Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 4278632, 13 pages
https://doi.org/10.1155/2018/4278632

Research Article

New Insights into Approaches to Evaluating Intention and Path for Network Multistep Attacks

Hao Hu,[1] **Yuling Liu**,[2] **Yingjie Yang**,[3] **Hongqi Zhang**,[4] **and Yuchen Zhang**[1]

[1]*Information Science and Technology Institute, Zhengzhou 450001, China*
[2]*Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China*
[3]*Henan Key Laboratory of Information Security, Zhengzhou 450001, China*
[4]*National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450001, China*

Correspondence should be addressed to Yuling Liu; ylliu@tca.iscas.ac.cn

Received 16 November 2017; Revised 22 April 2018; Accepted 26 April 2018; Published 10 July 2018

Academic Editor: Ivan Giorgio

The attack graph (AG) is an abstraction technique that reveals the ways an attacker can use to leverage vulnerabilities in a given network to violate security policies. The analyses developed to extract security-relevant properties are referred to as AG-based security evaluations. In recent years, many evaluation approaches have been explored. However, they are generally limited to the attacker's "monotonicity" assumption, which needs further improvements to overcome the limitation. To address this issue, the stochastic mathematical model called absorbing Markov chain (AMC) is applied over the AG to give some new insights, namely, the expected success probability of attack intention (EAIP) and the expected attack path length (EAPL). Our evaluations provide the preferred mitigating target hosts and the vulnerabilities patching prioritization of middle hosts. Tests on the public datasets DARPA2000 and Defcon's CTF23 both verify that our evaluations are available and reliable.

1. Introduction

Today's information systems face sophisticated hackers who combine multiple vulnerabilities to penetrate networks with devastating impact. Most network attacks are not single attack actions. They are multistage, multihost attacks, which are composed of a series of attack actions, leading to the network security facing huge threats and challenges. Attack intention and path evaluations aim to model and measure the security-related properties of hacker breaching the enterprise network from the attacker's perspective, which allows the administrator to quantitatively estimate the overall resilience of network systems against attacks

As a nice tool for modeling multistep attacks, attack graph (AG) [1] represents possible ways in which a potential attacker can break into the target network by exploiting a series of vulnerabilities on various network hosts. When using AG-based metrics, one can analyze security-relevant properties of a network. In particular, evaluations of attack intention and path aim to analyze the vulnerability exploiting relationship among the network nodes in the AG. From the attacker's perspective, we analyze possible attack paths, identify potential attack intention, and provide an indication of critical attack paths as well as the associated weakest vulnerability links. The estimations offer constructive guidance on security reinforcement and proactive defense.

The present AG-based security metrics commonly developed based on the "monotonicity" assumption of attackers, which is firstly proposed in the 2002 ACM Conference on Computer and Communications Security (CCS) [2]. They assumed that the attackers never have to backtrack during the network penetration and each node appears exactly once in any attack path. However, the ideal attack scenario may not be the real scenario launched by the attacker. We recognize that networks commonly have many host interconnections and network privileges obtained in many ways, leading to cycles in an AG. While it is possible to "unfold" the attack graph into an acyclic representation, the approach is impractical because the dramatic increasing in the size of graph structure will likely make the path computing inefficient. To address this issue, we model the complete AG as the absorbing Markov chain (AMC) for expressing the multistep attacks

so that we can handle the cycles, and give new insights into measuring intention and path based on this model. It improves the scientificity of metric. More specifically, we give new insights into evaluations towards cyclic attack graph releasing "monotonicity" assumption with two major highlights as follows:

(i) The expected success probabilities of attack intentions (EAIPs) for the attacker to compromise different attack intention nodes are estimated.

(ii) The expected attack path lengths (EAPLs) that the attacker needs to breach different attack intention nodes from different initial state node are calculated.

The former enables a manager to determine the prioritization of vulnerability patching regarding different attack intention hosts. The latter can be devoted to giving a better understanding concerning the counter steps the attacker needs to breach the goal and can further optimize the necessary steps to harden the enterprise network from external threats as well.

The rest of this paper is organized as follows. Section 2 describes related works, which include analysis of attack intention and path evaluations, respectively. Section 3 contains a detailed presentation of the preliminaries about this paper. In addition, the model of AMC-based AG is developed in Section 4. Section 5 performs deep analyses on metrics of attack intention and path using the probability inference over AMC, and two evaluation algorithms are proposed. Section 6 describes the experiments and analyses of the proposed algorithms on two public datasets. Finally, we conclude this paper in Section 7.

2. Related Works

Network security evaluation may provide quantifiable evidence to assist security practitioners in securing computer networks, which have received significant attention in recent years. For instance, Pendleton et al. [3] designed a security metrics framework. Behi et al. [4] provided a structure for quantitation of network security and prioritization of significant security metrics. In addition, Ramos et al. [5] presented a deep survey of the state-ofthe-art of existing model-based security metric from the aspects of classifications, advantages/disadvantages, characteristics, and open research issues.

Recent works mostly focus on the usage of AG for security metrics and monitoring, which makes it easier for administrators to directly understand the attacking process. In the AG, attack paths are described by using nodes and edges to represent vulnerabilities and exploits, respectively. For attack graph-based security metric, Kantar et al. [1] made a systematical study of potential challenges and open issues of AG. One of the important works related to AG focus on modeling and core building issues. They focus on solving the scalability problem for AG generation. A large number of commercially automatic builders were designed and commonly used in large-scale attacks. These automatic builders extended the limited capability for manual construction, which was tedious, error-prone, and impractical for attack graphs when the enterprise network has a large number of nodes. With the gradual development of construction technology of AG, the application of AG in the aspect of measurement of attack intentions and paths has attracted scholars' extensive attention.

For attack intention evaluation, intention recognition is the process of deducing an invader's ultimate goal from observed actions. The rapid development of network technologies has helped network attackers to hide their malicious intentions. The conventional Intrusion Prevention System (IPS) is capable of analyzing the actions of an attacker. However, IPS cannot infer intentions and predict a series of exploits. To improve the intelligence level of IPS, Cai et al. [6] constructed an intrusion prevention method based on Weighed Planning Knowledge Graph (WPKG), which is an acyclic graph essentially. Based on alert observations, Zhu et al. [7] identified the attacker's intention using alert correlation technology. However, it fails to reduce false positive alerts. Noel et al. [8] built a predictive model of possible attack paths and critical vulnerabilities, correlating alerts to known vulnerability paths. The model suggested best courses of action for responding to attacks. Ahmed et al. [9] analyzed attack types and classified them according to their malicious intentions, further used similarity metrics to recognize attacker plans, and predicted their intentions. Concerning that attack likelihoods are propagated through the attack graph, the probabilistic AG is proposed by Ou et al. [10] to calculate the cumulative probability of attack steps in the acyclic graph. Due to the drawback of static analysis in the above methods, Nayot et al. [11] used the dynamic Bayesian Attack Graph (BAG) to represent the causal relationships between preconditions, vulnerability exploits, and postconditions. The superiority is that the proposed approach can dynamically revise the likelihood of compromising intention via encoding the attack events into BAG. Besides, Ghasemigol et al. [12] introduced a comprehensive approach that can predict future attacks with higher precision and dynamically adapt to changes in the environment.

