. To reduce the computational complexity, the following recursive formula are used to update $\mu_d$(f,t) and $\sigma_d^2(f,t)$ at each time step.

$$\mu_d(f,t+1)=\mu_d(f,t)+\frac{\log(d_{t+1}^f)-\mu_d(f,t)}{t+1}$$

$$\sigma_d^2(f,t+1)=\frac{t-1}{t}\sigma_d^2(f,t)+\frac{[\log(d_{t+1}^f)-\mu_d(f,t)]^2}{t+1}$$

Then, the normalized logarithmic stack distance value $d^f{}_t$ can be calculated by the following equation.

$$d_t^f=[\log(d_{t+1}^f)-\mu_d(f,t)]/\sigma_d(f,t)$$

### *D. Hidden Semi MarkovModel*

The basic structure of the HSMM is illustrated in Fig.4. A HsMM consists of a pair of discrete-time stochastic processes $O_t$and $X_t$ ,t€{1,2,….. t} where t is the index of observation (also called event). The $O_t$ is the observed (or output) process and may either be discrete or continuous, univariate or multivariate hence$X_t$is the finite-state hidden semi-Markov chain.

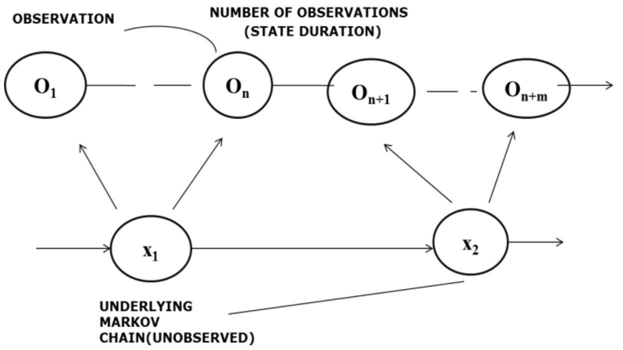

Fig. 4.Structure of HsMM

Here {$X_t$} is not observable directly through{ $O_{t\}}$ but can be estimated .To model a web proxy’s access behavior by an HsMM, each hidden semi-markov state represents a driving mechanism of the proxy-to-server traffic. The transition between two different markov states represents the changes of the driving mechanism. Duration of a particular semi-markov state represents the dwell time of its corresponding driving mechanism. Here two driving mechanisms are defined, namely normal and abnormal behavior.

### *E. Hidden Semi MarkovModel*

The stack distance reveals the frequency about the access of the particular string from the stack. Here the strings represents the number of URL. The probability about the number of strings accepted or rejected is obtained with the stack concept. This is given as a input to the HsMM. Through the modeling of the HsMM, the final probability about the acceptance and rejection of the string is obtained. From this the proxy-to-server traffic can be analyzed which represents the recessive attributes of the particular string (i.e) whether it is normal or abnormal. False Positive Rate is also obtained by comparing both the designed HsMM model with the arrival rate based method.

### *F. Web Server Model*

The stack distance reveals the frequency about the access of the particular string from the stack. Here the strings represents the number of URL. The probability about the number of strings accepted or rejected is obtained with the stack concept. This is given as a input to the HsMM. Through the modeling of the HsMM, the final probability about the acceptance and rejection of the string is obtained. From this the proxy-to-server traffic can be analyzed which represents the recessive attributes of the particular string (i.e) whether it is normal or abnormal. False Positive Rate is also obtained by comparing both the designed HsMM model with the arrival rate based method.

## III.IMPLEMENTATION

The implementation methodology of Distributed denial of service attack through web proxy system is discussed here.

### *A. Behavior Module*

The User submits the request for the required web page. The request is transferred from the client to the web server through the web proxy system. The proxy in turn then originates a new HTTP request and sends it to the web server in such a way that the requestor is not known. The web server then decides whether to serve the web page based on certain strategy [6][7][8].

### *B. Hidden Semi Markov Module*

This is implemented with the tool jahmm (An implementation of Hidden Markov Models in java. This class demonstrates how to build a HMM with known parameters, how to generate a sequence of observations with the given HMM, how to learn the parameters of a HMM given observation sequences, how to compute the probability of an observation sequence for a HMM[9],[10]. It uses the computer network that can experience jamming. When the wireless medium is jammed, a lot of web pages are lost. Thus, the HMMs built here have two states (congested/not congested).

The output probability of whether the string is accepted or rejected is obtained from the behavior model for the number of different requests. This probability is given as the initial value for the HMM. The transition between two different states in HMM is likely to accept or reject the string. The string here represents the URL of the requestor. The HMM returns the value after the observation sequence. From this the detection performance is compared with the arrival rate based scheme.

