International Journal of Hybrid Information Technology
Vol. 10, No. 6 (2017), pp.23-32
http//dx.doi.org/10.14257/ijhit.2017.10.6.03

# Generation Method of Network Attack Graph Based On Greedy Heuristic Algorithm

Yuan Feng [1], Lu Wang [2, 3], Jianwei Zhang[*1], Zengyu Cai[1] and Yong Gan[1]

[1]*School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China*
[2]*Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China*
[3]*Department of Police Technology, Railway Police College, Zhengzhou 450053, China*
*mailfengy@163.com*

***Abstract***

*State explosion has become a serious problem of attack graph generation method, which results in a large-scale attack graph. Attackers always try to infiltrate into the internal network quickly, access to the more important host directly and get higher access right. The model of network attack graph generation is established based on these premises. The model expands network state node according to the evaluation function. If the valuation function value is smaller, it is the priority to expand. The evaluation value is calculated by the path length, attack difficulty, type of target host and authority obtained after the attack. Experimental results show that the network attack graph generation method based on greedy heuristic algorithm can do well in network attack graph generation, and it has a lower time complexity and good scalability. The research for this article has a great significance to improve the usefulness of network attack graph.*

***Keywords****: network security, attack graph, heuristic, greedy algorithm*

## 1. Introduction

With the rapid development of Internet, computer and network systems against malicious attacks become more diverse and complex. Network attack techniques become more covert, efficient and intelligent. Traditional vulnerability scanning technology (such as Nessus, ISS Internet Scanner, *etc*.) cannot assess of potential threats from the interaction of vulnerabilities comprehensively, because it lacks of the associated analysis for vulnerability. Practice shows that although the isolated vulnerabilities have little effects, it can cause a huge network security risk if the associated of vulnerabilities is used by hackers. In order to analysis and assess network vulnerability more objectively, the analysis tools are required which can establish attack scenario automatically according loopholes, network services, physical links and authority in the target network. To do this, the researchers proposed the attack graph model by formal preconditions, processes and results of attacks. Model and analysis techniques attack graph [1-10] have become the research hot of network vulnerability analysis. Attack graph generation engine can find the state nodes and path through a series of searching and matching process. The state nodes are consistent with the associated setting attributes. The attack graph generation method can be divided into three kinds in according to the methods and tools used, namely model checking method [7], logical inference method based on the rules [1, 8] and searching algorithm based on graph theory [9].

Jianwei Zhang is the corresponding author.

ISSN: 1738-9968 IJHIT
Copyright © 2017 SERSC

Although the attack graph generation has achieved some progress, but the state explosion is still one of problems that must be overcome to practice attack graph [2]. In attack graph construction, the order of candidate nodes will affect the search efficiency. According to the characteristics of the Heuristic search method, it can judge the merits according to some properties and sort the queue according the judgment. This can make the most likely state of the nodes on the forefront of the queue. The heuristic search method can be found to meet the conditions of the fastest node and can reduce the number of searches to reduce improve search efficiency. Therefore, it uses a greedy heuristic algorithm to reduce the size of the attack graph and increases the attack graph generation speed in this paper.

## 2. Background Knowledge

There are always some security vulnerabilities in the network, while there may be some correlation relationship between these vulnerabilities. When vulnerability is successfully exploited, it could create favorable conditions for the use of another vulnerability.

To identify all the relationships completely, the most effective way is to find all paths to reach the target of the attack by simulating the process to attacking vulnerabilities in a network., These paths are represented in the form of graphs. This graph is the network attack graph, which also is called as attack graph. Compared with the attack tree or Petri nets, description of attack graph is stronger and has more extensive application range.

On the attack graph generation search strategies, reference [10] proposed using attack graph model to analyze network vulnerabilities. Attack graph generation method is based on the realization of its existing template attack, starting from the target state, using a depth-first search strategy to generate reverse network attack graph. Attack graph generated entirely by hand, not automated. Reference [11] generated attack graph based on graph theory on the assumptions of monotonicity. Each node in the attack graph is a property. It used forward breadth-first search method starting from an initial state property that is true and ended at all through post-attribute node exploits have been added. In order to solve the attack graph scalability issues. Reference [12] used a combination method of forward-search and reverse-search firstly. It found all attack paths using forward-search. And then it eliminated the nodes that can't reach the attack targets. In order to reduce the size of attack graph, the above-described methods did secondary process after attack graph generated.

