Hindawi
Security and Communication Networks
Volume 2019, Article ID 2031063, 16 pages
https://doi.org/10.1155/2019/2031063

Review Article

Survey of Attack Graph Analysis Methods from the Perspective of Data and Knowledge Processing

Jianping Zeng,[1,2] Shuang Wu,[1,2] Yanyu Chen,[1,2] Rui Zeng,[3] and Chengrong Wu[1,2]

[1]School of Computer Science, Fudan University, Shanghai 200433, China
[2]Engineering Research Center of Cyber Security Auditing and Monitoring, Ministry of Education, Shanghai 200433, China
[3]School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China

Correspondence should be addressed to Jianping Zeng; zjp@fudan.edu.cn

Received 1 October 2019; Revised 10 November 2019; Accepted 4 December 2019; Published 26 December 2019

Academic Editor: Pino Caballero-Gil

Attack graph can simulate the possible paths used by attackers to invade the network. By using the attack graph, the administrator can evaluate the security of the network and analyze and predict the behavior of the attacker. Although there are many research studies on attack graph, there is no systematic survey for the related analysis methods. This paper firstly introduces the basic concepts, generation methods, and computing tasks of the attack graph, and then, several kinds of analysis methods of attack graph, namely, graph-based method, Bayesian network-based method, Markov model-based method, cost optimization method, and uncertainty analysis method, are described in detail. Finally, comparative study of the methods and future work are provided. We believe that this work would help the research community to understand the attack graph analysis method systematically.

1. Introduction

Network security breach has become a potential danger that limits the further development of network applications. According to the "China Cyber security Report for the First Half of 2017," the number of port scan on MySQL and SQL Server in China was ranked first in the world [1]. Usually, network scanning is the first step to implement network attack; hence, we can be aware of the severe network security situation in China. Cyber attacks are also fatal for global enterprise networks, for example, the new ransomware virus Petya attacked several known organizations, such as the Ukrainian capital airport, Russian oil and gas giant Rosneft, American pharmaceutical company Merck, and so on. Because the virus can prevent the machine from booting properly to make the computer unusable, the network security issues greatly affect the normal operations of enterprise network systems. To solve such problem, it is important to analyze the networks and find out the weak nodes for security hardening.

There are many kinds of models for cyber attack evaluation, such as attack tree, Petri net, and attack graph. The attack graph model proposed by Swiler et al. [2] in 1997 has stronger ability in the description of network attack process. Hence, it becomes one of the most widely used tools for solving network security problems. When attackers launch network penetration, they usually start from gaining privilege to a node by exploiting vulnerabilities in the network, then gradually infiltrate into other nodes, and eventually reach the target node and obtain the required information. Therefore, an attack path from the initial node to the target node can be used to describe the attacker's specific attack behaviors. Since the network topology itself is of graph-based structure, nodes and attack paths can be represented by means of graphs. The attack graph model is designed to describe the abstracted network topology with a directed acyclic graph and to show the nodes, paths, and consequences of network attack. Each node in the attack graph can stand for host, vulnerability, or network device, according to different attack graph representation methods. The edge from node A to B indicates that from A, the attacker can reach node B. Thus, attack graph is similar to the network structure and can simulate attacker's attack steps. At the same time, there are many mathematical models that can formally represent and analyze the simulation. Thus, complex connections, vulnerabilities, and attack paths can be integrated together by modeling enterprise-level network as attack graph. With the help of various attack graph formal analysis methods, discovery of potential security problems become easy; hence, the attack graph plays a crucial role in network security analysis.

Since the proposal of the attack graph model, it has received extensive attention from the academic community. The methods in node representation, graph generation, and formal mathematical analysis have made great progress. In the previous review work, Kaynar [3] conducted a comprehensive analysis and comparison of attack graph generation, vulnerability classification, and application, Ye et al. and Shandilya et al. summarized the application scenarios [4, 5], and Chen et al. surveyed the attack graph generation methods [6]. The review work of the attack graph model is a summary of the outstanding research work at that time. Compared with these studies, the main contributions of this paper are as follows:

> Firstly, *the paper focuses on the analysis methods of the attack graph.* Previous review work mainly concentrated on the generation method and applications of the attack graph, which involves little about the analysis methods. Attack graph is a representation of network topology, and how to convert it into formal models is important for the actual applications. Although there exist many analysis methods, they are ignored by current survey work.
>
> Secondly, *this paper classifies the attack graph analysis methods according to the differences of research ideas*, so it can provide valuable reference for selecting appropriate analysis methods. Previous reviews mainly classified attack graph based on the node presentation methods, and they emphasized more on the representation of attack graphs rather than the difference between analysis methods.
>
> Finally, *this paper summarizes the uncertainty analysis methods of attack graphs.* The uncertainty in network attack stems from network structure, behavior of attack, and so on. For example, mobile devices frequently connect and disconnect with the network and thus lead to the connections varying a lot in the graph. The kind of uncertainty leads to great difficulty in dealing with network security, and thus the uncertainty analysis of the attack graph is an unavoidable problem. However, there is no systematic approach for this problem, and it is usually ignored in the existing review work.

This paper is organized as follows. The basic concepts, generation methods, and tasks of attack graphs are described in the next section. The attack graph analysis models and algorithms, including the graph algorithms, Bayesian and Markov model, cost-optimized analysis method, and uncertainty analysis method, are described in detail in the following sections. In the final section, the various analysis methods are compared in terms of advantages and disadvantages, and future research directions are pointed out.

2. Research Framework of Attack Graphs

The research framework of attack graphs is shown in Figure 1.

As can be seen from Figure 1, information about network topology, vulnerability, network configuration, and network connectivity should be firstly collected. Then, the information is used to generate the attack graph which can be visualized. According to the graph definition and requirements on security analysis, the graph should be described in mathematical formulation so that quantity analysis can be performed on nodes, edges, and attack path. Finally, the analysis results can provide a basis for various attack graph applications. The attack graph analysis method is the key to attack graph research and applications. Hence, we concentrate on the module of the "analysis method" in the framework.

2.1. Example of Attack Graph. Attack graphs are designed to represent the abstracted network topology with a directed acyclic graph. One of the main application scenarios is for network vulnerability analysis. The vertices of attack graphs can be related elements such as host, authority, vulnerability, service, and even some network security status, depending on the attack behavior analysis requirements. Unlike the diversity of vertex, the edges in attack graphs generally indicate the perpetration of attacks. As an example shown in Figure 2, we use the topology of a web Internet network presented in [7]. The network consists of three subnetworks, that is, the Internet, the DMZ (demilitarized zone), and the trust zone. The DMZ contains a DNS server and a web server. There are three servers in the trust zone, that is, the FTP server, database server, and administrative server.

The vulnerabilities on each server are listed in Table 1, and the communication rules between servers are presented in Table 2. Note that "CVE ID" is the vulnerability's identification in "Common Vulnerabilities and Exposures" library. Figure 3 shows the corresponding attack graph based on the network topology, vulnerabilities, and connections between the servers.

2.2. Attack Graph Generation Method. Attack graph generation generally contains three steps, that is, reachability analysis, attack template establishment, and attack graph construction [3]. For large-scale attack graphs, reducing the complexity of attack graph is necessary, and corresponding methods include path pruning, network properties compression, and property matching time reducing. Several tools are able to generate the graph automatically.

Sheyner et al. developed an attack graph generation tool [8], which is the first-generation product based on model checking technique. It takes the host state, state transition probability, and security attributes as inputs. The output is an attack graph containing paths that violate security attributes.

MulVAL (http://people.cs.ksu.edu/~xou/argus/software/mulval/readme.html) is a Linux-based attack graph

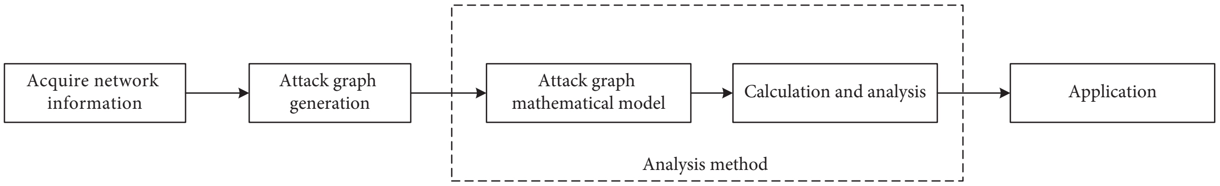


Figure 1: Attack graph research framework.