### *C. Web Server Module*

Modeling the varying process of a web proxy's recessive attribute can profile the proxy's real behavior better than the dominant attributes. The dominant attributes can be directly observed. It includes arrival rate, temporal locality, packet size and so on. Recessive attributes cannot be directly observed from the proxy-to-server traffic (e.g) the type(normal or abnormal).The main challenge of such a model is that the recessive attributes are unobservable to the victim server.

Here a web proxy is regarded as an invisible state machine whose state sequence represents the varying process of the proxy's recessive attributes. Since all recessive attributes are unobservable to the victim server, the state sequence can only be estimated via the observable dominant attributes of the proxy-to-server traffic. Hence after the probability value obtained from the HMM, the proxy decides whether to accept or reject the string based on normal or abnormal behavior. The following algorithms are used for evaluating the traffic behavior. This algorithm is used for evaluating whether the traffic is normal or abnormal.

**Algorithm 1:**

```
Name        : **Evaluation of normality**
Parameters : A proxy-to-server traffic
Input       : A trained model α, Judgement
            interval  T
Output      : Normal or abnormal behavior
//This Algorithm is for evaluating whether the
Traffic
is normal or abnormal.
Extract the observed variables of each request of the input
proxy-to-server traffic :
temporal locality and form a observation sequence $O=\{o_1, o_2, o_3……..o_t\}$
Calculate the judgment criteria of  O,
 If the judgment index falls in T then
return normal ;
else
return abnormal ;
end .
```

Judgment criteria of acceptance or rejection of web pages is based on the probability obtained from the HMM.

This illustrates the evaluation of normality based on the Judgment criteria.

**Algorithm 2:**

```
Name          : **Attack Algorithm**
Input         :  the entire attack duration w ;
the duration of each time unit u ;
Output        : Generates the attack request.
// This illustrates whether the web server accepts
or rejects the request.
Repeat
Randomly select an attack pattern P.
If constant-rate pattern is selected , the attack
rate. $r_c$has to be decided.
Randomly select the duration v for the select attack
pattern P with constraint
$v \leq w$;
for t=1 to v Step u do
if constant-rate pattern is selected then
                Send attack requests according to $r_c$;
else
                Randomly select a rate $r_r$ ;
Send attack requests according to $r_r$ ;
end if
end for
$w = w – v$ ;
Until $w \leq 0$
```

This illustrates whether the web server accepts or rejects the request based on the algorithm.

### *D. Performance Analysis*

The acceptance rate of the incoming request in the proposed scheme is compared with the commonly used arrival rate based method.

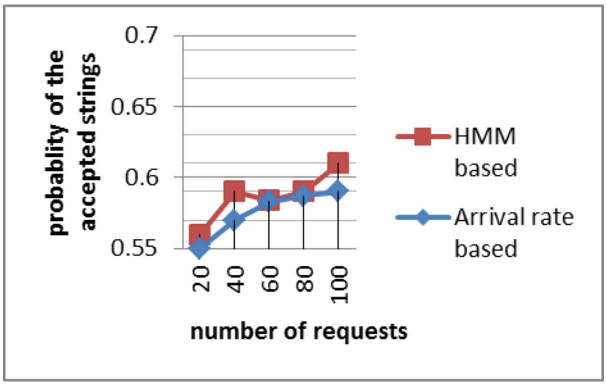

Fig: 5 .ROC curves of HMM and Arrival rate methods.

In this paper we use “normal” to denote the proxy to server traffic without malicious behavior, use “polluted” to denote the mixed traffic of normal and attack request. In Fig.5 The above curve denotes the normal curve of the proxy to server traffic with the accepted incoming files. Here the acceptance rate of the incoming request in the proposed scheme is compared with the commonly used arrival rate based method and is found to be better than the detection based on the arrival rate.

The values in the x-axis denote the number of incoming request with respect to respect to the memory. The values in the x-axis denote the probability of the acceptance rate.

In Fig.6 False Positive rate (FPR) of the arrival rate method is compared with the HMM based technique based on the arrival rate of the incoming request.

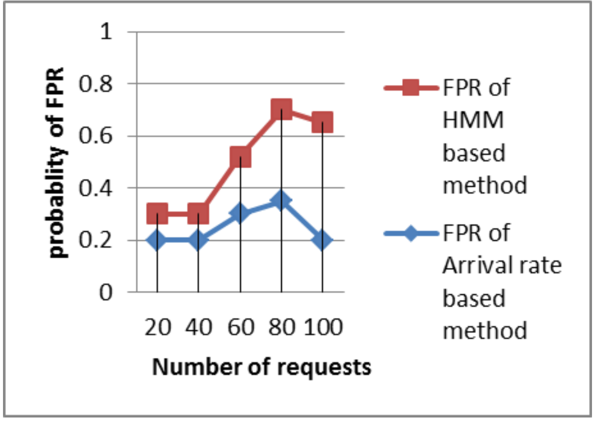

Fig: 6. FPR of HMM and Arrival rate methods.