In addition, the attackers often use a variety of greedy strategies to guide the attack when attack on the network. And in many network attack graph generation methods, the greedy strategy is used either explicitly or implicitly, the monotonic hypothesis is a typical example. These greedy strategies will be discussed separately. Reference [11] first proposed the hypothesis monotony. It means that the ability already obtained successful by an attacker will not be lost due to subsequent attacks. This hypothesis is consistent with the characteristics of the majority of the types of attacks [12], but also to reflect the attacker greedy strategy, in which an attacker before reaching the target will never give up ability it has obtained. This assumption is used by most related researchers. Due to use the assumptions, the complexity of attack graph generation is reduced to polynomial from into exponential order, which makes using attack graph in large-scale network security analysis is possible.

## 3. Network Attack Graph Generation Method Based on Greedy Heuristic Algorithm

How to control the scale of attack graph effectively is one research focus of attack graph generation. This paper attempts to apply greedy heuristic search strategy to attack graph generation. It can avoid low efficiency of the conventional method shortcomings.

 Copyright © 2017 SERSC

The new method is based on the following assumptions: the attacker has a strong attack capability, and can attack any host that satisfies the attack conditions. Although not all attacks can be completed, any potential danger should not be ignored for network administrators.

### 3.1. Attack Graph Modeling

To generate attack graph, the abstract things should be concretized with specific formal language and represented by the model. Firstly, it abstracts every elements of the network security into an abstract model. The security elements definitions are given at follow.

**Definition 1 Attack Complexity:** The complexity of the attack is used to measure degree difficulty of vulnerability. It is difficult to quantify the complexity of the attack weaknesses precisely. This paper uses the complexity measure methods as that in Reference [13].

**Definition 2 Host:** Host is a computer connected to the network and provides a service for the network. It can be expressed as a quintuple: (HID, NetID, HType, SSet, VSet). They respectively denote the host identification number, network segment identification, importance degree of host, running, the service set and the set of host vulnerabilities. Host identification number uniquely identifies a host identity. The external Network Segment identification is set as 0 and other internal segments are numbered successively in the order from outside to inside. The hosts are classified five types According to data and services in hosts. Service set is composed of the services running on the host. Set of host vulnerabilities is composed of the various services of the existence of running on the host.

**Definition 3 Service:** The service is a program running on the host program having a certain network function. It can be expressed as a five-tuple: (SID, Name, Pr, Po, RL), respectively, for service identification, service name, protocol, port, running privilege level.

**Definition 4 Vulnerability:** Vulnerability refers to the loopholes of service program, configuration and trust relationship. In this paper, it is expressed using a four-tuple as (VID, SID, VType, Range, VCom). Respectively, they are vulnerability identification, service identification, vulnerability type, local or remote, and complexity of attacking.

**Definition 5 Connection Relationship:** Host connection relationship refers to the network of any connection between the two hosts. Formal model used in this paper is expressed as (SHID, DHID, PPSet), respectively, the source host identification, target host identification and set of protocol ports allowing access.

**Definition 6 Attack Rules:** Attack rules refer to the rules that should be followed when attacking host. It can be formalized as a four-tuple (RID, VID, EPre, EPost), respectively, for the rule identification, vulnerability identification, preconditions, attacking results.

**Definition 7 Attack Action:** Attack action is an action that attacker can use it to Intrude one host from another host successfully. It is formalized as (SHID, DHID, RID, EPre, EPost). Respectively, they are source host identification, destination host identification and attack identification rules.

**Definition 8 Node State:** In the attack graph, Node state is represented as (SID, HID, VID, PL). SID is the identification of node state. HID is the identification of target host. VID is Vulnerability identification to reach this node state. PL is the rights level getting by attacker in this node state end level..