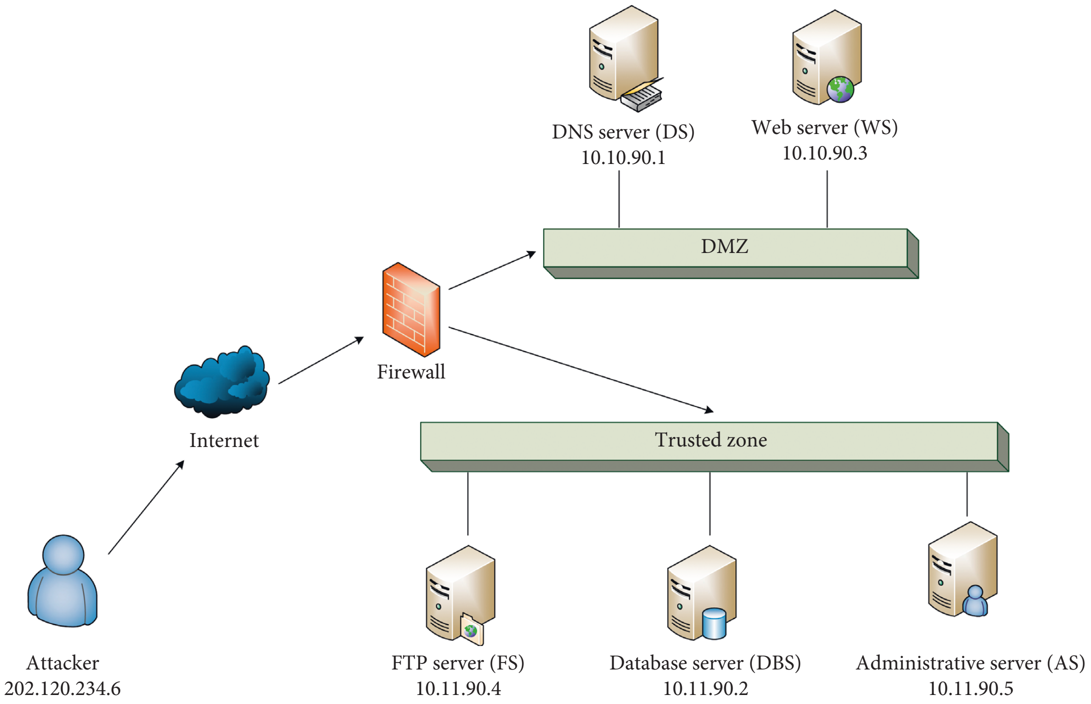


Figure 2: Example of network topology.

Table 1: Vulnerabilities on each server.

Server	Vulnerabilities	CVE ID
WS	Allow remote execution of code	CVE-2015-1635
DBS	Remote execution of SQL command	CVE-2014-1466
FS	Allow remote execution of code	CVE-2013-4465
	Allow remote execution of code	CVE-2012-2526
AS	Allow remote execution of code	CVE-2009-0241

Table 2: Communication rules between servers.

Source server	Destination server	Protocol and port
202.120.234.6	WS	HTTP (80)
WS	DBS	SQL (1433)
AS	DBS	SQL (1433)
	FS	FTP (21)
	WS	HTTP (80)
	DS	DNS (1024)

autogeneration tool introduced by Ou et al. [9]. It uses Prolog logic language to formally describe the configuration and vulnerability of nodes, then infers the entire attack process to generate attack paths, and uses Graphviz to draw the attack graph.

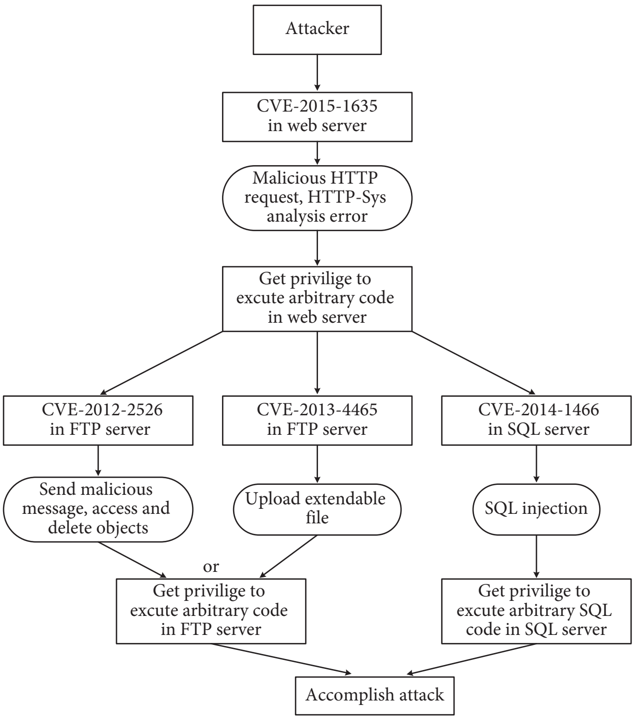

Attacker
CVE-2015-1635 in web server
Malicious HTTP request, HTTP-Sys analysis error
Get privilige to excute arbitrary code in web server
CVE-2012-2526 in FTP server
CVE-2013-4465 in FTP server
CVE-2014-1466 in SQL server
Send malicious message, access and delete objects
Upload extendable file
SQL injection
or
Get privilige to excute arbitrary code in FTP server
Get privilige to excute arbitrary SQL code in SQL server
Accomplish attack

Figure 3: The corresponding attack graph.

NetSPA (https://dspace.mit.edu/handle/1721.1/29899) is an attack graph generation tool designed by Lippmann et al. from MIT. It builds a network model by analyzing firewall rules and vulnerability information and performs reachability analysis [10]. Due to the lack of learning ability in attack mode, NetSPA needs to create a vulnerability rule set manually.

TVA uses the Nessus vulnerability scanner to automatically map the scanned vulnerabilities to the description of the network device [11]. In the generated attack graph, attack paths from the initial state to the target state are provided. Like NetSPA, TVA needs to manually create a rule set.

2.3. Attack Graph Calculation Task. The purpose of attack graphs is to quantify network security situation and find the weakness. Therefore, several important calculation tasks to be done on the attack graph include network vulnerability analysis, node security hardening selection, attack path prediction, and uncertainty analysis.

Vulnerability analysis includes two aspects. One is the analysis of possible attack paths before attack and defense of high-risk nodes on the path. The other is analyzing attack behavior, predicting subsequent target, and taking countermeasures for the attack [4].

In the network reinforcement, important steps include the selection of nodes needed to be strengthened, the balance between costs and benefits, and targeted network defense methods. All these tasks need rigorous modeling analysis.

For attack path prediction, since the network attack is usually systemic, the exploited vulnerability and the attack path can be traceable. While there are many attack paths in a particular attack, how to identify the paths that are the most likely used requires complete considerations. In addition, the dynamics of network configuration requires that network attack defense mechanism needs to be updated according to the development of network security technology and enterprise services. As a result, it is necessary for attack graphs to provide uncertainty analysis on the security problems caused by the network configuration.

2.4. Attack Graph Analysis Method. There is no such an analysis method that can fulfill with all of the above calculation tasks. Hence, the corresponding analysis method should be carefully selected according to the specific tasks. This paper systematically surveys on the analysis methods, which can be generally categorized into logic-based methods and probability-based methods. Probability-based analysis methods include Bayesian networks and Markov models, and the rest are based on logic. Among these methods, graph-based algorithms and Markov model-based analysis methods can be used to predict attack behavior and analyze the most likely attack path. Bayesian network-based analysis methods tend to identify high-risk nodes and the key nodes that should be reinforced. Cost-optimized algorithms have a huge advantage in balancing costs and benefits. The analysis method based on the uncertainty is used to study the influence of vulnerabilities, links, attack behaviors, and other factors on network attacks.

3. Attack Graph Analysis Methods Based on Graph Algorithm

Usually, an attack is launched at the initial node, and then the neighbor nodes which have weakness in security can be served as the next hop to finally reach the target node. Hence, by analyzing the characteristics of the attack graph, the effectiveness of each path and node in security assurance can be examined. Accordingly, the current research studies on attack graph algorithm can be summarized in two categories: one is based on graph path and the other is based on the node.

3.1. Graph Path Algorithm. The general research methods of attack graphs are based on various graph path algorithms of directed acyclic graphs. Several metrics, such as the shortest path, the average path length, and the extended security metric, have been devised to measure the network security in the algorithms.

The shortest path in the attack graph is the one that covers the least number of vulnerabilities in the process of reaching the attack target [12]. The idea is to utilize various graph algorithms, such as Dijkstra algorithm, Floyd algorithm, and so on [13]. This is a relatively straightforward method of attack graph analysis, but there are some problems. For example, it is suggested that this method does not consider the number of shortest paths in the attack graph [12]. Suppose two alternative topologies have the same shortest length, this method can lead to wrong results. Figures 4 and 5 represent two different networks. The topology, configuration, and version of applications in the two networks are different, so do the labels. Suppose the starting state is S, the target state is G. The shortest path length in both figures is 1, which means that the shortest path does not provide any reasonable reference. However, the paths with length 1 in Figure 5 are not only one. Hence, in order to increase the shortest path length of the attack graph, only one path needs reinforcement in Figure 4. For this reason, the network of Figure 4 is stronger than Figure 5. But the shortest path method cannot draw this conclusion. In addition, the shortest path is a coarse-grained metric that is not sensitive to small changes in network nodes. For example, in Figure 5, as long as one of the five paths from S to G is maintained, any changes in other paths will not affect the shortest path.

