Malware Containment Using Weight based on Incremental PageRank in Dynamic Social Networks

Jong-Hwan Kong and Myung-Mook Han*
Department of Computer Engineering, Gachon University
Seongnam, Korea
[E-mail: ball3314@naver.com, mmhan@gachon.ac.kr]
*Corresponding author: Myung-Mook Han

Received October 12, 2014; revised December 3, 2014; accepted December 18, 2014; published January 31, 2015

Abstract

Recently, there have been fast-growing social network services based on the Internet environment and web technology development, the prevalence of smartphones, etc. Social networks also allow the users to convey the information and news so that they have a great influence on the public opinion formed by social interaction among users as well as the spread of information. On the other hand, these social networks also serve as perfect environments for rampant malware. Malware is rapidly being spread because relationships are formed on trust among the users. In this paper, an effective patch strategy is proposed to deal with malicious worms based on social networks. A graph is formed to analyze the structure of a social network, and subgroups are formed in the graph for the distributed patch strategy. The weighted directions and activities between the nodes are taken into account to select reliable key nodes from the generated subgroups, and the Incremental PageRanking algorithm reflecting dynamic social network features (addition/deletion of users and links) is used for deriving the high influential key nodes. With the patch based on the derived key nodes, the proposed method can prevent worms from spreading over social networks.

Keywords: Malware Containment, Dynamic Social Networks, Incremental PageRank Algorithm, Subgroup Detection, Patch Distribution

This research was funded by the MSIP(Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2014.
A preliminary version of this paper was presented at APIC-IST 2014 and was selected as an outstanding paper.

http://dx.doi.org/10.3837/tiis.2015.01.026 **ISSN : 1976-7277**

1. Introduction

Social network services use the concept of social relations to build social relations over the Internet among online users. The Internet environment, web technology development, and the increasing widespread use of smartphones have led to fast-growing social network services, and today, users can easily communicate with their friends and share information by using these services. In addition, social networks allow users to convey a variety of information and news among users and participants; therefore, these networks have a great influence on the public opinion formed by social interactions among users as well as the spread of information. Major social network services include Twitter, Facebook, and Kakao Talk. As the number of social network users is rapidly increasing, businesses and advertisers are changing marketing strategies to focus on social network services. On the other hand, a key feature of social network services is relation formation based on trust among the users and information sharing. Hackers exploit the relational feature to spread viruses or worms infecting user computers with malware in order to gain personal information, to form a botnet for DDoS(Distributed Denial of Service) attacks, and to send spam messages. As a result, the variety of intelligent damaging attacks is rapidly growing, and social networks built on trust among users are becoming a perfect environment for the spread of malicious codes. For example, there is Koobface Worm, a type of Trojan Worm that appeared in 2008 [1]. Koobface Worm mostly spreads by delivering camouflage video link messages to users via social networks. The message recipients are prompted to download a video codec or an update required to play videos and to execute the file, but the file is actually a replica of a malicious code. It can be classified as a sociotechnological attack and shows the fast rate of infection since users tend to trust one another over social networks. These attackers target highly influential users with vulnerable security over social networks in order to spread the worms efficiently. Updating all the users' computers with security patches can be a simple solution to these SNS-based worms, but this method is very inefficient in terms of time, cost, and resources. An effective way of coping with SNS-based worms is to extract nodes that are most likely to spread these worms and to distribute the corresponding security patches to the nodes. It is possible to achieve effective security patch deployment when security patches are selectively distributed to the highly influential users over a social network. Immunity against these worms can quickly be formed, and the worms can be quickly responded to. In addition, it is effective to derive highly influential users by finding subgroups in the social network rather than by analyzing the entire nodes in the network. The method proposed in this paper uses modularity for the correct subgroup partitioning in a dynamic social network to generate subgroups and uses the incremental PageRank algorithm to quickly extract highly influential users from the given group. Therefore, a graph using the weighted direction and activity values is created in order to reflect the characteristics of a dynamic social network, and the incremental PageRank technique is used to reflect the characteristics of the dynamically changing social network efficiently.