**Definition 9 Directed Edge in Attack Graph:** In the attack graph, the directed edges are used to represent an attack action which makes the network security status changed. It can be expressed as (SSID, DSID, AID). Respectively, they are the source node identification, the destination node identification and the attack identification.

**Definition 10 Attack Graph:** Attack graph is a security state transition graph of a target network. It is represented as (S, E, $S_0$, $S_G$). S is the set of security state nodes of the network. E is the set of attack actions and E $\subseteq$S $\times$ S to change security state. It also is a

Copyright © 2017 SERSC 

set of directed edges. $S_0 \in S$ is the initial security state node. $S_G \in S$ is the set of the target states.

### 3.2. The Basic Idea and Valuation Function

In the attack graph generation process, it assumes that the attacker always selects the best hope to achieve optimal target node as the next extension nodes. This is the source of greedy heuristic algorithm in this paper. Meanwhile, in order to measure the "hope" of state node, it is necessary to design valuation function. If the hope degree of a node is greater, the valuation function value of it must be smaller. The design of valuation function will be discussed in detail at below.

In this paper, it uses heuristic greedy algorithm to reduce the search space to prevent the number of node states be too large. In design the heuristic function, it considers the following three factors. First, it considers the attack sequence length. The attack sequence length will lead attack graph scale exponentially increased. Second, it considers the cost of attack graph generation process nodes, namely the complexity of the attack. Third, it considers the value of nodes in attack graph based on the assumption that attacker will give attack the nodes that is relatively high or very important node.

In this article, node $n_i$ is extended from node $n_{i-1}$, then lets the evaluation value of $n_i$ as

$$f(n_i) = f'(n_{i-1}) + 1 + \text{cost}(n_i) \quad (1)$$

The $f'(n_{i-1})$ is the actual cost of the attack path $n_0, n_1, \ldots, n_{i-1}$. The *1* represents the increasing for search depth. The cost $(n_i)$ is the heuristic values of node. It is codetermined by attack complexity of $Q(n_i)$ and the attacks priority $P(n_i)$. The cost $(n_i)$ is calculated as follows:

$$\text{cost}(n_i) = Q(n_i) - P(n_i) \quad (2)$$

Among them, the attack complexity $Q(n_i)$ is same to that in Reference [13]. The attack category priority $P(n_i)$ is calculated by the following greedy strategy:

(1) Attacker can gain the access right of an new host. If the important degree increase one, the value of $P(n_i)$ increased *0.5*;

(2) Attacker can get more access right of the same host. The $P(n_i)$ is added *0.2*. If the access right of target host is reduced to User and the right of source host is Root, the $P(n_i)$ is subtracted *0.2*. To simple, there are only two kinds of rights, User and Root.

### 3.3. Generation Method of Network Attack Graph Based on Greedy Heuristic Algorithm

The process of Network attack graph generation based on heuristic greedy algorithm is as follows. First, it constructs the Open Table and Closed Table. The Open Table stores the nodes that are not expanded. The Closed Table stores node has been extended. The initial state of the network is in the Open Table. Second, it selects node which has the smallest evaluation value in the Open Table valued to extend. It extends a new node whose conditions are satisfied for all vulnerability. The new nodes are put into Open Table. The expanded node is removed to Closed Table from Open Table. And then it selects node which has the smallest evaluation value in the Open Table until the Open Table is empty or the current node is the target status node. The algorithm is shown in Figure 1.

 Copyright © 2017 SERSC

**Algorithm:** Network Attack Graph Generation Based on Greedy Heuristic Algorithm
**Inputs:** $C_0$, $S_0$, $S_G$, $R$, *NetTable*.
**Outputs:** $G$ ($S$, $E$, $S_0$, $S_G$).

