A Quantitative Method for Evaluating Network Security Based on Attack Graph

Yukun Zheng, Kun Lv(✉), and Changzhen Hu

School of Software, Beijing Institute of Technology, Beijing, China
{2120151096, kunlv, chzhoo}@bit.edu.cn

Abstract. With the rapid development of network, network security issues become increasingly important. It is a tough challenge to evaluate the network security due to the increasing vulnerabilities. In this paper, we propose a quantitative method for evaluating network security based on attack graph. We quantify the importance of nodes and the maximum reachable probability of nodes, and construct a security evaluation function to calculate the security risk score. Our approach focuses on the attacker's view and considers the most important factors that may affect the network security. The parameters we use are easily to be acquired in any network. Thus, the assessment score gotten through the evaluation function can comprehensively reflect the security level. According to the security risk value, security professionals can take appropriate countermeasures to harden the network. Experimental results prove that this model solves the security evaluation problem efficiently.

Keywords: Attack graph · Network security · Risk judgement · Vulnerability

1 Introduction

Network plays an increasingly important role in people's daily life with the rapid development of network technology. At the same time, the type of network attacks and the number of vulnerabilities are increasing rapidly. Many network and information systems are facing security threats. As more and more privacy are faced with the risk of being leaked, the responsibility of network security is more and more important accordingly. It is needed to evaluate the network security to measure the situation of the network. Evaluation of network security can help security professionals to optimize security configurations. Thus, an evaluation system is needed to solve all of the above problems.

In this paper, we construct a network security evaluation model for network based on attack graph. Firstly, the model is based on attack graph [1–3]. Attack graph can generate attack paths to analyze the network vulnerability. It shows users the weak point in the network analysis process for network security risk analysis. Secondly, in this model, the value of security risk is calculated by a function which is based on two parameters that are importance of nodes and the maximum reachable probability of nodes. The nodes in the attack graph have different effects on the security of the network. The higher the importance of node, the greater the impact on the network security of the node. Thus, we quantify the importance of nodes according to the major factors. The maximum reachable probability of nodes is an important factor of network

Z. Yan et al. (Eds.): NSS 2017, LNCS 10394, pp. 349–358, 2017.
DOI: 10.1007/978-3-319-64701-2_25

security as well. Then, we construct a security evaluation function which is based on the above two parameters and calculate the security risk value of the network. Finally, security professionals could distinguish the level of security according to the risk value and formulate their countermeasures. The parameters required by our approach are convenient to collect. The computational complexity of our model is relatively low. Thus, our method can be generalized to any network for measuring network security.

The organization of the paper is as follows. We discuss related works in Sect. 2. Then, we describe our model of evaluating the network security in Sect. 3. At Sect. 4, our model is tested based on a simple attack graph. We then present the conclusions in Sect. 5 and acknowledgement respectively.

2 Related Works

2.1 Attack Graph Generation

Various kinds of approaches have been proposed to generate attack graph automatically. Early approaches use network states, which result in the graphs growing exponentially.

Sheyner et al. uses model checking techniques to compute attack graphs [2]. Phillips and Swiler [3] developed a tool for generating attack graphs. Ritchey and Ammann [4] use model checking for vulnerability analysis of networks. X. Ou et al. [5] tried to generate logical attack graph and developed a tool named MulVAL. Now, it is an open source project in Kansas State University. Sheyner et al. [6] use a modified model checker, NuSMV, to produce attack graphs. Although their model could generate all the attack paths, the scalability problem is more serious. Then, P. Ammann et al. [7] introduce the monotonicity assumption into generation process, and reduce the computational cost to polynomial. Jajodia et al. [8] develop a tool named TVA (the Topological Vulnerability Analysis tool). It can analyze network vulnerability automatically and mine the weakness to generate the attack graph.

2.2 Network Security Analysis

Some researchers are also trying to evaluate network security quantitatively based on attack graphs [9–12]. Many mathematical algorithms have been applied in the field of network security evaluation.

J. Pamula et al. [9] describe a method to measure network security. Their method expresses the targets as the minimal sets of required initial attributes, and the security metric is the strength of the weakest adversary who can successfully penetrate the network. Wang et al. [10] make a further analysis on network metric with attack graphs, and they propose a simple security metric framework, which mainly describes the basic principles and the basic requirements of operators. Then, Wang et al. [11] give a metric example with probability of success, discussing the processing methods on cycles in attack graphs. But unfortunately, their method is suitable for single target, and is hard to describe a network's security as a whole. M. Frigault et al. [12] interpret attack graphs as special Dynamic Bayesian networks, and their outstanding contribution is considering the effect between the vulnerabilities in a dynamic environment.