The rest of this paper is organized as follows: In Section 2, the methods of social network graph analysis and those of user-to-user influence analysis are presented, and the PageRank algorithm is described. In Section 3, the existing research trends in countermeasures against SNS-based worms and the problems of the existing studies are described. In Section 4, the proposed method that applies the incremental PageRank algorithm over a dynamic social

network is described. In Section 5, the experimental results of the proposed method are described. In Section 6, the conclusions and the directions for future work are described.

2. Related Work

2.1 Methods of Social Network Graph Analysis

It is important to find the subgroups that exist in a social network for an efficient analysis of its structure. One subgroup means a community within the network and its nodes are strongly connected. There is a relatively weak connection between subgroups that are connected by a bridge node. Various methods of finding these groups have been proposed: graph partitioning, spectral bisection, hierarchical clustering, clustering, etc. [2]. In [3], a method of solving the problem of finding the community is proposed by combining the defined common neighborhood subgraph density and the affinity propagation algorithm.

2.2 Methods of User-to-User Influence Analysis

Social network analysis that analyzes the structure established by social relationships can analyze social roles and effects by means of relationship modeling, relationship strength, and density (high/low). The influence over nodes in the network can be seen by analyzing the links established to send and to receive information [4]. In the field of computer science, there are studies on efficient search methods expanding connections among the SNS users, and there are studies on social phenomena, social network analysis, and social network configuration [4-6]. Nodes represent actors, which are the basic configuration elements of the network, such as people, area, and resources. Links represent various relationships between these nodes. These links can represent the relationship status between nodes, the direction, the strength, etc. A network analysis of the actors can be made via indirect networks such as systems recommended on Amazon sites, where there are no direct links between the actors. There have been various studies on collaborative filtering and relationship inference via these indirect networks [7][8]. In addition, various types of subgroups can be found within the network. These subgroups can be observed within the network and used for identifying its community structure [9].

2.3 PageRank Algorithm

The PageRank algorithm is based on the following observations: more important web pages are more likely to receive more links from other sites. The PageRank algorithm measures the web page priorities in order to determine their relative importance based on the webgraph [10]. Each web graph has forward links that go to other web pages and backlinks that point to the web page. In **Fig. 1**, web page A has a large number of backlinks. In general, a web page with a large number of backlinks will have higher importance. However, there are many cases in which the number of backlinks does not match importance in the standard scene. For example, in **Fig. 1**, page B is an important page but does not have a large number of backlinks like E. However, it has to have a higher priority because it has backlinks from B. Therefore, the PageRank algorithm covers a page with many backlinks and a page with a few highly ranked backlinks. The PageRank value for a page can be calculated as shown in Equation (1).

$$PR(n_i) = \alpha \sum_{j \in B(n_i)} \frac{PR(n_j)}{|F(n_j)|} + \frac{1-\alpha}{N} \tag{1}$$

$PR(n_i)$ denotes the PageRank for n_i; α, the damping factor; N, the total number of web pages; $B(n_i)$, the set of pages that links to n_i; and $|F(n_j)|$, the number of forward links on page n_j. The damping factor is the probability that the person continues surfing the web by clicking on links, and the value of α is usually 0.85 [11]. The PageRank value for n_i is basically equal to the sum of the PageRank values for the pages that point to page n_i. When a page points to the number of pages N, the probability that the page heads for a particular page is $\frac{1}{n}$. In addition, the Page Rank value for page n_i is divided by the number of forward links on the page ($|F(n_j)|$). When it is assumed that the page can only go to the linked web pages, there can be a loop problem between web pages. Such a problem is called a rank sink. In order to resolve this problem, the probability of going to a random page without following the links $(1 - \alpha)$ is added.