Step0 Establish an empty Open Table as $O$ and CLOSED table as $C$. It sets the attack action set $A = \emptyset$, $S = \emptyset$ and $E = \emptyset$.
Step1 Initialize set $A$ according *NetTable* and the attack rule set $R$, It adds $S_0$ to $O$ and calculates the cost function f ($S_0$) = cost ($S_0$).
Step2 If the $O$ is empty, it returns the attack graph $G$ and exit the algorithm, and else it repeats the following operations:
Step2.1 Select the node that has smallest evaluation value in the $O$ as the current node $N_{now}$ and remove it from $O$;
Step2.2 If $N_{now}$ is in $S_G$, then goto Step2. Otherwise, repeat doing the following:
Step2.2.1 Add $N_{now}$ to the $C$;
Step2.2.2 For each Vulnerability of node $N_{now}$, if it fit the prerequisites of successful attacks, then perform the following steps:
Step2.2.2.1) According to result of this attack, generate the new state node $N_{new}$.
Step2.2.2.2) Calculate the evaluation value ion of $N_{new}$. as f ($N_{new}$) = f ($N_{now}$) + *1* + cost ($N_{new}$);
Step2.2.2.3) If $N_{new}$ is not in the $O$ or $C$, put $N_{new}$ into the $O$, add $N_{now}$ in $S$ and add ($N_{now}$, $N_{new}$) to $E$;
Step2.2.2.4) If $N_{new}$ is in $O$ or $C$ already, compare f ($N_{new}$) with the previous value. If f($N_{new}$) is smaller than previous value, updated evaluation valuation of $N_{NEW}$, delete edge with its parent node from $E$ and add ($N_{now}$, $N_{new}$) edge to $E$. If $N_{new}$ is in the $C$, then move it to $O$ again.

**Figure 1. The Algorithm of Network Attack Graph Generation Based On Greedy Heuristic**

## 4. Experimental Results and Analysis

### 4.1. Experimental Environment

We did the experiment to evaluate the effectiveness of our algorithms. It used the data and functions as Bugtraq. The network topology used in the experiment is shown in Figure 2. There are five hosts in internal network represented as from *IP1* to *IP5* and they connect to external network through a router and a firewall. The attacker is at outside external network. *IP1* and *IP2* are the servers and run operating system Linux. *IP3*, *IP4* and *IP5* are Windows hosts.

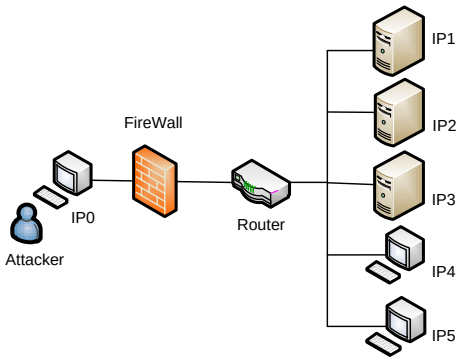


**Figure 2. Topology Graph of Experiment Network**

In this experiment, *IP1* provides FTP service and SMTP service. *IP2* provides HTTP service. *IP3* provides FTP service, *IP4* open Telnet and SSH services. *IP5* is the critical host which stores important information. Attacker used in *IP0* host external network and own Root privileges. Firewall policy allows external hosts to access *IP2* HTTP service and other access are blocked. The vulnerabilities and complexity for each are shown in Table 1.

**Table 1. Information of Hosts and Vulnerabilities**

| Hosts | Importance degree | Number of vulnerability | Range of hosts | Services on hosts | Authorities | Complexities |
|---|---|---|---|---|---|---|
| IP0 | 0 | - | - | - | - | - |
| IP1 | 2 | 15343 | Remote | FTP | Root | 0.1 |
| | | 8641 | Remote | SMTP | User | 0.3 |
| 1P2 | 1 | 11964 | Remote | HTTP | Root | 0.7 |
| 1P3 | 3 | 11826 | Remote | FTP | User | 0.5 |
| | | 8826 | Local | FTP | User | 0.3 |
| IP4 | 3 | 12815 | Remote | Telnet | User | 0.5 |
| | | 6274 | Remote | SSH | User | 0.7 |
| IP5 | 4 | 10707 | Local | - | Root | 0.5 |