The existing methods of network security analysis provide ideas for our work in combining attack graph and mathematical methods. Attack graphs provide the necessary context for correlating and prioritizing intrusion alerts, based on known paths of vulnerability through the network. The method of mathematics can make the evaluation of security risk more accurate.

3 Model Description

3.1 Description of Attack Graph

Our work is not focus on how to construct an attack graph automatically since there are so many articles devoted to this issue. We just analyze network security situation on the basis of the assumption that we have already gotten an attack graph. Here we generate our attack graph by MulVAL [4]. In this paper, the attack graph that we discuss refers to the acyclic attack graph. The definition of attack graph [1–3] is as follows.

Definition 1. The structure of attack graph is a directed graph. It can be defined as $G = (V_o \cup V_d, T, E)$, where the set $V_o \cup V_d$ of nodes represent vulnerable system and network configurations, the set T of nodes represent target nodes, the element $v_{ij} \in E$ in set E is a transition relation from v_i to v_j.

Furthermore, attack graph should meet the following conditions: an exploit cannot be realized until all of its previous conditions have been satisfied. A reachable condition can be satisfied if any of its previous exploits are realized.

3.2 The Node's Importance

The nodes in the attack graph have different effects on the security of the network. Standing on the shoulders of the meaningful results brought by previous works, we access the factors that may impact the node's importance from the view of attacker and propose a metric named *TNI* to measure the importance of the node's importance level.

Mehta et al. proposed using Google PageRank algorithm to assess importance of nodes in the attack graph [13], which considers mainly the topology and link relations. The PageRank algorithm is a link analysis algorithm and it assigns a numerical weighting to each element of a hyperlinked set of documents. Since the more steps the attack sequence is, the harder to attack success. Attackers prefer to choose the shortest attack path in an attack. Considering the shortest paths, we use betweenness centrality [16] to evaluate the node's importance level. Betweenness centrality is a measure of centrality in a graph based on shortest paths.

The mathematical descriptions about *TNI* are as follows. Firstly, we need to calculate the node's PageRank value and betweenness centrality respectively.

Then, we normalize the above two value in (0, 1) and get the average value of the above two parameters. The *TNI* of node v_i is denoted as $TNI(v_i)$:

$$TNI(v_i) = \frac{PR(v_i) + BC(v_i)}{2} \tag{1}$$

In this paragraph, we will discuss how to calculate PageRank value in detail. We use iteration algorithm to calculate PageRank value. The PageRank value of node v_i at time t is denoted as $PR(v_i, t)$:

$$PR(v_i, t+1) = \frac{1-d}{N} + d\sum \frac{PR(v_j, t)}{L(v_i)} \tag{2}$$

In (2), d is a constant; $M(v_i)$ is the set of node that links to node v_i; $L(v_i)$ is the number that node links to other nodes. Firstly $t = 0$, initialize the PageRank value of each node as $PR(v_i, 0) = \frac{1}{N}$. Then, iterative as (2) until (3) is satisfied. The values of the last iteration are the node's PageRank value.

$$|PR(v_i, t+1) - PR(v_i, t)| \leq \varepsilon \tag{3}$$

In (3), ε is a constant that can be adjusted in different situations. The smaller ε is, the harder formula convergences.

For every pair of nodes in a graph, there exists a shortest path between the nodes such that the number of edges that the path passes through is minimized. The betweenness centrality for each node is the number of shortest paths that pass through the node. The betweenness centrality of node is denoted as $BC(v_i)$:

$$BC(v_i) = \sum_{s \neq i \neq t} \frac{\sigma_{st}(v_i)}{\sigma_{st}} \tag{4}$$

In (4), $\sigma_{st}(v_i)$ is the number of shortest paths that pass through node v_i. $\sigma_{st}(v_i)$ is the total number of shortest paths from node to s node t. We use the Dijkstra algorithm to deal with the single-source shortest path and calculate the number $\sigma_{st}(v_i)$ and σ_{st}.

3.3 The Maximum Reachable Probability of Nodes

In the attack graph, there may be multiple paths from the initial node to the target node. For different attack sequences, attack difficulty is also different. Attacker prefer to choose the easiest path to attack, so the security of the network depends on the safety of its weakest part. We define the probability when attackers choose the easiest path to attack a node as the maximum reachable probability.