Through the above analysis, many works have been investigated from various aspects such as alert correlation, cumulative probability, evidence theory, and Bayesian inference. Although the above reports made significant progress in security metric using attack graph, the major limitation is that they do not allow cycles in attack graphs. Moreover, existing researches focus on analyzing attackers with just one attack intention. Few investigations have been provided on sophisticated scenarios with multiple attack intentions. How to quantify the reachable probabilities of different attack intentions and further rank all intentions to find out the preferred attack intention is still essential.

For attack path evaluation, nodes and edges in the AG describe vulnerabilities and their exploits, respectively. Path evaluation aims to analyze the vulnerability exploit relationship among the network nodes. Ritchey et al. [13] developed a mathematics model to determine if an intention state is reachable from the initial state. To identify the path of one-day attack, Sun et al. [14] described a prototype system called

ZePro to generate the path by taking a probabilistic approach using the Bayesian network. Wang et al. [15] described a novel security metric for zero-day attacks by counting how many such vulnerabilities are required for compromising network assets. A modified version of Floyd–Warshall and Dijkstra algorithm is proposed by Sarraute et al. [16] to compute the shortest attack path. To explore the fast and accurate solution of finding a potential vulnerable path in the network, Wang et al. [17] used the augmented road algorithm to find optimal attack path within the global paths. To integrate the above security metrics, Idika et al. [18] presented a suite of AG-based security metrics. For instance, the normalized mean of path lengths, the median of path lengths, mode of path lengths, and standard deviation of path lengths. The advantages are that multidimensional measurements of attack paths are achieved. Overall, the above investigations mainly focused on the path metrics upon the ideal attack scenario.

While the above achievements on evaluations of intention and path are abundant and useful, most of them miss out one major issue. Major existing metrics rely on the "monotonicity" assumption, which means that an attacker never needs to relinquish any obtained capability. The assumption of "monotonicity" means that an attacker never has to backtrack, which improves the scalability of the AG, but only reflects the ideal attack scenario. Bopche et al. [19] had reported that the ideal security metrics such as the shortest path and the number of paths cannot reflect the security strength of the network accurately. Ammann et al. [2] also explained that "there are certain attacks where monotonicity does not strictly hold".

Within this assumption, all the attack scenarios can be modeled as the ideal acyclic graphs. However, the real-world attackers may not be familiar with the given network topology. We recognize that networks have many host interconnections and network privileges obtained in many ways generally, leading to cycles in AG. When calculating the path length, existing reports omit the appearances of repeated nodes in the path. On the contrary, the action of vulnerability exploitation truly happens even if the attempt fails in the realistic attack scenario. Hence, the estimate of path length in the realistic scenario may be greater.

To accurately estimate the attacker's intention and measure the path length, we borrow a stochastic mathematical model AMC [20] from the attacker's perspective in this paper. AMC has been widely used in economics, which is capable of analyzing the potential rules of state transition behaviors. Inspired by this, we recognize that multistep and multihost attacks can also be modeled and analyzed using AMC. We analyze the propagation of probabilities along attack paths in the AMC and obtain a suit of metric. In detail, we use automatic tool Multihost Multistage Vulnerability Analysis (MulVAL) [21] to generate logical AG firstly. Then, we design a normalization algorithm with respect to state transition probability and prove that any complete AG can be converted to an AMC. In addition, with the inference process of AMC, we design the relevant matrices B and T for calculating EAIP and EAPL respectively. Additionally, we present two evaluation algorithms, which provide new insights into security metrics of EAIP and EAPL. Finally, we test our algorithms on the CTF and DARPA datasets.

Table 1: Symbols and their descriptions in this paper.

Symbol	Description
AG	Attack graph
AMC	Absorbing Markov chain
S	Set of all state nodes
O	Set of all transient state nodes
G	Set of all absorbing state nodes
A	Set of all atomic attack nodes
E	Set of total edges in AG
Δ	Set of total edge probabilities
n	Number of all states
t	Number of all transient states
r	Number of all absorbing states
S_i	The i-th attack state
E^i_{out}	Set of out-going edges of S_i
$e_{i,j}$	Edge between S_i and S_j
$a_{i,j}$	Dependent atomic attack from S_i to S_j
$\Delta(e_{i,j})$	Probability of $e_{i,j}$
EAIP	Expected success probability of attack intention
EAPL	Expected attack path length
P	State transition matrix
P^m	State transition matrix after launching m steps of atomic attacks
Q	Transition matrix of transient states
R	Transition matrix from transient states to absorbing states
N	Fundamental matrix
I	Identity matrix
C	Unit vector
B	Calculation matrix of EAIP
T	Calculation matrix of EAPL
ProbRank	Ranking of absorbing states according to their EAIPs
LengRank	Ranking of transient states according to their EAPLs

3. Preliminaries

Table 1 summarizes the primary symbols in this paper.

3.1. Motivation. In the above section, we explained that the circles in the AG are the key issue in our study. It is possible to unfold any cyclic graph into an equivalent acyclic graph such that each node appears exactly once in any path. However, this procedure is not necessary if we apply a stochastic model to the cyclic nodes so that we can evaluate the same probabilities as on the unfolded graph but without actually unfolding it. The key idea is to model the complete AG as the AMC, which plays an important role in our approach.

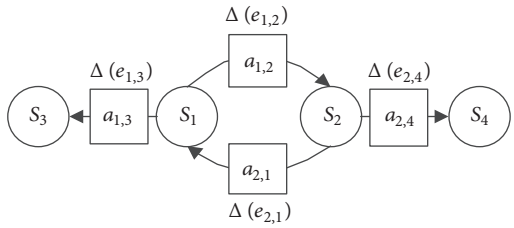

Figure 1: An example attack graph.

On the one hand, the Markov property of AMC is in line with the randomness of attack states transition. On the other hand, any network attacks have at least one ultimate state, which corresponds to the absorbing state in the AMC. Hence, we describe, analyze, and estimate the underlying rules of attack behaviors in the framework of AMC in this paper.

3.2. Attack Graph. As an effective method for modeling multistep attack behaviors, any real-world attack scenario can be abstracted as a logical AG such as Figure 1.

Definition 1. Atomic attack is a single step that cannot be broken anymore. It may be a host service scanning or an exploit of the vulnerability. Each atomic attack carries the attacker to a new attack state S.

Definition 2. Attack graph is a tuple $\mathrm{AG} = (S, E, A, \Delta)$, where S is the set of state nodes, A is the set of atomic attack nodes, E is the set of directed edges, and Δ is the set of probabilities of state transitions.

(1) $S = \{S_i \mid i = 1, 2, \ldots, n\}$ is the set of all state nodes in the AG, in which $G \subseteq S$ is the set of intention state nodes, and $O = S - G$ is the set of residual state nodes.

(2) $E \subseteq S \times S$, where the element in E is the inner product of S. $\forall e_{i,j} \in E$, $e_{i,j}$ is the edge connecting nodes S_i and S_j, where S_i is the former node of $e_{i,j}$ and S_j is the latter node of $e_{i,j}$.

(3) An atomic attack $a_{i,j} \in A$ allows an attacker to compromise the S_j from S_i with a nonzero probability of success. The attacker can reach certain privilege state by exploiting the relevant vulnerability.

(4) $\Delta(e_{i,j}) \in \Delta$ denotes the attack likelihoods propagated through the edge $e_{i,j}$. It measures the success probability associated with the atomic attack $a_{i,j}$. If the state transition $e_{i,j}$ is unreachable, assign $\Delta(e_{i,j}) = 0$.

