Study on the Application of Graph Theory Algorithms and Attack Graphs in Cybersecurity Assessment

Jinghan Zhang
Department of Mechanical Engineering
City University of Hong Kong
Hong Kong, China
jzhang2669-c@my.cityu.edu.hk

Wei Wang*
Department of Mechanical Engineering
City University of Hong Kong
Hong Kong, China
Shenzhen Research Institute of City University of Hong Kong
Shenzhen, China
wwang326@cityu.edu.hk
*Corresponding author

Enrico Zio
Department of Energy
Politecnico di Milano
Milano, Italy
MINES Paris - PSL Université
Centre de Recherche sur les Risques et les Crises (CRC)
Sophia Antipolis, France
enrico.zio@polimi.it

2023 7th International Conference on System Reliability and Safety (ICSRS) | 979-8-3503-0605-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSRS59833.2023.10381005

***Abstract*—Postulating the behavior of attackers is important in the design of cybersecurity protection measures. Attack graph is a technique employed for this purpose, which aids in identifying and modeling the potential attack paths an attacker could take to gain unauthorized access to a cyber network, exploit vulnerabilities, and compromise the system's confidentiality, integrity, and availability. In this study, we propose a framework aimed at identifying potential attack paths and determining the shortest path with the highest probability of a successful attack. Meanwhile, based on the attack graph determine the minimum patch sets with the most severity to protect the network security. Common Vulnerability Scoring System (CVSS) is utilized to quantify the exploitability and severity of each vulnerability. The Dijkstra algorithm is utilized to calculate the shortest path with the highest probability, and the Stoer-Wagner algorithm is utilized to calculate the minimum patch sets with the most severity. To demonstrate the proposed framework, we apply it to a simplified SCADA system within a corporate network susceptible to cyber attacks.**

Keywords—cybersecurity, attack graph, attack path, patch set, CVSS, Dijkstra algorithm, Stoer-Wagner algorithm

I. INTRODUCTION

With the development of the network, the network has more and more influence on people. While the Internet brings convenience to people's lives, the problem of network security is also more and more serious [1]. Whether it is an individual, a company or a government agency, they may suffer great losses due to cyber attacks, such as the theft or destruction of important information, and the paralysis of network systems due to attacks [2, 3]. In this case, it is urgent to seek the corresponding technical methods to solve the network security problem.

There are several techniques to protect networks, such as firewall, and intrusion detection system (IDS). Firewall technology can control network connections to effectively prevent network attacks (but in solving application-level security problems, there are some limitations). Intrusion detection technology finds network attacks by means of statistics or abnormal detection rules, and then takes defensive measures (but some covert network attacks are difficult to identify). Firewall and intrusion detection technology belong to the security technology that can be adopted to protect the network security when a network attack occurs. If we take precautions before the network attack, carry out security assessments on the network system, and take some repair and defense measures according to the evaluation results, network security will be improved to a greater extent. Based on this, people can scan the devices (hosts) and applications of the target network through the vulnerability scanner to find possible vulnerabilities, so that timely measures can be taken to patch or reinforce. Network security can be improved by scanning and fixing vulnerabilities, but only individual vulnerability information can be obtained based on this. In order to analyze the correlation between vulnerabilities and the mutual exploitation of vulnerabilities among various devices in the network system, many security assessment models have been proposed. Typical evaluation methods include Fault Tree [4], Attack Tree [5, 6], Petri Net [7, 8], Privilege Graph [9], and Attack Graph [10].

Attack graph has been widely recognized as a very effective evaluation model. The attack graph is used to represent the vulnerabilities in the evaluated network system, the relationship between the vulnerabilities, and the potential attack path for an attacker to exploit these vulnerabilities for a one - or multi-step attack [10]. To protect the security of the network system and prevent the key resources in the network system from being attacked, network security defenders need to take some security measures to increase the security of the network system. How to achieve the purpose of protecting network security with the set of security measures at the least cost, the attack graph can provide an important reference for solving this problem. There are two types of attack graphs: state attack graphs and attribute attack graphs. It is difficult to use state attack graphs in large-scale networks because of the state explosion problem. Now more and more researchers have begun to study attribute attack graph.