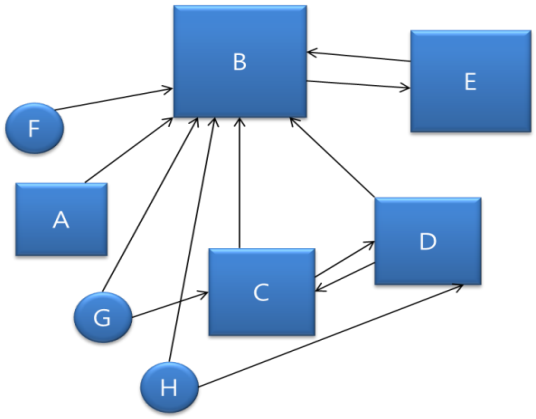

Fig. 1. PageRank Algorithm Principle

3. Analysis of Existing Research Trends and Issues

3.1 Existing Research Trends

The research trends in countermeasures against SNS-based worms can be mainly divided into the following two types: malware detection through log analysis and graph partitioning for effective security patch distribution with highly influential node extraction. According to the studies on malware detection based on log analysis, certain specific SNS data are collected, and worm log analysis and pattern analysis are conducted by using virtual machines to detect malware from the collected data [12]. On the other hand, according to the studies on countermeasures based on effective security patch distribution, the social network graph is partitioned for efficient social network analysis and effective patch distribution. The security patches are selectively distributed to the highly influential nodes extracted from the divided subgroups [13][14]. Malicious SNS-based worms first need to be analyzed before they can be handled properly. However, in order to effectively respond to SNS-based worms proactively, it is necessary to distribute the security patches faster than the spread of these worms.

3.2 Problem of existing studies

The method proposed in [14] uses modularity to identify the community based on the structure of the social network graph. In addition, it defines the features of dynamically changing social networks: newUser, newLink, removeUser, and removeLink. It proposes the adaptive community update algorithm to reflect dynamically changing social networks. These four dynamic characteristics can be used for graph partitioning to identify the network structure reflecting the changes in the dynamic graph structure without re-computation. After identifying the community, the most influential node is selected from the partition to cope with the worm in the process of selective patch distribution. The method proposed in [14] can rapidly identify the structure of dynamically changing social networks, and it basically uses an unweighted and undirected graph to perform its partitioning when the network graph is generated. It uses a quick greedy algorithm to select the key node from each partition. There is no direction and weighting in the graph. That is, the relationships between nodes are considered to be equal, and key nodes are selected on the basis of the degree of the nodes. However, this method has the following drawbacks: The reliability of key nodes may be reduced in actual social networks because the influence of the users is measured with unweighted links. It is important to consider the users' activities and structural features to select key nodes. In addition, the greedy algorithm is not used for the key node selection since it requires high computational complexity and a large amount of information.

4. Incremental PageRank–Based Response in Dynamic Social Networks

The method proposed in this paper consists of four steps like a conventional study. As shown in **Fig. 2**, the four-step strategy consists of the graph generation step to analyze the structure of the social network, the subgroup detection step to find the optimal subgroup within the network, the key node selection step to select the key node within the subgroup, and the patch deployment step to center on the selected key nodes for security patch distribution.

A previous method using modularity [14] adopted the structure in which graph data are input and a non-directed graph is generated to perform modularity. Further, a method for performing patches by selecting many connections or nodes connected with other communities in the process of selecting influential users in the modularized subgraph has been proposed. The previous method has problems such as difficulties in reflecting the influence of actual nodes due to the non-existence of the directionality of graphs, degradation in the performance of modularity when considering the value of modularity and the time taken by the search algorithm for detecting subgraphs, and the consideration of only the degree of connection and the connectivity with other communities in the method of selecting influential nodes. For estimating the actual influences of a social network, node activities (postings of entries) of the corresponding nodes rather than the degree of connections should be considered. Moreover, there is a disadvantage of requiring a considerable amount of information and calculation due to the use of greedy algorithms in selecting nodes having much influence in each community. Because the number of nodes and the amount of information generated is very large in actual social networks, efficiency can be reduced and overhead can be generated when greedy algorithms are used. There is also the disadvantage of increased calculation costs for recalculating subgroups and selecting key nodes, which are characteristics of frequently changing dynamic social networks. In order to improve these problems, in this paper, we propose a modularity method, which is faster than conventional methods in a dynamic social network environment, and an incremental PageRank-based patch method, which is efficient in selecting key nodes for a patch in a dynamic social network. A flowchart of the proposed

method is shown in **Fig. 2**.