The logical attack graph is generated using a network model builder (e.g., MulVAL, TVA, and NETSPA) by taking the network topology, services running logs, and the firewall policy as input. In this paper, we adopt MulVAL to generate the logical AG. The MulVAL is an efficient polynomial-time builder [21]. Given an attack graph, the edge probability can be computed by diverse technologies (e.g., prior domain knowledge, alert sequence, CVSS metric, and data mining). For the specific surveys, the reader can refer to [1].

Definition 3. Attack intention is a certain ultimate state that attacker wants to achieve.

Definition 4. Attack path is the transition sequence of the attacker starting from an initial state to an intention state. The number of edges equals the length of the attack path.

Definition 5. Expected success probability of attack intention (EAIP) is the mathematical expected value of the probability that the attacker can reach his intention. ProbRank denotes the ranking of intentions according to EAIPs.

Definition 6. Expected attack path length (EAPL) is the mathematical expected value of the number of attack steps the attacker needs to take from his initial state to his intention state. LengRank denotes the ranking of state nodes according to their EAPLs.

Remarks

(1) Assume that an attacker can exploit a total of two independent attack paths with length 1 and 4 to reach his intention with probability 1/6 and 1/3, respectively; then $\mathrm{EAIP} = 1/2, \mathrm{EAPL} = (1/6) \cdot 1 + (1/3) \cdot 4 = 3/2$. In other words, the attacker will reach the intention node with a sum of probability of 1/2. Meanwhile, the mean number of steps of reaching the intention is 3.

(2) The path in ideal attack scenario does not consider the repeated appearing nodes, so there is no circle in the acyclic scenario graph. The EAPL and EAIP equal the arithmetic mean of statistics.

(3) In a real-world attack scenario, the number of possible paths is uncertain due to the cyclic path. Hence, the calculation of EAPL is difficult. To the best of our knowledge, so far little researches have been devoted to this issue.

3.3. Absorbing Markov Chain

Definition 7. Markov chain (MC) [20] is a discrete sequence satisfying the following condition: MC contains a finite number of random states, and each state is only related to the predecessor state. Formally

$$p\left(x_{i+1} \mid x_i, x_{i-1}, \ldots, x_1\right) = p\left(x_{i+1} \mid x_i\right). \tag{1}$$

Definition 8. State transition matrix P [20] is an $n \times n$ adjacency matrix of MC; the (i, j) entry $p_{i,j}$ is the probability of state transition $x_i \longrightarrow x_j$. If $x_i \longrightarrow x_j$ is unreachable, assign $p_{i,j} = 0$. The matrix P satisfies

$$\sum_{j=1}^{n} p_{i,j} = 1, \quad 0 \le p_{i,j} \le 1, 1 \le i, j \le n. \tag{2}$$

Definition 9. Absorbing state is the state where the security intention is violated. This state node only has in-going edges but does not have out-going edges.

INPUT: AG = (S, A, E, Δ)
OUTPUT: The matrix P of AMC
BEGIN
(1) Initialize i, $j = 1$, $k = 0$, $E^i_{out} = \varnothing$
(2) **FOR** $i = 1$ **to** n {
(3) **FOR** $j = 1$ **to** n {
(4) Select $S_j \in S$
(5) **If** $\Delta(e_{i,j}) \neq 0$ {
(6) $k = k + 1$
(7) $e_{i,j_k} = e_{i,j}$
(8) Add e_{i,j_k} to E^i_{out} }
(9) **else** $p_{i,j} = 0$ }
(10) **FOR** $t = 1$ to k {
(11) $p_{i,j_t} = \dfrac{\Delta(e_{i,j_t})}{\Delta(e_{i,j_1}) + \ldots + \Delta(e_{i,j_k})}$ }
(12) $k = 0$ }
(13) Return P
END

Algorithm 1: Pseudocode for state transition probability normalization algorithm.

Definition 10. Transient state is the nonabsorbing state, which has at least one out-going edge.

Definition 11. Absorbing Markov chain (AMC) [20] is a special Markov chain containing at least one absorbing state. For the AMC including r absorbing states and t transient states, the standard form of the state transition matrix is

$$P = \left[\begin{array}{c|c} Q & R \\ \hline 0 & I \end{array}\right] \tag{3}$$

where Q is a $t \times t$ matrix representing transition probabilities of the transient states, 0 is a $r \times t$ zero matrix, R is a $t \times r$ matrix representing the transition probabilities between transient states and absorbing states, and I is an $r \times r$ identity matrix. The total number of states is $n = t + r$.

4. Model of AMC-Based AG

In this section, we first present a normalized algorithm for state transition probabilities in the AG and then construct the state transition matrix P of AMC. On this basis, we prove that the complete AC can be converted to the AMC.

The pseudocode describing the normalization approach for edge probabilities in AG is presented in Algorithm 1. The variables i and j label the ith row and jth column of the matrix P, respectively, where $1 \leq i, j \leq n$. The variable k labels the kth out-going edge of the node S_i. The set E^i_{out} is the set of total out-going edges of the node S_i.

We initialize the variables firstly as depicted in line (1), and then each row vector of matrix P is generated orderly. For the ith row vector of P, as shown in lines (3)–(11), it is generated according to the node S_i. More specifically, we first select all the out-going edges e_{i,j_k} of S_i and add them into the set E^i_{out}, then we calculate the normalized probability of state transition $S_i \longrightarrow S_{j_t}$ in the AMC, and assign the (i, j_t) entry of P as $\Delta(e_{i,j_t})/(\Delta(e_{i,j_1}) + \ldots + \Delta(e_{i,j_k}))$, $1 \leq t \leq k$. The residual entries in the ith row vector are assigned with 0. We perform the above recursive process until the matrix P is fully constructed.

According to the number of layers of the recursive algorithm, the time complexity is $O(n^2)$. Our algorithm needs to store $n \times n$ matrix P and $1 \times k$ vector E^i_{out}; thus the space complexity is also $O(n^2)$. Therefore, the proposed Algorithm 1 is reachable in polynomial time.

A complete attack graph contains at least one absorbing state node. The absorbing state is the attack intention node where the security goal is violated. It is possible to go to an absorbing state starting from any transient state in a finite number of steps in a complete AC. Once the attacker reaches the intention node, then the system is considered to be in a breached state and the attacker realizes his goal. Therefore, the attacker will continue to remain in this state until preventive measures are taken by the security team to remove the attacker's presence from the system. Hence, the absorbing state corresponds to the attack intention, and any complete AC contains at least one absorbing state.

A complete AG satisfies the following two conditions and is therefore an absorbing Markov chain: (i) The sum of probabilities of all out-going edges from the node S_i is equal to 1. (ii) The AG contains at least one absorbing state node. The above conditions hold as follows.

Using Algorithm 1, for any node S_i, we can obtain $\Delta(e_{i,j_1})/(\Delta(e_{i,j_1}) + \ldots + \Delta(e_{i,j_k})) + \Delta(e_{i,j_2})/(\Delta(e_{i,j_1}) + \ldots + \Delta(e_{i,j_k})) + \ldots + \Delta(e_{i,j_k})/(\Delta(e_{i,j_1}) + \ldots + \Delta(e_{i,j_k})) = 1$, which indicates that Definition 8 holds. Consequently, condition (i) holds. Since any complete AC contains at least one absorbing state, we can derive that condition (ii) holds. Accordingly, we can get that the proposed proposition holds by (1) and (2).