The proposed scheme is based on proxy behavior instead of the traffic volume. It does not depend on traffic intensity but only compares the proxy’s current behavior with the normal behavior profile. Although traffic volume are not sutaible for the detection when attacks are based on low traffic, the existence of attack requests will distort a proxies access behavior. This enables the proposed scheme to achieve detection. The detection becomes easier when proxies access behavior is distorted seriously by the heavy attack traffic.

The proposed scheme is compared with detection based on arrival rate and the performance of the HMM based behavior is found to be better than the existing system.

## IV.CONCLUSIONS

The proposed method focuses on a new latent attack that exploits the communication mechanism of proxy servers to achieve the web proxy based distributed denial of service attacks. The temporal locality is used to extract the access behavior of the web proxy. From the depth of the stack the recent usage of the files is obtained. Based on time deviation between the present requested file and the previous requested file the normality of the string is detected.

An improved hidden semi markov model is proposed. It demonstrates how to build the HMM and the observation sequence with the known parameters. The output probability obtained from the behavior model is given as the input to the initial states of the HMM. The transition between two different states in HMM represent that the incoming request is likely to accept or reject. Hence the web proxy’s access behavior is modelled by mapping it to the HMM. The recessive attributes which is not directly observed is obtained from the observed proxy-to-server traffic. The driving mechanism of whether the traffic is normal or abnormal is also obtained from the incoming request.

The acceptance rate of the incoming request in the proposed scheme is compared with thecommonly used arrival rate based method and the performance of HMM based technique is found to be better than the detection based on the arrival rate.

False Positive rate with respect to the number of the incoming request is identified and the performance of the HMM based technique is found to be better than the detection based on the arrival rate.

As a result the issue about the False Positive rate and the Acceptance rate is addressed.

In future, the proposed method will be implemented in a real communication platform to test other new strategy . We propose to investigate other attacks on web proxy based distributed denial of service.

## REFERENCES

[1] Yi Xie, S.Tang,Y. Xiang and J. Hu,“ Resisting Web Proxy-Based HTTP Attacks by Temporal and Spatial Locality Behavior” IEEE Transactions On Parallel And Distributed Systems, Vol. 24, No. 7, July 2013.

[2] S. Triukose, Z. Al-Qudah, and M. Rabinovich, “Content Delivery Networks: Protection or Threat?” Proc. 14th European Conf.Research in Computer Security (ESORICS), pp. 371-389, 2009.

[3] Z.L.C. Zhonghua and W. Xiaoming, “Research on Detection Methods of CC Attack,” Telecomm.Science, pp. 62-65, 2009.

[4] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E.Vazquez, “Anomaly-Based Network Intrusion Detection:Techniques, Systems and Challenges,” Computers and Security,vol. 28, nos. 1/2, pp. 18-28, 2009.

[5] J. Ferguson, “Variable Duration Models for Speech,” Proc.Symp. Application of Hidden Markov Models to Text and Speech,pp 143-179, 1980.

[6] S.Yu, “Hidden Semi-Markov Models,” Artificial Intelligence,vol. no. 2, pp. 215-243, 2010.

[7] Y. Xie and S. Yu, “A Large-Scale Hidden Semi-Markov Model forAnomaly Detection on User Browsing Behaviors,” IEEE/ACM Trans. Networking, vol. 17, no. 1, pp. 54-65, Feb. 2009.

[8] S. Levinson, “Continuously Variable Duration Hidden Markov

Models for Automatic Speech Recognition,” Computer Speech and Language, vol. 1, no. 1, pp. 29-45, 1986.

[9] S.-Z. Yu and H. Kobayashi, “An Efficient Forward-Backward Algorithm for an Explicit-Duration Hidden Markov Model,” IEEE Signal Processing Letters, vol. 10, no. 1, pp. 11-14, Jan. 2003.

[10] L. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” Proc. IEEE, vol. 77, no. 2,pp. 257-286, Feb. 1989.

[11] “DoSHTTP,” http://www.socketsoft.net/, 2013.

[12] Y. Xie and S. Yu, “Measuring the Normality of Web Proxies Behavior Based on Locality Principles,” Network and Parallel Computing, vol. 5245, pp. 61-73, 2008.

[13] S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, and F.Tang,“DiscriminatingDDos Attacks from Flash Crowds Using Flow Correlation Coefficient,” IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 6, pp. 1073-1080, June 2012.

[14] A. Mahanti, D. Eager, and C. Williamson, “Temporal Localityand Its Impact on Web Proxy Cache Performance,” PerformanceEvaluation, vol. 42, nos. 2/3, pp. 187-203, 2000.

[15] S. Yu, W. Zhou, R. Doss, and W. Jia, “Traceback of DDosAttacks Using Entropy Variations,” IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 3, pp. 412-425, Mar. 2011.