Attack Strategy Prediction with Precisely Estimated Probability and Evidence Mapping

Chih-Hung Wang and Ren-Wei Liou

***Abstract*—With the advance on computer science technology, the attackers' ability has continuously improved as well, and their artifices are getting more and more sophisticated. To secure the system, an intrusion detection system is usually deployed to notice the administrator when abnormal events happen. However, due to the poor quality of IDS alerts, massive quantity of alerts will hold analysts back when defending. Furthermore, it is not enough to avoid the attacking threats only knowing current crisis. With the assist of prediction, defenders can one step ahead of the attackers to increase the successful defense rate. In this paper, we propose a network attack prediction framework based on the improved attack graph. Two probability propagation functions are designed to update the probabilities for the next connected nodes. Through the pre-drawn probability attack graph, the administrator can realize all possible weak spots and attack paths in the system. Moreover, this paper provides a mapping function to map the exact alert as an evidence to the node of the attack graph to recalculate the probabilities in the attack graph and predict the most likely attacking path in the current situation.**

***Index Terms*—Attack graph, intrusion detection system, evidence mapping, network vulnerability analysis.**

I. Introduction

Intrusion detection system (IDS) plays an important role in information system security as it can raise alerts when detecting abnormal events. However, the IDS has a serious weakness; once a series attacks take place, there will be a large quantitative of alerts generated by IDSs while most of them are false and redundant. The high false positive rate will make it difficult to reveal the truth behind the scenes; therefore, it is not efficient to defense only by using IDSs. On the other hand, through the analysis of machine configurations and vulnerabilities, the attack graph can be established to model all the potential paths that the attackers may use to intrude the system. According to this kind of graph, the administrator can know 3e weak spots and further repair them in order to strengthen the system. But in some cases, vulnerabilities are inevitable, therefore to predict possible incoming attacks for a specific situation is an important issue.

It is quite efficient to adopt attack graph in prediction since it shows all attack routes. Because each path is drawn based on the vulnerabilities' characteristics and machines' configurations, so using attack graph in prediction has a high credibility. For instance, an attack may effect on UNIX system but has no impact on Windows system. Once an attacker tries to launch this attack against Windows system, even the action being defiantly failed, the IDSs still raise alerts. We believe this will bring troubles to the administrator when defending attacks. With the aid of attack graph, we can filter out the attacks that we do not need to concern about and make defending process more efficient.

In 2005, Ou *et al*. introduced a tool called "MulVAL", an open source attack graph generator, which makes the generation process easier. However, this attack graph does not concern about the corresponding occurrence probabilities for the existing evidences and thus it does not meet the prediction purpose. Many previous works have been proposed to address this limitation by appending likelihood on nodes in the attack graph. Despite the concept of probability in attack graph has been proposed in the previous literatures, there are still some drawbacks needed to be improved. Some works designed their probability values by their subjective observation, while some works failed to consider the probability propagation between nodes. Further, some previous works did not consider the probability updated when the attacks occur.

To eliminate the limitations mentioned above, we present a scheme to perform attack strategy prediction. In our work, we use MulVAL as guideline and Snort's alert as evidences. Our scheme queries the well-known vulnerability database "CVE details" and vulnerability scoring system "CVSS" (Command Vulnerability Scoring System) to obtain objective values as our initial probabilities, and the propagation function is designed to reasonably modify the probabilities in the attack graph as the evidences continually appear. Furthermore, our mapping function maps IDS alerts in the attack graph in order to update the initial probabilities.

The rest of this paper is organized as follows. The next section describes the related works. The proposed scheme is illustrated in Section 3 and the experiments are described in Section 4. Finally, Session 5 gives conclusion remarks.

II. Related Work

It will elevate the successful rate of defense if knowing enemies' tricks. Employing graphical approach can offer this kind of knowledge to the administrator. Graphical approaches works are similar to the scenario-based approach but more precise to the attack illustration. Several previous works have been proposed based on this approach, such as attack tree model [1]-[4] and attack graphs [5]-[10].

In 2005, a logical attack graph were introduced, an open source called "MulVAL". According to the input file and reasoning rule set, MulVAL can output a complete attack graph. There are two approaches to generate MulVAL input file: (1) Use the adapter to scan the system and automatically generate the input file; (2) Manually create

Manuscript received March 12, 2018; revised June 20.