### 4.2. Example Analysis

The topology of experimental network is shown in Figure 2 and the vulnerabilities and complexity for each are shown in Table 1. It generated a complete attack graph without heuristic algorithm. The complete attack graph has *106* nodes and *105* edges and there are *46* paths reaching target. Heuristic attack graph generation process is shown in Figure 3. The traditional heuristic approach in [14] only considered the complexity of attacks. So the priority extending selections are *node 2*, *node 3*, and *node 5* in step two. When using greedy heuristic algorithm proposed this paper, it considers the importance degree of host and the authorities change. So the priority extending selections are *node 8*, *node 2* and *node 5*. Compared with reference [14], our algorithm can attack the most important host *IP5* as early as possible and spending least costs. So, greedy heuristic algorithm is more reasonable and faster than reference [14].

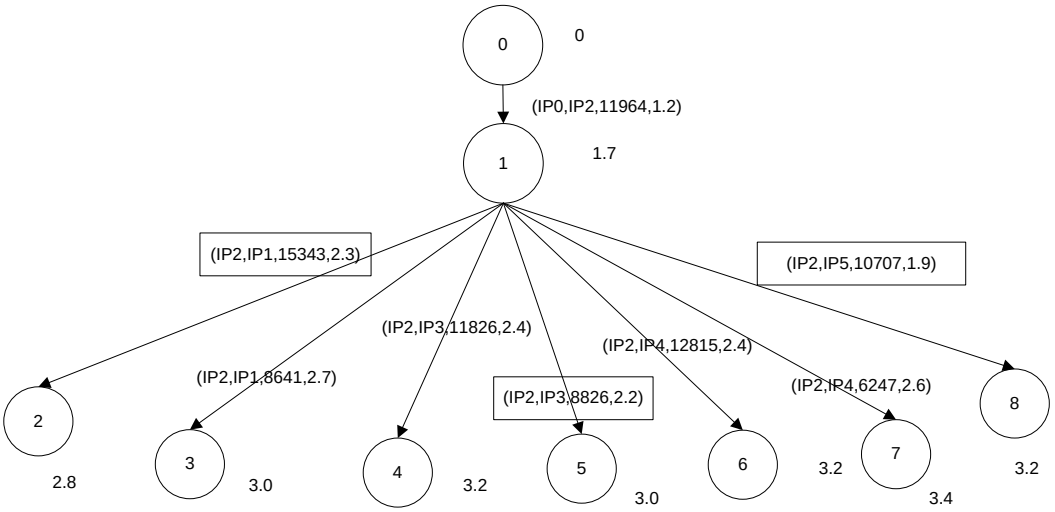


**Figure 3. An Example of an Attack Graph Generation Based On Greedy Heuristic Algorithm**

 Copyright © 2017 SERSC

### 4.3. Experiments on CPU Consuming

At present, many attack graph construction methods have been proposed. The Sheyner [6] and Ou [1] are two typical methods. The simulation network has only one subnet and there are not firewalls. Any host has no more than five vulnerabilities. For different network size, it generates attack graph respectively using the methods of Sheyner, Ou and ours. Figure 4 shows the cooperation of three methods on CPU performance.

It is obviously Sheyner method has the worst performance because the CPU time consuming is increased exponentially as the size of the network. The CPU times consuming increase polynomially in Ou and our methods. Meanwhile, the performance of the method proposed in this paper is better than Ou, mainly due to the using of heuristics and optimization strategies to limit the graph growing effectively.

