Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186–5140
Volume 3, Number 1, pages 57–62, January 2014

Abnormal Web Traffic Detection Using Connection Graph

Manh Cong Tran
Department of Computer Science
National Defense Academy
Yokosuka Shi, Kanagawa, Japan
Email: manhtc@gmail.com

Lee Heejeong
Department of Computer Science
National Defense Academy
Yokosuka Shi, Kanagawa, Japan
Email: heejery@gmail.com

Yasuhiro Nakamura
Department of Computer Science
National Defense Academy
Yokosuka Shi, Kanagawa, Japan
Email: yas@nda.ac.jp

***Abstract*—Internal network security threats are becoming increasingly dangerous and difficult to detect when cyber criminals tend to take advantage of web technology as a medium for communication. Web traffic generated by non-human activities such as bot-nets or worms exhausts a network's resources, deludes people and affects network security. This paper proposes a new method to detect abnormal web traffics in a network. It introduces two features:** *malicious-server-degree* **and** *abnormal-traffic-score*, **those are based on characteristics of a connection graph model for web access data. These features filter out suspicious clients generated abnormal traffics. The experiment specifically shows different levels of potential anomalous traffics for each suspicious client. The detected abnormal web traffic is easy to be visually seen, and the method is simply implemented even in large networks.**

***Keywords*—Abnormal detection; Web traffic; Connection graph; Intrusion detection**

I. Introduction

In the modern era along with a innovative growth up of the network technology, cyber security is the highest priority for any fields of life. While external threats are always dangerous and need to be shielded with firewalls and other defenses, internal weaknesses misleads people and difficult to detect. Sophisticated infiltration techniques always improved and applied to transform external threats to internal threats. In web environment, digital spies and thieves can cover their identities, conceal their physical locations, and create the malicious code from side to side. Cybercriminals or the Internet spiders are taking advantage of web technology and using it as a medium for communication to lurk malwares or bot-nets activities to outside. A new type of attack named APT (Advanced Persistent Threat) is a vivid example in exploiting users web behaviors to infect malicious code to their computers. Moreover, due to prevent the detection depend on content inspection, obfuscation techniques and primarily encryption are also used. Thus, it is frequently impossible to pinpoint specific attackers, however analyzing the Internet traffic to detect the malicious or abnormal traffic is useful method to block and prevent the future attacks.

In this paper, with a simple approach, we present a new method in detection of abnormal web traffics using connection network. Web requests are extracted from web or proxy server repositories where are great source of knowledge about web usage patterns of different web users. We proposed two new features *malicious-server-degree* (*MSD*) and *abnormal-traffic-score* (*ATS*). *MSD*, which helps to evaluate how much a server is malicious, is built based on the popular of web servers among client. *ATS* points out which suspicious clients can generate abnormal web traffics. As can be filtered in web log data, there are almost of favorite servers are accessed daily from multiple clients such as $www.yahoo.co.jp$ or $www.youtube.com$, however some seem to be private which have only one or very few accessible from clients. In another view, if all clients and servers are considered as a set of vertices in a network then it can be seen that popular servers have multiple in-connection edges, and vice versa. In Fig.1, a connection graph is set where each node present a server or client, and node s is most popular server with 6 connections from clients.

The rest of the paper is structured as follows: section 2, we review the related work; section 3 presents our proposed method for abnormal web traffic detection; data preparation and experimental results are analyzed in section 4. Conclusion is written in section 5 along with our future works.

II. Related Works

There are many researches using web log data as main resources to analysis user's the Internet access behavior for Spam or bot-nets detection or to do data mining to find improvement solutions for websites performance [8], [9], [10]. In [8], based on analyzing web usage navigation behaviors Pedram Hayati et al proposed an action set as a new feature set for spambots detection. Classifying algorithm SVM (Support Vector Machine) is also used to train feature set and classify spambots. Classification algorithm to identify malicious data stealing attempts within web traffic was presented in [9]. Areej Al-Bataineh et al uncovered the network behavior of web-based data stealing bot-net, including patterns of communication, and most importantly the use of encryption to hide the stolen data. The classifier analyzes entropy and byte frequency distribution of HTTP POST request contents as features.

