Attack Graph Based on Vulnerability Correlation

Subjects: Computer Science, Information Systems

Contributor: Gun-Yoon Shin, Sung-Sam Hong, Jung-Sik Lee, In-Sung Han, Hwa-Kyung Kim, Haeng-Rok Oh

As network technology has advanced, and as larger and larger quantities of data are being collected, networks are becoming increasingly complex. Various vulnerabilities are being identified in such networks, and related attacks are continuously occurring. To solve these problems and improve the overall quality of network security, a network risk scoring technique using attack graphs and vulnerability information must be used. This technology calculates the degree of risk by collecting information and related vulnerabilities in the nodes and the edges existing in the network-based attack graph, and then determining the degree of risk in a specific network location or the degree of risk occurring when a specific route is passed within the network.

Keywords: network security ; common vulnerability scoring system

1. Introduction

Recently, as an increasing variety of attacks have been actively performed, a growing number of methods for detecting them have been studied. The attack graph-based method uses information such as the existing network elements (nodes) and relationships (edges) between elements to identify the optimal intrusion route. By blocking in advance, it is possible to quickly and accurately detect attacks and then defend against them. In particular, it creates an attack graph based on the existing network and uses it to create an optimal attack route. Therefore, the security system that should be applied to the existing network is also determined by calculating the optimal attack route and the expected damage in the network. With this background, in fields such as national defense and security, research is being conducted to perform optimized attack and target removal by applying it to the enemy's network where the attack is being performed. Related research includes the creation of an automated attack/defense agent based on reinforcement learning [1], the establishment of a cyber battlefield based on the attack graph [2][3], and network attack and defense based on the attack graph [4][5][6][7][8][9].

Attack graph is mainly used when a user, who plays the role of an attacker or intruder, attacks a particular network to reach a target and attempts to compensate for each node's vulnerability by utilizing the intrusion results. In this context, it is important to create an attack graph, as well as determine whether it is possible to construct and detect an optimal route to a target node, at a minimum cost. The information used at that time includes network security conditions, node-specific vulnerabilities, input/output information, protocols, and IP addresses. In previous research works, it was difficult to analyze and create the optimal route because they analyzed the network and found the attack route within the vast network. To overcome this problem, recent research has studied an automatic attack route and defended the attack using it [1][10][11].

To identify and supplement vulnerabilities as well as prevent and detect attacks occurring in the network, it is necessary to collect and analyze information related to each node and edge in the network to build an attack graph and detect the optimal route accordingly. Representative network information related to this includes network connection information, access authority, network type, confidentiality, integrity, availability, vulnerability score, etc. Using this information, most existing studies that have built attack graphs and evaluated network vulnerabilities have used CVSS scores. CVSS performs an assessment of each vulnerability and calculates the risk based on information that can be analyzed or collected in common between networks. However, the problem with this is, because common information is used, the exact score cannot be calculated by not considering the special situations or information that each network has, and the association between network nodes is not considered because each vulnerability is evaluated individually. Therefore, further research is needed to solve this problem.

The number of attacks using vulnerabilities in the network is increasing rapidly every year, and attacks using vulnerabilities in the network are mainly used. In addition, CVSS scores, which are mainly used to detect such attacks, have a problem in that they do not reflect the specialized parts of each security environment, because they are a universal evaluation method. Therefore, researchers propose a method that can determine the risk of attacks utilizing network vulnerabilities on nodes and edges, and researchers calculate them comprehensively to calculate the risk of each attack route and accordingly select the priority. The CVSS score, attack type, and vulnerability association analysis were applied to calculate the risk of nodes, edges, and attack routes. CVSS only used access location and CIA scores, and the impacts of the attacks were determined through attack types that occurred during the year collected in advance. In addition, the effect of vulnerabilities between nodes connected to each other through experts was judged and analyzed. Experiments were conducted through a self-generated network to verify the proposed method.

2. Attack Graph

An attack graph is used in various fields, such as attack detection, identification, and defense. In the security field, it helps establish an effective defense system by identifying the intention, attack range, and vulnerability, and providing this information to the defender. It also creates attack scenarios that extract possible routes that attackers can use to break into target networks, which are then used to predict and block future attacks. This approach creates an attack graph for the target, and it conducts attack simulation and attack optimal route analysis. This approach makes it possible to understand what components the network has, and the related cyber assets and problems (damage, situation, etc.) that occur when each asset is attacked.