Chih-Hung Wang and Ren-Wei Liou are with Department of Computer Science and Information Engineering, National Chiayi University, Chiayi, Taiwan (e-mail: wangch@mail.ncyu.edu.tw, s1040489@mail.ncyu.edu.tw).

doi: 10.18178/jacn.2018.6.1.248

the input file by the administrator. The input file describes the details of the system such as the connections between machines, existing vulnerabilities, user accounts, used protocols, etc. Since MulVAL is an open source software and easy for installation, many approaches have adopted it to carry out their developments, e.g., [11]-[14].

However, most of the attack graph models only show all possible attack paths to the administrator, and they could not tell the administrator which is most likely path to be used by the attacker for some given conditions. Some researches tried to append probabilities on the attack graph, such as [15]-[17], but some of them estimated the probabilities by their individual experiences. For instance, the scheme of [17] has applied the well-known CVSS to set up the initial probabilities; however, they only took one metric, Access Complexity, into consideration. We think the result is inaccurate since the other two metrics, Access Vector and Authentication, are also major factors for successfully exploiting the vulnerability.

It is not enough only to append probabilities on the attack graph. Some of the works allowed probability propagation between connected nodes [12], [18]-[20], and some of works applied Bayesian networks technique on the probability attack graph [15], [21], [22].

Our purpose is to employ MulVAL to generate the attack graph, and upgrade it into probability attack graph by visiting CVE database and CVSS. Furthermore, if the IDS alerts appear and can be matched to some specific nodes in the attack graph, the system updates the corresponding probabilities and performs the propagation function to make the prediction dynamically changed and keep the outputted graph up-to-date.

III. Proposed Scheme

The process of proposed scheme is shown in Fig. 1. We first improve the original attack graphs by querying CVE database and CVSS calculator to transform the attack graphs (AGs) into probability attack graphs (PAGs). In the detection part, while the IDSs found suspicious behavior traffic flows, some alerts will be raised to warn the administrator. The alerts aggregation process helps the administrator remove duplicate alerts and store the aggregated alerts in the alert database.

A. Alerts Aggregation

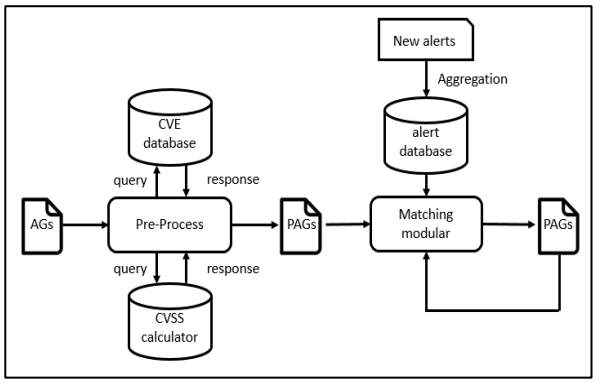

Fig. 1. The process of the proposed method.

A large amount of alerts make it difficult to analyze the attacks; in order to make the correlation process efficiency, we filter out alerts with the same pair of source-destination IP addresses, pair of source-destination port numbers, alert classification and timestamp within a pre-defined period. Then store them in a database table named "Aggregated". This process mainly filters out the redundant alerts.

B. Attack Graph Preprocess

As mentioned in previous section, the MulVAL does not concern about the probability so that it cannot meet our expectations. In this process, we show how to employ CVE database and CVSS calculator to transfer MulVAL attack graphs into probability attack graphs.

The CVSS (Command Vulnerability Scoring System) is built for measuring how dangerous the vulnerability is. The CVSS is composed of three metric groups: Base, Temporal and Environmental, and each metric group consists of several metrics. The Base metric group consists of six metrics: Access Vector *AV*, Access Complexity *AC*, Authentication *Au*, Confidentiality Impact *C*, Integrity Impact *I* and Availability Impact *A*. TABLE I describes the values of each metric.

In this paper, we take the metrics: Access Vector *AV*, Access Complexity *AC*, Authentication *Au*, in Base metric group into consideration. These three metrics represent the probabilities of a single vulnerability being able to be successfully exploited by attackers. The standard of grading is shown in TABLE II.