Another path algorithm attempts to find the number of attack paths which represents how many different methods an attacker can choose to reach the target [14]. The number of attack paths reflects the exposure degree of the network to attackers. The more the attack path number, the lower the security of the network. Compared to the shortest path method, the attack graph path number is more sensitive and performs better in the prediction of attack behavior. However, the shortcomings of this method are also very obvious. For example, the number of attack paths does not

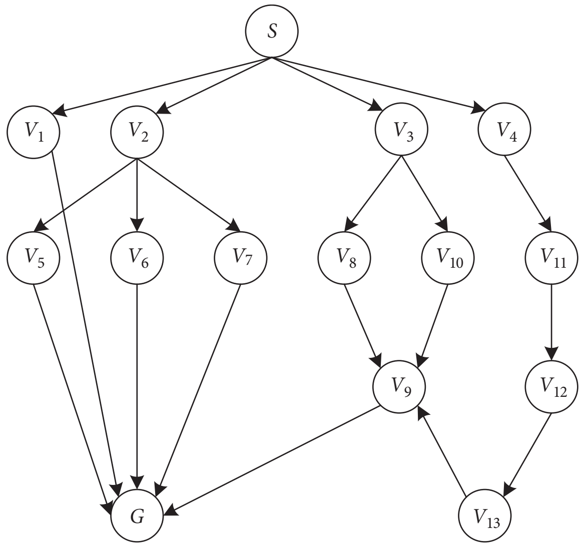


Figure 4: Optional attack graph A.

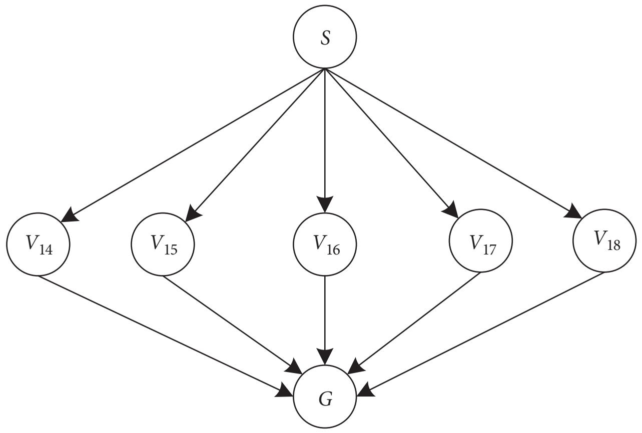


Figure 5: Optional attack graph B.

reflect the cost of the attack spent, nor does it reflect the difficulty of each path to the target.

An average path length metric that averages the length of all attack paths to measure the cost of attacks on the target network is proposed [15]. This approach is slightly different from the assumptions on which the shortest path method is based. The shortest path method assumes that the attacker will choose the shortest path, which is subject to many other factors in reality, such as the skills the attacker has and the utilization of the vulnerability on the shortest path. Therefore, the average path length metric uses the average length instead of the smallest one. This method has a good performance in network security reinforcement. However, it is not sensitive for this method to the path change of the attack graph.

Based on the above three metrics, the improved security metric was proposed by Idika and Bhargava [14]. Firstly, the average of all attack path lengths in the graph is normalized. The normalization compensates for the lack of path number in the average path length metric. At the same time, the normalization makes the comparison between different attack graphs more reasonable and comprehensive. The attack graph with less normalized average path length is more likely to be risky. Secondly, the mean of the path lengths is not able to describe the variance of path lengths. Therefore, the authors introduce three criteria, that is, the standard deviation of path lengths, the distribution of path lengths, and the median of path lengths. Through the distribution of path lengths, the most typical path length in the attack graph can be revealed, and it suggests a likely amount of effort for attackers. As for the median of path length, it suggests an average effort for attackers. For standard deviation and the median, administrators should focus on those paths whose length is less than the average during network hardening.

The above methods only consider paths and ignore the role of nodes in path selection. To overcome this problem, some researchers recently proposed a path analysis method for large-scale networks [16]. This method combines path lengths and the danger coefficient of nodes represented by CVSS (https://www.first.org/cvss/) (Common Vulnerability Scoring System) scores. The danger coefficient of the whole path can be expressed by the product of all node scores on the path. The higher the danger coefficient is, the higher the probability the path will be utilized by attackers. Based on the results, administrators can select the path with a greater danger coefficient than the predefined threshold for optimization.

3.2. Importance Sorting Algorithm for Graph Nodes. In order to capture the uniqueness of different nodes for security reinforcement, other studies focus on the node sorting method.

Inspired by the PageRank algorithm which measures the importance of the webpages in search engine [17], Mehta et al. [18] improved the PR algorithm to solve the problem in analyzing the large scale of the attack graph. This algorithm first analyzes the path that a particular network may be attacked to generate an attack model, and then it was transformed into an attack graph. The nodes represent a particular state, for example, a database server port is open or not. The edges represent a transition between states, for example, an open port state might cause the invasion of database server. The leaf nodes represent an error state which is a kind of privilege the attacker finally gets. The sorting of attack graph nodes is essentially a prediction of state transitions, similar to the Markov model introduced in Section 5. The sorting algorithm obtains the leaf node with the highest PR value in the graph, and the error state represented by the node is the most likely privilege gained by the attackers. Sorting other nonleaf nodes will reveal the attack path that is the most likely to be exploited by the attacker. For example, if the node that represents the open state of a port on the database server has a high PR value, it means that the attacker is more likely to use this port to launch attack.

The PageRank algorithm will find out the way most likely to be attacked through and provide advice for reinforcing vulnerable nodes. However, network is usually changed dynamically. As a result, the corresponding attack graph should be reconstructed frequently, and a lot of computation for the algorithm should not be ignored.

To solve the problem, Lu et al. [19] employed GNN (Graph Neural Network) [20] to sort the attack graph nodes. GNN learns the topological dependence of objects, such as the Ranking of a node relative to its adjacent nodes. The reason why authors used GNN is that, compared to other machine learning algorithms, GNN does not need normalized vector data. Secondly, GNN guarantees convergence. Although this method is the same as the basic idea of PageRank, it provides a better solution for dynamic network changes. The experimental results show that the accuracy of training with GNN is similar to the PageRank algorithm. GNN takes a long time to train, but after that, the attack graph can be tested quickly, so it is more suitable to deal with the frequent dynamic changes.

3.3. Comparisons between the Two Methods. The path analysis methods do not need to investigate and assign the probability of node states. Therefore, the variables in the methods are more explicit and can be solved by algebraic methods. However, the sensitivity of path analysis algorithm is generally low, and the difficulty of exploiting is also ignored. The analysis method for sorting node importance takes the general states of the nodes into consideration, and thus it can be utilized to overcome the problems in path analysis methods.

4. Attack Graph Analysis Method Based on Bayesian Network

Bayesian network (BN) is a kind of probabilistic graph network. It is commonly used in the field of uncertainty analysis and reasoning. The Bayesian network uses causal relationships to estimate the probability of an unknown event based on events that have occurred. The attack graph based on the Bayesian network is represented by a triple (Node, Edge, and PTable). The nodes in the attack graph denote the vulnerabilities, privileges, etc. The edges are the dependencies between the nodes. PTable is the conditional probability distribution, which is used to record the conditional probability of nodes that are being attacked. The value of the probability is usually determined by experts in the professional field.

An example of an attack graph in Bayes-based attack graph analysis is shown in Figure 6. Five nodes from A to E indicate the vulnerabilities of the system or the privilege obtained through vulnerabilities. PTable is presented in the figure. For example, when node C is successfully attacked by an attacker, the probability that node E is attacked is 0.5. When a node is identified as an evidence node, which means that the attack event has happened, then the probability of other node status can be obtained by using the Bayesian formula.

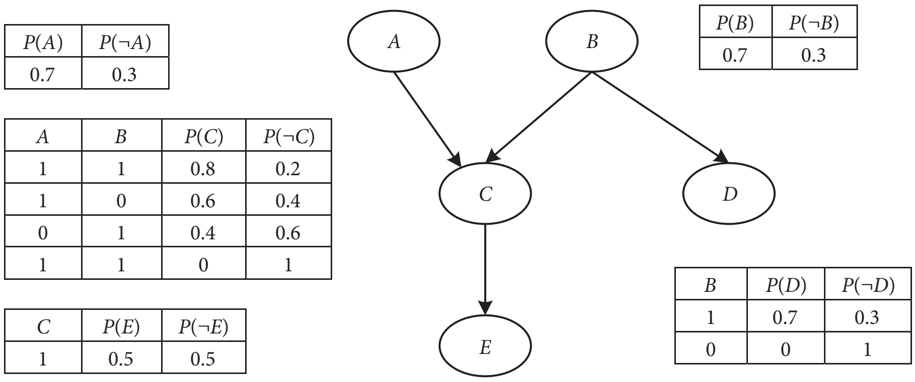
Liu and Man applied Bayesian network to the attack graph for the first time [21]. The authors incorporated CVSS-based scores and the casual relationships to update the posterior probabilities of nodes, and then the attack path can be exploited. Later, the huge advantages of solving uncertainty make BN widely used in attack graphs.