If node $v \in V_d$, the maximum reachable probability of node v can be calculated by its parent node. If node c is the one of the parent node of node v and $P(c)$ is the maximum reachable probability of node v, then we can get the reachable probability of node from node c is $P(v) = d(v) * P(c)$. Here $d(v)$ is the access probability of node v.

We get the data of access probability from the CVSS based database. For node v, the maximum reachable probability is $P(v) = d(v) * Max\{P(c)|c \in Pre(v)\}$.

If node $v \in T$, all conditions of it's parent node must be qualified according to the definition of attack graph. We can deduct the formula contrasting to the initial nodes. The maximum reachable probability is that $P(v) = d(v) * \prod_{c \in Pre(v)} P(c)$.

For initial nodes and target nodes, we define their maximum reachable probability as follows. The initial nodes represent the initial conditions that an attacker can exploit. The target nodes are the target of attackers attacking the network. It is necessary to calculate the probability of initial nodes for calculating the maximum reachable probability of target nodes.

Definition 2. In an attack graph $G = (V_o \cup V_d, T, E)$, $d(v_i)$ is the access probability of its own, $Pre(v_i)$ are the parent nodes of node v_i. The maximum reachable probability of node v_i is defined as $P(v_i)$: if node $v_i \in V_d$, the maximum reachable probability of node v_i is $P(v_i) = d(v_i) * Max\{P(c)|c \in Pre(v_i)\}$; if node, the maximum reachable probability of node v_i is $P(v_i) = d(v_i) * \prod_{c \in Pre(v_i)} P(c)$.

We can calculate the maximum reachable probability of all nodes according to the Definition 2. To describe the attacker's multistep attack process, we use the breadth first search algorithm. In order to simulate the attacker's choice of multiple paths and the limit of longest attack path, we define the longest attack length L and search the best path step by step in our algorithm. The length L represent an attacker's ability to attack the network and should be revised according to the actual situation. Meanwhile, we define a data structure to record the trace of the attack sequence to avoid the loop path. When a loop path appears, we should find the equivalent path of the loop path instead of simply cancel the loop path. When there are multiple paths, our algorithm will choose the one with the highest success rate according to Definition 2.

The probability of the node own can be queried from the standard CVE (Common Vulnerabilities and Exposures) name [14]. We adapt different methods according to the Definition 2 to get the node's maximum reachable probability after its parent nodes are calculated. Finally we get the probability of each nodes as all the procedures finished.

3.4 Construct Evaluation Model

To sum up the opinions above, two parameters decide the network security situation: the node's importance and the maximum reachable probability of nodes in the attack graph. Let $V(v_i)$ be a function to calculate the security risk score of node v_i, and then we can see that:

When $TNI(v_i)$ do not change, the higher $P(v_i)$ is, the higher $V(v_i)$ becomes. $V(v_i)$ is proportional to $P(v_i)$.

When $P(v_i)$ do not change, the $TNI(v_i)$ higher is, the higher $V(v_i)$ becomes. $V(v_i)$ is proportional to $TNI(v_i)$.

When $P(v_i)$ and $TNI(v_i)$ do not change, the higher λ is, the more obvious the change of $V(v_i)$ is.

Based on the above judgment and the monotone character of the function, we could construct a security risk function:

$$V(v_i) = \lambda * TNI(v_i) * P(v_i) \tag{5}$$

We can also assess the security situation of the whole network. Let V represents a function that calculates security risk score of the whole network. We just add up the risk score to get the risk score of the whole network. Then, we normalize the score in (0, 1) for comparison and evaluation. According to expressions (2), (4) and (5), we conclude that the mathematical model describing network security risk is as follows:

$$V = log_N\left(\sum_{i\in[1,N]} e^{\lambda * TNI(v_i) * P(v_i) - 1}\right) \tag{6}$$

The security risk score distinguishes the levels of network security. Using our method, managers could check the network regularly and collect the security risk information. According to the security risk value, administrators are aware of the situation of the whole network, and then take up the corresponding measures, to ensure the safety of the network relatively.

The safe intervals of different network are also different. What is more, calculating different network security scores need different scoring criteria. So, our approach is not suitable for comparing security between different networks directly. But, our method is very suitable for security monitoring and evaluation for the same network. Using our method to deal with the security of the same network, changes of the security risk value represent changes in the level of network security. Managers can judge whether the network is safe after many evaluations.