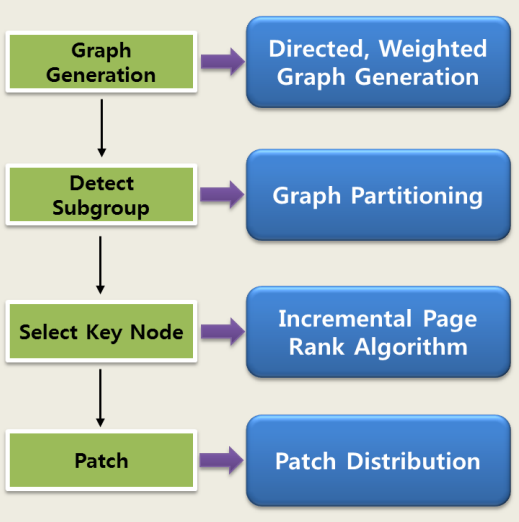

Fig. 2. Flowchart of Proposed Method

The method proposed in [14] has been used for relatively efficient subgraph detection, and influences are measured for each node by extracting users with higher influences in the social network by using the PageRank method based on the information of user activities. Further, an incremental PageRank method for a subgraph is proposed and can reflect the characteristics of dynamic social networks.

4.1 Graph Generation

Social network data are collected, and a directed graph is generated using the collected data. In order to reflect the user activities, the links between the nodes within the network are assigned weights, which are calculated on the basis of the frequency of measurable data such as the number of messages sent between users.

4.2 Detect Subgroup

In this paper, the proposed method uses modularity, which is mainly used to accurately partition a subgroup within the network [15]. The modularity value Q is defined in equation (2):

$$Q = \sum_{S \in \mathcal{S}} \left[\frac{m_s}{M} - \frac{d_S^2}{4M^2} \right] \tag{2}$$

Modularity is a measure representing the property of existence of many links in a subgroup, such as a community and a small number of connections between groups. Subgroups are generated through combinations of each node in the network by using these modularity values, and a subgroup having a high modularity value Q is generated. This method has the advantage of enabling very fast modularization and extending it to a large number of nodes.

4.3 Incremental PageRanking

A PageRank value indicates the relative importance of a web page and is used as an important factor in web page search. The influence of a node belonging to a subgroup in a network can be calculated using the same principle. In addition, the incremental PageRank

algorithm is used for efficiently reflecting dynamic social network features such as node addition and deletion, and relationship changes. It is very inefficient and time consuming to calculate the PageRank whenever there are structural changes such as node addition and node deletion. It is assumed that the short-term impact of the newly added node in the social network is not greater than that of the existing nodes. The PageRank is calculated only for the newly added structural changes, and hence, the dynamic social network features can be reflected. If the number of newly added nodes is considerably less than the number of existing nodes, it is assumed that changes in the PageRank values for the existing nodes are small. Therefore, it is possible to use the incremental PageRanking algorithm. For the incremental PageRank computation, the initial values for the newly added nodes are set and the sum of all nodes is assumed to be 1. PageRank converges on one value irrespective of the setting of the initial values. There is a difference only in the number of iterations required for the convergence [16]. For faster convergence, the initial value is set to compute the rank between the nodes.

Table 1. Incremental PageRank Algorithm for Dynamic Social Graph

Incremental PageRank algorithm for dynamic social graph

Input: A subgroup network. G, q damping factor ≈ 0.15
Output: Updated users' PageRank pr
If $pr(v) = Null$ then
Set $pr(v_n)$
End If
If (new node || Remove node) then
While PageRank converging do
For each $p \in G$ do
$\Gamma^+(p) \leftarrow$ pages pointed by p
For each $p' \in \Gamma^+(p)$ do
$Aux_{p'} = Aux_{p'} + \frac{pr_p}{\lvert\Gamma^+(p)\rvert}$
End For
End For
For each $p \in G$ do
$pr_p = \frac{q}{N} + (1-q)Aux_p$
$Aux_p = 0$
End For
End While
End

4.4 Patch Distribution Method

The patch distribution method is based on the following two assumptions:

• When a social network-based worm is detected, the patch system issues a warning and carries out patchwork.