Researchers have found that the impact of vulnerabilities would change over time. For example, if the vendor releases a patch to fix vulnerability, the exploitability of the vulnerability will be greatly reduced. Therefore, it is insufficient to use the CVSS-based score to evaluate vulnerabilities without considering the time evolution. Thus, Frigault et al. [22] employed the factor of time, such as the availability of exploitation or patch, to establish a dynamic Bayesian attack model (DBN). In this model, the attack graph is composed of multiple Bayesian attack graphs. Each BN corresponds to a specific time, and nodes are connected by edges in consecutive time slice. The DBN model satisfies the Markov properties, that is, the system state only depends on the previous state. According to the probability distribution of the initial and the adjacent time, a joint probability distribution can be obtained.

In addition, only using CVSS scores to estimate the probability in the attack graph does not make full use of other information of the network. For this reason, Wu et al. [23] added three environmental factors to the Bayesian attack graph to improve the inference ability. These factors are the value of assets in the network, the usage of the network, and the attack history of the network. The authors believed that the network which is of higher asset values and used more frequently is more likely to be attacked again. The experiments support this conclusion, so considering environmental factors will make the analysis results more accurate.

Although the introduction of the Bayesian network to analyze attack graph is more comprehensive compared to the graph algorithm, all of the above works did not propose a reasonable and effective model that can apply to the process of reasoning. To overcome the problem, Liu and Man [21] put forward a variable elimination (VE) algorithm, but the complexity of VE algorithm highly depends on the order of variable elimination which is random and has low computational efficiency. Therefore, it can only be used in a small-scale network. In addition, the VE algorithm can only calculate the unconditional probability of one node at a time. Therefore, Munoz-Gonzalez et al. [24] used the joint tree algorithm (JT) as an improvement on the VE algorithm. The JT algorithm can achieve the convergence state through the message passing mechanism. Once the model achieves the convergence state, all the probabilities no longer change. At the same time, the introduction of the prior probability value of one node would only affect several probability values other than the total graph. The experiment results show that the JT algorithm is superior to the VE algorithm in terms of time complexity and space complexity and is more suitable for actual situations.

Although there are a lot of research studies on attack graph analysis based on Bayesian network, little work pays attention to the attack time consumed which is important for administrators to predict when the next attack will happen. Hu et al. [25] presented a method to calculate the time



$P(A)$	$P(\neg A)$
0.7	0.3

A	B	$P(C)$	$P(\neg C)$
1	1	0.8	0.2
1	0	0.6	0.4
0	1	0.4	0.6
1	1	0	1

C	$P(E)$	$P(\neg E)$
1	0.5	0.5

$P(B)$	$P(\neg B)$
0.7	0.3

B	$P(D)$	$P(\neg D)$
1	0.7	0.3
0	0	1

FIGURE 6: An example of an attack graph in Bayes-based attack graph analysis.

consumed. They calculated the average time consumption by weighting on history attack and took the expected time based on the probability of future attack.

5. Attack Graph Analysis Method Based on Markov Model

Markov models are widely used in attack graph analysis. They can be divided into four categories, that is, Markov chain (MC), Markov decision process (MDP), hidden Markov model (HMM), and partially observable Markov decision process (POMDP). Their relationship is shown in Table 3.

All of the above models have no aftereffects. Given the known information, the past state is irrelevant for predicting future states, and the future state is only relevant to the present one. In this section, the attack graph studies are, respectively, reviewed based on these four models.

5.1. Markov Model. By means of Markov chain, a triple (S, P, Q) is used to represent the attack graph, where S denotes all possible states in the system, including absorbing states and transient states. States consist of network assets, user privilege, etc. P denotes the state transition probability matrices and Q denotes the initial probability of states. The absorbing Markov chain has two properties. First, an attack graph has at least one absorbing state. Second, in an attack graph, it is possible to go from every state to an absorbing state. As shown in Figure 7, node 4 is absorbing because it is impossible to leave it once entered.

In the attack graph, the absorbing state is regarded as the attack target. Once the node is reached, the attack is done successfully. For any network, the attack path is from the initial node to the target node through the transient states, and thus state transition can be used to indicate the change of the network security. The network state, state transition relationship, initial state, and target state can be abstracted from the network to construct a state transition system of the network. By analyzing the attack path, it is possible to conduct the network security assessment.

Abraham and Nair [26] modeled the attack graph as an absorbed Markov chain. The transition probabilities of the Markov chain are calculated by CVSS scores, and the probability $P(i, j)$ from i to j is defined as the score of j divided by the sum of the scores of all of the next nodes of i. Through the model, the authors can perform a security assessment on the network, such as calculating the expected path length and probabilistic path metric.

Abraham and Nair later introduced time factors into the Markov chain and presented a network security metric model [27]. The time factors can capture the probability that the vulnerability exploited by the attacker changes with time. The authors used the results of Frei's vulnerability lifecycle model [28] to calculate the likelihood of an exploit or patch being available a certain number of days after its disclosure. Actually, the impact of the vulnerability will gradually decrease over time. Then, the probability of the state transition from this vulnerability should be reduced. Thus, it is useful to combine time weights and the CVSS score into the transition matrix of the Markov model.

5.2. Markov Decision Process. The analysis method based on Markov decision process (MDP) uses a five-tuple (S, A, P, R, γ) to describe attack graph. S denotes the set of states that may appear in the system, A denotes the action set, P denotes the state transition matrix, R is the benefit of state transfer by performing the action, and γ is a discount factor indicating the uncertainty about the future. MDP can be utilized to represent attacker's decision. The benefit is the attacker's cost or the reward if the attack is successful. In the attack graph, the attacker tends to choose a path that minimizes the cost of the attack or gets the highest reward. Markov decision process can select the most profitable set of actions in a series of random action sequences based on Markov properties.

Sheyner et al. [8] calculated the likelihood of the successful attack based on the MDP modeling of attack graph. The method of value iteration is used to select the optimal action strategy based on MDP. However, as the scale of the network increases, the great challenges in the calculation are obvious. Therefore, simplifying the calculation has become one of the issues when using MDP to solve optimization problems.

Durkota et al. [29] adopted a pruning strategy for MDP in response to the above problem. The authors considered the MDP problem from the attacker's point of view and used *Sibling-Class Pruning* and the *Branch-and-Bound* methods

Table 3: Four categories of Markov models.

	No consideration of decision action	With consideration of decision action
Visible state	Markov chain (MC)	Markov decision process (MDP)
Invisible state	Hidden Markov model (HMM)	Partially observable Markov decision process (POMDP)

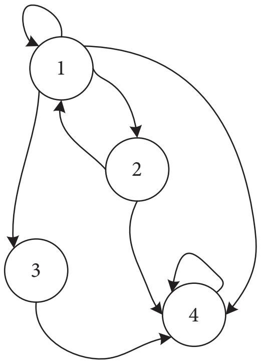


Figure 7: Absorption Markov chain.

to speed up the search process. With the pruning techniques, the amount of calculation is greatly reduced, and the model can be applied to large-scale network analysis.

5.3. Hidden Markov Model. Hidden Markov model (HMM) adds a hidden state to the Markov chain and can be represented by a five-tuple (*S, O, A, B, PI*). *S* is the set of hidden states, expressed as the state of the system, i.e., the attack state. *O* is the set of observation states, expressed as physical components (such as hosts and servers), network assets, privileges, or vulnerabilities. *A* is the state transition probability matrix. *B* is the observation symbol probability matrix and PI is the initial state distribution. Taking Figure 8 as an example, the upper layers X_1, X_2, and X_3 are hidden states, and the bottom layers Y_1, Y_2, Y_3, and Y_4 are observation states. There is a certain relationship between hidden states. For example, if X_1 represents state of port scanning, then the next state is more likely to be "sending error packets." The relationship between hidden states is described by the transition probability matrix *A*. At the same time, each hidden state corresponds to several observation states. For instance, at the state of X_1 (port scan), we can observe both Y_1 (warning from snort and other detection tool) and Y_2 (honeypot capture) states with a certain probabilities which are represented by the observation symbol probability matrix *B*.

Wang et al. [30] first proposed a method for quantitative analysis of the attack graph under the framework of the hidden Markov model. In the research, nodes such as network assets, system vulnerabilities, and user rights are taken as observations, and the system state of attack and nonattack is set as hidden states. The observations are associated with the hidden system state by a certain probability. Based on the proposed HMM model, the next system state can be predicted by capturing a series of observable values. The most probable attack sequence for a specific

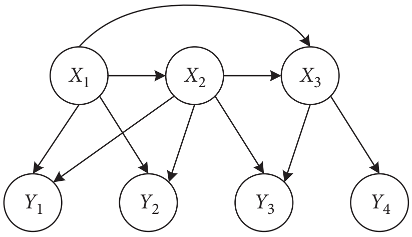


Figure 8: Hidden Markov model.

observation sequence is attained by Viterbi algorithm which takes both vulnerability scores and defense cost into consideration. In this way, it is possible to select the most necessary path for network hardening.

5.4. Partially Observable Markov Decision Process. The partial observable Markov decision process (POMDP) is represented the model by using a seven-tuple (*S, A, P, R,* Ω*, O,* γ). Ω is the set of observations. *O* is the conditional observation probability, indicating how likely it is in a particular state after observing Ω. Because the administrator is unsure about the current state, he needs to perceive the environment to determine which state he is in. Then, the concept of a belief state space is introduced, which is to estimate the current state and then the POMDP problem can be converted into an MDP problem.

