Detecting Web based DDoS Attack using MapReduce operations in Cloud Computing Environment

Junho Choi, Chang Choi, Byeongkyu Ko, Dongjin Choi, and Pankoo Kim*
Chosun University, Gwangju, Republic of Korea
xdman@paran.com, {enduaranceaura, byeongkyu.ko, dongjin.choi84}@gmail.com,
pkkim@chosun.ac.kr

Abstract

A distributed denial of service attacks are the most serious factor among network security risks in cloud computing environment. This study proposes a method of integration between HTTP GET flooding among DDOS attacks and MapReduce processing for a fast attack detection in cloud computing environment. This method is possible to ensure the availability of the target system for accurate and reliable detection based on HTTP GET flooding. In experiments, the processing time for performance evaluation compares a patten detection of attack features with the Snort detection. The proposed method is better than Snort detection method in experiment results because processing time of proposed method is shorter with increasing congestion.

Keywords: DDoS Attack, HTTP GET Flooding Attack, Web Security, MapReduce

1 Introduction

DDos (Distributed Denial of Service) attacks are the most serious factor among network security risks in cloud computing environment. DDos attack is an attempt to make a machine or network resource unavailable to its intended users. Although the means to carry out, motives for, and targets of a DoS attack may vary, it generally consists of efforts to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet [18] [16] [3].

The number of such DDoS incidents[4] [17] is steadily increasing. For example, the attacks against large e-commerce sites in February 2000 and the attacks against root DNS servers in 2003 and 2007 have drawn public attention to the problem of DDoS attacks. Today, mainly mid-sized websites are attacked by criminals in order to extort protection money from their owners without attracting too much public attention. Besides that, also Internet Service Providers (ISP) have to deal with the problem that DDoS traffic is congesting their link bandwidths [9].

In the past, Layer 4 was a main target for depletion of connection resources between network equipment and server such as, TCP, UDP and so on.

IP blocking and detour were the most popular methods through decision whether the user or the attacker. The defensive approach used methods of traffic analysis, packet counting, the behavior pattern analysis of browser and so on.

However, web applications was a main target recently because DDoS corresponding solutions were effectively respond to the existing DDoS attacks.

Web applications attacks are difficult to distinguish between normal traffic and DDoS. Also, the Target system can be affected regardless of hardware performance because target server can be damaged by small connections and traffics.

Journal of Internet Services and Information Security (JISIS), volume: 3, number: 3/4, pp. 28-37

*Corresponding author: Chosun University, No. 8111, Computer Engineering, 309, Pilmun-daero, Dong-gu, Gwangju, 501-759, Republic of Korea, Tel: +82-(0)62-230-7799

Many companies have introduced DDoS detection system but they are difficult to predict for the cost effectiveness of operations by DDoS detection system.

This study proposes a method of integration between HTTP GET flooding among DDOS attacks and MapReduce processing [7] [2] for a fast attack detection in cloud computing environment. This method is possible to ensure the availability of the target system for accurate and reliable detection based on HTTP GET flooding.

The composition of this paper is as follows. Section 2 describes DDoS and HTTP GET Flooding attacks. In section 3, we introduce the large packet analysis for DDoS attack detection. Also, we explain the packet information gathering and preprocessing. And then, section 3.2 described the detection method of HTTP Get flooding in cloud computing environment. Section 3.3 proposed the MapReduce applying method of HTTP Get flooding detection. Section 4 discussed the experiment and evaluation results using the processing time for performance evaluation between the patten detection of attack features and the Snort detection. Finally, section 5 discussed about conclusion and future study direction.

2 Related Work

2.1 DDoS attack

A DoS attack is a malicious attempt by a single person or a group of people to cause the victim, site, or node to deny service to its customers. When this attempt derives from a single host of the network, it constitutes a DoS attack. On the other hand, it is also possible that a lot of malicious hosts coordinate to flood the victim with an abundance of attack packets, so that the attack takes place simultaneously from multiple points. This type of attack is called a Distributed DoS, or DDoS attack [11] [6] [15].