4 Experiments

In this section, we study a relatively realistic network to validate the rationality and feasibility of the algorithm we propose. The experimental environment are Intel Pentium E5400 (2.70 GHz), 2 GB of memory, Window XP. The algorithm is implemented in Eclipse 3.2.

In the experimental network shown in Fig. 1, there are two hosts in this mini-network, which are a web server and an Apache server. The link-layer connectivity between the two hosts is provided by a switch. In addition, an attacker's client, is connected to the switch directly. Two firewalls connecting to the switch are used to protect the network respectively. The vulnerabilities which would be exploited by attackers are shown in Table 1. For each vulnerability, the CVE name, CVSS score, vulnerability location, and exploiting probability of success are also described as follows. The CVSS score represent the risk level of single vulnerability. The access complexity represent the success rate of attackers to penetration the corresponding vulnerability.

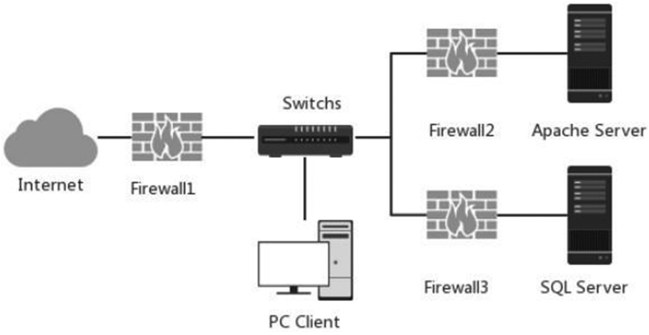

Fig. 1. Experimental network topology

Table 1. Vulnerability information

Cve name	CVSS score	Vulnerability location	Access complexity
CVE-2006-3747	7.5	Apache Server	High
CVE-2008-4250	10	Apache Server	Low
CVE-2012-0021	2.6	Apache Server	High
CVE-2012-0578	4.0	SQL Server	Low
CVE-2011-4671	7.5	SQL Server	Low

The information in Table 1 are obtained from the National Vulnerability Database published by National Institute of Science and Technology (NIST). The vulnerabilities are stored using the standard Common Vulnerabilities and Exposures (CVE) name [15]. For each vulnerability in the database, NVD provides CVSS [16] scores in the range 1 to 10. We use specific numerical values to characterize the access complexity of vulnerabilities. In detail, we use 0.37 to represent the High level of access complexity, 0.61 to represent the Medium level and 0.71 to represent the Low level and undefined level.

Figure 2 shows the attack graph we generated using MulVal. Since the semantics of each node in the graph is too long, we replace them with different letter number. In this attack graph, there are 12 intermediate node and 6 target nodes. We use the algorithm we proposed in this paper to calculate the two parameters and then get the security risk score. The result of TNI and the maximum reachable probability are shown in Table 2.

The security value of this network is 0.62, which is relatively high. According to the maximum reachable probability of different nodes, administrator know which nodes are easy to be leaked, and then take up corresponding measures, to ensure the safety of the network. When some vulnerability are fixed, the security value will decline sharply. Our methods can be used to check the network regularly and collect the security risk information. By comparing the historical security risk information and the risk value, network administrators are able to understand the security situation of the network.

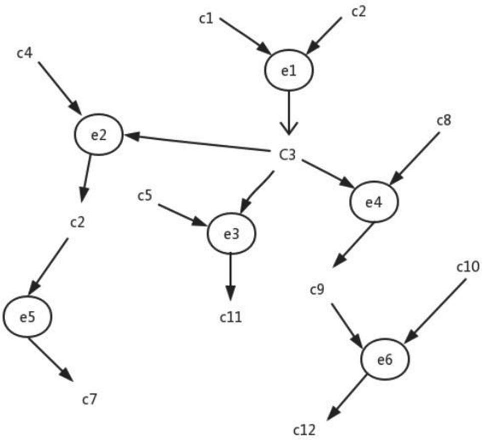

Fig. 2. Attack graph of the network

Table 2. Node information

Node	TNI	TMRP
c1	0.1	1
c2	0.1	1
c3	0.83	1
c4	0.1	1
c5	0.1	0.35
c6	0.54	0.35
c7	0.43	0.35
c8	0.1	1
c9	0.52	0.71
c10	0.1	1
c11	0.34	0.35
c12	0.5	0.49
e1	0.64	0.71
e2	0.52	0.35
e3	0.41	0.35
e4	0.44	0.71
e5	0.53	0.35
e6	0.54	0.49