Miehling et al. used POMDP to develop optimization strategies for attack graph analysis [31]. It is assumed that the defender can only partially observe the attacker's action at any given time and needs to make decisions when the information is incomplete, and thus POMDP problem can be formulated. In the attack graph, the nodes represent system attributes, such as, attacker permission levels on a given machine, vulnerabilities of a service or system, information leakage, and so on. Exploits which are represented by the edges in the graph are events that allow the attacker to use their current set of capabilities to obtain further capabilities. The probability of the observed events can be used to estimate the attacker's ability. The authors used POMDP solver developed by Cassandra to obtain the optimal defense policy [32].

In addition to the incomplete observations, the utility function might be unknown. For example, for the zero-day vulnerability in the attack graph, the information about the vulnerability is unknown so that the cost or reward cannot be estimated. However, defenders can get some feedback after taking defensive measures. Therefore, Hu et al. [33] used a POMDP algorithm for solving unknown utility. The authors divided the time equally and calculated the benefit as the average of the benefit for each time period. Finally, dynamic programming is used to estimate the optimal utility function. The simulation results show that when the utility function is unknown, the algorithm can help the defender identify an effective defense strategy.

Recently, Miehling et al. developed a new POMDP model [34], which is more general than the previous one [31]. The authors considered more complex dependencies between vulnerabilities, such as the successful exploitation of a vulnerability that would create multiple attack conditions. At the same time, more real situations such as false positives of alerts are also discussed. In this method, only the state related to the current defense decision, regardless of the entire state space, is considered, and the experiment shows the calculation efficiency is improved.

5.5. Comparisons of the Methods. The four methods of MC, MDP, HMM, and POMDP are employed to denote the attack behavior. These models can be distinguished by whether the state is visible or not and whether to consider decision actions. MC only makes use of state transition probability, while MDP introduces utility function. However, the utility function and state transition probability are known over all time periods. MDP adds the attacker's decision-making behavior into the Markov chain. From the actual application point of view, the defender has to use the existing partial information, historical state sequence, and reward function to make decisions so that POMDP becomes potentially effective.

6. Attack Graph Analysis Method Based on Cost Optimization Algorithm

Apart from finding the security condition of specific network, another important task of attack graph is to determine how to implement target reinforcement. When it comes to network hardening, it is necessary to consider the cost. Any measures adopted have cost, for example, deploying a new packet filtering firewall will cost, and filtering out packages by mistake also costs. If the impact of a network attack is not severe or the attack is easy to be repaired, then the cost of network hardening can be higher than repairing it after attack. From the perspective of attackers, if the gain from attacking is far less than the time and other cost of attacking, the attackers may give up attacking the network. Therefore, it is important to decide whether it needs reinforcement or not and which nodes should be reinforced.

In this section, cost optimization algorithms are introduced. The general idea is to obtain the attack paths and the probability of each node to be exploited and then calculate the cost of node hardening.

6.1. Attack Graph Analysis Method Based on Cost Minimization Algorithm. For a particular network, finding the least cost in network optimization is an NP-complete problem. Cost minimization algorithms simultaneously take the vulnerability exploitation, the time spent, and economic cost into account.

In the early stage of the attack graph research, the general idea to solve this problem is to find the smallest set of vulnerability [35]. The smallest set means that the target attack state becomes unreachable if all the vulnerabilities in the set are fixed or removed. However, the parent nodes for each node in the set are ignored in this method. Hence, the conditions that vulnerability happens still exist and can become a potential security risk.

Approaches based on cost distribution along paths are another attempt for the problem. The motive is that removing the starting nodes with serious vulnerabilities can effectively improve the network security. Islam and Wang [36] proposed a heuristic algorithm for searching such initial nodes. Each initial node in the attack graph has a value of effective cost which is defined as the ratio of node cost and the number of vulnerabilities in it. Then, the cost is distributed to the next nodes according to several rules, and finally, the distribution reaches the target node that needs to be defended. As a result, the initial node with greatest impact on the target node and minimal initial cost should be selected for hardening. Wang et al. proposed a disjunctive normal form (DNF) representation method for attach graph [37]. In this way, the target node is transformed into the DNF of its preorder nodes. The decomposition of the target node contains only the initial conditions, and each disjunction in the DNF provides a different condition in network hardening. Options with the minimum costs are chosen by the given assumptions on the cost of initial conditions.

Reduced ordered binary decision diagram (ROBDD) [38] is a new idea to tackle the cost minimization problem. ROBDD provides an efficient graphical way for representing and manipulating Boolean functions, which include one source and two sinks labeled with 0 and 1. There are two types of relations between exploit nodes and condition nodes in an attack graph, namely, AND relation and OR relation. Each internal node N_i in ROBDD has a high edge pointing to node N_i^h and a low edge pointing to node N_i^l. By performing iterative Shannon decomposition on each node as the function, we get

$$\mathrm{MinCost}\left(N_j\right) = \min\left\{\mathrm{MinCost}\left(N_j^h\right), \mathrm{MinCost}\left(N_j^l\right) + C\left(N_j\right)\right\}. \quad (1)$$

For example, in the attack graph shown in Figure 9(a), c_i indicates initial condition, A and B are middle nodes, G is the target node, and there are two paths lead to the target node. Figure 9(b) shows the corresponding ROBDD of Figure 9(a), and the available path is labeled 1, while the unavailable one labeled 0. Assume the hardening cost of c_i is $C(c_1) = 10$, $C(c_2) = 1$, and $C(c_3) = 15$. The cost of network hardening is $\min\{C(c_2), C(c_3), C(c_2) + C(c_1), C(c_3) + C(c_2)\}$, so the minimum cost is 1 and c_2 is supposed to be reinforced. The ROBDD method does not need graph traversal to reach the target state, and thus the complexity is $O(n)$, where n is the number of nodes in ROBDD.

In addition, intelligent algorithms can also be employed to the cost minimization problem. Genetic algorithm can be used to solve the minimum cost network hardening of attack

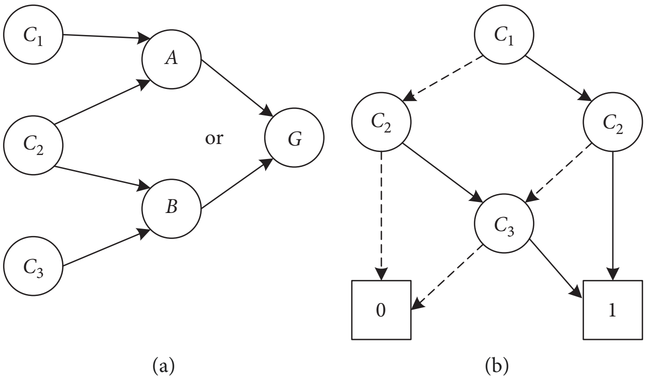


Figure 9: Example of ROBDD attack graph.

graphs [39]. Firstly, the attack graph is binary coded: 0 means the node does not need to be changed while 1 means it needs to be changed. The second step is to initialize population, define fitness function and objective function, and iterate according to the presupposed parameters. The fitness function is the cost of the optimized state represented by each chromosome, and the objective function represents the expected minimum cost. The final result of multiple iterations is the approximate optimal result. Sequential linear programming is another approach to get the cost minimization solution [40]. The theoretical complexity of this algorithm is high, but with the appropriate parameters, the running time in practice can be accepted.

6.2. Attack Graph Analysis Method Based on Game Theory. The essence of cyber security analysis is the game among individuals, that is, attackers and defenders. Game theory provides a reasonable mathematical framework for analyzing network security, and it can help to choose the best strategies with considerations of defense cost and profit.

In the process of network reinforcement, honeypot is a common facility for defenders to reduce the risk of network attacks. Legitimate users do not interact with the honeypot, and thus the honeypot can act as bait to draw the attackers' attention. At the same time, it can send intrusion alerts to the defender. However, the construction cost and maintenance cost of honeypots are very high. It is very important for defenders to consider how to properly deploy honeypots. On the other hand, for attackers, they need to predict and avoid honeypots. Therefore, game theory is used to simulate the offensive and defensive interaction, and the best way to deal with the attackers can be determined by calculating the gains attained by the offense and defense.

Durkota et al. extended Stackelberg models, which is a leader-follower game [29]. In the game, the defender is the leader and the attacker is a follower. The authors suppose that the attacker knows the number of honeypots and their types, but not sure where the honeypots are. The defender strengthens the security defense by placing a honeypot, and the attacker selects the optimal attack path by analyzing the defender behaviors. There are two kinds of nodes in the attack graph, that is, fact nodes and action nodes. Fact nodes are used to represent the logical structure of the entire network, and the action nodes represent the attack behavior, which is accompanied by the probability and cost of successful attacks. The attacker would choose the attack path with the highest profit. Once the attacker enters the honeypot, the attack ends. Therefore, the authors transform the attack graph with the game theory model into a MDP problem to solve this complex problem and introduce some pruning techniques to effectively reduce the amount of computation. However, the assumption of this method has limitations, for example, the attacker needs to know most of the information and just cannot distinguish the truth of the host.