• The node that received the patch carries out the task of transferring the patch to the corresponding node and the neighboring nodes.

Response is possible through an efficient security patch based on these general security

patch matters. In the previous steps, patch is carried out on the basis of the key nodes in each subgroup, which are extracted through a gradual PageRank algorithm. By distributing efficient security patches centering around the nodes with high influences for each community, it is possible to efficiently respond to social network-based worms. Further, key nodes are calculated for a fast selection of key nodes in a subgroup by applying a quick sort on the value of the PageRank of each node. This makes it possible to respond efficiently on the basis of the modified page rank values when modifications such as the addition of new nodes or the deletion of the existing key nodes have occurred among various characteristics of a dynamic social network. Table 2 shows the algorithm in which the quick sort method is applied to a PageRank list for an efficient selection of key nodes and patches.

Table 2. Patch Distribution

Patch distribution

Input: A subgroup PageRank list pr[]={pr[]$_1$, pr[]$_2$, …, pr[]$_p$}
Output: Set of influential users
 	
quicksort(pr[], top, end)
if(top<end) then
p ← partition(pr, top, end);
quicksort(pr[], top, p-1);
quicksort(pr[], p+1, end);
end if
end

5. Experimental Results

In this section, the experimental results for the proposed method will be described. The Facebook network dataset [18] is used as the data for the experiments; experiments of comparison have been performed between the previous method [14] and the proposed method through a modularity performance evaluation, and experiments on the infected rate through a patch for the methods. The Facebook dataset is a set of data collected from September 2005 to January 2009, which is composed of 63,731 nodes and 1,445,687 edges. R [19], an open-source statistical analysis program, is used for the experiment. In this experiment, a stepwise evaluation of performance has been made according to the modularity steps and the selection of key nodes and patch methods for comparing the previous method [14] and the method proposed. Each experiment has been repeated for 50 times, and the mean value is used for the evaluation.

5.1 Evaluation of modularity performance

In the previous method [14], the algorithm proposed in [17] is used as the algorithm for generating the subgroup, and comparisons of performance have been made between representative algorithms [20][21] for detecting subgroups and the algorithm for detecting subgroups proposed in this paper. Table 3 shows the experimental results for comparing performances of the algorithms used in the four methods. The results that can be derived from the table make it possible to perform a comparative analysis of the algorithm that enables an efficient search of subgroups in large-scale social networks. Although the previous method provides the advantage of generating subgroups by designating the number of subgroups, there is a disadvantage of making it impossible to efficiently generate subgroups when the number of subgroups is incorrectly designated. The application of the previous method can be

regarded as a non-efficient method considering the extensibility of social networks. The algorithm used in this study automatically adjusts the number of subgroups and can obtain modularity values similar to the previous method; although the values are slightly lower by 0.01, the values are relatively high as compared to other algorithms. In addition, the time taken for generating the subgroups has also been compared. The time taken for generating subgroups can be considered the important criterion for rapidly identifying the structure of a social network. The previous method took an average of 1,572.7 s to generate subgroups for the same graph, while the proposed method took an average of 5.54 s. Although the previous method [21] shows the fastest execution time, it is not considered an appropriate method due to the relatively low value of modularity. Further, the number of clusters is set to 79 for the comparison of the previous method [14] and the proposed method. As can be seen in the table, the algorithm proposed in this paper can be regarded as a more efficient and appropriate algorithm for identifying the structure of a social network. Furthermore, when comparing the number of clusters, the number of clusters detected by the proposed algorithm, which is different from the previous method generating subgroups by designating the number of clusters, is less than that for other algorithms for detecting subgroups. It can be seen that the proposed method satisfies the criteria of carrying out the patch on the basis of a minimum number of subgroups and key nodes while having a relatively high value of modularity for an efficient patch strategy. This could be attributed to the fact that more resources and costs are consumed for the patch strategy when there are more subgroups and key nodes. Therefore, in this study, we use a multi-level modularity algorithm, which exhibits excellent performance as compared to the previous methods, as shown in Table 3. It can be concluded that the multi-level modularity algorithm used in this study is better than the previous methods in terms of the modularity value and the algorithm performance time. Therefore, it is suitable for executing an efficient patch strategy.