After this, there are many researchers to study how to generate attack graphs. Ritchey proposed to use SMV (Symbolic Model Verifier) method to automate the construction of an attack graph [11]. Sheyner and Richard .etl. used model checker NuSMV to evaluate the corporate networks [12, 13]. Ammann proposed the monotonicity hypothesis of the attack graph, reduced the complexity of generating an attack graph from the exponential level to the polynomial level, and proposed a graph theory-based method to generate an attack graph [14, 15]. Lippmann [16] and Ingols [17] et al. also used graph theory to generate attack graphs and pruning techniques to reduce the complexity of attack graphs. Ou et al. use logic programming to generate attack graphs, which further reduces the complexity of generating attack graphs, and the computational complexity is only O(n2) [18, 19]. The attack graph established based on model checking is the state attack graph. Model checking tools can be used to automatically generate all possible attack paths, so as to automatically construct attack scenarios and generate attack graphs. However, the attack graph based on model checking has the problem of state explosion, which makes the established state attack graph very large when it is applied to large-scale networks and has limitations in scalability. The monotony hypothesis of the attack graph proposed by Ammann is very important [14, 15]. The so-called monotonicity assumption refers to the fact that the attacker will not lose the conditions gained from the previous attack in a subsequent attack. Based on this assumption, the attribute attack graph is built, and the complexity of the attack graph is greatly reduced.

Analyzing attack graphs will help us to obtain qualitative and quantitative results. Quantitative evaluation of network security is very important. Attack probability, that is, the probability of a successful attack, reflects the security condition of a network system. There are many researchers have researched how to calculate the successful attack probability. Phillips et al. proposed to use the vulnerability on the attack path to calculate the probability of the attack path by exploiting complexity [10]. Jha et al. believe that the attacker's attack selection is Markov and calculate the attack probability by using the Markov decision model [20]. Mehta et al. calculated the attack probability by using the Google Page Rank algorithm [21, 22]. These three examples are used to calculate the attack probability of the state attack graph. In addition, Bayesian network pages are often used to calculate attack probabilities. Dantu et al. described the behavior and capability of the attacker and used the Bayesian network method to calculate the node attack probability [23, 24]. Frigault et al. used Bayesian networks to analyze the inherent risks of network systems, and adopted dynamic Bayesian networks to integrate the characteristics of vulnerability changes over time into the assessment process [25, 26]. Poolsappasit uses Bayesian attack graphs for static and dynamic evaluation of network systems [27]. In these methods used to analyze the attack graphs, the graph-based approaches (graph path algorithm), such as the Dijkstra algorithm, Bellman-Ford algorithm, Floyd-Warshall algorithm, and Stoer-Wagner algorithms, are often considered simpler and more intuitive compared to Bayesian and Markov models. Additionally, the graph approach offers advantages in terms of scalability as it does not necessitate model training and is minimally affected by node modifications. In this study, we integrate graph-based approaches and attack graphs to analyse network security.

In this study, we contribute solutions to the shortest attack path problem and the minimum patch set problem based on attribute attack graphs. The shortest attack path indicates the path an attacker is most likely to take, and the minimum patch set can provide an important reference for resource allocation of security measures. We aim to use the graph theory algorithm applicable to attack graphs to solve the corresponding problems.

Based on these objectives, part I, is a summary of past literature, including Network security technology, attack graph generation method, and attack graph analysis method; part II, is the basic definition of attack graph and strategy of cyber security assessment based on attack graph are introduced; in part III, the analysis of attribute attack graph and framework of searching for the shortest attack path and minimum patch set are introduced; in part IV, a case study is used to illustrate the framework; in part V, the possible drawbacks of the framework and future research possibilities are introduced.

II. ATTACK GRAPH

A. Basic Definition

When attackers are penetrating the network, a series of continuous attacks taken by the attacker is an attack path from the attacker node to the target node. An attack graph is a representation of all possible paths. Depending on the different meanings expressed by the vertices and edges in the attack graph, it can be divided into multiple types, mainly including state attack graphs and attribute attack graphs [28].

In state attack graphs, vertices represent network state information such as hostnames and provided services, while directed edges represent transitions between states [29]. A state attack graph can be represented as AG=(E, V), where E is the set of edges or atomic attack sets, and any edge $e \in E$ represents the transition of the global state. V represents the set of state vertices, and for any vertex $v \in V$, it can be represented using a quadruple <h, srv, vul, x>, where h represents the involved host in that state, srv represents the involved service, vul represents the vulnerabilities existing in that state, and x represents other relevant information that needs to be considered. For large-scale network systems, the rapid growth of states during transitions leads to an excessively large-scale state attack graph.