The following is the state transition matrix of AMC associated with Figure 1 calculated by Algorithm 1.

$$P = \left[\begin{array}{cc|cc} 0 & \dfrac{\Delta(e_{1,2})}{\Delta(e_{1,2}) + \Delta(e_{1,3})} & \dfrac{\Delta(e_{1,3})}{\Delta(e_{1,2}) + \Delta(e_{1,3})} & 0 \\ \dfrac{\Delta(e_{2,1})}{\Delta(e_{2,1}) + \Delta(e_{2,4})} & 0 & 0 & \dfrac{\Delta(e_{2,4})}{\Delta(e_{2,1}) + \Delta(e_{2,4})} \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \tag{4}$$

5. New Insights into Evaluations of Attack Intention and Path

In the previous section, we model the AG as the AMC and construct the state transition matrix of AG-based AMC. In this section, some lemmas and theorems for state transition are deduced based on this model. In addition, we present new insight into evaluations to quantify EAIP and EAPL.

5.1. Evaluation of EAIP

Lemma 12. *Let P be the transition matrix of AMC meeting Definition 11. The entry $p_{i,j}$ denotes the probability of $S_i \longrightarrow S_j$ and $p_{i,j}^m$ denotes the probability of $S_i \longrightarrow S_j$ after launching m steps of atomic attack. P^m is the mth power of matrix P. Then one has*

$$P^m = \begin{bmatrix} Q^m & \sum_{k=0}^{m-1} Q^k \cdot R \\ 0 & I \end{bmatrix}. \tag{5}$$

Proof. Using mathematical induction, we have the following:

(1) When $m = 2$, (5) holds as follows:

$$\begin{aligned} P^2 &= \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix} = \begin{bmatrix} Q \cdot Q & Q \cdot R + R \cdot I \\ 0 & I \cdot I \end{bmatrix} \\ &= \begin{bmatrix} Q^2 & \left(Q + Q^0\right) \cdot R \\ 0 & I \end{bmatrix} = \begin{bmatrix} Q^2 & \sum_{k=0}^{1} Q^k \cdot R \\ 0 & I \end{bmatrix}. \end{aligned} \tag{6}$$

(2) Assume (5) holds if $m = h - 1$; thus $P^{h-1} = \begin{bmatrix} Q^{h-1} & \sum_{k=0}^{h-2} Q^k \cdot R \\ 0 & I \end{bmatrix}$, and then we can derive

$$\begin{aligned} P^h = P^{h-1} \cdot P &= \begin{bmatrix} Q^{h-1} & \sum_{k=0}^{h-2} Q^k \cdot R \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} Q^{h-1} \cdot Q & Q^{h-1} \cdot R + \sum_{k=0}^{h-2} Q^k \cdot R \\ 0 & I^2 \end{bmatrix} \\ &= \begin{bmatrix} Q^h & \sum_{k=0}^{h-1} Q^k \cdot R \\ 0 & I \end{bmatrix}. \end{aligned} \tag{7}$$

Therefore, the assumption follows. From (1) and (2), we can obtain that Lemma 12 holds. □

Lemma 13. *The attacker will reach an ultimate absorbing state from any initial transient state if the number of attack actions is not limited. In other words, the transition probabilities between transient states are 0. Mathematically,*

$$\lim_{m \longrightarrow \infty} P^m = \begin{bmatrix} 0 & \sum_{k=0}^{m-1} Q^k \cdot R \\ 0 & I \end{bmatrix}. \tag{8}$$

Proof. By (3) and (5), if the above formula holds, then we only need to prove $\lim_{m \longrightarrow \infty} Q^m = 0$. The ijth element $Q_{i,j}^m$ of the matrix Q^m gives the probability that the Markov chain, starting in the state S_i, will be in the state S_j after m steps. Then we need to prove $\lim_{m \longrightarrow \infty} Q_{i,j}^m = 0$. It gives us an idea about the convergence of the AMC. Suppose the probability of reaching an absorbing state is nonzero; let it be u, where $0 < u \leq 1$. Then, initially, the probability that the process will not be absorbed is $(1 - u)$. After m steps, this probability is equal to $(1 - u)^m$. Note that as $m \longrightarrow \infty$, we can derive $(1 - u)^m \longrightarrow \infty$. Thus, for every transient state S_i, the probability that the attacker remains in the state S_i is 0. Hence, Lemma 13 holds. □

Lemma 14. *Given a $t \times t$ fundamental matrix N, the (i, j) entry $N_{i,j}$ denotes the expected number of visits to the transient node S_j from the initial transient node S_i before absorption. Then we can derive $N = (1 - Q)^{-1}$.*

Proof. (1) If we want to compute the expected number of steps until the chain enters a recurrent class, assuming starting at state S_i, we only need to sum $N_{i,j}$ over all transient states S_j. Thus, the sum of number of visits the chain is in the state S_j, given that the chain, starting in the state S_i, through m steps of atomic attack is $N = I + Q^1 + Q^2 + Q^3 \cdots = \sum_{m=0}^{\infty} Q^m$.

(2) As $\sum_{k=0}^{m-1} Q^k = I + Q + \ldots + Q^{m-1} = (I - Q)^{-1} \cdot (I - Q^m)$, by (5) and (8), we can derive $\lim_{m \longrightarrow \infty} Q^m = 0$; it implies that all the eigenvalues of Q have absolute values strictly less than 1. Hence, $I - Q$ is an invertible matrix. Thus, $\lim_{m \longrightarrow \infty} (I - Q^m) = I$. Furthermore, we can obtain $\sum_{m=0}^{\infty} Q^m = \lim_{m \longrightarrow \infty} \sum_{k=0}^{m-1} Q^k = (I - Q)^{-1}$.

According to (1) and (2), Lemma 14 holds. □

Theorem 15. *Given a $t \times r$ matrix B, the (i, j) entry $B_{i,j}$ denotes the EAIP of the attacker absorbing in S_j, given that the chain started in S_i, where $1 \leq i \leq t$, $1 \leq j \leq r$. Then one can derive $B = N \cdot R$.*

Proof. By Lemmas 13 and 14, we can obtain $\lim_{m \longrightarrow \infty} P^m = \begin{bmatrix} 0 & N \cdot R \\ 0 & I \end{bmatrix}$. The ith row of $N \cdot R$ gives the probabilities of ending up in each of the absorbing states, given that the chain started in the ith transient states. Hence, we can derive $B = N \cdot R$, and the theorem holds. □

To simplify the presentation, we use the attack graph in Figure 1 as an example. We manually assign $\Delta(e_{1,2}) = 3/4, \Delta(e_{1,3}) = 3/4, \Delta(e_{2,1}) = 1/2, \Delta(e_{2,4}) = 1/4$, and then we can obtain the corresponding state transition matrix P using Algorithm 1. Additionally, combined with (3), we can

construct Q and R and further calculate the matrix B as follows:

$$P = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ S_4 \end{array} \begin{array}{c} \begin{array}{cccc} S_1 & S_2 & S_3 & S_4 \end{array} \\ \left[\begin{array}{cc|cc} 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{2}{3} & 0 & 0 & \frac{1}{3} \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \end{array},$$

$$Q = \begin{array}{c} \\ S_1 \\ S_2 \end{array} \begin{array}{c} \begin{array}{cc} S_1 & S_2 \end{array} \\ \left[\begin{array}{cc} 0 & \frac{1}{2} \\ \frac{2}{3} & 0 \end{array}\right] \end{array}, \quad (9)$$

$$R = \begin{array}{c} \\ S_1 \\ S_2 \end{array} \begin{array}{c} \begin{array}{cc} S_3 & S_4 \end{array} \\ \left[\begin{array}{cc} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{array}\right] \end{array},$$

$$B = (I - Q)^{-1} \cdot R = \begin{array}{c} \\ S_1 \\ S_2 \end{array} \begin{array}{c} \begin{array}{cc} S_3 & S_4 \end{array} \\ \left[\begin{array}{cc} \frac{3}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} \end{array}\right] \end{array}.$$

From the first row of B, if the attacker started at S_1, the EAIPs of S_3 and S_4 are 3/4 and 1/4 respectively. Since the EAIP of S_3 is higher, the most likely attack intention is S_3 and the rank of intentions is ProbRank = $S_3 > S_4$. By the second row of B, if the initial attack state is S_2, it is observed that the reachable probabilities of S_3 and S_4 are equal to 1/2. Moreover, for the above two examples, we can see that the attacker will finally reach the absorbing state nodes with the definite probability 1 regardless of its initial state.

5.2. Evaluation of EAPL

Theorem 16. *Given a $t \times 1$ matrix T, where the entry T_i denotes the EAPL the attacker needs to take from the initial state node S_i to the intention state, $1 \le i \le t$, let C be a $t \times 1$ unit vector, and then one has $T = N \cdot C$.*

Proof. According to Lemma 14, for attacker starting from the initial node S_i, the visits to $S_1, S_2, \ldots, S_t$ before absorption are $N_{i,1}, N_{i,2}, \ldots, N_{i,t}$, respectively. As a result, the EAPL of S_i is $T_i = N_{i,1} + N_{i,2} + \ldots + N_{i,t}$ (equals to the sum of visits to the appearing nodes in the path). Hence, we can derive $T = [T_i] = [N_{i,1} + N_{i,2} + \ldots + N_{i,t}] = N \cdot C$ and that Theorem 16 holds. □

Going on with the example in Section 5.1, we can derive that

$$T = N \cdot C = \begin{bmatrix} \frac{3}{2} & \frac{3}{4} \\ 1 & \frac{3}{2} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{array}{c} S_1 \\ S_2 \end{array} \begin{bmatrix} \frac{9}{4} \\ \frac{5}{2} \end{bmatrix}. \quad (10)$$

Obviously, EAPLs of S_1 and S_2 are 9/4 and 5/2, respectively. Assume that the time-cost for each step attack is equal; then the penetration starting from S_1 is quicker than that from S_2. Therefore, the crucial degree of S_1 is higher, and the mitigation priorities of the nodes are LengRank = $S_1 > S_2$.

5.3. Algorithm Complexity Analysis. For the above-mentioned two algorithms, the space and time complexity are as follows:

(1) The algorithms need to maintain $n \times n$ matrix P, $t \times t$ matrix Q, $t \times t$ matrix N, $t \times r$ matrix B, $t \times 1$ matrix C, and $t \times 1$ matrix T. Since $n = r + t$, the space complexity is $O(n^2)$.

(2) The operations of the algorithm include matrix inversion, matrix addition, and matrix multiplication. Among them, the time complexity of matrix multiplication is the highest. When two $n \times n$ matrices are multiplied, there are $2n^3$ fundamental operations, so the time complexity is $O(n^3)$.

To sum up, the proposed algorithms achieve polynomial complexity.

6. Experiments and Analyses

The Defcon's Capture the Flag (CTF) contest is the largest open computer security hacking game in the world. The game is adversarial, with multiple potentially competing intentions. As demonstrated in [22], since all players are skilled in attack and defense, the attacking process is of great significance for intrusion analysis. The CTF23 dataset [23] released recently, in 2015, contains a large number of attack scenarios. The classical DARPA2000 dataset [24] by the MIT Lincoln laboratory is the standard test set for DDoS attack scenarios.

In this section, deep performances are presented based on the above two datasets by using our metrics.

6.1. CTF 23. Defcon is the largest Internet security community in the world. Defcon provides a "Capture the Flag" (CTF) contest, which is a contest of computer security attack and defense skills, as shown in Figure 2. CTF attracts several expert intruders with a legal opportunity to deliver their skills in a public forum. Each team has to defend its own flag, while trying to corrupt as many of the other teams' flags as possible. A flag is a data file on the team's server.

During the game, intruders seek to replace the flag on someone else's server with their own flag, while defenders try to preserve their flags on their own server. Defcon has recently published the archive of CTF23, which includes all the traffic generation during the game. The size of CTF23 is more than hundreds of GB. There are many attackers starting from different initial states with different attack intentions in attack scenarios.

Firstly, we use TCPReplay tool [25] to replay the CTF23 dataset and detect the alert data by Snort. By extracting the alert sequences using the analyzer ArcSight [26], we

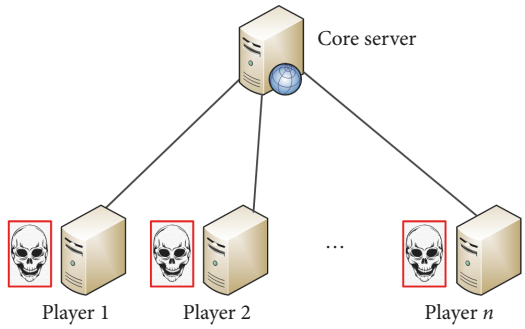

Figure 2: Defcon's Capture the Flag contest network.

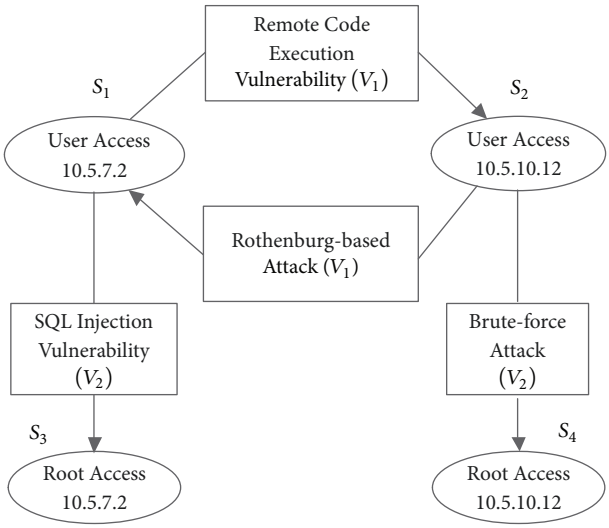

Figure 3: A simple attack scenarios extracted from CTF23.

reconstruct more than 80 different attack scenarios. Among them, the minimum attack scenario only has 3 nodes, and the maximum attack scenario has no more than 20 nodes. Then we select a familiar attack scenario depicted in Figure 3 as our test scenario. Although the attack process is simple, it helps to analyze the impact of single attack step over the whole network.

The set of transient states is $O = \{S_1, S_2\}$ and the set of the intention states is $G = \{S_3, S_4\}$. Assume that the attacker's initial state is S_1, and his intention is S_4; one possible attack path is $S_1 \longrightarrow S_2 \longrightarrow S_4$. The description of this path is as follows: The intruder (IP: 10.31.10.8) breaks into the server with IP address 10.31.1.2 using remote code execution vulnerability and gets its user privilege. After launching a brute-force attack to log in the MySQL server, the attacker can obtain the root privilege finally.

To simplify the discussion, we give some enumerations and summarize the relationships between attack path and edge probabilities. Suppose that the success probabilities of "remote code execution" and "Rothenburg-based attack" are the same and equal to V_1, the success probabilities of "SQL Injection" and "brute-force" are equal to V_2, where $V_1, V_2 \in (0, 1]$. We first construct the state transition matrix P using Algorithm 1. Afterwards, the EAIP matrix B and EAPL matrix T can be calculated using Algorithms 2 and 3, respectively, as follows:

$$P = \begin{bmatrix} 0 & \frac{V_1}{V_1+V_2} & \frac{V_2}{V_1+V_2} & 0 \\ \frac{V_1}{V_1+V_2} & 0 & 0 & \frac{V_2}{V_1+V_2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$B = \begin{matrix} S_1 \\ S_2 \end{matrix} \begin{bmatrix} \frac{V_1+V_2}{2V_1+V_2} & \frac{V_1}{2V_1+V_2} \\ \frac{V_1}{2V_1+V_2} & \frac{V_1+V_2}{2V_1+V_2} \end{bmatrix}, \tag{11}$$

$$T = \begin{matrix} S_1 \\ S_2 \end{matrix} \begin{bmatrix} 1+\frac{V_1}{V_2} \\ 1+\frac{V_1}{V_2} \end{bmatrix}.$$

6.1.1. Evaluation of EAIP. From the first row vector of matrix B, we can obtain that if the initial states is S_1, the EAIPs of S_3 and S_4 are $B_{1,1} = (V_1 + V_2)/(2V_1 + V_2)$ and $B_{1,2} = V_1/(2V_1 + V_2)$, respectively. Some enumerations of EAIP calculation are organized in Table 2. The corresponding 3D figure of V_1, V_2, and EAIP is illustrated in Figure 4. The EAIP of S_3 is higher than 0.5, and the EAIP of S_4 is lower than 0.5. Hence, the attacker is more likely to compromise the node S_3. We can identify that the possible attack intention is S_3. Meantime, the intentions rank is ProbRank $= S_3 > S_4$. It means that the first suggested vulnerability to patch is SQL Injection Vulnerability on the server 10.5.7.2. Moreover, we can observe that the value of EAIP(S_3) − EAIP(S_4) increases as the parameter V_2/V_1 increases.

Affected by the attacker's own characteristics (e.g., knowledge level, professional skills, and attack experience), a different attacker has a different success probability on the vulnerability. In practical application, the probability can be estimated by the attacker's historical security events. Afterwards, the EAIPs of different intentions can be calculated using Algorithm 1. The security engineer can take precautions for the preferred attack intention nodes.

6.1.2. Evaluation of EAPL. From the matrix T, we observe that the EAPLs of S_1 and S_2 are both $1 + V_1/V_2$. Therefore, the security rank of the transient nodes is LengRank $= S_1 = S_2$.

Some enumerations of EAPL are listed in Table 3. The 3D figure of EAPL is illustrated in Figure 5. There is a relatively big difference of EAPL with different V_1 and V_2. For example, from Table 3, when $V_1 = 0.8, V_2 = 0.2$, EAPL $= 5$. On the contrary, when $V_1 = 0.2, V_2 = 0.8$, EAPL $= 1.25$. Therefore, the probabilities of V_1 and V_2 have significant impact on the number of steps the attacker needs to reach his intention. Meanwhile, from the formula of $1 + V_1/V_2$, the value of EAPL increases as V_1/V_2 increases. And the minimum EAPL $= 1$ when $V_1 = 0$. In particular, we can obtain EAPL $= 2$ when $V_1 = V_2$, indicating that the attacker should perform an average of two atomic attacks to achieve his intention.

INPUT: AG = (S, E, A, Δ)
OUTPUT: Matrix B, ProbRank
BEGIN
(1) Use **Algorithm** 1 to construct the matrix P.
(2) Construct $t \times t$ matrix Q, $t \times r$ matrix R from matrix P according to Definition 11
(3) Calculate $t \times r$ matrix $B = (I - Q)^{-1} \cdot R$
(4) **FOR** $i = 1$ to r {
(5) Rank $B_{i,1}, B_{i,2}, \ldots, B_{i,r}$ in descending order and record as ProbRank }
(6) Return B and ProbRank
END

Algorithm 2: Pseudocode for EAIP evaluation algorithm.

INPUT: AG = (S, A, E, Δ)
OUTPUT: Matrix T, LengRank
BEGIN
(1) Use **Algorithm** 1 to construct the matrix P
(2) Construct $t \times t$ matrix Q, $t \times r$matrix R from P by Definition 11
(3) Construct $t \times 1$ matrix $C = [1, 1, \ldots, 1]^{\mathrm{T}}$
(4) Calculate $t \times 1$ matrix $T = (I - Q)^{-1} \cdot C$
(5) **FOR** $i = 1$ to t {
(6) Rank $T_1, T_2, \ldots, T_t$ in in descending order and record as LengRank }
(7) Return T and LengRank
END

Algorithm 3: Pseudocode for EAPL evaluation algorithm.

One can make use of EAPL to quantify the average number of atomic attacks for the attacker to achieve his intention. During the application, by analyzing the ongoing attack events, the security manager can locate the current state of the attacker. Through calculating the EAIPs of the attacker's different intentions, one can also identify the attacker's preferred target. This information is valuable for a security engineer to prioritize which intention node needs to be patched first and how it will affect the strength of the network against attacks.

6.2. DARPA 2000. The DARPA 2000 is a classical dataset for DDoS attack analysis, which is a commonly acknowledged. Similarly, we first use the TCPReplay tool to replay DARPA2000 dataset. Then with the automatic AG generation tool, we constructed the attack scenario graph of LLDOS1.0 in Figure 6. For the edge probabilities, we borrow the results in [7], where

$$\begin{aligned} e_{2,5} &= 0.15, \\ e_{2,3} &= 0.28, \\ e_{3,5} &= 0.1, \\ e_{1,2} &= 0.29, \\ e_{2,2} &= 0.2 \\ e_{2,4} &= 0.28, \\ e_{4,2} &= 0.18, \\ e_{4,3} &= 0.42, \\ e_{4,5} &= 0.15, \\ e_{5,5} &= 1 \\ e_{1,4} &= 0.33, \\ e_{4,4} &= 0.18. \end{aligned} \tag{12}$$

Firstly, we use Algorithm 1 to generate the state transition matrix P of AMC and then perform Algorithms 2 and 3 to calculate the matrices B and T as follows:

$$P = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{array} \begin{array}{c} \begin{array}{ccccc} S_1 & S_2 & S_3 & S_4 & S_5 \end{array} \\ \left[\begin{array}{ccccc} 0 & 0.47 & 0 & 0.53 & 0 \\ 0 & 0.22 & 0.31 & 0.31 & 0.16 \\ 0 & 0 & 0.87 & 0 & 0.13 \\ 0 & 0.19 & 0.45 & 0.19 & 0.17 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array},$$

$$B = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ S_4 \end{array} \begin{array}{c} \\ \left[\begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \end{array}\right] \end{array},$$

Table 2: Enumerations of EAIPs in Figure 3.

V_1	V_2	EAIP(S_3)	EAIP(S_4)	V_1	V_2	EAIP(S_3)	*EAIP*(S_4)
0.2	0.2	0.67	0.33	0.4	0.2	0.6	0.4
0.2	0.4	0.75	0.25	0.4	0.4	0.67	0.33
0.2	0.6	0.8	0.2	0.4	0.6	0.71	0.29
0.2	0.8	0.83	0.17	0.4	0.8	0.75	0.25
0.2	1	0.86	0.14	0.4	1	0.78	0.22
...	...	...	...	...	...	...	...
V_1	V_2	EAIP(S_3)	EAIP(S_4)	V_1	V_2	EAIP(S_3)	EAIP(S_4)
0.6	0.2	0.57	0.43	0.8	0.2	0.56	0.44
0.6	0.4	0.63	0.37	0.8	0.4	0.6	0.4
0.6	0.6	0.67	0.33	0.8	0.6	0.64	0.36
0.6	0.8	0.7	0.3	0.8	0.8	0.67	0.33
0.6	1	0.73	0.27	0.8	1	0.69	0.31
...	...	...	...	...	...	...	...

Table 3: Enumerations of EAPLs in Figure 3.

V_1	V_2	EAPL	V_1	V_2	EAPL
0.2	0.2	2	0.4	0.2	3
0.2	0.4	1.5	0.4	0.4	2
0.2	0.6	1.33	0.4	0.6	1.67
0.2	0.8	1.25	0.4	0.8	1.5
0.2	1	1.2	0.4	1	1.4
...	...	...	...	...	...
V_1	V_2	EAPL	V_1	V_2	EAPL
0.6	0.2	4	0.8	0.2	5
0.6	0.4	2.5	0.8	0.4	3
0.6	0.6	2	0.8	0.6	2.33
0.6	0.8	1.75	0.8	0.8	2
0.6	1	1.6	0.8	1	1.8
...	...	...	...	...	...

$$T = \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{matrix} \begin{bmatrix} 8.198 \\ 7.200 \\ 7.692 \\ 7.197 \end{bmatrix}. \tag{13}$$

In this above scenario, the transient state set is $O = \{S_1, S_2, S_3, S_4\}$ and the absorbing state set is $G = \{S_5\}$.

According to the matrix B, the attacker will definitively reach the intention node S_5 with the probability 1 regardless of initial state. By the matrix T, we can derive that, for different initial state nodes S_1, S_2, S_3, and S_4, the calculated EAPLs to S_5 are 8.198, 7.200, 7.692, and 7.197, respectively. Hence, the security rank of the middle nodes in the attack path is LengRank = $S_1 > S_3 > S_2 > S_4$.

The security metrics under ideal and realistic scenarios in Figure 6 are presented in Table 4. Some conclusions can be summarized as follows:

(i) For the ideal scenario, each node appears exactly once in any attack path, and the attacker never has to backtrack during the network penetration, Therefore, there is no circles in ideal path, and the number of total paths is 8. Among them, the shortest paths are $S_1 \longrightarrow S_2 \longrightarrow S_5$ and $S_1 \longrightarrow S_4 \longrightarrow S_5$ with length 2. The mode of path lengths is 3, and the median of path lengths is 3. The mean path length

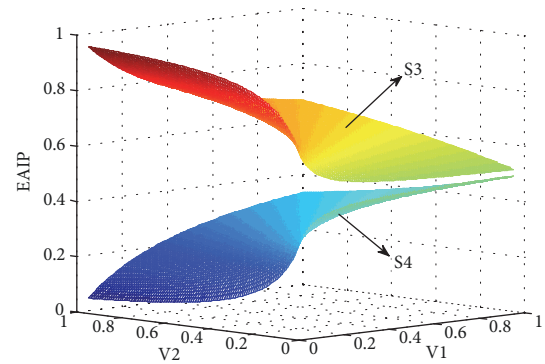

Figure 4: Distribution of EAIPs in Figure 3.

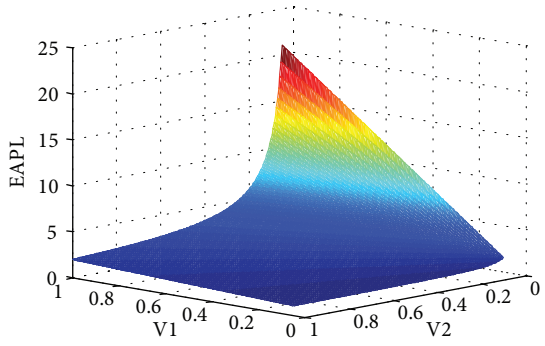

Figure 5: Distribution of EAPLs in Figure 3.

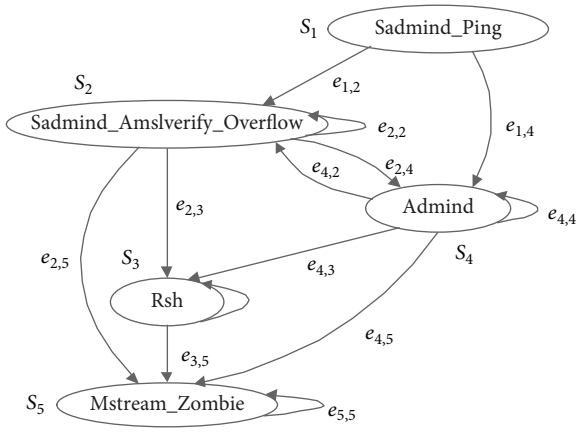

Figure 6: LLDOS1.0 attack scenario.

Table 4: Structural measurements of LLDOS1.0 attack scenario in Figure 6.

Types	Measurement	Value
Ideal attack scenario	Shortest path length	2
	Median of path lengths	3
	Mean path length	3
	Number of paths	8
	Mode of path lengths	3
	Success probability of intention	0.14
Realistic attack scenario	EAIP of intention S_5	1
	Maximum EAPL	8.198
	Minimum EAPL	7.197

is 3. The ideal success probability of S_5 is 0.14, which is the cumulative probabilities of all the paths under the "monotonicity" assumption, which indicates that each step of the attack launched in the path is indeed successful.

(ii) For the realistic scenario, even a failed action of state transition is still handled as an occurrence of the atomic attack. Since the number of atomic attacks is not limited in actual practice, the occurrence probability of S_5 is 1, which is bigger than that of 0.14 in the ideal attack scenario. Moreover, the maximum EAPL is 8.2 when the attacker's initial state is S_1, and the minimum EAPL is 7.197 when the attacker's initial state is S_4. The maximum 8.2 indicates that the attacker needs to launch an average of 8.2 atomic attacks to breach his intention. However, in the ideal attack scenario, the result is 3 since the repeated nodes appearing in the path are excluded.

6.3. Comparisons and Discussions. The detailed comparisons of security metrics among ours and other related methods are summarized in Table 5. The security parameters under the ideal and realistic attack scenarios are fully analyzed. As explained in paragraph 3 of Section 1, the metrics under the ideal attack scenario is limited by the "monotonicity" assumption of attackers. Therefore, new insights into EAIP and EAPL in the cyclic graph under realistic attack scenario are provided for the first time in this paper. The major merits are as follows:

(i) For the ideal attack scenario, the common metrics involve the most likely attack path [17–19], success probability of intention [10, 16, 18, 19], and the number of attack paths [16–19], which are investigated in the acyclic graph. Furthermore, [18, 19] further analyzed the mean, median, and mode of path lengths. All these metrics do not take into account the circles and instead assume that each node appears exactly once in any attack path.

(ii) For the realistic attack scenario, two new insights into security metrics are given by using AMC to handle the circles in the cyclic graph. First, we investigate the EAIPs of different intentions, which helps to identify the most possible attack target and determine the priorities of critical destination hosts. Second, by locating the current state of the attacker, we can calculate EAPL. The measurement of EAPL enables the administrator to comprehend a reliability number of attack steps that the attacker needs to complete his goal and make the appropriate protection decisions.

To conclude, our new insights provide more accurate and reliable quantification into the security metrics of multistep attacks.

7. Conclusion and Future Work

Quantifying security with metrics is important since we want to have a scoring system to evaluate the strength of the security. Although many investigations have been made, they are more or less subjected to the attacker's "monotonicity" assumption. To overcome the limitations, we employ a mathematical model AMC to handle the circles in AG and present the realistic metrics for calculating EAIP and EAPL. In addition, we aggregate existing approaches and give a suite of evaluation methods for both ideal and realistic attack scenario towards multistep attacks. Experiments verify the validity and accuracy of the proposed model and algorithms.

Large-scale networks are mostly generated by integrating multiple small-scale local networks. Due to the scalability of

Table 5: Comparisons of security metrics among our method and others.

Types	Ref. [10]	Ref. [16]	Ref. [17]	Ref. [18]	Ref. [19]	Ours
Most likely attack path			√	√	√	√
Cumulative success probability of intention	√	√		√	√	√
Number of attack paths		√	√	√	√	√
Mean of path lengths				√	√	√
Median of path lengths				√	√	√
Mode of path lengths				√	√	√
EAIP of intention						√
EAPL of breaching intention						√

security attacks, our metrics can be extended to the large-scale network systems. Trying to capture the large dataset of the enterprise to test the scalability of the proposed methodology is the future work. Meanwhile, the emphasis is on measuring the state transition probability based on the observed alerts from IDS, firewall, system logs, etc. so that we can improve the flexibility and applicability in a further step.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Key Research and Development Program of China (Grants nos. 2016YFF0204002 and 2016YFF0204003), the Equipment Pre-Research Foundation during the 13th Five-Year Plan period (Grant no. 6140002020115), the CCF-Venus "Hongyan" Scientific Research Plan Foundation (Grant no. 2017003), and the Science and Technology Leading Talent Project of Zhengzhou (Grant no. 131PLJRC644).

References

[1] K. Kaynar, "A taxonomy for attack graph generation and usage in network security," *Journal of Information Security and Applications*, vol. 29, pp. 27–56, 2016.

[2] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graph-based network vulnerability analysis," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 217–224, ACM, Washington, DC, USA, November 2002.

[3] M. Pendleton, R. Garcia-Lebron, J.-H. Cho, and S. Xu, "A survey on systems security metrics," *ACM Computing Surveys*, vol. 49, no. 4, article no. 62, 2016.

[4] M. Behi, M. GhasemiGol, and H. Vahdat-Nejad, "A new approach to quantify network security by ranking of security metrics and considering their relationships," *International Journal of Network Security*, vol. 20, no. 1, pp. 141–148, 2018.

[5] A. Ramos, M. Lazar, and R. H. Filho, "Model-based quantitative network security metrics: a survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2704–2734, 2017.

[6] Z. Cai, Q. Zhang, and Y. Gan, "Intrusion intention recognition and response based on weighed plan knowledge graph," *Computer Modeling New Technologies*, vol. 18, no. 12B, pp. 151–157, 2014.

[7] B. Zhu and A. A. Ghorbani, "Alert correlation for extracting attack strategies," *International Journal of Network Security*, vol. 3, no. 3, pp. 244–258, 2006.

[8] S. Noel, E. Harley, K. H. Tam, M. Limiero, and M. Share, "Cygraph: graph-based analytics and visualization for cybersecurity," *Handbook of Statistics*, vol. 35, pp. 117–167, 2016.

[9] A. A. Ahmed, "Investigation approach for network attack intention recognition," *International Journal of Digital Crime and Forensics*, vol. 9, no. 1, pp. 17–38, 2017.

[10] X. Ou and A. Singhal, "Security risk analysis of enterprise networks using probabilistic attack graphs," *National Institute of Standards and Technology*, pp. 13–23, 2012.

[11] 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.

[12] M. Ghasemigol, A. Ghaemi-Bafghi, and H. Takabi, "A comprehensive approach for network attack forecasting," *Computers & Security*, vol. 58, pp. 83–105, 2016.

[13] R. W. Ritchey and P. Ammann, "Using model checking to analyze network vulnerabilities," *2000 IEEE Symposium on Security and Privacy*, pp. 156–165, 2000.

[14] L. Wang, S. Jajodia, and A. Singhal, *Using Bayesian Networks to Fuse Intrusion Evidences And Detect Zero-Day Attack Paths*, Network Security Metrics, Springer International Publishing, Cham, Switzerland, 2017.

[15] L. Wang, S. Jajodia, and A. Singhal, *K-Zero Day Safety: Evaluating The Resilience of Networks against Unknown Attacks*, Network Security Metrics, Springer International Publishing, Cham, Switzerland, 2017.

[16] C. Sarraute, G. Richarte, and J. Lucángeli Obes, "An algorithm to find optimal attack paths in nondeterministic scenarios," in *Proceedings of the ACM workshop on security and artificial intelligence, (AISec '11)*, pp. 71–80, Chicago, Ill, USA, October 2011.

[17] H. Wang, Z. Chen, J. Zhao, X. Di, and D. Liu, "A vulnerability assessment method in industrial internet of things based on attack graph and maximum flow," *IEEE Access*, vol. 6, pp. 8599–8609, 2018.

[18] N. Idika and B. Bhargava, "Extending attack graph-based security metrics and aggregating their application," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 75–85, 2012.

[19] G. S. Bopche and B. M. Mehtre, "Graph similarity metrics for assessing temporal changes in attack surface of dynamic networks," *Computers & Security*, vol. 64, pp. 16–43, 2017.

[20] G. F. Lawler, *Introduction to Stochastic Processes*, Chapman and Hall/CRC, Taylor and Francis Group, London, UK, New York, NY, USA, 2nd edition, 2006.

[21] X. Ou, S. Govindavajhala, and A. W. Appel, "MulVAL: a logic-based network security analyzer," in *Proceeding of the 14th conference on USENIX Security Symposium*, vol. 14, p. 8, 2005.

[22] H. Huang, J. Ding, and W. Zhang, "A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation, ICRA 2011*, pp. 1451–1456, China, May 2011.

[23] DEFCON, "Capture the flag traffic dump," http://www.defcon.org/html/links/dc-cft.html.

[24] MIT Lincoln Lab, "2000 DARPA intrusion detection scenario specific datasets," http://ll.mit.edu/IST/ideval/data/2000/2000_data_index.html.

[25] MIT Lincoln Lab, "TCPdump file replay utility," http://ideval.ll.mit.edu/IST/ideval/tools/tools index.html.

[26] ArcSight, "ESM enterprise security manager," http://www8.hp.com/us/en/software/enterprise-software.html.