Table 3. Modularity Test

	Modularity value	Minimum time elapsed	Maximum time elapsed	Average time elapsed	Number of clusters
Multi-level modularity algorithm [15]	0.59	4.89 s	6.72 s	5.54 s	79
Spin-glass algorithm [17]	0.60	1478.72 s	1722.56 s	1572.7 s	79
Walk trap algorithm [20]	0.56	664.3 s	822.59 s	734.5 s	5719
Label propagation algorithm [21]	0.038	2.46 s	4.15 s	3.8 s	380

5.2 Method for selecting key nodes for each cluster and performance evaluation of reflecting characteristics of dynamic social networks

In deriving key nodes for each cluster, it is necessary to obtain information that enables the measurement of the influences per node. Information on the activities of users is used as an important factor in measuring influences in social networks. In order to acquire this information, posting frequencies of Facebook data are measured to obtain weights for each user node. Based on the weights measured, the PageRank values are derived for each subgroup previously generated. Further, in order to reflect the characteristics of dynamic social networks, situations such as addition of new nodes and deletion of existing nodes are

determined first, and key nodes are selected for each cluster by comparing the values of ranks of each node based on the incremental PageRank algorithm mentioned above. While the method can be inefficient when the task of PageRank is performed for the entire graph since a considerable amount of calculation is required to reflect the information of social networks that are frequently modified, it can be used efficiently for deriving key nodes when the PageRank task is performed for each subgroup and the information of frequently modified social networks can be easily reflected. The average time for deriving key nodes in conventional networks is 2.21 s, which is very fast. The average time for performing the tasks such as addition of arbitrary nodes and deletion of existing nodes as the characteristics of dynamic networks is 4.72 s, which is considered to be relatively fast.

Table 4. Selecting key node performance

	Static network	Dynamic network
Proposed method	2.21 s	4.72 s
Nguyen's method	3.47 s	27.2 s

5.3 Infection rate and recovered rate according to patch method

Experiments on infection rate have been carried out assuming that 0.5% of the dataset is initial users infected by a worm. The experiment on the infection rate compares the maximum infection rates generated by the method discussed in [14] and the proposed method by using the SIR model [22], an influential model, when the worm starts propagating. The infection rate is the ratio of the number of infected users to the total number of users. The subgroups for the comparison are set to have the same number for each patch method. The results of each experiment are used in the performance evaluation on the basis of the maximum infection rate for carrying out patch tasks and the total time (unit time) consumed for restoration.

In **Fig. 3**, it can be seen that the maximum infection rate of the proposed method is lower than that of the previous method [14] by 14%. This can be attributed to the fact that the maximum value of the infection rate is lowered by performing the patch steps according to the network analysis method, which is faster than the previous methods. **Fig. 4** presents the result of the recovery rate through the patch steps. As for the time taken in processing the patch for the entire node, the proposed method took 55 unit time, and the previous method took 101 unit time. Therefore, the proposed method is considered to show excellent performance as compared with the previous method.