A new game theory model for attack graph analysis is proposed based on reasonable assumption [41]. In this model, the attacker only knows the total number of honeypots but does not know their types. However, the optimal strategy for finding a defender's honeypot is NP-hard and cannot be directly calculated in larger networks. The problem can be converted into an approximate model of a perfect information game where the attacker is supposed to know the defense strategy of the defender. Experiments show that the strategy is very close to the original model.

6.3. Comparisons of the Methods. The goals of two methods are the same, that is, to find the proper security reinforcement under restriction of cost. The cost minimization algorithms take the vulnerability exploitation, the time spent, and economic cost into account. The game theory-based method further considers the interaction between offensive and defensive sides. Therefore, it can attain more useful results for reinforcement. However, both of the two methods should face the same questions, such as the setting of cost, the computation complexity, and so on.

7. Uncertainty Analysis Based on the Attack Graph

Dealing with uncertainty in the attack process is important for network security. New methods should be introduced into the attack graph to process the uncertainties. The uncertainties come from several sources, such as the network structure, attack actions, device configurations, and so on [42]. Therefore, according to the types of uncertainty that can be analyzed, three analysis methods towards uncertain path, uncertain node, and zero-day attack are summarized in the section.

7.1. Analysis of Attack Graph with Uncertainty Paths. The uncertain graph plays an important role in graph structure uncertainty analysis and is widely applied in many areas of uncertainty analysis [43, 44]. Nguyen et al. [45] attempted to model uncertainties in the existence of vulnerabilities and network connections by uncertain graphs. In the uncertainty graph model, the existence of each edge is unknown and can be described by probability. The original uncertain graph model uses a triple (V, E, P) to represent the existence probability: V for the node, E for the edge, and p for the existence probability. Whether the attack path can be utilized is based on the reachability of initial node to target node in corresponding uncertainty graph [46], which can be calculated by summarizing the path existence probability in all possible worlds of the uncertain graph.

The original uncertainty graph assumes that the probabilities of edges are independent of each other. However, this is not always true in attack graph. Therefore, Nguyen et al. extended the form of the uncertain graph and described the relationship between edge existence by a quintuple (V, E, p, X, q). X is a Boolean variable indicating whether the edge exists or not. The edge existence probability of X is $p = P[X]$; q is a Boolean function indicating the relationship between edges. At the same time, the article demonstrates that when Boolean function is monotonous, uncertainty analysis of attack graph transforms from NP-complete to the calculation of confidence interval of the path's initial probability distribution. The reachability from the initial node to target node is positively correlated with the uncertainty of the path; hence, the impact of possible vulnerabilities and configuration changes in the network on the reachability of the target node can be analyzed.

7.2. Analysis of Attack Graph with Uncertainty Nodes. Mobile terminal which can freely connect to the network provides a new way for attackers to invade the network. The attackers can even attack other network nodes through the vulnerability on mobile devices instead of merely attacking the device itself. It is imperative to introduce the mobile device nodes in the attack graph.

The analysis method for attack graphs with mobile devices is of great challenge. Firstly, whether the mobile device exists in the network, when and how long it exists, and which network services are open on it are all unknown. Secondly, many questions are still difficult to resolve, for example, how to determine whether the mobile device has exploitable vulnerabilities and how to measure the vulnerabilities on the mobile devices.

A scalable probabilistic graph model that incorporates dynamic network features into modeling is proposed [40]. The model used probabilities to represent the possible usage of mobile devices and their properties, such as connection duration and connection frequency. The scalable probabilistic attack graph adds a node represented as device Online (H, P), in which H stands for a mobile device and P stands for the operating system of H. The node is assigned a Bernoulli variable as the probability of its connection to network. In the analysis phase, it is assumed that the mobile device connects to the network according to the defined probability and then determines how the connection impacts on the security of whole network. The experimental results show that after the introduction of mobile devices, great changes have happened in the distribution of security threats across the network. Experiment also finds that the mobile phone's attack expectation is far beyond other nodes. The results also confirm that mobile devices have a great impact on traditional network defense strategies.

7.3. Analysis of Attack Graph with Zero-Day Attack. Zero-day vulnerability refers to the vulnerability that has been discovered but possibly not known to public, and the official has not released a related patch. For this reason, it is a serve threat to network security. By borrowing the idea of attack surface [47–49] and k-anonymity in privacy protection [50], Wang et al. proposed the k-zero-day method to model the zero-day attacks in network defense [51]. According to the basic idea of k-anonymity, the analysis supposes the existence of zero-day vulnerability on each node and then counts the number of possible zero-day vulnerabilities on each attack path. If the number of zero-day vulnerabilities is less than k, then the attack graph is k-safety. The greater the value of k is, the more unknown vulnerabilities that an attacker needs to exploit to invade the network and the more secure the network is. Therefore, the defender can properly arrange the devices on the network so that the requirement of k-safety is satisfied. This strategy for defending zero-day attack not only preserves the possibility of zero-day attack in attack graphs but also finds a more feasible way to solve the uncertainty brought by zero-day attack.

Sun et al. [52] introduced zero-day vulnerabilities in Bayesian networks, also attempting to combine zero-day vulnerabilities with Bayesian-based attack graph analysis methods. This approach is remarkable, but how the zero-day vulnerability impacts on other vulnerabilities needs further evaluation.

7.4. Comparisons of the Methods. The uncertainty analysis provides an effective remedy for dealing with special scenarios. The analysis method of attack graph with uncertain paths focuses on the uncertainties in the existence of vulnerabilities and network connections, the analysis of attack graph with uncertain nodes pays attention to the uncertainties raised from mobile devices, and the analysis method of attack graph with zero-day attack deals with those vulnerabilities that have been discovered but possibly not known to public.

8. Applications of Attack Graph Analysis Method

Currently, applications of attack graph can be categorized into four types, that is, network risk assessment, network security hardening, prediction of attack behavior, and the uncertainty analysis of network security.

8.1. Network Risk Assessment. The administrator can analyze the network security by assigning probabilities or damage values to the edges and nodes in the attack graph and calculating the security indicators. These metrics can be used to determine whether a host or network is under attack, such as risk analysis and reliability analysis [53]. In the graph algorithm, the corresponding metric, such as the number of attack paths in the graph, can be used as the criterion for whether the network is secure. Otherwise, the attack graph analysis methods based on Bayesian network, Markov models, and other probability-based methods infer the possibility of attacks through probability distributions.

8.2. Security Hardening. Network security hardening aims to adopt an optimal security policy to improve network security. Repairing all vulnerabilities is meaningless and infeasible. A more suitable choice is to find some high-risk vulnerabilities and measure the cost of attack and defense, respectively, and then remove some appropriate vulnerabilities according to the cost. When selecting the node to harden, a heuristic optimization algorithm can be used to compare the optimal strategy. For example, Wang et al. integrated attack graph and the hidden Markov model to explore the probabilistic relation between system vulnerabilities and attack states, and then a heuristic searching algorithm is employed to automatically infer the optimal security hardening through cost-benefit analysis [30]. Based on the Bayesian network analysis method, Poolsappasit et al. proposed to multiply three key factors, that is, the probability of exploiting the vulnerability, the expected return of vulnerability elimination, and hardening cost [54]. Then, the result is considered as the hardening return to find out the best hardening strategies by using genetic algorithm.

8.3. Prediction of Attack Behavior. In some occasions, the administrator needs to predict the next targets once a host has been found compromised. By applying attack graph analysis methods to the scenario, it is able to get the prediction results. For graph algorithms, the prediction can be inferred from the kinds of path, such as the shortest path. For Bayesian network-based method, the next behavior can be predicted by calculating the posterior probability for neighbor nodes. Moreover, the method provides a good way to update the prediction model by adding the new attack instances into training set and then updating the model. In applications, the alarm message provided by security infrastructures, such as intrusion detection system, system log, and so on, can be taken as the current attack situation [4].

8.4. Uncertainty Analysis Based on Attack Graph. The network structure and configurations are usually changing dynamically and frequently. Software update, application, or configuration changes may cause minor changes on network topology, but the changes on attack graph are enormous. Therefore, the attack graph of the network is more sensitive to uncertainty. The uncertainties usually are caused by several factors. The vulnerabilities in the new version of software are unknown and probably not the same as the old version. As another example, the mobile device connections are usually not kept and difficult to predict, which lead to the great uncertainty on network security. The task of uncertainty analysis based on attack graphs is thus to analyze the uncertain phenomena mentioned above, evaluate the influence of these uncertain factors, and attempt to conduct quantitative analysis. These uncertainties can be solved by uncertain attack graphs. The scalable attack graph analyzes the connection probability of mobile devices in the network, evaluating security status of whole network. The zero-day attack graph can deal with the unknown vulnerabilities that may exist in the network.

9. Conclusion and Future Work

9.1. Conclusion. This paper focuses on the analysis method of attack graphs. The main research work can be grouped into five types, namely, graph-based attack graph analysis method, Bayes-based attack graph analysis method, Markov model-based attack graph analysis method, cost optimization methods, and uncertainty analysis methods. To summarize the survey, the comparison among these methods is provided in Table 4.