5 Conclusion

We propose a quantitative method for evaluating network security based on attack graph. We analyze the host information, topology information and vulnerability information of the network, get all possible attack paths and generate the attack graph. In this paper, we construct a network security evaluation model for network based on attack graph. Then, we define and calculate the importance of nodes and the maximum reachable probability of nodes in an attack graph. Finally, we construct the model based on the above two parameters. The approach provides a method to analyze attack paths, compute the security risk value of the network and help security professionals to choose appropriate countermeasures based on conditional decision preferences of relevant factors. Our future work is to optimize the algorithms of attack probability assessment and to test our method on large scale networks.

Acknowledgments. This paper is partially supported by the Basic Scientific Research Program of Chinese Ministry of Industry and Information Technology (Grant No. JCKY2016602B001) and National Key R&D Program of China (Grant No. 2016YFB0800700)

References

1. Phillips, C.A., Swiler, L.P.: A graph-based system for network vulnerability analysis. In: Workshop on New Security Paradigms, pp. 71–79 (1998)
2. Sheyner, O., Haines, J., Jha, S., Lippmann, R., Wing, J.M.: Automated generation and analysis of attack graphs. In: Proceedings of the 2002 IEEE Symposium on Security and Privacy, pp. 254–265 (2002)
3. Swiler, L., Phillips, C., Ellis, D., Chakerian, S.: Computer attack graph generation tool. In: Proceedings of DARPA Information Survivability Conference and Exposition II (2001)
4. Ritchey, R.W., Ammann, P.: Using model checking to analyze network vulnerabilities. In: IEEE Symposium on Security and Privacy, pp. 156–165 (2000)
5. Ou, X., McQueen, A.: A scalable approach to attack graph generation. In: Proceedings of the 13th ACM Conference on Computer and Communications Security (2006)
6. Sheyner, O.M.: Scenario graphs and attack graphs. Ph.D. dissertation, Pittsburgh, PA, USA, chair-Jeannette Wing (2004)
7. Ammann, P., Wijesekera, D., Kaushik, S.: Scalable, graph-based network vulnerability analysis. In: CCS 2002: Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 217–224. ACM, New York (2002)
8. Jajodia, S., Noel, S., O'Berry, B.: Topological analysis of network attack vulnerability. In: Kumar, V., Srivastava, J., Lazarevic, A. (Eds.) Managing Cyber Threats: Issues, Approaches and Challenges. Kluwer Academic Publisher (2003)
9. Noel, S., Jajodia, S., O'Berry, B., Jacobs, M.: Efficient minimum-cost network hardening via exploit dependency graphs. In: ACSAC, pp. 86–95. IEEE Computer Society (2003)
10. Wang, L., Islam, T., Long, T., Singhal, A., Jajodia, S.: An attack graph-based probabilistic security metric. In: Atluri, V. (ed.) DBSec 2008. LNCS, vol. 5094, pp. 283–296. Springer, Heidelberg (2008). doi:10.1007/978-3-540-70567-3_22
11. Pamula, J., Jajodia, S., Ammann, P., Swarup, V.: A weakest-adversary security metric for network configuration security analysis. In: Karjoth, G., Massacci, F. (Eds.) QoP, pp. 31–38. ACM (2006)
12. Frigault, M., Wang, L., Singhal, A., Jajodia, S.: Measuring network security using dynamic bayesian network. In: Ozment, A., Stølen, K. (Eds.) QoP, pp. 23–30. ACM (2008)
13. Mehta, V., Bartzis, C., Zhu, H., Clarke, E., Wing, J.: Ranking attack graphs. In: Zamboni, D., Kruegel, C. (eds.) RAID 2006. LNCS, vol. 4219, pp. 127–144. Springer, Heidelberg (2006). doi:10.1007/11856214_7

14. NVD Homepage, CVSS. http://nvd.nist.gov/cvss.cfm. Accessed 09 Jun 2017
15. Scarfone, K., Mell, P.: An analysis of CVSS version 2 vulnerability scoring. In: Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement, pp. 516–525 (2009)
16. Mantrach, A.: The sum-over-paths covariance kernel: a novel covariance measure between nodes of a directed graph. IEEE Trans. Pattern Anal. Mach. Intell. **32**, 1112–1126 (2010)