Attribute attack graphs contain two types of vertices and two types of edges [30]. The two types of vertices include condition vertices and vulnerability vertices. Condition vertices represent the permissions currently possessed by the attacker, while vulnerability vertices represent existing vulnerabilities and the atomic attacks that successfully exploit those vulnerabilities. The two types of edges include edges from condition vertices to vulnerability vertices, indicating that the vulnerability is successfully exploited when all the preconditions are met, and edges from vulnerability vertices to condition vertices, indicating the postconditions obtained after successfully exploiting the vulnerability. An attribute attack graph can be represented as AG=(C, V, E), where C represents the set of conditions, including preconditions and postconditions, V represents the set of vulnerabilities, and E represents the set of edges.

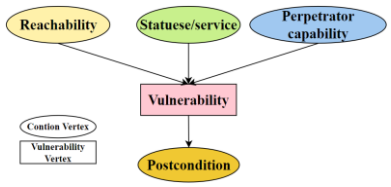
B. *Strategy of Cyber Security Assessment Based on Attack Graph*

The overall strategy for utilizing the attack graph method in evaluating cyber security is, the first step involves understanding the actual system configuration, while the second step entails constructing an attack graph based on the network topology. In the third step, attack profiles are assigned, with the objective of identifying the shortest attack path with the highest probability. The fourth step involves assigning attributes to each vertex, encompassing vulnerabilities within the network system and the conditions necessary for the successful exploitation of these vulnerabilities. In the fifth step, the results are evaluated, and if they meet the safety requirements, the process concludes; otherwise, defenders must undertake enhancements and repeat the cybersecurity assessment.

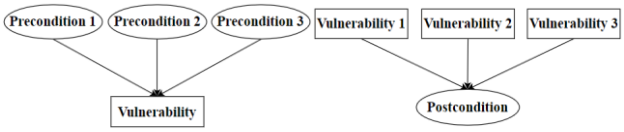
III. Attack Graph Analysis

A. *Attribute Attack Graph*

In this study, the attack graph is an attribute attack graph. Vulnerabilities are exploitable weaknesses in the design, implementation, or management of a system. Preconditions are a set of system properties that must exist in the network system, so that the exploits can be successful. An initial precondition is a system property that exists inherently in a system. Preconditions can include three types of preconditions: statuses/services, reachability, and perpetrator capability. Statuses/services mean the target holds particular operating systems, software/applications, services, or is in a particular state. Reachability means the target is reachable. Perpetrator capability means the perpetrator has the ability to exploit vulnerability or privilege levels.



(a) Precondition, vulnerability, and postcondition



(b) Multiple preconditions (c) Multiple vulnerabilities

Fig. 1. Schematic diagram of attack graph.

Figure 1 depicts the schematic diagram of the attack graph employed in this study. In Figure 1 (a), the diagram illustrates the elements comprising the graph, namely the precondition, vulnerability, and postcondition. Within the attack graph, the condition (comprising the precondition and postcondition) is represented by an ellipse-shaped vertex, while the vulnerability is represented by a rectangular vertex. Figure 1 (b) demonstrates that an atomic attack necessitates the fulfillment of multiple preconditions. This implies that the successful exploitation of the vulnerability entails the satisfaction of all these preconditions, establishing a logical relation of conjunction (AND). Figure 1 (c) showcases a scenario where a postcondition can be triggered by multiple vulnerabilities. This implies that various distinct vulnerability attacks can lead to the same postcondition, signifying a logical relation of disjunction (OR).

B. *Strategy of Searching for Attack Paths and Patch Sets*

In this study, we integrate the graph theory with an attack graph. The objective is to search the potential attack paths and patch sets [31, 32].

- Attack path: the hackers may take to exploit weaknesses in the network systems. An attack graph typically contains many attack paths, each representing a series of exploits or atomic attacks.

  Shortest path with the highest probability: among all attack paths, the path with the highest probability of success and the shortest attack steps.

- Patch set: a set of atomic attacks, patching these attacks can make the network system safe. A patch set is critical if and only if the attacker cannot reach his goal, removing any one of the patch sets is likely to enable an attacker to successfully attack.