In recent years, many researchers are applying theory of graph [1], [2], [3], [4], [5], [6], [7] to develop variety of techniques to analysis network user behaviors or to detect worms [11], [12]. In order to detect hit-list worms, a kind of worms that bases theirs scanning strategy on the sequential probing of a predefined list of hosts, Collins et al [11] propose a graph-based algorithm solution [13] that divides the network

according to a monitored protocol such as HTTP, FTP, SMTP, or Oracle. By monitoring connections in the network, two of situations are defined to detect worms are graph inflation and component inflation. Graph inflation occurs when an attacker communicates with servers that are not active during the observation period (the interval which traffic records are observations). In this case, the cardinality of the vertex set in the protocol graph will increase. Component inflation occurs when the attacker communicates with servers already present in traffic records during the observation period , which means two or more connected components will be reduce to one. For each worms using different protocols, the detection method needs a separate observation. CAI Jun et al [11] present an analysis of the structure characters, which does not try to detect worm, but in attempt to analysis of user behaviors for web traffic. To do that, by using the algorithm of finding the community structure in bipartite network, clients are divided into variation of interest communities, clients are co-visit in the same set of servers are in the same community. Based on comparing the number of hosts and clients in one group to another, the abnormal behaviors can be defined, however need some more steps to specify which hosts in detected abnormal communities have malicious traffic.

In this work, we propose two new evaluable parameters named *MSD* and *ATS* to detect abnormal traffic based on characteristics of a connection graph which is modeled for web traffic. The method is simple to implement in real environment even for a large network. It is also useful when providing focused abnormal behavior results for each suspect clients and servers.

III. Abnormal Traffic Detection

In this section, we consider the data model and the methodology which is applied. Then, abnormal traffic detection method is introduced. First and foremost, we explain an important step known as preprocessing data to convert web traffic log from raw to formatted data which can be modeled as a connection graph. There are some steps in preprocessing data to process web log such as: data cleaning - removes log entries that are not needed, user identification.

A. Preprocessing Web Log Data

Before going to analyze the data, preprocessing data is an important step for some purposes as: remove unnecessary information, count a number of requests from a client to servers, and calculate amount of data sent between them. Web server log maintains a history of page requests. World Wide Web Consortium (W3C) is a organization to provide standard format for web server log files, but there exist some other proprietary formats also. For example IIS provides six different log file formats which are used to track and analyze information about IIS-based sites and services such as: W3C Extended Log File Format, W3C Centralized Logging, NCSA Common Log File Format, IIS Log File Format, ODBC Logging, and Centralized Binary Logging. In this work, we use NCSA common log format to analysis with format as follow: Remote host address, Remote log name (This value is always a hyphen), User name, Date, time, and Greenwich mean time (GMT) offset, Request and protocol version, Service status code (A value of 200

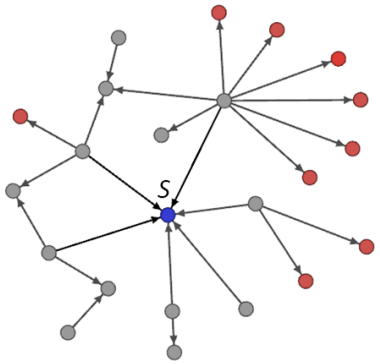

Fig. 1. A sample of client-to-server network based on web log data

```
10.24.31.38 - - [30/Oct/2012:02:00:01 +0900] "GET
http://a0.twimg.com/profile_images/2009600787/i_nor
mal.gif HTTP/1.1" 200 5914
10.31.0.160 - - [30/Oct/2012:02:00:01 +0900] "GET
http://swupmf.adobe.com/webfeed/oobe/aam10/win/upda
terfeed.xml HTTP/1.1" 304 270
10.43.0.58 - - [30/Oct/2012:02:00:04 +0900]
"CONNECT accounts.google.com:443 HTTP/1.1" 200 4470
10.37.0.154 - - [30/Oct/2012:02:00:04 +0900] "GET
http://notify8.dropbox.com:80/subscribe?10.33.0.46
- - [30/Oct/2012:02:00:06 +0900] "OPTIONS
http://eeebc28/ HTTP/1.1" 404 801
```