When carrying out an attack on a target, the attack graph provides priority to decision makers, thus enabling quick and accurate decision-making. By evaluating the attack graph-based target attack method, it provides the decision-maker with the expected effect according to the attack route and the future occurrence situation. Related technologies include Multihost, multistage Vulnerability Analysis (MulVAL), NETSPA, Dijkstra, and Floyd. Jha et al. [4] proposed a method for determining the possibility of an attack through an attack graph to which minimum hitting and greedy algorithm could be applied, so that the minimum security measures to ensure the safety of the network could be determined. Jajodia et al. [5] described a tool that implemented an integrated topology approach to network vulnerability analysis. Then, an attack graph was created to analyze the network security conditions of the Topological Vulnerability Analysis (TVA) tool, Exploit, Nessus vulnerability scanner, and attack route analysis leading to a specific attack target, and the vulnerabilities for each network were analyzed. Ingols et al. [6] proposed a method for analyzing multiple prerequisite graph-based attacks that linearly expanded as the size of the network increased. Ammann et al. [7] proposed a more concise and expandable method with which to solve the expandability and complexity problems that occurred in the existing attack graph generation method. Thus, an attack graph was created and applied to a large network to identify useful information. Wang et al. [8] proposed an attack graph-based automation method that strengthened the network in preparation for intrusions performed in multiple stages, and unlike previous methods, it minimized the cost with satisfactory conditions at the beginning. Sheyner et al. [9] defined the attack model and attack graph in network security, proposed a method for accordingly generating and visualizing the attack graph, and built an attack graph tool to automatically generate the attack graph. This was visually shown to the user, so that vulnerability analysis could be easily performed. Yoon et al. [12] proposed an attack graph-based moving target defense (MTD) technique that used software-defined networking (SDN) to change the host network configuration according to the host importance, and they built a hierarchical attack graph model that provided network topology and network vulnerabilities that could be used to make MTD shuffling decisions. Gonda et al. [13] proposed a method of inferring the importance of vulnerabilities in a LAG or connection graph analysis and relaxed planning graph using node centrality measurement. Thus, the number of meaningless attack vectors was quickly reduced, and the vulnerabilities of each node were properly reflected. In Lu et al. [14], a graph neural network (GNN)-based attack graph ranking method was proposed, and the optimal route of the attack graph was measured through the suitability analysis of the GNN.

3. Common Vulnerability Scoring System

Starting with the Internet worm in November 1988, there has been increasing demand for computer accident response over time, and many accident response teams have provided their own security incident results, but the vulnerability analysis standards were different from one another, thus resulting in the problem that the standards for vulnerabilities became unclear. To solve this problem, the CVSS method was proposed.

CVSS is an open-source-based framework that can calculate vulnerability risk, and it performs risk assessment using items such as access route, complexity, authentication, confidentiality, integrity, availability, etc. The National Institute of Standards and Technology’s National Vulnerability Database (NVD) also provides relevant information. This has the advantage of providing a standard for measuring the vulnerability score, providing an open framework for evaluating the vulnerability score, and prioritizing and supporting vulnerabilities through vulnerability score evaluation. CVSS upgrades vulnerability evaluation items through continuous updates; in this method, three types of matrix information (base, temporal, and environmental) are calculated, and these are used to calculate a vulnerability value. This method is also useful for using the network vulnerability level or detailed values (CIA, access vector, complexity, authentication, etc.) as they are. A recent study improved the CVSS and suggested a more suitable formula for each network.

Yang et al. [15] proposed a DBRank algorithm that calculated the vulnerability of a node by considering the gain and spread of the vulnerability according to the attacker's characteristics. Spanos et al. [16] proposed the Weighted Impact Vulnerability Scoring System (WIVSS) with improved CVSS, and in their method, the CIA weight was modified from the existing base matrix. They proposed new rule about CIA. For example, the weights were defined higher in the order of confidentiality, integrity, and availability, partial value was multiplied by 0.5 of complete value, and impact score range was defined from 0 to 7. Based on these rules, new confidence, integrity, availability values were proposed. In CVSS 2.0, when the CIA value were 'None', 'partial', and 'complete', their weights defined 0.0, 0.275, 0.660. And the weights in confidence, integrity, availability were all the same. But in this method, confidence was 0.0, 1.5, 3.0, integrity was 0.0, 1.2, 2.4 and availability was 0.1, 0.8, 1.6. Through this, more detailed vulnerability risk scores could be obtained.

Jacobs et al. [17] proposed a system for predicting exploits using the information provided by NVD's CVSS and MITER's Common Vulnerabilities and Exposures (CVE), along with other information. The features used in that study were selected through three steps. First, all the available information was extracted from the collected data set, and then information having a share of less than 1% in the entire vulnerability database was removed. Then, meaningless information was removed through expert opinion. Gallon et al. [18] applied CVSS to the attack graph to increase accuracy. The CVE identifier and the base, temporal, and environmental scores of CVSS were used. As such, rather than applying the existing CVSS as is, it was necessary to improve CVSS, and apply a method suitable for each network or attack graph. Gencer et al. [19] proposed a method for determining a fuzzy-based vulnerability score using CVSS score. To define the relationship between the exact inputs and fuzzy multiple outputs, they used a fuzzy logistic regression (FLR). They also used the least squares method to estimate the parameters of the presented model. Ref. [20] proposed a system utilizing a Markov chain and CVSS that identified and evaluated vulnerabilities that occur frequently on the Internet of Medical Things (IoMT). They analyzed representative vulnerabilities and weaknesses from IoMT, and they measured scores by defining CVSS information according to the IoMT network. Moreover, the probability for IoMT threats was calculated based on the Markov transition probability matrix. Ref. [21] proposed a security system for detecting vulnerabilities that exist in the government's website. They used vulnerability scanners to analyze vulnerabilities that existed on websites and measured CVSS scores for vulnerabilities. Through this method, they were able to identify what risks existed on the website and accordingly propose recommendations. Ref. [22] proposed a method to reduce the CVSS matrix by using a decision tree (DT) and reduce the error of the resulting score. A group of 15 experts calculated a vulnerability assessment using the basic CVSS matrix and found that most vulnerabilities scored differently. Therefore, to reduce these errors, basic CVSS metrics were classified, and overlapping parts were removed through a correlation analysis between metrics. In addition, the score was calculated by constructing an attribute subset through a DT-based attribute selection. Ref. [23] proposed a method of detecting vulnerabilities and determining priorities by analyzing the variables and the characteristics of the network environment. Various information was collected through open-source intelligence (OSINT), defined as global, IP, and vulnerability variables, and the risk was calculated using the CVSS based on the variables.

4. Vulnerability Correlation

Most vulnerability research has focused on vulnerability classification and the vulnerability analysis of security assets. However, by applying the attack graph to the network and comparing the vulnerabilities of each node, it has been confirmed that they are affected by the vulnerabilities of other nodes, rather than independent vulnerabilities [24]. This correlation is mainly expressed as a matrix, and refs. [25][26][27][28] also studied how to advance this matrix. For example, in a specific network, if an attacker gains access by exploiting the vulnerability of one node, then in this state, the attacker continues to perform additional attacks using other weaknesses to reach the desired target node. This method shows that there is a correlation between vulnerabilities, and it can be confirmed that vulnerabilities are used as a precondition for attacking other vulnerability. Therefore, it is possible to obtain an optimized attack route or pattern in a vast network by analyzing and understanding the correlations between vulnerabilities.

Li et al. [24] performed a network security analysis by creating a matrix showing CVE correlation information between two nodes. At that time, a vulnerability analysis graph (VCG) was defined to identify the vulnerability correlation, and the correlation was defined as one metric based on the precondition between the CVEs of the two nodes. Liang et al. [29] proposed a method with which to evaluate a network security risk using VCG. They generated VCG, and the node in the graph included the attacked node IP, the name of the vulnerability, the permission obtained through the attack, and the security state at the edge. Nan et al. [30] performed a vulnerability correlation analysis for a network situation analysis while using a correlation coefficient. Further, Debnath et al. [31] proposed a CVSS-based vulnerability and risk assessment (HPCvul) to analyze and evaluate vulnerabilities in high-performance computing (HPC) networks. Standardization was performed to enable smooth sharing of data, and the possibility of an attack using vulnerabilities was identified using an attack graph. Static and dynamic risk assessments were conducted; the static risk assessments were used to explore known vulnerabilities in the network and evaluate the relationship between vulnerabilities to derive the possibility of successful exploitation, while the dynamic risk assessment was used to perform a real-time risk analysis.

References

1. Strickland, E. AI agents play hide-and-seek: An OpenAI project demonstrated “emergent behavior” by AI players-. IEEE Spectr. 2019, 56, 6–7.
2. Pridmore, L.; Lardieri, P.; Hollister, R. National Cyber Range (NCR) automated test tools: Implications and application to network-centric support tools. In Proceedings of the 2010 IEEE AUTOTESTCON, Orlando, FL, USA, 13 September 2010; pp. 1–4.
3. Yamin, M.M.; Katt, B.; Gkioulos, V. Cyber ranges and security testbeds: Scenarios, functions, tools and architecture. Comput. Secur. 2020, 88, 101636.
4. Jha, S.; Sheyner, O.; Wing, J. Two formal analyses of attack graphs. In Proceedings of the 15th IEEE Computer Security Foundations Workshop, Washington, DC, USA, 24 June 2002; pp. 49–63.
5. Jajodia, S.; Noel, S.; O’Berry, B. Topological Analysis of Network Attack Vulnerability. In Managing Cyber Threats; Springer: Boston, MA, USA, 2005; pp. 247–266.
6. Ingols, K.; Lippmann, R.; Piwowarski, K. Practical attack graph generation for network defense. In Proceedings of the 2006 22nd Annual Computer Security Applications Conference (ACSAC’06), Miami Beach, FL, USA, 11 December 2006; pp. 121–130.
7. Ammann, P.; Wijesekera, D.; Kaushik, S. Scalable, graph-based network vulnerability analysis. In Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, USA, 18 November 2002; pp. 217–224.
8. Wang, L.; Noel, S.; Jajodia, S. Minimum-cost network hardening using attack graphs. Comput. Commun. 2006, 29, 3812–3824.
9. Sheyner, O.; Wing, J. Tools for generating and analyzing attack graphs. In International Symposium on Formal Methods for Components and Objects; Springer: Berlin/Heidelberg, Germany, November 2003; pp. 344–371.
10. Walter, E.; Ferguson-Walter, K.; Ridley, A. Incorporating Deception into CyberBattleSim for Autonomous Defense. arXiv 2021, arXiv:2108.13980.
11. Hammar, K.; Stadler, R. Finding Effective Security Strategies through Reinforcement Learning and Self-Play. In Proceedings of the 2020 16th International Conference on Network and Service Management (CNSM), Izmir, Turkey, 2–6 November 2020; pp. 1–9.
12. Yoon, S.; Cho, J.-H.; Kim, D.S.; Moore, T.J.; Free-Nelson, F.; Lim, H. Attack Graph-Based Moving Target Defense in Software-Defined Networks. IEEE Trans. Netw. Serv. Manag. 2020, 17, 1653–1668.
13. Gonda, T.; Pascal, T.; Puzis, R.; Shani, G.; Shapira, B. Analysis of Attack Graph Representations for Ranking Vulnerability Fixes. In Proceedings of the Global Conference on Artificial Intelligence, Luxembourg, 17–19 September 2018; pp. 215–228.
14. Lu, L.; Safavi-Naini, R.; Hagenbuchner, M.; Susilo, W.; Horton, J.; Yong, S.L.; Tsoi, A.C. Ranking attack graphs with graph neural networks. In Proceedings of the International Conference on Information Security Practice and Experience, Xi’an, China, 13–15 April 2009; pp. 345–359.
15. Yang, X.; Shunhong, S.; Yuliang, L. Vulnerability ranking based on exploitation and defense graph. In Proceedings of the 2010 International Conference on Information, Networking and Automation (ICINA), Kunming, China, 17–19 October 2010; pp. V1-163–V1-167.
16. Spanos, G.; Sioziou, A.; Angelis, L. WIVSS: A new methodology for scoring information systems vulnerabilities. In Proceedings of the 17th Panhellenic Conference on Informatics, Thessaloniki, Greece, 19–21 September 2013; pp. 83–90.
17. Jacobs, J.; Romanosky, S.; Edwards, B.; Adjerid, I.; Roytman, M. Exploit Prediction Scoring System (EPSS). Digit. Threat. Res. Pract. 2021, 2, 1–17.
18. Gallon, L.; Bascou, J.J. Using CVSS in attack graphs. In Proceedings of the 2011 Sixth International Conference on Availability, Reliability and Security, Vienna, Austria, 22–26 August 2011; pp. 59–66.
19. Gencer, K.; Başçiftçi, F. The fuzzy common vulnerability scoring system (F-CVSS) based on a least squares approach with fuzzy logistic regression. Egypt. Inform. J. 2020, 22, 145–153.

20. Allouzi, M.A.; Khan, J.I. Identifying and modeling security threats for IoMT edge network using Markov chain and common vulnerability scoring system (CVSS). arXiv 2021, arXiv:2104.11580.

21. Putra, F.G.; Soewito, B. Measurement of Security System Performance on Websites of Personnel Information Systems in Government Using Common Vulnerability Scoring System. J. Pendidik. Tambusai 2022, 6, 2949–2957.

22. Kai, S.; Zheng, J.; Shi, F.; Lu, Z. A CVSS-based Vulnerability Assessment Method for Reducing Scoring Error. In Proceedings of the 2021, 2nd International Conference on Electronics, Communications and Information Technology (CECIT), Sanya, China, 27–29 December 2021; pp. 25–32.

23. Reyes, J.; Fuertes, W.; Arévalo, P.; Macas, M. An Environment-Specific Prioritization Model for Information-Security Vulnerabilities Based on Risk Factor Analysis. Electronics 2022, 11, 1334.

24. Li, Z.-Y.; Xie, C.-H.; Tao, R.; Zhang, H.; Shi, N. A Network Security Analysis Method Using Vulnerability Correlation. In Proceedings of the 2009 Fifth International Conference on Natural Computation, Tianjian, China, 14–16 August 2009; pp. 17–21.

25. Ali, M.U.; Aydi, H.; Batool, A.; Parvaneh, V.; Saleem, N. Single and Multivalued Maps on Parametric Metric Spaces Endowed with an Equivalence Relation. Adv. Math. Phys. 2022, 2022, 6188108.

26. Zhou, M.; Saleem, N.; Liu, X.-L.; Özgür, N. On two new contractions and discontinuity on fixed points. AIMS Math. 2022, 7, 1628–1663.

27. Saleem, N.; Zhou, M.; Bashir, S.; Husnine, S.M. Some new generalizations of F-contraction type mappings that weaken certain conditions on Caputo fractional type differential equations. AIMS Math. 2021, 6, 12718–12742.

28. Kalsoom, A.; Saleem, N.; Işık, H.; Al-Shami, T.M.; Bibi, A.; Khan, H. Fixed Point Approximation of Monotone Nonexpansive Mappings in Hyperbolic Spaces. J. Funct. Spaces 2021, 2021, 3243020.

29. Liang, L.; Yang, J.; Liu, G.; Zhu, G.; Yang, Y. Novel method of assessing network security risks based on vulnerability correlation graph. In Proceedings of the 2012 2nd International Conference on Computer Science and Network Technology, Changchun, China, 29–31 December 2012; pp. 1085–1090.

30. Nan, X.; Chen, R.; Tian, H.; Liu, Y. Network Situation Risk Assessment Based on Vulnerability Correlation Analysis. In Proceedings of the 2021 IEEE International Conference on Progress in Informatics and Computing (PIC), Shanghai, China, 17–19 December 2021; pp. 330–334.

31. Debnath, J.K.; Xie, D. CVSS-based Vulnerability and Risk Assessment for High Performance Computing Networks. In Proceedings of the 2022 IEEE International Systems Conference (SysCon), Montreal, QC, Canada, 25 April–23 May 2022; pp. 1–8.