  Minimum patch set with the most severity: among all patch sets, the patch set with the most severity of vulnerabilities and the smallest number of vulnerabilities.

The process of searching for the shortest path with the highest probability and minimum patch sets in the attack graph is depicted in Figure 2. In an attack graph, it is important to determine the potential attack paths and needful patch sets, and this is based on the probabilities of successfully attacking nodes and the severity of vulnerability nodes. In this case, the Dijkstra algorithm [33, 34], a graph theory algorithm, was selected as the approach to finding the shortest path with the highest probability; the Stoer-Wagner algorithm [35], a graph theory algorithm, was selected as the approach to finding the minimum patch sets. Regarding the probabilities of the nodes being attacked, it is related to vulnerability's exploitability. Regarding the severity of vulnerability nodes being attacked, it is related to the vulnerability severity. The exploitability and severity of each vulnerability are assessed using the Common Vulnerability Scoring System (CVSS) [36], which provides a method for capturing the fundamental characteristics of a vulnerability and assigning it a numerical score that reflects its exploitability and severity.

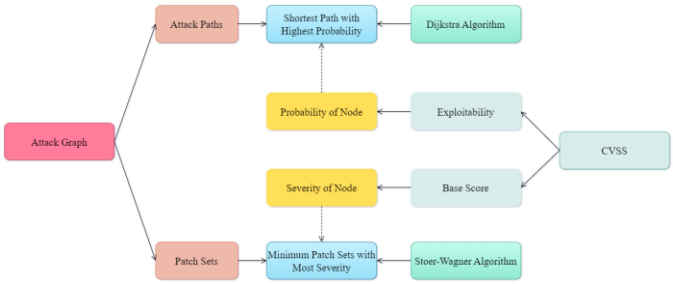


Fig. 2. Strategy of searching for the shortest path with the highest probability and minimum patch sets with the most severity.

C. *Shortest Path with Highest Probability Identification*

When using the Dijkstra algorithm to find the shortest path with the highest probability of a successful attack, the steps are:

1) Assign probabilities to all arcs in the network;

2) Label arcs and nodes;

3) Convert probabilities to natural logarithms;

4) Multiply probabilities by actually adding logs: a*b=c →ln(ab) = ln(c), ln(a) + ln(b) = ln(c);

5) Maximize the probability of success by actually minimizing the negative probability of success;

6) Convert all the logs to positive numbers by multiplying by (-1);

7) Give integer arc lengths to the shortest path code;

8) After the shortest path is obtained, take the total distance, multiply by (-1), and raise e to this power to obtain the highest probability path length.

D. *Minimum Patch Sets Identification*

When using Stoer-Wagner to find the minimum patch sets with the most severity, the steps are:

1) Assign severities to all arcs in the network;

2) Label arcs and nodes;

3) Convert severities to natural logarithms;

4) Compare severities by adding logs: a<b → -ln(a) > -ln(b);

5) Minimize the patch sets with most severities by minimizing the severities of vulnerabilities;

6) Give integer arc lengths to the minimum patch sets code.

IV. Case Study

A. *Network Structure*

The approach is illustrated as a simplified SCADA (supervisory control and data acquisition) [37, 38], as shown in Figure 3. There are several components, Firewall1 (F1), Firewall2 (F2), Host1 (H1), Host2 (H2), and Internal Sever (S). The communication rules (protocol) in this network case are shown in Table I.

This simplified system pertains to the Ukraine power grid SCADA system [39]. It involves three types of networks: the outer network, which represents the internet, the corporate network, and the SCADA network. The attack target is to obtain privileges within the SCADA network. To protect the network, two firewalls are deployed to separate the different network types. Within the corporate network, there exists an internal server and two hosts. The vulnerabilities corresponding to CVEs (Common Vulnerabilities and Exposures) are shown in Table II [40]. In CVSS [36], the Base Score serves as an indicator of the severity of a vulnerability based on its intrinsic characteristics, which remain constant over time. Moreover, it assumes a reasonable worst-case scenario in terms of the impact across various deployed environments. The Base Score consists of two distinct sets of metrics: Exploitability metrics and Impact metrics. The Exploitability metrics aim to gauge the ease and technical means through which the vulnerability can be exploited. They essentially encapsulate the attributes of the vulnerable component, formally referred to as such. On the other hand, the Impact metrics focus on the direct consequences resulting from a successful exploit. They represent the repercussions endured by the impacted component, which is the formal terminology used to describe the affected entity.

Within the given assumption that the administration privilege is Host1 < Internal sever < Host2, the attack graph can be established accordingly. In this scenario, if hackers successfully acquire the privilege of the server initially, they will not attempt to obtain the privilege of Host1 but will persist in pursuing the privilege of Host2. Based on these preconditions, the vulnerability vertex can be conditioned, resulting in the following attack graph:

- Initial Access: Exploit a vulnerability or bypass Firewall1.
- Privilege Escalation: Attain the privilege of Host1 or sever or Host2.
- Lateral Movement: Transition among Host1, server and Host2.
- Further Access: Exploit a vulnerability or bypass Firewall2.

This conditioned attack graph delineates the potential attack path, where the primary objective is to bypass Firewall1 initially, followed by gaining the privilege of Host1 or sever or Host2, and ultimately exploiting a vulnerability or bypassing Firewall2.

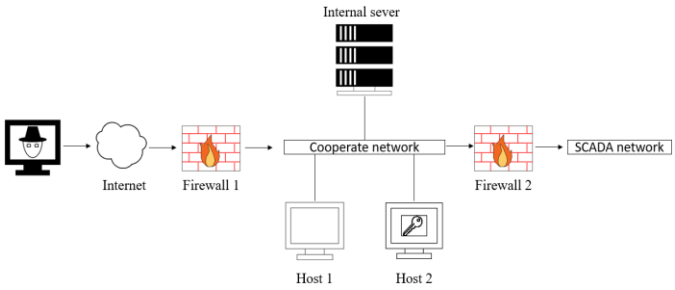


Fig. 3. Simplified case of network topology [16-17].

TABLE I. Communication Rules (Protocol) in This Case

Source	Destination	Protocol
Firewall 1	Host 1	<F1, H1>
	Host 2	<F1, H2>
	Sever	<F1, S>
	Firewall 2	<F1, F2>
Host 1	Host 2	<H1, H2>
	Sever	<H1, S>
	Firewall 2	<H1, F2>
Internal server	Host 2	<S, H2>
	Firewall 2	<S, F2>
Host 2	Firewall 2	<H2, F2>

TABLE II. CORRESPONDING CVES FOR THE CASE

Components	Vulnerabilities	Base Score	Exploitability Score	Impact Score
Firewall 1	CVE-2022-3480	7.5	3.9	3.6
	CVE-2022-30276	7.5	3.9	3.6
Firewall 2	CVE-2022-47361	7.8	1.8	5.9
	CVE-2019-0039	8.1	2.2	5.9
Host 1	CVE-2021-41192	6.5	2.8	3.6
	CVE-2021-22909	7.5	1.6	5.9
	CVE-2021-39155	7.5	3.9	3.6
	CVE-2021-21220	8.8	2.8	5.9
	CVE-2021-35395	9.8	3.9	5.9
Sever	CVE-2022-33139	9.8	3.9	5.9
	CVE-2020-4610	7.8	1.8	5.9
	CVE-2020-17533	8.1	2.8	5.2
	CVE-2021-37147	7.5	3.9	3.6
	CVE-2021-29529	7.8	1.8	5.9
Host 2	CVE-2021-38759	9.8	3.9	5.9
	CVE-2022-3480	7.5	3.9	3.6
	CVE-2020-3812	5.5	1.8	3.6
	CVE-2020-9054	9.8	3.9	5.9
	CVE-2021-30860	7.8	1.8	5.9

B. Attack Graph for the Case

Based on the vulnerabilities, an attack graph has been constructed to visually illustrate the potential attack paths and their dependencies, as depicted in Figure 4. The attack graph serves as a graphical representation of the sequential stages or steps that attackers may undertake to exploit the vulnerabilities and accomplish their objectives. Within the attack graph, two primary types of vertices are present: one represents the vulnerabilities inherent in the network system, and the other represents the conditions necessary for a successful attack, including the statuses/services of network components, reachability, and the perpetrator capabilities. Preconditions refer to the conditions required for a successful attack on the vulnerabilities, while postconditions denote the conditions obtained after the successful exploitation of the vulnerabilities.

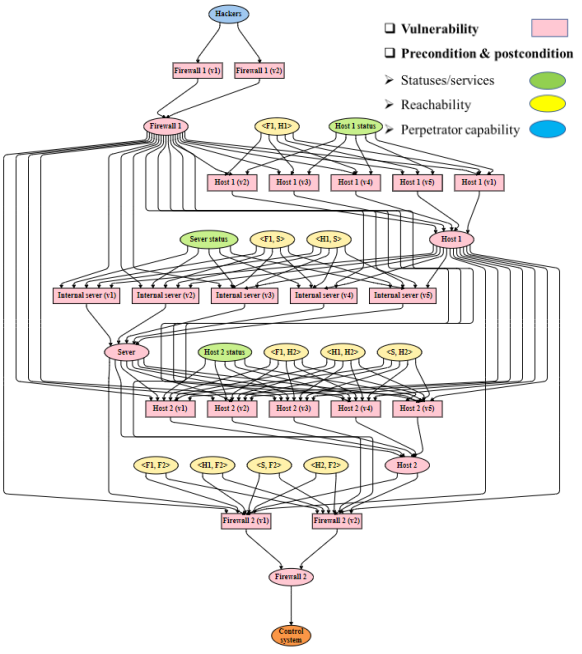


Fig. 4. Attack graph for the case study.

C. Results for this Case

1) Shortest attack path

Based on Dijkstra algorithm, the qualitative result (the shortest attack path) is: ['A', 'B', 'D', 'H', 'e', 'g'], ['A', 'C', 'D', 'H', 'e', 'g'], ['A', 'B', 'D', 'H', 'f', 'g'], ['A', 'C', 'D', 'H', 'f', 'g']

To obtain quantitative results, it is necessary to assign exploitability and propensity values to each node. Exploitability can be obtained from the Common Vulnerability Scoring System (CVSS). At this stage, by solely incorporating CVSS exploitability scores into the attack graph, the quantitative outcome of the shortest attack path with the highest exploitability can be represented consisting of the nodes:

['A', 'B', 'D', 'H', 'f', 'g'], ['A', 'C', 'D', 'H', 'f', 'g']

With an exploitability of 0.564, corresponding to a 2.2 exploitability score in CVSS.

Note: A = Attacker, B = Firewall1 (v1), C = Firewall1 (v2), D = Firewall1, H = <F1, F2>, e = Firewall2 (v1), f = Firewall2 (v2), g = Firewall2, herein, “v” represents vulnerability.

2) Minimum patch sets

Based on the Stoer-Wagner algorithm, the qualitative result (the minimum patch sets) is: {'B', 'C'}, {‘e’, ‘f’}

To obtain quantitative results, it is necessary to assign the severity to each vulnerability node. Severity can be represented by the Base Score from the Common Vulnerability Scoring System (CVSS). At this stage, by solely incorporating CVSS Base scores into the attack graph, the quantitative outcome of the minimum patch sets with the most severity can be represented consisting of the nodes: {'B', 'C'}

Note: B = Firewall1 (v1), C = Firewall1 (v2), e = Firewall2 (v1), f = Firewall2 (v2), g = Firewall2.

V. CONCLUSIONS

This study focused on the importance of understanding attacker behavior in the context of designing effective cybersecurity protection measures. The utilization of attack graphs proved to be a valuable technique for identifying and modeling potential attack paths that could be exploited by unauthorized individuals seeking access to a cyber network. The main contribution of this research was the proposal of a framework aimed at identifying potential attack paths and determining the shortest path with the highest probability of a successful attack; identifying possible patch sets and determining the minimum patch set with the most severity. By leveraging the Common Vulnerability Scoring System (CVSS), the framework quantified the exploitability and severity of each vulnerability, enabling a more comprehensive analysis. The study employed the Dijkstra algorithm to calculate the shortest path with the highest probability, thus providing insights into the most likely attack routes. Additionally, the Stoer-Wagner algorithm was utilized to determine the minimum patch sets with the most severity, helping prioritize security measures and protect the network effectively. To illustrate the applicability of the proposed framework, the study applied it to a simplified SCADA system within a corporate network, which is known to be vulnerable to cyber attacks. By demonstrating the effectiveness of the framework in this specific context, we

showcased its potential for enhancing network security and mitigating potential threats.

This study contributes to the field of cybersecurity by providing a systematic approach to understanding and mitigating potential attack paths, and identifying the possible patch sets. The proposed framework, along with the algorithms and methodologies employed, offer valuable insights for security practitioners and researchers alike, enabling them to make informed decisions in protecting critical systems and networks from malicious actors.

At this stage, the main purpose is to study the graph theory algorithms that are suitable for analyzing the attack graph to get the shortest attack path and the minimum patch set. This study has a lot of room for further research. In reality, it is not enough to only consider the shortest path and the minimum patch set, because many factors in reality will affect the choice of the attacker's attack path, which will easily affect the final result. Therefore, it is also necessary to consider other relatively short paths and relatively serious weaknesses. In addition, it is important to allocate resources according to the actual resources used for security measures.

ACKNOWLEDGEMENT

This work was supported by National Natural Science Foundation of China (Project no. 72101221) and GRF – RGC General Research Fund CityU 11215323 (Project no. 9043545).

REFERENCES

[1] World Economic Forum. The Global Risks Report 2022, 17th Edition. World Economic Forum, 2022.

[2] Zhao Y, Adina N, Blanc K, et al. Dynamic Probabilistic Risk Assessment for Cyber Security Risk Analysis of the Electric Grid// Proceedings of the 29th European Safety and Reliability Conference (ESREL). 2020.

[3] Anand K, Duley C, Gai P. Cybersecurity and financial stability. Discussion Papers, 2022.

[4] Helmer G, Wong J, Slagell M, et al. A Software Fault Tree Approach to Requirements• Analysis of all Intrusion Detection System. Requirements Engineering Journal, 7(4), 2002,pp.207-220.

[5] Schneier B. Attack Trees. Dr. Dobb's Journal, 24(12), 1999, pp.21-29.

[6] Moore A, Ellison R, Linger R. Attack Modeling for Information Security and Survivability. Technical Note, CMU/SEI-2001-TN-001, 2001.

[7] Mcdermott J. Attack Net Penetration Testing. In: Proc of the 2000 New Security Paradigms Workshop, Ballycotton, County Cork, Ireland, ACM Press, 2000, pp.15-22.

[8] Laborde R, Nasser B, Grasset F, et al. A Formal Approach for the Evaluation of Network • Security Mechanisms Based on RBAC Policies. Electronic Notes in Theoretical Computer Science, 2005, pp.117-142.

[9] Dacier M, Deswartes Y, Kaaniche M. Quantitative Assessment of Operational SecurityModels and Tools. Technical Report Research Report 96493, LAAS, 1996.

[10] Phillips and Swiler are the first to put forward the concept of attack graph [Phillips C, Swiler L P. A graph-based system for network-vulnerability analysis//Proceedings of the 1998 workshop on New security paradigms. 1998: 71-79.

[11] Ritchey R, Ammann P. Using Model Checking to Analyze Network Vulnerabilities. In: Proc. Of the IEEE Symposium on Security and Privacy, 2000, pp.156-165.

[12] Oleg Sheyner, Joshua Haines, Somesh Jha, Richard Lippmann, and Jeannette M. Wing. 2002. Automated Generation and Analysis of Attack Graphs. In Proceedings of the IEEE Symposium on Security and Privacy.

[13] Oleg Mikhail Sheyner. 2004. Scenario graphs and attack graphs. Ph.D. Dissertation. Carnegie Mellon University. AAI3126929.

[14] Ammann P, Wijesekera D, Kaushik S. Scalable, graph-based network vulnerability analysis. In: Proc. of the 9th ACM Conf. on Computer and Communications Security, New York, ACM Press, 2002, pp.217-224.

[15] Ammann P, Pamula J, Street J, et al. A host-based approach to network attack chaining analysis. In: Proc. of the 21st Annual Computer Security Applications Conference, 2005, pp. 71-84.]

[16] Lippmann, R, Ingols K, Scott C, et al. Validating and Restoring Defense in Depth Using Attack Graphs. In: Proc. of the Military Com. Conf., 2006, pp.1-10.

[17] Ingols K, Lippmann R, Piwowarski K. Practical Attack Graph Generation for Network Defense. In: Proc. of Comp. Sec. App. Conf., 2006, pp.121-130.

[18] Ou X, Govindavajhala S, Appel A. MulVal: a logic-based network security analyzer. In: the 14th USENIX Security Symposium, MD, USA, ACM Press, 2005, pp.113-128.