An example [23]. Consider CVE-2017-7269, buffer overflow vulnerability in the ScStoragePathFromUrl function. It belongs to the WebDAV service of Internet Information Services (IIS) 6.0 installed in Microsoft Windows Server 2003 R2. This vulnerability allows remote attackers to execute arbitrary code via a long header beginning with "If: <http://" in a PROPFIND request, as exploited in the wild in July or August 2016. Through the CVE database, we know the base vector for this vulnerability can be exploited from network, the level of Access Complexity is Low and the attacker is not required to be authenticated as carrying out the exploits for this vulnerability. The metrics of CVE-2017-7269 is shown in TABLE III.

TABLE I: Base Metric Group [24]

Metric	Metric symbol	Metric value	Simple description
Access Vector	*AV*	Local (L)	Vulnerability can only be exploited by local access.
		Adjacent Network (A)	Vulnerability can exploited with adjacent network access.
		Network (N)	Vulnerability can be remotely exploited through network access.
Access Complexity	*AC*	High (H)	Specialized access condition exists when exploiting vulnerability.
		Medium (M)	Specialized access condition exists with extend.
		Low (L)	No specialized condition exists.
		Multiple (M)	Attackers are required

Authentication	*Au*		to authenticate identity two or more times.
		Single (S)	Attackers are required to authenticate identity once.
		None (N)	Authentication is not required to exploit the vulnerability.
Confidentiality Impact	*C*	None (N)	No information leakage when system compromised.
		Partial (P)	There is considerable informational disclosure.
		Complete (C)	All the system files being revealed.
Integrity Impact	*I*	None (N)	There is no impact to the integrity of the system.
		Partial (P)	Some of the system file or information can be modified.
		Complete (C)	Total compromise of system integrity.
Availability Impact	*A*	None (N)	System still function.
		Partial (P)	Part of the services malfunction.
		Complete (C)	System is total shutdown.

TABLE II: CVE NODE SCORING STANDARD [24]

Metric	Evaluation	Score
Access Vector	Local (L)	0.395
	Adjacent Network (A)	0.646
	Network (N)	1
Access Complexity	High (H)	0.35
	Medium (M)	0.61
	Low (L)	0.71
Authentication	Multiple (M)	0.45
	Single (S)	0.56
	None (N)	0.704
Probability of cve node being exploited = $AV \times AC \times Au$		

TABLE III: AN EXAMPLE OF INITIAL PROBABILITY ASSIGNMENT

Metric	Evaluation	Score
Access Vector	Network (N)	1
Access Complexity	Low (L)	0.71
Authentication	None (N)	0.704
Probability of CVE-2017-7269 being exploited = 0.49984		

Some of the previous researches stop at this phase since they only append probabilities on partial nodes without considering the probability propagation, e.g., [17]. In [17], it is hard for administrator to realize the actual risk at a glance on their graph since the authors only attached probability values on some nodes.

In our work, we think the probability needs to be propagated from the current position to the root node in the attack graph. We design two propagate functions to fit our requirement.

Definition 1 [Probability Attack Graph]. An acyclic Probability Attack Graph is defined as tuple $PAG = (N, E, Pr, a)$, where:

1. $N = N_{configuration} \cup N_{reasoning\_rule} \cup N_{reasoning\_result}$. $N_{configuration}$ denotes a set of rectangle nodes (namely primitive node), $N_{reasoning\_rule}$ denotes a set of oval nodes (namely derivation node) and $N_{reasoning\_result}$ denotes a set of diamond-shaped nodes (namely derived node).
2. E represents a set of edges between nodes. $E \subseteq N \times N$. An ordered pair $(N_{pre}, N_{post}) \in E$ if $N_{pre} \mapsto N_{post}$. Further, for $N_j \in N$, the set $Pa[N_j] = \{N_i \in N | (N_i, N_j) \in E\}$ represents a parents set of N_j.
3. Pr represents the probability; it is shown by the numeric behind a colon in the annotation for a specific node.
4. a represents the matched alerts generated by IDSs.