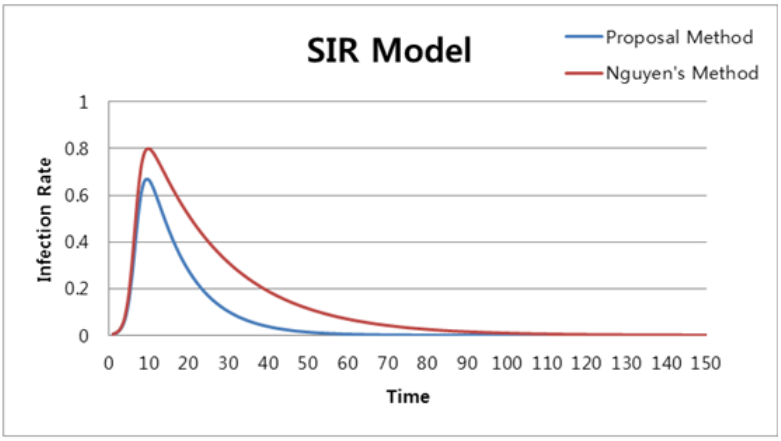

Fig. 3. Result of Infection Rate Test

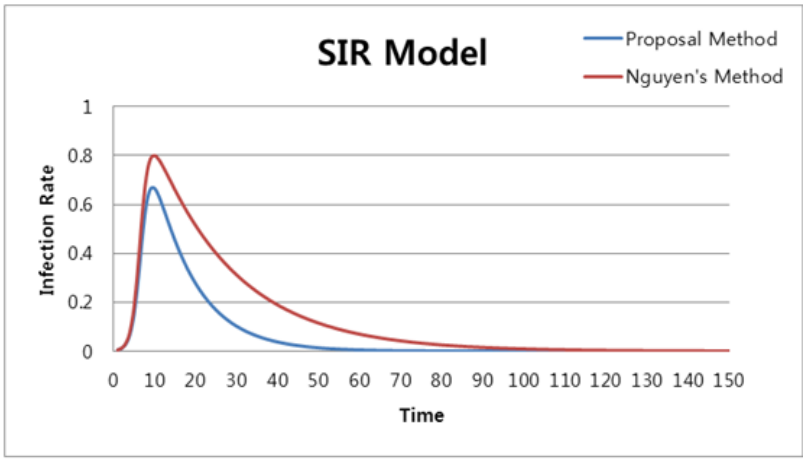

Fig. 4. Result of Recovery Rate Test

6. Conclusion and Future Work

In this study, in order to respond to a social network-based worm more promptly, graphs are generated on the basis of social network data, and subgroups having higher modularity values are generated using modularity. Key nodes are derived by reflecting the influences between the nodes belonging to the subgroup generated by the gradual PageRank algorithm that can reflect the characteristics of a dynamic social network (addition and deletion of nodes), and a patch strategy is proposed to carry out the patch on the basis of the derived key nodes. In order to use the gradual PageRank, the directionality of graphs and weights on the user activities are given, generating more reliable results in selecting key nodes than in the methods using a conventional non-directed, non-weighted graph. By carrying out a patch based on the key nodes in the subgroup by using the proposed method, it is possible to respond to the propagation of a social network-based worm more promptly. Further, the experimental results showed that the proposed method decreased the maximum infection rate by 14% as compared to the previous method and that the time for carrying out a patch for the entire node was excellent.

In a future study, research will be conducted on a response method that can reflect optimal subgroup division methods for an efficient structure analysis of social networks, and features for each type of worm.

References

[1] Thomas, Kurt and David M. Nicol., "The Koobface botnet and the rise of social malware," in *Proc. of Malicious and Unwanted Software (MALWARE), 2010 5th International Conference on*. IEEE, 2010. Article (CrossRef Link).

[2] M.E.J. Newman, "Detecting Community structure in networks," *European Physical Journal B-Condensed Matter and Complex Systems*, Vol.38, No.2, pp.321-3, 2006. Article (CrossRef Link).

[3] Yoonseop Kang and Seungjin Choi, "Social Network Analysis using Common Neighborhood Subgraph Density," *Journal of KIISE:Computing Practices and Letters*, Vol. 16, No. 4, pp. 432-436, 2010. Article (CrossRef Link).