Fig. 2. An example of web log data file (raw data)

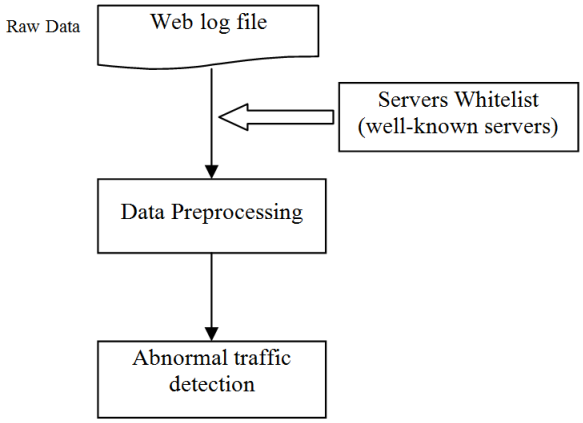

Fig. 3. Preprocessing flow of web log data

indicates that the request was fulfilled successfully), Bytes sent [14]. Fig.2 shown a part of web log data file.

One of the unnecessary information is safe requests which are requests from internal clients to well-known servers (such as $www.microsoft.com$, $www.yahoo.com$). Simply, in this step, a servers whitelist is built and with each request, which has server exists in this list, will be ruled out and known as normal traffic. This preprocessing flow is shown in Fig.3. Servers whitelist can be built on rule-based as Snort [15] typically provide rules to allow servers and server-connections on specific ports to be whitelisted [16]. Known causal accesses in experience of administration like checking update of some software should be removed from consideration. The more number of servers in whitelist, the more little real web traffic

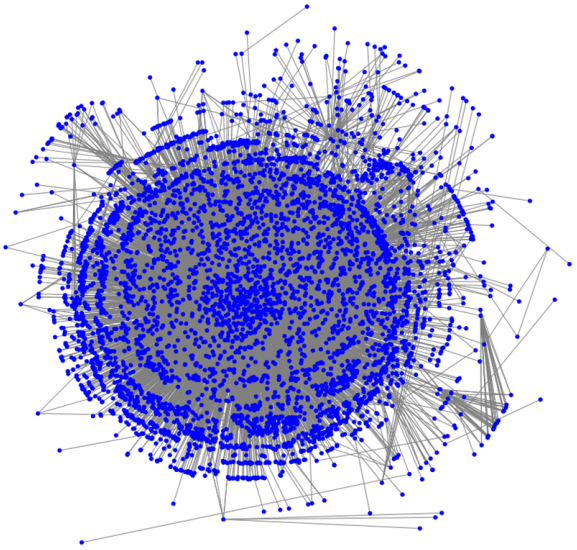

Fig. 4. Example of a graph of traffic data before preprocessing

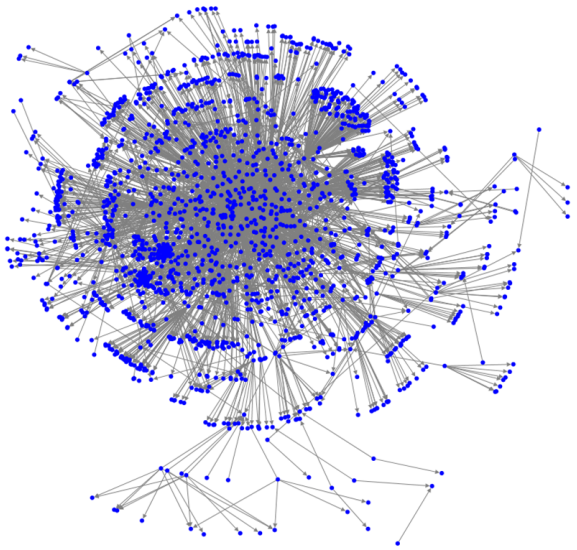

Fig. 5. Graph traffic data in Fig.4 after preprocessing

data have been processed.

For purpose of building a connection graph model, we just need some information: set of clients, set of servers and the data sent between them. Thus, in this step, data should also be grouped by couple of client and server which having connections between them.

The Fig.4, and Fig.5 illustrates the using servers whitelist in preprocessing data step help reduce a valuable amount of data which known as normal traffic. Fig.4 indicates a graph before preprocessing which includes 4842 nodes and 1606 edges. Fig.5 shows preprocessed graph, the nodes and edges were reduced 1782 and 3427 respectively.

B. Data Model and Terminology

To perform the analysis web traffic, we define a connection graph from web log data as known as directed

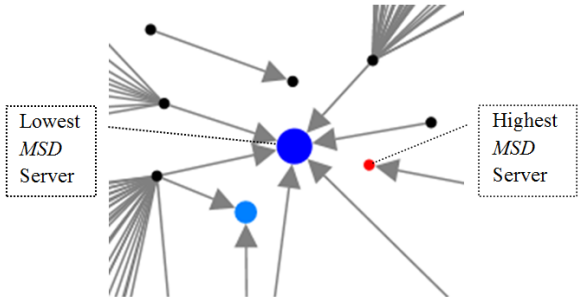

Fig. 6. Lower *MSD* servers (bigger nodes), and higher *MSD* server (smaller nodes)

client-to-server networks $G = (C, S, E)$ where $C = \{Set\ of\ all\ clients\}$, $S = \{Set\ of\ all\ servers\}$ and $E = \{(c_i, s_j)|c_i \in C, s_j \in S\}$.

The number of nodes in the graph will reduce after data preprocessing step using servers whitelist. Graph is a client-to-server bipartite graph, so out-degree of a client $DegOut(c|c \in C) \geq 1$. With a server, there is at least one connection that starts from a client so out-degree $DegOut(s|s \in S) = 0$ but in-degree $DegIn(s|s \in S) \geq 1$. Based on in-degree and out-degree, we define two parameters addressed in the paper:

Malicious Server Degree (*MSD*): a server $s \in S$ which has many accesses from clients (greater $DegIn(s)$) means server s is a popular server and vice versa, if s have few connections from clients then s is a suspect of malicious server. For example, $www.youtube.com$ is a very popular website which many people are interest to access, so $DegIn(www.youtube.com)$ get a high value. If a web server is created by cyber criminals with the purpose of sending or receiving the information to or from the bot-net infected computer then there will be only one or few access to that server. We define $MSD(s)$ that server s is a malicious server at possibility of $MSD(s)$.

$$MSD(s) = \frac{1}{DegIn(s)} \tag{1}$$

where $0 < MSD(s) \leq 1$ because of $DegIn(s) \geq 1$. In Fig.6, biggest size node server is having 5 accesses so it has lowest *MSD*, and smallest size node with one access so it has highest *MSD*.

Abnormal Traffic Score (*ATS*): for a client $c \in C$, if there are many connections to servers which are malicious suspect then the web traffic generated from c are abnormal traffic with high probability. Consideration $S_c \subseteq S$ is a set of servers which have connections from c, we define:

$$ATS(c) = \sum_{s_i \subseteq S_c} MSD(s_i) = \sum_{s_i \subseteq S_c} \frac{1}{DegIn(s_i)} \tag{2}$$

We have $Max(MSD(s_i)) = 1$, so $Max(ATS(c)) = DegOut(c)$. A client c with higher $ATS(c)$ will point out that the traffic from c is abnormal traffic with high probability. Bigger client nodes shown in Fig.7 is having higher *ATS*.