Definition 2 [Probability Distribution]. Given a $PAG = (N, E, Pr, a)$, let $N_i \mapsto N_j$ and $Pr(N_{reasoning\_result})$ denote the probability of $N_{reasoning\_result}$, the cumulative functions are represented as follows:

1. $Pr(N_j) = \prod_{N_i \in Pa[N_j]} Pr(N_i)$

For $N_{reasoning\_result}$:

2. $Pr(N_{reasoning\_result}) = \cup_{N_i \in Pa[N_{reasoning\_result}]} Pr(N_i)$

The operation $\cup_{N_i \in Pa[N_{reasoning\_result}]} Pr(N_i)$ is denotes as:

$$1 - \prod\nolimits_{N_i \in Pa[N_{reasoning\_result}]} [1 - Pr(N_i)]$$

Fig. 2 illustrates how the cumulative functions work. In this situation, exploit e_1 and the results are placed in c_1. The attacker either exploits e_2 or e_4 and will cause results of c_2. The formula of calculating probability of each node also can be found in Fig. 2.

C. Evidence Mapping

In this section, we define a function for evidence mapping (by IDS alerts) to the specific nodes in Probability Attack Graph.

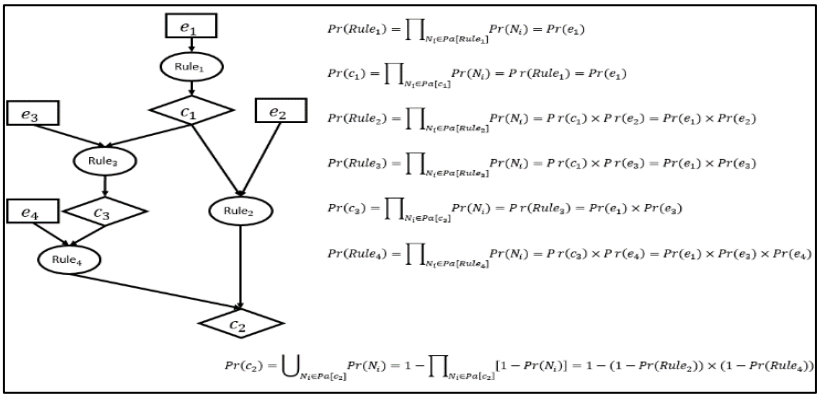

Fig. 2. Example of probability attack graph.

Definition 3 [Mapping Function]. Given a $PAG = (N, E, Pr, a)$, the followings are the major mapping processes for evidences.

1. N_{cve} represents nodes that contain CVE information, $N_{cve} \in N_{configuration}$.
2. $G\_Pa[N_j] = \{N_g \in N | (N_g, Pa[N_j]) \in E\}$ represents a set of N_j's grandparents.
3. $Sibling(N_{cve})$: $N_s \in N_{reasoning_{result}} \wedge (N_s, N_{cve}) \in Pa(N)$.
4. $host(N)$ denotes as specific host describe in the attack graph node.
5. $cveID(N)$ denotes as the CVE identity.
6. $src(a)$ denotes as the source IP address of alert.
7. $dst(a)$ denotes as the destination IP address of alert.
8. The mapping function is defined as:
   $map_a: a \mapsto \{n \in N_{cve} | \phi(a, n)\}$
   Where:
   $\phi(a, n): (dst(a) = host(n)) \wedge (cveID(a) = cveID(n)) \wedge ((src(a), dst(a) \subset G\_Pa[Sibling(n)])$

Once an alert is mapped to a specific node as an evidence, it is regarded that the vulnerability has already been exploited successfully, so that we then update the probability of the matched node into 100% and re-calculate the probabilities of related nodes. Through the updated PAG, the administrator can easily know how possible the attackers achieve their goal and what vulnerabilities they are going to take advantage for compromising the system.

IV. EXPERIMENT

To test our scheme, we use DARPA 2000 LLDOS scenario 2.0.2 dataset. DARPA 2000 is a well-known dataset to evaluate intrusion detection technique. Based on the information released on the DARPA official website and information from the CVE website "cve-detail", we construct the input file for MulVAL and generate our attack graph. However, it is difficult for us to illustrate our work through the original attack graph in this paper, and thus we decided to prune it and reserve the main multi-step attack in the scenario. In this multi-step attack, the attacker first occupies host "mill (172.16.115.20)" by exploiting vulnerability "CVE-1999-0977" for gathering more details about intranet, then he uses host mill as a stepping stone to compromise a host called "pascal (172.16.112.50)".