[4] Eun-Young Kang and Kee-Young Kwahk, "Managing Duplicate Memberships of Websites : An Approach of Social Network Analysis," *Journal of Intelligence and Information Systems*, Vol. 17,

No. 1, pp. 153-169, 2011. Article (CrossRef Link).
[5] C.T. Butts, "Social network analysis: A methodological introduction," *Asian Journal of Social Psychology*, Vol. 11, pp. 13-41, 2008. Article (CrossRef Link).
[6] Hyunjin Lee and Taechang Jee, "Social Networks Analysis using External Community Relationship," *Journal of Digital Contents Society*, Vol. 12, No. 1, pp. 69-75, 2011. Article (CrossRef Link).
[7] Hyoung-Do Kim, "Collaborative Filtering by Consistency Based Trust Definition," *Journal of Society for e-Business Studies*, Vol. 14, No. 1, pp. 1-11, 2009. Article (CrossRef Link).
[8] Seung-Hoon Lee, *et al*., "Inferring and Visualizing Semantic Relationships in Web-based Social Network," *Journal of Intelligence and Information Systems*, Vol.15, No. 1, pp. 87-102, 2009. Article (CrossRef Link).
[9] M. Girvan and M.E.J. Newman, "Community structure in social and biological networks," in *Proc. of the National Academy of Science*, Vol. 99, No.12, pp. 7821-7826, 2002. Article (CrossRef Link).
[10] Larry Page, *et al*., "The PageRank Citation Ranking : Bringing Order to the Web," *Stanford Digital Library Technologies Project*, 1998. Article (CrossRef Link).
[11] A. Esuli, *et al.*, "PageRanking WordNet synsets: An application to opinion mining," in *Proc. of Association for Computational Linguistics*, pp. 424–431, 2007. Article (CrossRef Link).
[12] Erhan J. Kartaltepe, *et al*., "Social Network-Based Botnet Command-and-Control: Emerging Threats and Countermeasures," *Applied Cryptography and Network Security*. Springer Berlin Heidelberg, p.511-528, 2010. Article (CrossRef Link).
[13] Z.Zhu, G. Cao, *et al*, "A social network based patching scheme for worm containment in cellular networks," in *Proc. of IEEE Infocom*, 2009. Article (CrossRef Link).
[14] Nam P. Nguyen, *et al*. "A Novel Method for Worm Containment on Dynamic Social Networks," in *Proc. of The 2010 Military Communications Conference*, 2010. Article (CrossRef Link).
[15] V.D. Blondel, *et al*., "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, P10008, 2008 Article (CrossRef Link).
[16] G.Salton and M.McGill, "Introduction to Modern Information Retrieval," McGraw-Hill, New York, NY, 1983. Article (CrossRef Link).
[17] M. E. J. Newman and M. Girvan "Finding and evaluating community structure in networks," *Physical review E* 69(2), 026113, 2004. Article (CrossRef Link).
[18] B.Viswanath, A. Mishlove, M. Cha and K. P. Gummadi, "On the evolution of user interaction in facebook," in *Proc. of 2nd ACM SIGCOMM Worshop on Social Networks*, Aug, 2009. Article (CrossRef Link).
[19] http://www.r-project.org/ Article (CrossRef Link).
[20] Pascal Pons and Matthieu Latapy, "Computing communities in large networks using random walks," *Computer and Information Sciences-ISCIS 2005*, Springer Berlin Heidelberg, pp.284-293, 2005. Article (CrossRef Link).
[21] Raghavan, U.N., Albert, R. and Kumara, S. "Near linear time algorithm to detect community structures in large-scale networks," *Physics review E* 76, 036106, 2007. Article (CrossRef Link).
[22] Brauer and Fred, "The Kermack–McKendrick epidemic model revisited," *Mathematical Biosciences 198.2*, pp.119-131, 2005. Article (CrossRef Link).

Jong-Hwan Kong received the Bachelor degree in Computer Software from Kyungwon University, Korea in 2012 and Master degree Computer Engineering from Gachon University, Korea in 2014. He is currently a Ph.D. candidate in the Department of Computer Engineering, Gachon University, Korea. His research interests include Network Security, Information Security, Data Mining, Internet of Things Security.

Myung-Mook Han received MS degree in computer science from New York Institute of Technology in 1987 and Ph.D. degree in information engineering from Osaka City University in 1997, respectively. From 2004 to 2005, he was a visiting professor at Georgia Tech Information Security Center(GTISC), Georgia Institute of Technology. Currently, he is a professor in the Department of Computer Engineering, Gachon University, Korea. His research interests include Information Security, Intelligent System, Data Mining, Big Data. He is a member of IEEE and IEICE.