C. Proposed Method

The steps of abnormal web traffic detection is described as bellow:

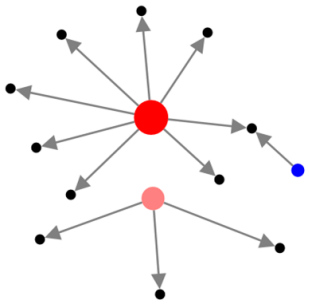

Fig. 7. Higher *ATS* clients will connect to many higher *MSD* servers

1) Input: input web log files (raw data)
2) Data preparation using server whitelist
3) Group data by clients and servers, define the connections between each client and server
4) Building directed client-to-server networks $G = (C, S, E)$ from data of step 3.
5) For each client and server, calculate $DegOut(c \in C)$, $DegIn(s \in S)$, *MSD* and *ATS* as in (1) and (2).
6) Return list of suspicious clients, sorted by *ATS*, which can generate abnormal web traffic.

Suppose that n is the number of clients and servers in the network, and k is the safe server counted in servers whitelist, m is the number of edges in network after data preparation. We focus on two big steps are data preparation and calculation of $DegOut$ and $DegIn$. In the step of data preparation, we just need to compare each server from network with servers in whitelist so running time is equal to $O(nk)$. About degree of server and client calculation, each computation we need $O(mn)$ time. We also need a $O(mn)$ to figure *ATS* for each client. Thus, total running time is $O(nk) + O(mn)$.

IV. Experimental Results

A. Data Preparation and Malicious Servers Analysis

File of web log data are collected from proxy server in 1 day worth, on Monday 01 October 2012, of web traffic in a large university environment serving a total user population in excess of 2000 clients, and accounted for over 4.2 million records (847 MB). Analyzed data is extracted within 2 hours from 08:00 AM to 10:00 AM which is in working time and people accessed the Internet very much.

As can be seen from the Table I, the number of vertices in network is decreased much after preprocessing by using servers whitelist with 200 experiential safety servers. Generated client-to-server graph is shown in Fig.8. With preprocessed pattern data (from 08:00 AM to 10:00 AM), based on *MSD*, servers are sorted out and summarized in Table II. The rate of server nodes with 80 percent, which just have 1 or 2 connection(s) shown in the table, presents that suspect servers are really filtered out in data preprocessing using servers whitelist. The visualization of suspicious servers are illustrated in Fig.9. In that, the left server nodes with bigger size are having smaller *MSD*, and we can see that the number of suspicious server nodes accounted for most of the graph space for server's side.

TABLE I. Network Information in One Day, on Monday 01 October 2012 Before and After Preprocessing

Number of Vertices	Original	After Preprocessing
Clients	1439	958
Servers	33851	12750
Edges	178655	49204

TABLE II. Classification of Servers Based on *MSD*

MSD	Number of Server Nodes	Overall Percentage of Server Nodes
0.500 $\sim$ 1.000	2710	80%
0.040 $\sim$ 0.333	600	17%
0.005 $\sim$ 0.038	73	3%

TABLE III. List Top 5 of Clients Having Highest *ATS*

No	Client IP	*ATS*	Out-Degree	Data Sent (MB)	Connections Analysis
1	10.24.11.111	263	263	1776.9	100% of connections are to servers having *MSD*=1 (highest *MSD* value)
2	10.37.0.67	101.6	206	33.2	above 80% connections are to servers having $0.1 < MSD \leq 1$
3	10.19.0.185	88.4	162	170.9	
4	10.32.0.215	68.9	185	13.3	
5	10.24.42.37	66.6	81	14.5	

B. Results Analysis

To detect abnormal web traffic, *ATS* of each client nodes are calculated. Based on *ATS*, graph is redrawn and client nodes size is proportional with *ATS* value in Fig.10. The results are shown in Table III, which is observed that based on *ATS* we can filtered out fishy clients having much abnormal web traffic. The client with IP *10.24.11.111* has the highest *ATS* value is also synonymous with he or she has many connections to servers having high *MSD* value. In the case of client IP *10.24.11.111*, 100 percent of client's requests connect to private servers with highest *MSD* value (*MSD*=1.0, which means that there is only one access to these servers).

The results still do not point out what kind of abnormal traffic from a detected client. However, by manual examination of clients we can invest that they have some deviant behaviors, such as a daily outgoing traffic to a private server which resembles a bot-net traffic, or brief communication with a large number of destinations, which looks like scanning traffic.

V. Conclusion and Future Works

In this paper, we analysed the characteristics of connection for user web accesses to detect abnormal traffic. The method is effective with two proposed parameters *MSD* and *ATS* which help to measure how much a malicious servers and abnormal traffic. By using servers whitelist in preprocessing data step also reduce the data which have to analysis. The main properties of the method are scalability to large network, simplicity of implementation, and results that are easily interpreted real time.

The results are good for pointing out which connections between clients and servers are candidates for detection as abnormal traffic, however reality abnormal traffic is still not found out. Therefore, for the future works, we are going to

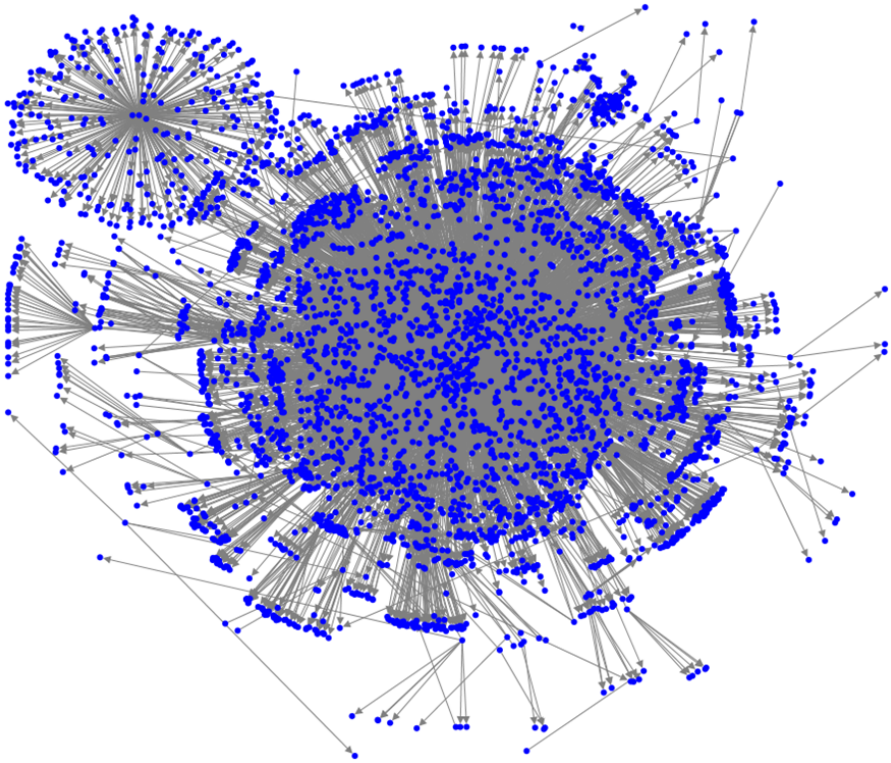

Fig. 8. Client-to-server graph after data preprocessing (data from 08:00 AM to 10:00 AM, 3949 nodes and 11980 edges)

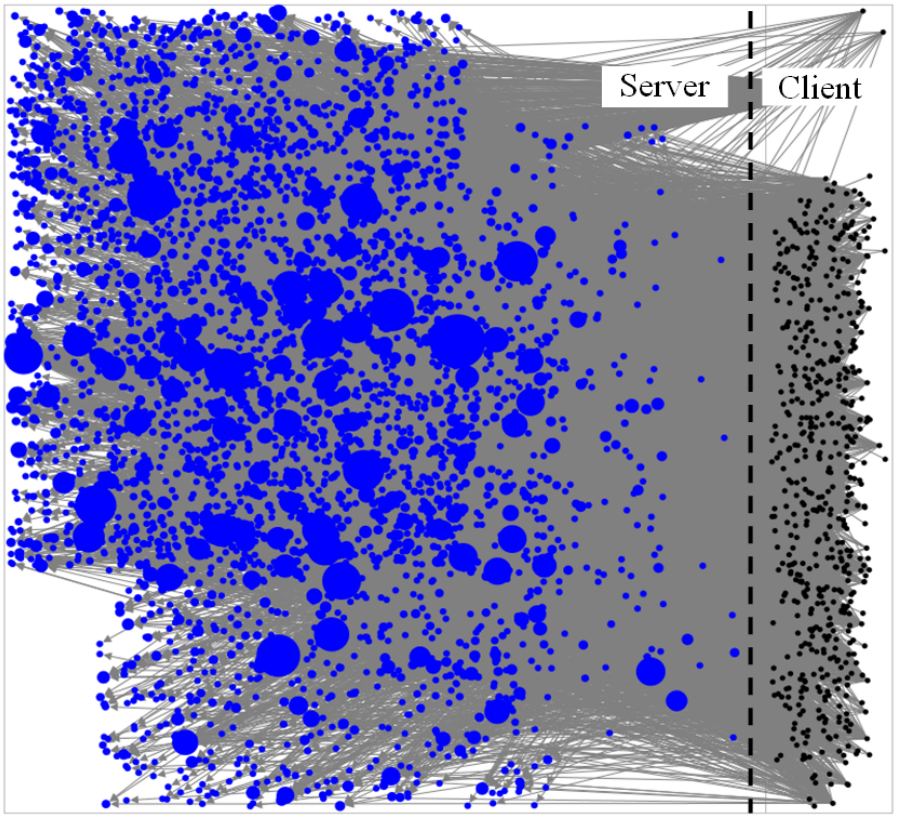

Fig. 9. Suspicious server nodes are drawn in smaller size but higher *MSD* value

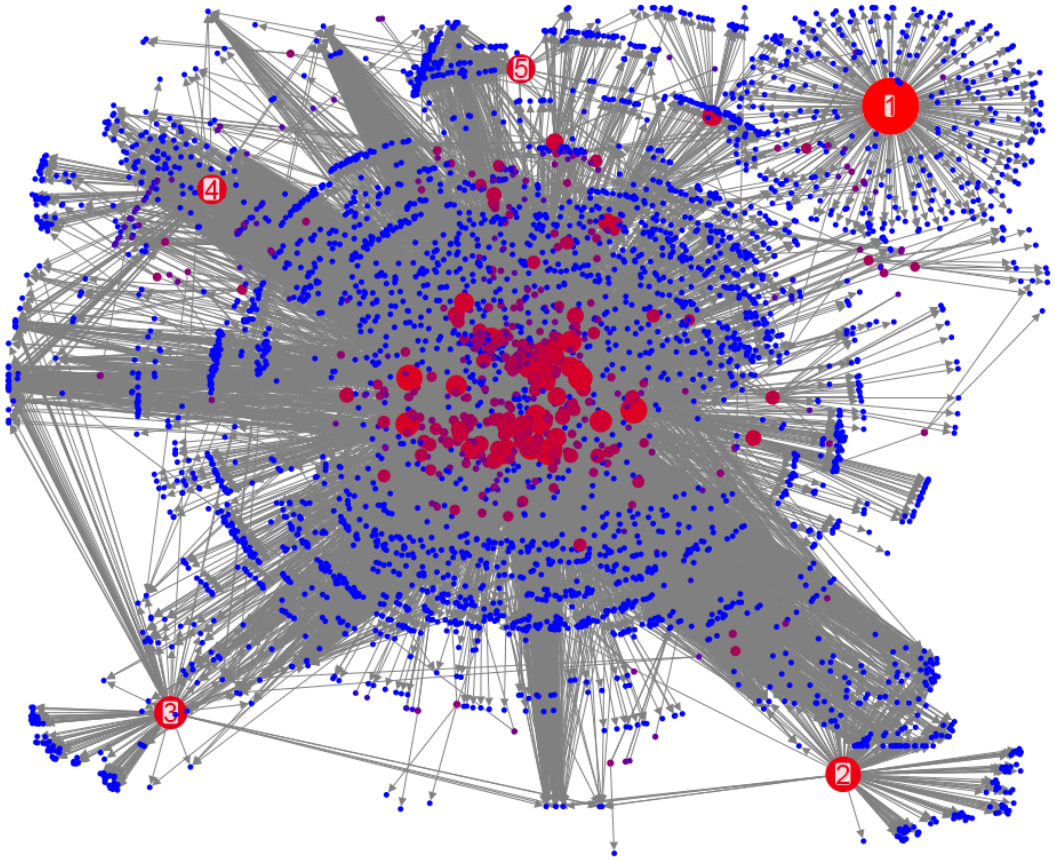

Fig. 10. Bigger client nodes mean that generated traffic are more abnormal

inherit this result to propose an algorithm in analyzing web access in real time to detect promptly the risk from internal. Our continuing research aims to model a signature web access for each host and apply it to network management, distribute of resources for security purposes.

REFERENCES

[1] M. E. J. Newman, The structure and function of complex networks, SIAM Review, Vol.45, No.2, PP.167-256 (2003).

[2] Mark Newman, Networks: An Introduction, Oxford University Press, 2010.

[3] Maarten van Steen, Graph Theory and Complex Networks: An Introduction, 2010.

[4] Ernesto Estrada, The Structure of Complex Networks: Theory and Applications, Oxford University Press, 2011

[5] M. E. J. Newman, "A measure of betweenness centrality based on random walks", Social Networks, Vol.27, pp. 39–54, 2005.

[6] Linton C. Freeman, "Centrality in Social Networks Conceptual Clarification", Social Networks, Vol.1, No.3, pp.215-239, 1979.

[7] Ulrik Brandes, "A Faster Algorithm for Betweenness Centrality", Journal of Mathematical Sociology, Vol.25, No.2, pp.163–177, 2001.

[8] Pedram Hayati, Kevin Chai, Vidyasagar Potdar and Alex Talevski, Behaviour-Based Web Spambot Detection by Utilising Action Time and Action Frequency, Proceeding ICCSA'10 Proceedings of the 2010 international conference on Computational Science and Its Applications - Volume Part II, pp.351-360, 2010.

[9] Areej Al-Bataineh and Gregory White, Analysis and Detection of Malicious Data Exfiltration in Web Traffic, 2012 7th International Conference on Malicious and Unwanted Software (MALWARE), pp.26–31, 2012.

[10] Baoyao Zhou and Jinlin Chen, "User behavior based website link structure evaluation and improvement", Proceedings of the IADIS International Conference on WWW/Internet, p.168–175, 2002.

[11] M. Patrick Collins and Michael K. Reiter, Hit-List Worm Detection and Bot Identification in Large Networks Using Protocol Graphs, Recent Advances in Intrusion Detection(RAID)'07 Proceedings of the 10th international conference on Recent advances in intrusion detection, pp.276–295, 2007.

[12] CAI Jun, YU Shun-Zheng and WANG Yu, The Structure Analysis of User Behaviors for Web Traffic, Computing, Communication, Control, and Management, ISECS International Colloquium, Vol.4, pp.501–506, 2009.

[13] Reinard Diestel, Graph Theory, Graduate Texts in Mathematics, Vol.173, Springer-Verlag, Heidelberg, ISBN 978-3-642-14278-9, 2010.

[14] Dilip Singh Sisodia and Shrish Verma, "Web Usage Pattern Analysis Through Web Logs: A Review", 2012 International Joint Conference on Computer Science and Software Engineering(JCSSE), pp.49–53, 2012.

[15] Martin Roesch, "Snort - Lightweight intrusion detection for networks", Proceedings of LISA '99: 13th Systems Administration Conference, pp.228-238, 1999.

[16] Vyas Sekar, Yinglian Xie, Michael K. Reiter and Hui Zhang, Is Host-Based Anomaly Detection + Temporal Correlation = Worm Causality?, Technical Report, CMU-CS-07-112, 2007.