Fig. 3 shows the PAG after our Pre-Process phase, the probability of node is located right beside the colon, and descriptions of each node are shown in Fig. 4. Through this graph, we can realize that there are 16 ways for attacker to compromise the host "pascal" and execute code with root privilege on it, so the probability of attackers to achieve their goal is 92%. This graph is drawn before the intrusion takes place, and this will help the administrator observe the weak spots on the system and offer him/her chances to repair the system before putting the services online.

However, some vulnerabilities are inevitably. By mapping the correct alert to the exact node in PAG, our work will notice administrator which machine has already been occupied. Through the node probability, the administrator will realize which host may be the next target, and this information will offer him chances to prepare for defending the incoming attack. In TABLE IV the first column represents the probability of considerable node before alert mapping, second column represents the probability of the node after the vulnerability exploited and the corresponding alert mapped to PAG, and the third column shows the final probability after the attacker captures the host "pascal" as well as the vulnerability alert appears.

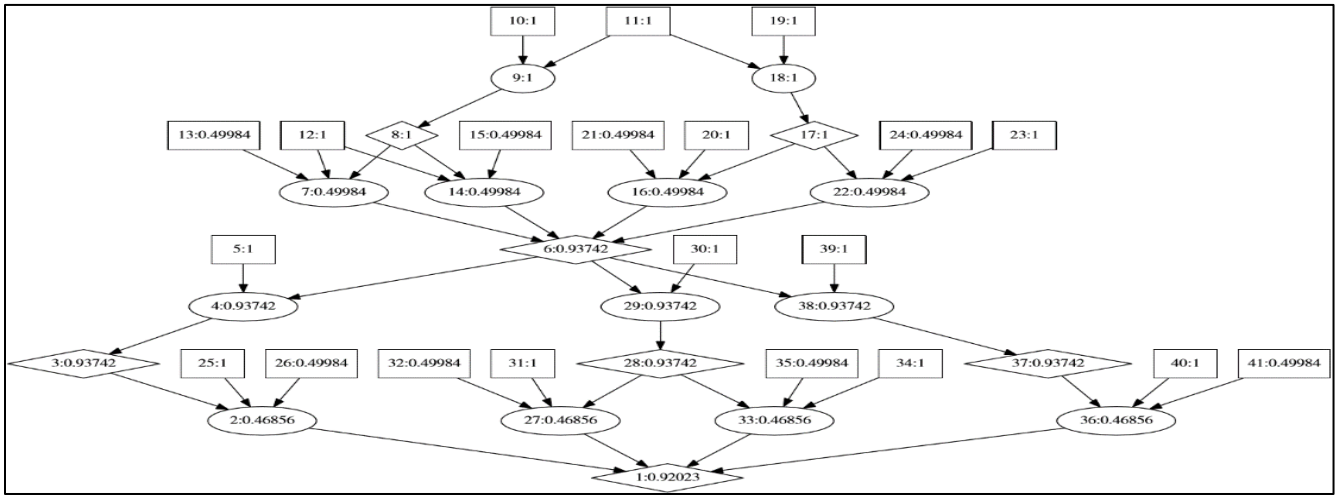

Fig. 3. Probability attack graph for partial scenario.

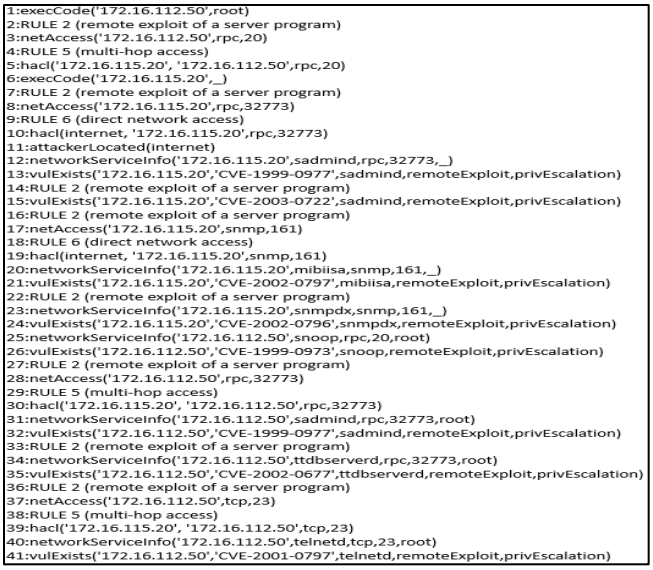

Fig. 4. Description of each nodes in PAG.

TABLE III: PROBABILITIES OF CONSIDERABLE NODES

Nodes	Original	After 'CVE-1999-0977' exploited on host "mill"	After 'CVE-1999-0977' exploited on host "pascal"
1	92.023%	93.742%	100%
2	46.856%	49.984%	49.984%
3	93.742%	100%	100%
4	93.742%	100%	100%
6	93.742%	100%	100%
7	49.984%	100%	100%
13	49.984%	100%	100%
14	49.984%	49.984%	49.984%
15	49.984%	49.984%	49.984%
16	49.984%	49.984%	49.984%
21	49.984%	49.984%	49.984%
22	49.984%	49.984%	49.984%
24	49.984%	49.984%	49.984%
26	49.984%	49.984%	49.984%
27	46.856%	49.984%	100%
28	93.742%	100%	100%
29	93.742%	100%	100%
32	49.984%	49.984%	100%
33	46.856%	49.984%	49.984%
35	49.984%	49.984%	49.984%
36	46.856%	49.984%	49.984%
37	93.742%	100%	100%
38	93.742%	100%	100%
41	49.984%	49.984%	49.984%

In Fig. 3, the nodes of 13, 15, 21 and 24 describe different vulnerabilities that can allow the attacker taking control of host "mill" while the nodes of 26, 32, 35 and 41 are the vulnerabilities that exist in the host "pascal". Since the vulnerability "CVE-1999-0977" has been exploited on the host "mill", the probability of node 13 increases from 50% to 100% that can describe exact CVE on the host "mill" and further show that the probabilities of nodes affected by the node 13 will rise as well. Similar case can be found when the attacker exploited "CVE-1999-0977" against host "pascal".

V. CONCLUSION

In this paper, we present an approach for attack strategies prediction and mapping intrusion detection alerts as evidences. We use well-known open source MulVAL to generate attack graphs and refine it by assigning credible probability value, revise the problem on probability. Furthermore, we design two functions for probability propagation that the previous works did not consider. Our mapping function can map the exact alert to the attack graph as a sign of machine occupied by someone with bad intention. Therefore, the administrator can realize the current state about their system and prepare for resisting the incoming attack.

In the future work, we are planning to map more kinds of alerts, not specified alerts containing CVE identities. This will make our attack prediction more precise, achieve alert correlation property, and help the analyst on discovering the hidden malicious activities behind the large amount of alerts.

ACKNOWLEDGMENT

This research is supported by Ministry of Science and Technology, Taiwan under the grant: MOST 106-3114-E-110-001.

REFERENCES

[1] J. Dawkins, C. Campbell, and J. Hale, "Modeling network attacks: Extending the attack tree paradigm," *Workshop Statistical Machine Learning Techniques in Computer Intrusion Detection*, 2002.

[2] A. P. Moore, R. J. Ellison, and R. C. Linger, "Attack modeling for information survivability," *Technical Note CMU/SEI-2001-TN001, Carnegie Melon Univ. / Software Eng. Inst.*, Mar. 2001.

[3] I. Ray and N. Poolsappasit, "Using attack trees to identify malicious attacks from authorized insiders," in *Proc. 10th European Symp. Research in Computer Security (ESORICS '05)*, 2005, pp. 231-246.

[4] B. Schneier, *Attack Trees*, Dr Dobb's J., Dec. 1999.

[5] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graph-based network vulnerability analysis," *Ninth Conf. Computer and Comm. Security*, 2002, pp. 217-224.

[6] S. Jajodia and S. Noel, "Topological vulnerability analysis: A powerful new approach for network attack prevention, detection, and response," *Algorithms, Architectures and Information Systems Security*, pp. 285-305, 2009.

[7] S. Mathew, R. Giomundo, S. Upadhyaya, M. Sudit, and A. Stotz, "Understanding multistage attacks by attack-track based visualization of heterogeneous event streams," in *Proc. of the 3rd International Workshop on Visualization for Computer Security*, ACM, 2006, pp. 1-6.

[8] X. Ou, S. Govindavajhala, and A. W. Appel, "MulVAL: A logicbased network security analyzer," in *14th Conf. USENIX Security Symp.*, 2005, pp. 113-128.

[9] C. Phillips and L. P. Swiler, "A graph-based system for network vulnerability analysis," *New Security Paradigms Workshop*, 1998, pp. 71-79.

[10] D. Saha, "Extending logical attack graph for efficient vulnerability analysis," in *15th ACM Conf. Computer and Comm. Security*, 2008, pp. 63-73.

[11] J. Homer, A. Varikuti, X. Ou, and M. A. McQueen, "Improving attack graph visualization through data reduction and attack grouping," *Visualization for Computer Security*, Springer, Berlin, Heidelberg, pp. 68-79, 2008.

[12] C. Liu, A. Singhal, and D. Wijesekera, "Mapping evidence graphs to attack graphs," in *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, pp. 121-126, 2012.

[13] X. Ou, W. F. Boyer, and M. A. McQueen, "A scalable approach to attack graph generation," in *13th ACM Conf. on Computer and Communications Security*, ACM, 2006, pp. 336-345.

[14] S. Roschke, F. Cheng, and C. Meinel, "A new alert correlation algorithm based on attack graph," *Computational Intelligence in Security for Information Systems*, Springer, Berlin, Heidelberg, pp. 58-67, 2011.

[15] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using bayesian attack graphs," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 61-74, 2012.

[16] J. Sembiring, M. Ramadhan, Y. S. Gondokaryono, and A. A. Arman, "Network security risk analysis using improved MulVAL Bayesian Attack graphs," *International Journal on Electrical Engineering and Informatics*, vol. 7, no. 4, 2015.

[17] A. Singhal and X. Ou, "Security risk analysis of enterprise networks using probabilistic attack graphs," *Network Security Metrics*, Springer, Cham, pp. 53-73, 2017.

[18] S. Jha, O. Sheyner, and J. Wing, "Two formal analyses of attack graphs," in *Proc.15th IEEE Computer Security Foundations Workshop. CSFW-15*, IEEE, 2002, pp. 49-63.

[19] O. Sheyner, J. Haines, and S. Jha, "Automated generation and analysis of attack graphs," in *Proc. 2002 IEEE Symposium on Security and Privacy*, IEEE, 2002, pp. 273-284.

[20] L. Wang, T. Islam, A. Singhal, and S. Jajodia, "An attack graph-based probabilistic security metric," in *Proc. IFIP Annual Conference on Data and Applications Security and Privacy*, Springer, Berlin, Heidelberg, pp. 283-296, 2008.

[21] R. Dantu, K. Loper, and P. Kolan, "Risk management using behavior based attack graphs," in *Proc. International Conference on Information Technology: Coding and Computing, 2004*. pp. 445-449.

[22] Y. Liu and H. Man, "Network vulnerability assessment using Bayesian networks," *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, International Society for Optics and Photonics*, 2005, pp. 61-72.

[23] CVE details. *The Ultimate Security Vulnerability Datasource*. [Online]. Available: https://www.cvedetails.com/

[24] NVD CVSS national vulnerability database cvss support. [Online]. Available: http://nvd.nist.gov/cvss.cfm

Chih-Hung Wang was born in Kaohsiung, Taiwan in 1968. He received the BS degree in information science from Tunghai University and MS degree in information engineering from National Chung Cheng University, Taiwan in 1991 and 1993, respectively. He received the Ph.D degree in information engineering from National Cheng Kung University, Taiwan in 1998. He is presently a professor of Department of Computer Science and Information Engineering, National Chiayi University, Taiwan. His research interests include cryptography, network security, intrusion detection and data compression.

Ren-Wei Liou was born in Taichung, Taiwan in 1992. He received the BS degree from National Ilan University and is presently a master student of Department of Computer Science and Information Engineering from National Chiayi University, Taiwan. His research interests include intrusion detection, attack graph and information security.