DDoS attack can be performed by using automated attacking tools. Some attacking tools are agents based in which agents and handlers know each other's identity while in IRC (Internet relay chat) based attacking tools, communication is done indirectly in which they do not know each other identity Some automated attack tools are Trinoo, TFN, TFN2K, Stacheldraht, Shaft, Knight and so on [10].

In the past, Layer 4 was a main target for depletion of connection resources between network equipment. However, web applications was a main target recently because DDoS corresponding solutions were effectively respond to the existing DDoS attacks. Application level DoS attacks emulate the same request syntax and network level traffic characteristics as those of legitimate clients, thereby making the attacks much harder to detect and counter. Application-level attacks are HTTP GET Flooding, Refresh attack, SQL Injection attack, CC attack and so on [5] [17] [1].

Web applications attacks are difficult to distinguish between normal traffic and DDoS. Also, the Target system can be affected regardless of hardware performance because target server can be damaged by small connections and traffics.

The best solution to the DDoS problem seems to be the following: victims must detect that they are under attack as early as possible. Then they must trace back the IP addresses that caused the attack and warn zombies administrators about their actions. In that way, the attack can be confronted effectively [11].

2.2 DDoS Attack Detection Method based on Web

Firstly, detection method based on signature can identify an attack if the monitored traffic matches known characteristics of malicious activity. In practice, bandwidth attacks do not need to exploit software vulnerabilities in order to be effective. It is relatively easy for attackers to vary the type and content of attack traffic, which makes it difficult to design accurate signatures for DoS attacks. While signature-based detection can be used to detect communication between attackers and their “zombie” computers for known attack tools, in many cases this communication is encrypted, rendering signature-based detection ineffective. This limits the effectiveness of signature based detection for DoS attacks [12].

The next is DDoS detection method based on a threshold for HTTP GET Request. The HTTP Get Flooding attack is the most critical and frequently attempted attacks and the threshold is generated from the characteristics of HTTP GET Request behaviors [14]. DDoS detection method based on a threshold for HTTP GET Request is short of accurateness since the threshold is bound to be high. Especially, the conventional method is vulnerable to up-to date DDoS attack that paralyzes the system with small amount of HTTP requests.

Finally, detection method based on user behavior is a methodology for disadvantage between the above methods. The main research contents of this method are a classification method through applying user pattern from web page, a method applying Markov model through web browsing pattern analysis, a detection method of falsification HTTP message attack through monitoring HTTP Request and so on.

2.3 HTTP GET Flooding attack

An HTTP flood is an attack method used by hackers to attack web servers and applications. It consists of seemingly legitimate session-based sets of HTTP GET or POST requests sent to a target web server. These requests are specifically designed to consume a significant amount of the server's resources, and therefore can result in a denial-of-service condition (without necessarily requiring a high rate of network traffic). Such requests are often sent en masse by means of a botnet, increasing the attack's overall power [8].

Fig 1 illustrates the packet flow of HTTP GET request attack after single TCP connection. This case is from a Single TCP connection to processing of HTTP GET request and attack Victims get damaged by less attack bandwidth.

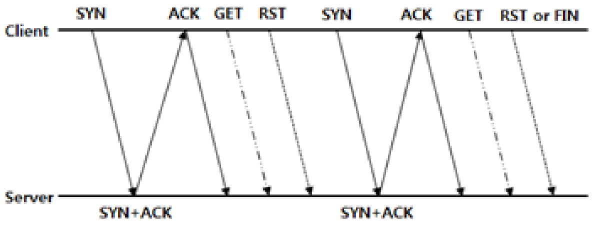


Figure 1: Packet flow of HTTP GET request attack

The attackers continually sends victim servers the HTTP GET request through session for the response. Also, multiple HTTP GET request in TCP connection are new HTTP attack format using HTTP 1.1 feature. Fig. 2 is the multiple HTTP GET request in single TCP connection.

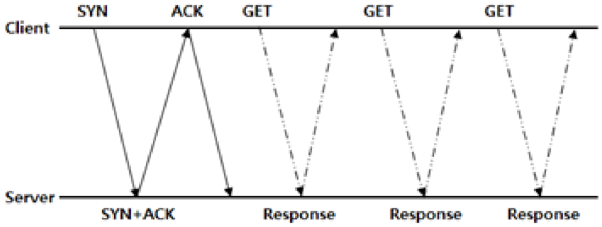


Figure 2: Multiple HTTP GET request in single TCP connection

This attack can not detect using SYN rate limit detection method. Also, it can request to change the number of pages for avoiding detection. This process is for giving a heavy load to same page or database server through multipe HTTP GET request in single connection.

HTTP flood attacks may be one of the most advanced non-vulnerability threats facing web servers today. It is very hard for network security devices to distinguish between legitimate HTTP traffic and malicious HTTP traffic, and if not handled correctly, it could cause a high number of false-positive detections. Rate-based detection engines are also not successful at detecting HTTP flood attacks, as the traffic volume of HTTP floods may be under detection thresholds. Because of this, it is necessary to use several parameters detection including rate-based and rate-invariant [8].

3 Large Packet Analysis for DDoS attack Detection

3.1 Packet Information Gathering and Pre-processing

DDos threats can be classified as an attack of Zombie Cloud Client, DDoS attack for user VM(Virtual Machine) attack between VM through partial domination of cloud server, Hypervisor attack of VM through most domination of cloud server and so on [19] [13]. Among them, hypervisor attack performs the service and data loss through authorization. Fig 3 illustrates flooding DDoS attack in cloud computing environment.

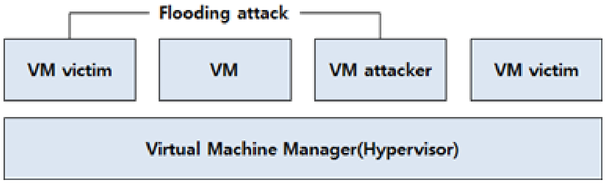


Figure 3: Flooding DDoS attack in Cloud Computing Environment

DDos attack is occured the resource imbalance between victim client and internet through packet transmission from infected zombie cloud client to victim client system. Also, transmitted huge traffics from infected host interrupt to connect victim clients. DDoS attacks have occurred, the source address is widely distributed. However the port can be changed by attack packets and attack tool. As a result, the distribution of destination address converges on small point.

The packet analysis is consist of packet loader and packet collector in cloud computing environment. Packet loader stores the files about collected packets using packet capture tool in HDFS. Packet collector performs packet information gathering through Libcap and Jpcap module from live interface. Fig. 4 is the module of the packet collector and loader.

3.2 Detection Method of HTTP GET flooding

The most typical response method of HTTP GET flooding is the analysis of request value based on packet checking. And then, it performs detection and blocking by threshold. Policy based on threshold performs a web server protection and traffic blocking using threshold through monitoring of HTTP GET Request by source terminal and destination.

Detection of DDoS attack analyzes to check the normal state of each network. Next step is a definition of parameter for network attack analysis. Final Step is a definition of threshold by parameter of normal state. In this paper, analyzed parameter is classified CPU usage, Load, packet size, information distribution of packet header, protocol distribution for classification distribution of network service, the

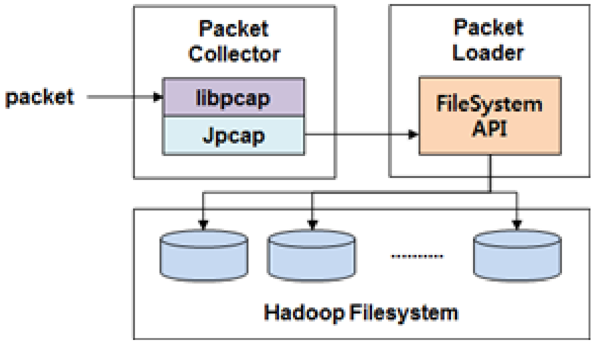


Figure 4: Module of the Packet Collector and Loader

maximum value, minimum value of traffic, monitoring of flow using spoofing address and so on. The traffic analysis through combination of parameters can be improved a reliability of traffic features.

The reliability is measured using extracted Parameters based on entropy statistical method. The entropy is a metric based on uncertainty of random variable and it is defined as Formula 1.

$$H(X) = \sum_{i=1}^{n} p_i \log p_i \tag{1}$$

p_i is the probability mass function and it gives the probability that a discrete random variable is exactly equal to some value.

GET Flooding can be divided by normal user and zombie PC attack through calculation of GRPS (GET Request per Second) because normal user does not continually request a same page at the same time.

The excess threshold by GRPS sets to source address of attacker. However, low threshold can be occurred the false-positive. Also, high threshold can be occurred false-negative. As a result, the threshold must be set carefully because extraction of GRPS value is hard.

In addition, strange detection is decided by normal range using the mean and standard deviation. However, detection methods using only the mean and standard deviation are not complete because there are the potential for that to happen in out of the normal range even if there are the packets in the normal range. Therefore, these disadvantages can be overcome through the rate of traffic. If the packets of the normal range or abnormal range during cycle availability are indicated, they can be seen as a sign of attack.

3.3 MapReduce Applying Method for HTTP GET Flooding Detection

In this paper, confrontation method of HTTP GET flooding attack is as following:

1. The suspected IP by DDoS attack is sent challenge values.

2. The IP by normal response is allowed the connection but another IP is filtered over a period of time.

3. The depletion of TCP connection are checked and huge HTTP Request are confirmed. (Creation of HTTP Request such as, image, jsp, html and so on.)

4. The detection method of DDoS attack by packet analysis is used input values of MapReduce for strange detection rule analysis using a statistical analysis and threshold.

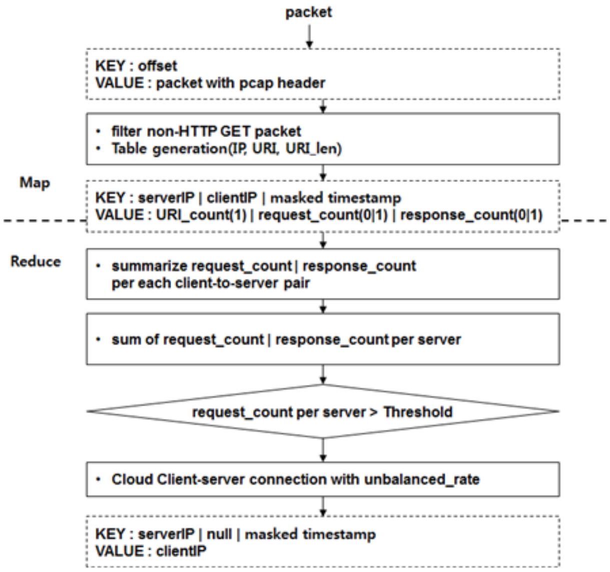

packet
KEY : offset
VALUE : packet with pcap header
• filter non-HTTP GET packet
• Table generation(IP, URI, URI_len)
Map
KEY : serverIP | clientIP | masked timestamp
VALUE : URI_count(1) | request_count(0|1) | response_count(0|1)
Reduce
• summarize request_count | response_count per each client-to-server pair
• sum of request_count | response_count per server
request_count per server > Threshold
• Cloud Client-server connection with unbalanced_rate
KEY : serverIP | null | masked timestamp
VALUE : clientIP

Figure 5: MapReduce for HTTP GET flooding DDoS detection

In fig. 5, Get_count of Map region table is increased after 3 ways hand shake if incomplete GET or POST has been entered. Finally, Get_count value of excess threshold is indicated that this case is a malicious packet. Also, Get packet is continually used a same URI. The Get packet is continually used a same URI in HTTP GET flooding attack. Therefore, malicious packet can be divided using threshold of IP, Port, URI and so on.

4 Experiment and evaluation

This study proposes a method of integration between HTTP GET flooding attack among DDOS attacks and MapReduce processing for a fast attack detection in cloud computing environment. In experiment, we use the authorized traffics among user request traffics following firewall policy and the authorized traffics is sent to master node in Hadoop cluster. The transmission traffics is performed the preprocessing after classification using packet. The Signature transmits preprocessed packet to data node using Map Function and Pattern matching is performed in parallel. The packet log information is saved the distributed file in data node. The master node using the MapReduce Function is performed anomaly detection by same file size and time schedule. This proposed system is consist of 8 computers. Gigabit ethernet and switch of each node are connected in parallel. Also, each node is consist of master, secondary and data node.

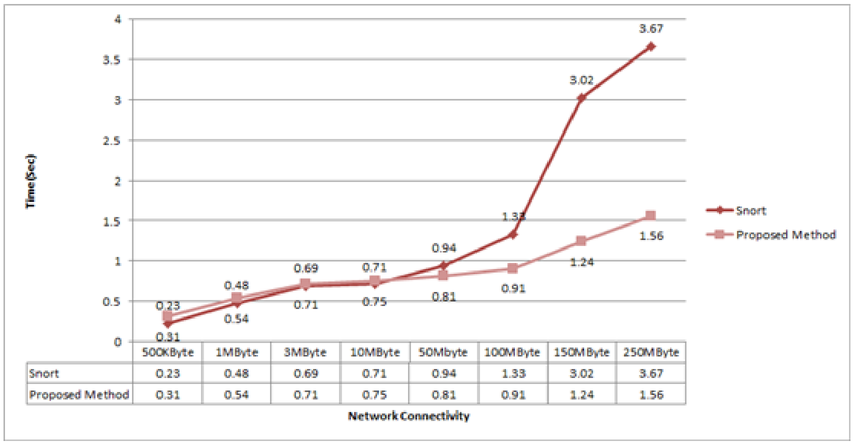
The performance evaluation of proposed method were tested accuracy, reliability and rapid detection of HTTP GET flooding attack. We use the NetBot attack tool for experiment and table 1 is the probability result of attack detection.

Table 1: Probability Results of Attack Detection

Factor	True Positive	False Positive
HTTP GET request	74.7%	0%
Incomplete get	26.1%	0%
URI	98.3%	0%

The result of detection is same in all section. Other sections showed the same detection rates because characteristic pattern of attack occurred around 1,000 in 2 second. Also, detection was possible in less than 0.01 seconds. Finally, normal packet did not show characteristics of anomaly packet.

The Experiment of detection method using MapReduce algorithm was measured the performance of detection time and rates between pattern rules of proposed system and external packet Signature. Also, the environment of evaluation was measured according to the normal environment and network congestion. The detection rule is same between pattern rules proposed system and Snort rules. Also, result analysis was used Wireshark packet analysis tool and log files in each node. Finally, The comparison experiment of detection time according to the network congestion compares the average detection time by attack of 50 times between proposed method and Snort detection method.



	500KByte	1MByte	3MByte	10MByte	50Mbyte	100MByte	150MByte	250MByte
Snort	0.23	0.48	0.69	0.71	0.94	1.33	3.02	3.67
Proposed Method	0.31	0.54	0.71	0.75	0.81	0.91	1.24	1.56

Figure 6: Comparison of detection time(Network Connectivity)

The Fig. 6 is a comparison evaluation of detection time(Network Connectivity). The proposed method is better than Snort detection method in experiment results because processing time of proposed method is shorter with increasing congestion. The proposed system are reduced about 2.11 second in case of 250Mbyte/s and This is the result of parallel processing by MapReduce.

Table 2: Comparison Evaluation between Previous Method and Proposed Method

Detection Method	**Comparison Item**	**Comparison Result**	
		Previous Method	Proposed Method
Signature based detection	Detection of varietal attack	Not detected	Detectable
Threshold based detection	Error rate	High	Low
User behavior based detection	The learning process of all site	Complex algorithm	simple algorithm

The table 2 is comparison Evaluation between Previous Method and Proposed Method. Firstly, detection method based on signature can not detect new patterns and varietal but proposed method can detect them using analysis of normal CPU usage, packet information, protocol distribution and so on. Secondly, DDoS detection method based on a threshold simply detects to use only threshold/second but proposed method shows a low error rate by threshold checking based on HTTP Response analysis. Finally, detection method based on user behavior needs definition of browsing model and analysis of all web site structure but proposed method has the advantage of simple algorithm because our method can detect a attack except analysis of web site.

5 Conclusion

This study proposes a method of integration between HTTP GET flooding among DDOS attacks and MapReduce processing for a fast attack detection in cloud computing environment. This method is possible to ensure the availability of the target system for accurate and reliable detection based on HTTP GET flooding. In experiments, the processing time for performance evaluation compares a patten detection of attack features with the Snort detection. The proposed method is better than Snort detection method in experiment results because processing time of proposed method is shorter with increasing congestion. Future work needs the study of various pattern recognition for DDoS attack detection in cloud computing environment.

Acknowledgments

This research was financially supported by the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea(NRF) through the Human Resource Training Project for Regional Innovation and also by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2013R1A1A2A10011667).

References

[1] A. Bakshi and Y. B. Dujodwala. Securing cloud from ddos attacks using intrusion detection system in virtual machine. In *Proc. of the 2010 2nd International Conference on Communication Software and Networks (ICCSN'10), Singapore, Singapore*, pages 260–264. IEEE, February 2010.

[2] J. CHOI, C. CHOI, K. YIM, J. KIM, and P. KIM. Intelligent reconfigurable method of cloud computing resources for multimedia data delivery. *Informatica*, 24(3):381–394, 2013.

[3] Y. FENG, R. GUO, D. WANG, and B. ZHANG. A comparative study of distributed denial of service attacks, intrusion tolerance and mitigation techniques, intrusion tolerance and mitigation techniques. In *Proc. of the 5th International Conference on Natural Computation (ICNC'09), Tianjian, China*, pages 628–632. IEEE, August 2009.

[4] G. Gandhi and S. Srivatsa. An entropy architecture for defending distributed denial-of-service attacks. *International Journal of Computer Science and Information Security (IJCSIS)*, 6(1):129–136, March 2009.

[5] K. Internet and S. Agency. Study on the detection and mitigation algorithm for session consuming ddos attacks on web service. Technical report, Korea Internet and Security Agency, 2010.

[6] C.-H. Lin, C.-Y. Lee, J.-C. Liu, C.-R. Chen, and S.-Y. Huang. A detection scheme for flooding attack on application layer based on semantic concept. In *Proc. of 2010 International Computer Symposium (ICS'10), Tainan, Taiwan*, pages 385–389. IEEE, December 2010.

[7] R. Lämmel. Google's mapreduce programming model — revisited. *Science of Computer Programming*, 70(1):1–30, January 2008.

[8] R. Ltd. DDoSPedia - HTTP Flood. `http://security.radware.com/knowledge-center/DDoSPedia/http-flood/`, 2013.

[9] R. Mehran, G. Markus, S. Armin, and B. Thomas. Method and system for throttling or blocking geographical areas for mitigation of distributed denial of service attacks using a graphical user interface. `http://www.freepatentsonline.com/EP2109280.html`, September 2013.

[10] A. Mishra, B. B. Gupta, and R. C. Joshi. A comparative study of distributed denial of service attacks, intrusion tolerance and mitigation techniques, intrusion tolerance and mitigation techniques. In *Proc. of the*

2011 European Intelligence and Security Informatics Conference (EISIC'11), Athens, Greece, pages 286–289. IEEE, September 2011.

[11] C. Patrikakis, M. Masikos, and O. Zouraraki. Distributed denial of service attacks. *The International Protocol Journal*, 7(4):13–35, December 2004.

[12] T. Peng, C. Leckie, and K. Ramamohanarao. Survey of network-based defense mechanisms countering the dos and ddos problems. *Journal of ACM Computing Surveys (CSUR)*, 39(1):1–42, April 2007.

[13] R. N. C. Rajkumar Buyya, Rajiv Ranjan. Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities. In *Proc. of the 7th High Performance Computing and Simulation (HPCS'09), Leipzig, Germany*, pages 1–11. IEEE, June 2009.

[14] Y. seo Choi, I.-K. Kim, J.-T. Oh, and J.-S. Jang. Aigg threshold based http get flooding attack detection. In *Proc. of The 13th International Workshop on Information Security Applications (WISA'12), Jeju Island, Korea, LNCS*, volume 7690, pages 270–284. Springer-Verlag, August 2012.

[15] F. B. Shaikh and S. Haider. Security threats in cloud computing. In *Proc. of 6th International Conference on Internet Technology and Secured Transactions (ICITST'11), Abu Dhabi, United Arab Emirates*, pages 214–219. IEEE, December 2011.

[16] B. s. Sumit kar. An anomaly detection system for ddos attack in grid computing. *International Journal of Computer Applications in Engineering, Technology and Sciences (IJ-CA-ETS)*, 1(2):553–557, April-September 2009.

[17] S. Suriadi, D. Stebila, A. Clark, and H. Liu. Defending web services against denial of service attacks using client puzzle. In *Proc. of the 9th International Conference on Web Services (ICWS'11), Washington DC, USA*, pages 25–32. IEEE, July 2011.

[18] t. f. e. Wikipedia. Denial-of-service attack. `http://en.wikipedia.org/wiki/Denial-of-service_attack`, 2013.

[19] M. Zakarya and A. A. Khan. Cloud qos, high availability and service security issues with solutions. *International Journal of Computer Science and Network Security (IJCSNS)*, 12(7):71–79, July 2012.

Author Biography

Junho Choi received a doctoral degree in the Department of Computer Engineering at Chosun University of Korea in 2012. Currently, He is working as a lecturer at the same university. His research interests include semantic information processing, semantic web, multimedia processing and system security.

Chang Choi received a doctoral degree in the Department of Computer Seience at Chosun University of Korea in 2004. Currently, He is working as a lecturer at the same university. His research interests include multimedia processing, semantic information processing, ontology engineering and semantic web.

Byeongkyu Ko is a student for the doctoral degree in computer engineering from Chosun University of Korea. He is received a master degree at the same university in 2012. His research interests include web documents classification, Natural Language Processing, semantic information processing and semantic web.

Dongjin Choi is a Ph.D student in the Department of Computer Engineering at Chosun University in Korea. His research interests include Semantic information processing, Text mining and Natural language processing. Contact him at 8109 IT Building Chosun University, 375 Seoseok-dong Dong-gu Gwangju, 501-759, Korea.

Pankoo Kim received his M.S. and Ph.D. degrees in Computer Engineering from Seoul National University, Korea in 1994. He is a full professor in the Department of Computer Engineering at Chosun University. He is an editor-in-chief of IT CoNvergence PRActice(INPRA) Journal. His specific interests include semantic web techniques, semantic information processing and retrieval, multimedia processing, semantic web and system security.