[19] Ou X, Boyer W, McQueen M. A scalable approach to attack graph generation. In: Proc. of the 13th ACM Conf. on Computer and Communications Security, Alexandria, ACM Press,2006, pp.336-345.

[20] Jha S, Sheyner O, Wing J. Two formal analyses of attack graphs. In: Proc. of the 15th IEEE Computer Security Foundations Workshop, Cape Breton, IEEE Computer Society,2002, pp.49-63.

[21] Mehta V, Bartzis C, Zhu H. Ranking Attack Graphs. in: Proc of the 9th International Symposium on Recent Advances in Intrusion Detection, Hamburg, Germany, Springer Press,2006, pp.127-144.

[22] Monica B, Marci G, Franco S. Inside Pagerank. ACM Transactions on IntemetTechnology, 2005, pp.92-128.

[23] Dantu R, Loper K, Kolan P. Risk Management Using Behavior based Attack Graphs. In: •Proc of 2004 International Conference on Information Technology, Coding and Computing(ITCC 2004), Las Vegas, Nevada, USA, IEEE Press, 2004.

[24] Dantu R, Kolan P. Risk management using behavior based Bayesian networks. In: IEEEInternational Conference on Intelligence and Security Informatics, May, 2005.

[25] Frigault M, Wang L. Measuring Network Security Using Bayesian Network-Based Attack Graphs. In: Proc. 32nd Ann. IEEE Int'l Computer Software Applications Conf., 2008, pp.698-703.

[26] Frigault M, Wang L. Singhal A, et al. Measuring Network Security Using Dynamic Bayesian Network. In: Proc. of 14th ACM Workshop Quality of Protection, 2008.

[27] Poolsappasit N, Dewri R, Ray 1. Dynamic Security Risk Management Using Bayesian Attack Graphs. IEEE Transactions on Dependable and Secure Computing, 9(1), 2012, pp.61-74.

[28] Hong J B, Kim D S, Chung C J, et al. A survey on the usability and practical applications of graphical security models. Computer Science Review, 2017, 26: 1-16.

[29] Lippmann R P, Ingols K W. An annotated review of past papers on attack graphs. 2005.

[30] Lallie H S, Debattista K, Bal J. A review of attack graph and attack tree visual syntax in cyber security. Computer Science Review, 2020, 35: 100219.

[31] Liu X. A network attack path prediction method using attack graph. Journal of Ambient Intelligence and Humanized Computing, 2020: 1-8.

[32] Jha S, Sheyner O, Wing J. Two formal analyses of attack graphs//Proceedings 15th IEEE Computer Security Foundations Workshop. CSFW-15. IEEE, 2002: 49-63.

[33] Al-Tameemi H L H. Using dijkstra algorithm in calculating alternative shortest paths for public transportation with transfers and walking, 2014.

[34] Noto M, Sato H. A method for the shortest path search by extended Dijkstra algorithm//Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0. IEEE, 2000, 3: 2316-2320.

[35] Arikati S R, Mehlhorn K. A correctness certificate for the Stoer–Wagner min-cut algorithm. Information Processing Letters, 1999, 70(5): 251-254.

[36] Duy Le T, Ge M, The Duy P, et al. Cvss based attack analysis using a graphical security model: Review and smart grid case study//Smart Grid and Internet of Things: 4th EAI International Conference, SGIoT 2020,

TaiChung, Taiwan, December 5–6, 2020, Proceedings. Springer International Publishing, 2021: 116-134.

[37] Ghosh S, Sampalli S. A survey of security in SCADA networks: Current issues and future challenges. IEEE Access, 2019, 7: 135812-135831.

[38] Urooj B, Ullah U, Shah M A, et al. Risk Assessment of SCADA Cyber Attack Methods: A Technical Review on Securing Automated Real-time SCADA Systems//2022 27th International Conference on Automation and Computing (ICAC). IEEE, 2022: 1-6.

[39] Whitehead D E, Owens K, Gammel D, et al. Ukraine cyber-induced power outage: Analysis and practical mitigation strategies//2017 70th Annual Conference for Protective Relay Engineers (CPRE). IEEE, 2017: 1-8.

[40] Stouffer K, Falco J, Scarfone K. Guide to industrial control systems (ICS) security. National Institute of Standards and Technology, 2008.