Compared with Bayesian and Markov models, the graph algorithm is more simple and intuitive. Meanwhile, the graph algorithm does not need to carry out model training, and the influence of node changes is limited, so the graph algorithm performs better in terms of scalability. Bayesian network has great advantages in solving uncertainties and correlation problems. Markov-based models need enough training data. When using HMM to infer the distribution over hidden states, it is necessary to enumerate all the observation sequences. It is difficult to obtain an accurate result by using MDP and POMDP since the problem is NP-hard, so some approximation algorithms are used instead. The complexity of the optimization algorithm is generally higher than the graph algorithm.

9.2. Future Work. The attack graph analysis method is still developing continuously, and the future development will focus on the following aspects:

(1) Integrate different attack graph analysis methods to improve the ability of attack representation and modeling analysis.

The existing analysis methods have respective advantages and disadvantages. Therefore, in the future, the researchers can try to integrate logic-based path analysis and probability-based node analysis. An effective integration framework can be designed to

Table 4: Comparison of attack graph analysis methods.

Analysis method	Advantage	Disadvantage	Calculation tasks	Complexity	Scalability
Graph algorithm	Intuitive, portable	Insufficient combination with exploit utilization	Identify the most likely path and high-risk node, predict attack behavior	$O(n^2)$	Strong
Bayesian network	Flexible, easy to train	Complicated analytical calculations	Analyze vulnerability, identify high-risk nodes, network hardening, and predict attack behavior	$O(n^2)$	General
Markov model	Easy to train, better prediction	More restrictions	Identify the most likely paths, identify high-risk nodes, network hardening, and predict attack behavior	$O(n^2)$	General
Cost optimization algorithm					
Game theory	Strong portability	Slight discrepancy with actual results	Network hardening, predict attack behavior	$O(n^2)$	General
Cost minimization algorithm	Strong portability	Limited application, difficult model selection	Network hardening, predict attack behavior	$O(n)$	Strong
Uncertainty algorithm	Solved problems that other algorithms cannot solve	Limited application areas	Analyze vulnerability, identify high-risk nodes	—	Strong

take full advantages of the two analysis methods. For example, the complexity of Bayesian and Markov algorithms can be reduced by pruning attack graph based on probability of vulnerability exploiting.

(2) Combine the attack graph analysis method with big data technology to improve the accuracy of model training and parameter setting.

Probability-based analysis methods, such as Bayesian, Markov, and game theory, need to accurately estimate the parameters which are difficult to be set. In future research, these parameters can be obtained through big data analysis to avoid errors caused by subjective allocation. The general idea is to collect a large amount of network attack data first. Then, clean, extract features, and explore the relations between different vulnerabilities by big data technology, providing a basis for determining the transition probability.

(3) Introduce more uncertainty theory to attack graph analysis and enhance the ability to analyze uncertainty in attack behavior.

In addition to the system uncertainty, there are also the attacker uncertainty and system environment uncertainty. Due to the complex characteristics and dynamics, existing methods are difficult to deal with all kinds of uncertainties exactly. Therefore, it is necessary to introduce more uncertainty theory and techniques, such as fuzzy cognitive maps, rough sets, D-S theory, and rule-based systems, to enhance the ability to analyze uncertainty in attack behavior.

(4) Combine the attack graph technology with other security technologies to solve the difficult problems in network security.

The typical examples are APT attacks and zero-day vulnerability detection. The APT attack is tough in vulnerability identification and network defense because of its long duration and hard to be detected form the abnormality of single node. The attack graph technology can infer the change of the network state and discover the abnormal operation of system in the graph. Therefore, the combination of state analysis in attack graphs and network intrusion detection technology can improve the detection of APT attacks. Meanwhile, it is possible to combine the code detection and attack graph technology to discover suspicious parts in the network and improve the ability to identify zero-day vulnerabilities.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China (grant nos. 2017YFB0803203 and 2016YFB0800101) and Shanghai Municipal Natural Science Foundation (grant no. 15ZR1403700).

References

[1] Beijing Rising Information Technology Limited by Share Ltd., *The Information And Network Security of the National Information Center (BU): China's Network Security Report in the First Half of 2017*, pp. 15–89, People's Post and Telecommunications Press, Beijing, China, 2017.

[2] L. P. Swiler, C. Phillips, and T. Gaylor, "A graph-based network-vulnerability analysis system," Tech. Rep. SAND97-3010/1, Sandia National Laboratories, Livermore, CA, USA, 1997.

[3] 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.

[4] Z. W. Ye, Y. B. Guo, C. D. Wang, and A. K. Ju, "Survey on application of attack graph technology," *Journal of Communications*, vol. 38, no. 11, pp. 121–132, 2017.

[5] V. Shandilya, C. B. Simmons, and S. Shiva, "Use of attack graphs in security systems," *Journal of Computer Networks and Communications*, vol. 2014, Article ID 818957, 13 pages, 2014.

[6] F. Chen, H. D. Mao, W. M. Zhang, and C.-H. Lei, "Survey of attack graph technique," *Chinese Computer Science*, vol. 38, no. 11, pp. 12–18, 2011.

[7] N. Gao, L. Gao, Y. Y. He et al., "Dynamic security risk assessment model on Bayesian attack graph," *Journal of Sichuan University (Engineering Science Edition)*, vol. 48, no. 1, pp. 111–118, 2016.

[8] O. Sheyner, J. Haines, S. Jha et al., "Automated generation and analysis of attack graphs," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 273–284, Berkeley, CA, USA, May 2002.

[9] X. Ou, W. F. Boyer, and M. A. Mcqueen, "A scalable approach to attack graph generation," in *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 336–345, Alexandria, VA, USA, October 2006.

[10] R. Lippmann, K. Ingols, C. Scott et al., "Validating and restoring defense in depth using attack graphs," in *Proceedings of the Military Communications Conference*, pp. 1–10, Washington, DC, USA, October 2006.

[11] S. Noel, M. Elder, S. Jajodia et al., "Advances in topological vulnerability analysis," in *Proceedings of the Conference For Homeland Cybersecurity Applications & Technology*, pp. 124–129, Washington, DC, USA, March 2009.

[12] R. Ortalo, Y. Deswarte, and M. Kaâniche, "Experimenting with quantitative evaluation tools for monitoring operational security," *IEEE Transactions on Software Engineering*, vol. 25, no. 5, pp. 633–650, 1999.

[13] P. Höfner and B. Möller, "Dijkstra, Floyd and Warshall meet Kleene," *Formal Aspects of Computing*, vol. 24, no. 4-6, pp. 459–476, 2012.

[14] N. Idika and B. Bhargava, "Extending attack graph-based security metrics and aggregating their application," *IEEE Transactions on Dependable & Secure Computing*, vol. 9, no. 1, pp. 75–85, 2011.

[15] W. Li and R. B. Vaughn, "Cluster security research involving the modeling of network exploitations using exploitation graphs," in *Proceedings of the IEEE International Symposium on Cluster Computing and the Grid*, p. 26, Singapore, May 2006.

[16] C. Zhao, H. Q. Wang, J. Y. Lin, H. Lv, and J. Han, "Attack graph analysis method for large scale network security hardening," *Journal of Frontiers of Computer Science and Technology*, vol. 12, no. 2, pp. 263–273, 2018.

[17] S. Brin, R. Motwani, L. Page, and T. Winograd, "What can you do with a web in your pocket?," *Data Engineering Bulletin*, vol. 21, no. 2, pp. 37–47, 1998.

[18] V. Mehta, C. Bartzis, H. Zhu, E. Clarke, and J. Wing, "Ranking attack graphs," in *Proceedings of the International Conference on Recent Advances in Intrusion Detection*, pp. 127–144, Hamburg, Germany, September 2006.

[19] L. Lu, R. Safavi-Naini, M. Hagenbuchner et al., "Ranking attack graphs with graph neural networks," in *Proceedings of the 5th International Conference on Information Security Practice and Experience*, pp. 345–359, Xi'an, China, April 2009.

[20] F. Scarselli, A. C. Tsoi, M. Gori et al., "A new neural network model for graph processing," Technical Report DII 1/05, University of Siena, Siena, Italy, 2005.

[21] Y. Liu and H. Man, "Network vulnerability assessment using Bayesian networks," in *Proceedings of the SPIE-the International Society for Optical Engineering*, Bellingham, WA, USA, March 2005.

[22] M. Frigault, L. Wang, A. Singhal, and S. Jajodia, "Measuring network security using dynamic Bayesian network," in *Proceedings of the ACM Workshop on Quality of Protection*, pp. 23–30, Alexandria, VA, USA, October 2008.

[23] J. Wu, L. Yin, and Y. Guo, "Cyber attacks prediction model based on Bayesian network," in *Proceedings of the IEEE International Conference on Parallel and Distributed Systems*, pp. 730-731, Singapore, December 2012.

[24] L. Munoz-Gonzalez, D. Sgandurra, M. Barrere, and E. C. Lupu, "Exact inference techniques for the analysis of Bayesian attack graphs," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 2, pp. 231–244, 2019.