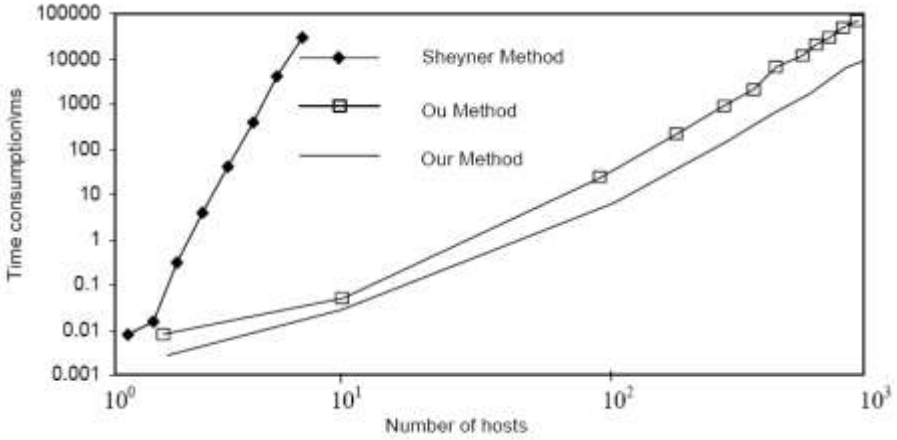


**Figure 4. Compared Ou and Methods of Sheyner, Ou and Our On Cpu Performance**

### 4.4. Experiments on Scalability

In this paper, it tests the scalability of our methods by increasing the number of hosts within the network. The simulation network has only one subnet and there are not firewalls. Any host has no more than five vulnerabilities and can access the web server. The results are shown in Figure 5.

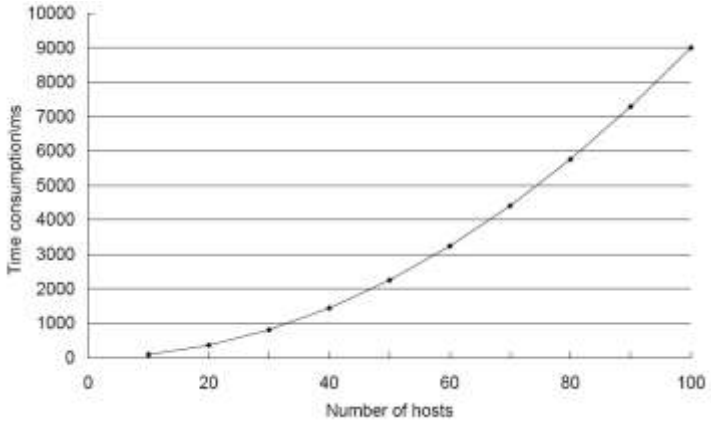


**Figure 5. Curve between Network Size and CPU Time Consumption**

Figure 5 shows curve between network size and CPU time consumption. As can be seen from the figure, CPU time consumption is increasing with the expansion of the

Copyright © 2017 SERSC

network size. But the rate of increase is in the acceptable range. When the number of hosts reached 100, the generation time is no more than 10 seconds. It can be seen that the time complexity of our algorithm is low from the experimental results. So it can be used in large networks.

## 5. Conclusion

Network security research has great significance in the information society. But it is also a challenge task. The key of constructing attack graph automatically is the generation algorithm and it directly determines efficiency and accuracy attack graph generation. An attack graph generation method based on greedy heuristic algorithm is proposed. It not only reduces the amount of calculation required to generate attack graph, but also greatly reduces the complexity of attack graph and retains important information for each attack path. The experimental results show that the new method based on greedy heuristic algorithm has a good accuracy in generating attack graph. It can predict the path tacked by the attacker effectively. It will help the network administrators to making manage network decision and providing guidance.

## Acknowledgements

This work is supported by National Natural Science Foundation of China under Grant (No. 61572445), Outstanding Youth Science and Technology Innovation Project of Henan Province: research of the macro scenario fitting routing technology centering on content.

## References

[1] X. Ou, W.F. Boyer and A. McQueenM, "A scalable approach to attack graph generation", Proceedings of the 13th ACM Conference on Computer and Communications security, Alexandria, Virginia, USA, **(2006)**, pp. 336-345.
[2] S. Wang, Z. Zhang and Y. Kadobayashi, "Exploring attack graph for cost-benefit security hardening: A probabilistic approach", Computers & Security, vol. 32, no. 1, **(2013)**, pp. 158-169.
[3] N. Ghosh and S. K. Ghosh, "A planner-based approach to generate and analyze minimal attack graph", Applied Intelligence, vol. 36, no. 2, **(2012)**, pp. 369-390.
[4] L. Xie, X. Zhang and J. Zhang, "Network Security Risk Assessment Based on Attack Graph", Journal of Computers, vol. 8, no. 9, **(2013)**, pp. 2339-2347.
[5] I. Kotenko and E. Doynikova, "Evaluation of Computer Network Security based on Attack Graphs and Security Event Processing", Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, vol. 5, no. 3, **(2014)**, pp. 14-29.
[6] Q. Y. Wei, "Attack Graph Analysis for Network Anti-Forensics", International Journal of Digital Crime & Forensics, vol. 6, no. 1, **(2016)**, pp. 28-50.
[7] H. Rattikorn and K. Phongphun, "Host-centric model checking for network vulnerability analysis", ACSAC. Anaheim, CA, **(2008)**, pp. 225-234.
[8] H. Mao and W. Zhang, "An approach for network security analysis using logic exploitation graph", Proceedings of the 7th IEEE International Conference on Computer and Information Technology, Aizu-Wakamatsu City, Fukushima, Japan, **(2007)**, pp. 761 -766.
[9] S. Q. Zhong, D. F. Yan and C. Liu, "Automatic generation of host-based network attack graph", Proceedings of the 2009 World Congress on Computer Science and Information Engineering, Los Angeles, California USA, **(2009)**, pp. 93 -98.
[10] L. P. Swiler and C. Phillips. "A Graph-based System for Network Vulnerability Analysis Report", Proceedings of the Workshop on New Security Paradigms, Charlottesville, VA, September 22-26, pp. 71-79.
[11] P. Ammann, D. Wijesekera and S. Kaushik, "Scalable, graph-based network vulnerability analysis", Acm Conference on Computer & Communications Security, Washington, DC, USA, **(2002)**, pp.217-224.
[12] F. Zhao, X. Chen and J. Li, "Generation Method of Network Attack Graphs Based on Privilege Escalation", Computer Engineering, vol. 34, no. 23, **(2008)**, pp. 185-160.
[13] Y. Zhang, X. Yun and M. Hu, "Research on privilege-escalating based vulnerability taxonomy with multidimensional quantitative attribute", JOURNAL OF CHINA INSTITUTE OF COMMUNICATIONS, vol. 25, no. 7, **(2004)**, pp. 108-115.

 Copyright © 2017 SERSC

[14] F. Liu, M. Lin and M. Tan, "Generation Algorithm of Network Attack Graph Based on Sequential Search", Journal of University of South China (Science and Technology), vol. 28, no. 3, **(2014)**, pp. 82-86.
[15] W. U. Huan, L. Pan and X. Z. Wang, "A Risk Assessment Model Using Incomplete Attack Graphs Analysis", Journal of Beijing University of Posts & Telecommunications, vol. 33, no. 3, **(2010)**, pp. 57-61.

## Authors

**Yuan Feng**, she received her master degree in communication and information system from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2006. She is a lecturer at Zhengzhou University of Light Industry. Her research interests include mobile communication, network engineering and information security.

**Lu Wang**, he received his master degree in computer software and theory from Zhengzhou University of Light Industry, Zhengzhou, China, in 2010. He is a lecturer at Railway Police College. His research interests include network information security and computer forensics.

**Jianwei Zhang**, he received his Ph.D. degree in computer application technology from from PLA Information Engineering University in 2010. He is a professor at Zhengzhou University of Light Industry. His research interests include broadband informationnetwork and network security.

**Zengyu Cai,** he received his master degree in computer application technology from Northeast Normal University, Changchun, China, in 2006. He is an associate professor at Zhengzhou University of Light Industry. His research interests include trusted computing, plan recognition and information security.

**Yong Gan,** he received his Ph.D. degree in computer application technology from Xi`an Jiaotong University, Xi'an, China, in 2013. He is a professor at Zhengzhou University of Light Industry. His research interests include information security, cryptography, multimedia communications and network engineering.

Copyright © 2017 SERSC

 Copyright © 2017 SERSC