[25] H. Hu, H. Q. Zhang, Y. Liu, and Y. Wang, "Quantitative method for network security situation based on attach prediction," *Security and Communication Networks*, vol. 2017, Article ID 3407642, 19 pages, 2017.

[26] S. Abraham and S. Nair, "Cyber security analytics: a stochastic model for security quantification using absorbing Markov chains," *Journal of Communications*, vol. 9, no. 12, pp. 899–907, 2014.

[27] S. Abraham and S. Nair, "A predictive framework for cyber security analytics using attack graphs," *International Journal of Computer Networks & Communications*, vol. 7, no. 1, pp. 1–17, 2015.

[28] S. Frei, *Security econometrics—the dynamics of (in)security*, Ph.D. dissertation, Createspace Independent Pub., Scotts Valley, CA, USA, 2009.

[29] K. Durkota, V. Lisy, B. Bošansky, and C. Kiekintveld, "Optimal network security hardening using attack graph games," in *Proceedings of the International Conference on Artificial Intelligence*, pp. 526–532, Buenos Aires, Argentina, July 2015.

[30] S. Wang, Z. Zhang, and Y. Kadobayashi, "Exploring attack graph for cost-benefit security hardening: a probabilistic approach," *Computers & Security*, vol. 32, no. 1, pp. 158–169, 2013.

[31] E. Miehling, M. Rasouli, and D. Teneketzis, "Optimal defense Policies for partially observable spreading processes on Bayesian attack graphs," in *Proceedings of the ACM Workshop on Moving Target Defense*, pp. 67–76, Denver, CO, USA, October 2015.

[32] T. Cassandra, *pomdp-solve: POMDP Solver Software, v5.4*, https://rdrr.io/cran/pomdp/man/solve_POMDP.html, 2003–2015.

[33] Z. Hu, M. Zhu, and P. Liu, "Online algorithms for adaptive cyber defense on Bayesian attack graphs," in *Proceedings of the Workshop on Moving Target Defense*, pp. 99–109, New York, NY, USA, October 2017.

[34] E. Miehling, M. Rasouli, and D. Teneketzis, "A POMDP approach to the dynamic defense of large-scale cyber networks," *IEEE Transaction on Information Forensics and Security*, vol. 13, no. 10, pp. 2490–2505, 2018.

[35] S. Jha, O. Sheyner, and J. Wing, "Two formal analyses of attack graphs," in *Proceedings of the Computer Security Foundation Workshop*, Cape Breton, Canada, June 2002.

[36] T. Islam and L. Wang, "A heuristic approach to minimum-cost network hardening using attack graph," in *Proceedings of the IEEE New Technologies, Mobility and Security*, pp. 1–5, Tangier, Morocco, November 2008.

[37] L. Wang, S. Noel, and S. Jajodia, "Minimum-cost network hardening using attack graphs," *Computer Communications*, vol. 29, no. 18, pp. 3812–3824, 2006.

[38] F. Chen, L. Wang, and J. Su, "An efficient approach to minimum-cost network hardening using attack graphs," in *Proceedings of the International Conference on Information Assurance and Security*, pp. 209–212, Naples, Italy, September 2008.

[39] M. Jun-Chun, W. Yong-Jun, S. Ji-Yin, and C. Shan, "A minimum cost of network hardening model based on attack graphs," *Procedia Engineering*, vol. 15, no. 1, pp. 3227–3233, 2011.

[40] H. M. J. Almohri, L. T. Watson, D. Yao, and X. Ou, "Security optimization of dynamic networks with probabilistic graph modeling and linear programming," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 4, pp. 474–487, 2016.

[41] K. Durkota, V. Lisý, B. Bošanský, and C. Kiekintveld, "Approximate solutions for attack graph games with imperfect information," in *Proceedings of the International Conference on Decision and Game Theory for Security*, pp. 228–249, London, UK, November 2015.

[42] P. Xie, J. H. Li, X. Ou et al., "Using Bayesian networks for cyber security analysis," in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems & Networks*, pp. 211–220, Chicago, IL, USA, July 2010.

[43] J. Ghosh, H. Q. Ngo, S. Yoon, and C. Qiao, "On a routing problem within probabilistic graphs and its application to intermittently connected networks," in *Proceedings of the IEEE International Conference on Computer Communications*, pp. 1721–1729, Barcelona, Spain, May 2007.

[44] W. Segev, G. Avigdor, and E. Opher, "Inference of security hazards from event composition based on incomplete or uncertain information," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1111–1114, 2008.

[45] H. H. Nguyen, K. Palani, and D. M. Nicol, "An approach to incorporating uncertainty in network security analysis," in *Proceedings of the Hot Topics in Science of Security: Symposium and Bootcamp*, pp. 74–84, Hanover, MA, USA, April 2017.

[46] S. W. Zeng, Z. H. Wen, L. W. Dai et al., "Analysis of network security based on uncertain attack graph path," *Computer Science*, vol. 44, no. S1, pp. 351–355, 2017.

[47] M. Howard, J. Pincus, and J. M. Wing, "Measuring relative attack surfaces," in *Computer Security in Century*, pp. 109–137, Springer, Boston, MA, USA, 2003.

[48] P. Manadhata, J. Wing, M. Flynn et al., "Measuring the attack surfaces of two FTP daemons," in *Proceedings of the ACM Workshop on Quality of Protection*, pp. 3–10, Alexandria, VI, USA, October 2006.

[49] P. Manadhata and J. M. Wing, “Measuring a system’s attack surface,” *Advances in Information Security*, vol. 54, pp. 1–28, 2004.

[50] P. Samarati, “Protecting respondents identities in microdata release,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[51] L. Wang, S. Jajodia, A. Singhal, P. Cheng, and S. Noel, “k-zero day safety: evaluating the resilience of networks against unknown attacks,” in *Network Security Metrics*, pp. 75–93, Springer, Cham, Switzerland, 2017.

[52] X. Sun, J. Dai, P. Liu, A. Singhal, and J. Yen, “Using Bayesian networks for probabilistic identification of zero-day attack paths,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2506–2521, 2018.

[53] J. Somesh and J. M. Wing, “Survivability analysis of networked systems,” in *Proceedings of the International Conference on Software Engineering*, pp. 307–317, Toronto, Canada, May 2001.

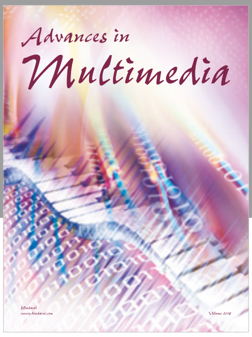
[54] N. Poolsappasit, R. Dewri, and I. Ray, “Dynamic security risk management using Bayesian attack graphs,” *IEEE Transactions on Dependable & Secure Computing*, vol. 9, no. 1, pp. 61–74, 2012.

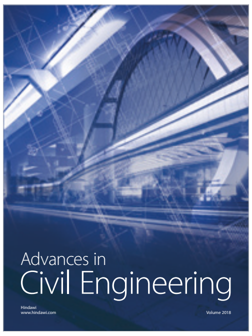
Journal of
Engineering

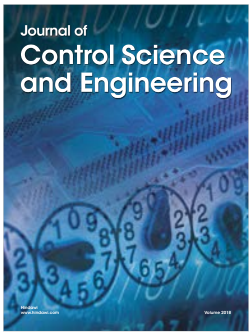
The Scientific
World Journal

International Journal of
Rotating
Machinery

Journal of
Sensors

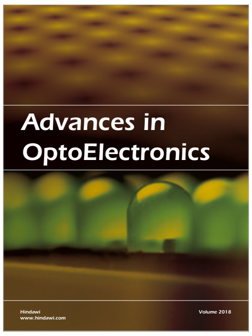

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

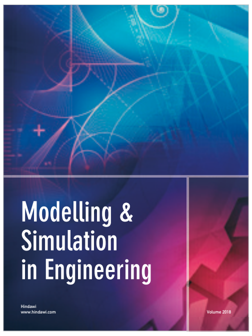
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

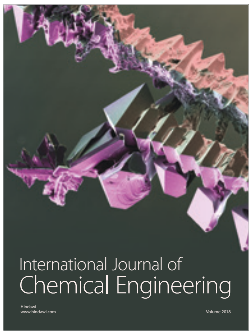

Advances in
OptoElectronics

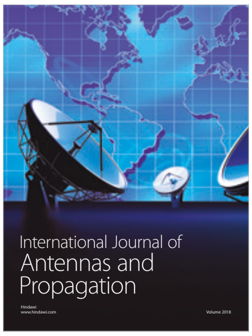
VLSI Design

International Journal of
Navigation and
Observation

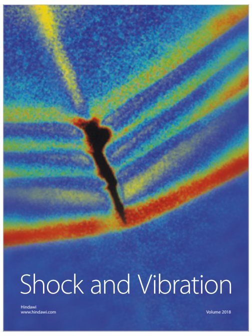

Modelling &
Simulation
in Engineering

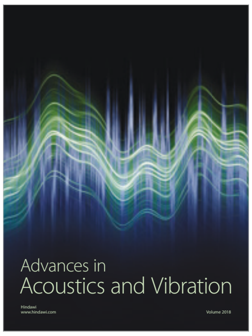
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration