Send Orders for Reprints to reprints@benthamscience.ae

The Open Cybernetics & Systemics Journal, **2015,** *9,* **953-960**

Open Access

An Approach of Security Risk Evaluation Based on the Bayesian Attack Graph

Wang Hui*, Chen Fuwang and Wang Yunfeng

College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, 454000, P.R. China

Abstract: The evaluation of network risk is a vital task. Nevertheless, there‘s no approach for looking both the severity of the vulnerabilities and the general status of network security. It can not handle with uncertainty occurred in the process of evaluation. This paper proposes a practical approach named HTV to solve the upper two problems. First, an algorithm using the Bayes Theorem is designed to check the causal dependencies of attack events and their evidence. Then, a model that we call Bayesian Attack Graph (BAG) is proposed to model the attack events and the vulnerabilities and the attack evidence so that the vulnerabilities in system could be identified correctly and availably. Finally, we adapt the definition that we call the degree threat of vulnerability (DVT) to quantify the severity of vulnerabilities. Results in experiments show that this approach can split the vulnerabilities into various levels, so it can help assess the severity of the vulnerabilities and the general status of network security availably.

Keywords: Bayesian attack graph, causal dependencies, severity, uncertainty.

1. INTRODUCTION

As networks continue to grow both in size and complexity, network security mainly caused by vulnerabilities has become a rising concern. From a security perspective, a strong network is one in which almost all of vulnerabilities can be logically identified and almost all of threats can be comprehensively evaluated. Conversely, current scanning and detecting techniques, such as X-Scan can evaluate threats, but can’t comprehensively evaluate them. They can also identify vulnerabilities, but can not availably identify them because of the hardware upgrading and software release. So it is essential to have an availably assessment method to address them.

At present, attack graphs and Bayesian network have been the most popular techniques evaluating network security [1-4]. The attack graph can be applied to identify potential vulnerabilities and comprehensively evaluate network security. In addition, it offers the capability showing all attack paths by analyzing the dependency of vulnerabilities and configurations within networks. However, as it stands in assessment process, lots of uncertainty needs to be dealt with. One of them is the causation of attacks and attack evidence. While for the attack graph, it only shows what is possible without any indication of what is likely [5]. It is often without any way to describe the non-determinism of dependency.

In contrast to the attack graph, Bayesian network has the capability of describing and dealing with uncertainty. And it is an attribute attack graph in essence. Therefore, Bayesian network is more effective than the attack graph for addressing above issues. However, it must achieve the measurement metrics of individual vulnerability.

Currently, one of the standardization of measurement metrics for an individual vulnerability is Common Vulnerability Scoring System (CVSS) [6]. It has the capability of quantifying metrics of vulnerabilities and atomic attacks and providing basic data for assessment of network security. In reality, threat level caused by different vulnerabilities is commonly different considerably in different cost and severity. So we need targeted give priority to dealing with vulnerabilities posing a worst threat to systems based on Bayesian network and CVSS.

2. LITERATURE REVIEW

Recent years have seen meaningful effort on the progress of combination of Bayesian network and the CVSS in solving above issues. There are also some methods and models on feasibility of network security assessment. Jone *et al.* [5] present a method utilizing the combination of Bayesian network and the CVSS to measure probability which vulnerabilities might be exploited successfully. Sawilla *et al.* [7] consider the importance of graph nodes for attackers to gain privileges of network and apply the Google PageRank algorithm to compute their numeric ranks. Mehta *et al.* [8] also utilize Google PageRank on different types of attack graphs to evaluate network security. While there still exists drawbacks for posing a worst threat to the system.

To alleviate such drawbacks, Nayot *et al.* [9] propose a probabilistic model to assess Dynamic Security Risk. By applying logical Bayesian network to analyze vulnerabilities, they model networks to compute probability of multi targets. However, they don’t tell us how to give a specific evaluation to overall security of networks. Anoop *et al.* [10] present a

1874-110X/15 2015 Bentham Open

method to quantity threats for an individual vulnerability based on attack graphs. They quantitatively input possibility that vulnerabilities can be exploited successfully into its interval and have an effective evaluation to network. But it fails to present an effective way to evaluate the severity of vulnerability. Holm *et al.* [11] analyze influences that basis data from CVSS have on assessment result and present a quantitative method. However, the quantitative method is so complex that it will be only applied to small networks.

In this paper, we propose a practical method named the hierarchical threat of vulnerability (HTV) to address these limitations. Our major contributions are definitions of Bayesian Attack Graph (BAG) and the degree threat of vulnerability (DTV). We also give a detection algorithm CRDA and a revised algorithm BAGA generating BAG. By defining DVT, our method can be used to divide vulnerabilities in networks into different levels and give an effective assessment to the overall security risk and the severity of vulnerability simultaneously. Compared with [10], the BAGA can be used to generate Bayesian network because it can eliminate cycle paths in attack graphs.

3. THE CORRELATION BETWEEN ATOMIC ATTACKS AND ATTACK EVIDENCE

Within networks, causation of network attacks and attack evidence will be inevitably non-determinate. To correctly account for it, we will define some notions on network security and design a detection algorithm.

Definition 1 Vulnerability: Let V be the set of vulnerabilities and $v_i \in V$. For each $v_i \in V$, We define v_i as a vulnerability in the set V.

Definition 2 Atomic Attack: Let A be the set of atomic attacks and $a_j \in A$. For each $a_j \in A$, We define a_j as an atomic attack in the set A.

Within the actual network, it is necessary to utilize a series of attack actions exploiting vulnerabilities to gain a single privilege sequentially. So we define an individual action as the atomic attack. Further, it is essentially a set composed by a number of command or operations in a sorted order. That means we can uniformly name the command or operations as elementary operations, represented within the concept as mi. If elementary operations and their built-up sequences are different, as well atomic attacks consisted of them. In Fig. (**1**), for example, $\{m_1 \rightarrow m_2 \rightarrow m_3 \rightarrow m_4\}$ and $\{m_1 \rightarrow m_2 \rightarrow m_3 \rightarrow m_4\}$ are two different atomic attacks.

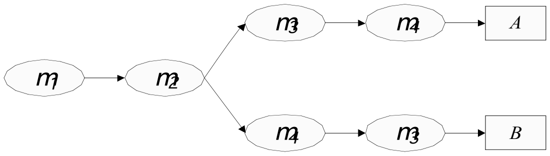


Fig. (1). Atomic attack.

Definition 3 Attack Evidence: Let O be the set of attack evidence and $o_l \in O$. For each $o_l \in O$, We define o_l as one attack evidence in the set O.

In terms of attack evidence, each of them is composed of a series of elementary operations, recorded by monitoring system as the observation data in System log. Because of system malfunction, few of elementary operations must be missing within the System log. Like that, we come to believe that atomic attacks are composed by the same elementary operations, but in different order. That is to say attack evidence from above atomic attacks is always the same. One atomic attack will induce some different attack evidence, but the occurrence of one attack evidence may also depend on multiple atomic attacks. The indeterminacy of their causation is shown in Fig. (**2**).

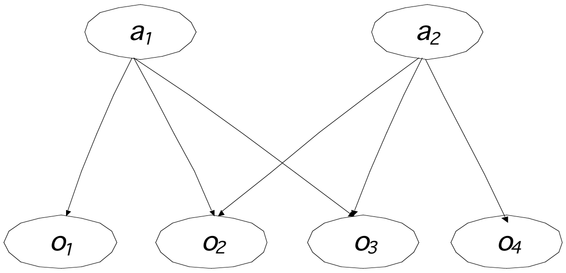


Fig. (2). Causation bayesian network.

In Fig. (**2**), we can clearly know that a_1 is likely to induce o_1 and o_2 and o_3 and that a_2 is likely to induce o_1 and o_2. However, the occurrence of o_2 and o_3 depends on atomic attacks both a_1 and a_2. The causation between o_2 and o_3 and a_1 and a_2 is their indeterminacy. That's to say we don't know whether a_1 will induce o_2 or o_3, as well a_2.

To address above indeterminacy, there is a need for an accurate metric measuring possibility that atomic attacks will actually induce attack evidence, namely $P(o_1,o_2...o_l| a_1,a_2...a_j)$. Here, we will employ following notions.

Proposition 1 Here, we assume that the set X consists of a collection of X_i and that the set Y consists of a collection of Y_m. Let the set Z be the subset of Y, namely $Z \subseteq Y$. For any integer r, with the constraints of C_1 ($1 \le r \le n$) andC_2 $(1 \le i_1 < i_2 ... < i_r)$, if X_{i1}, $X_{i2}...X_{ir}$ are independent of each other when the events which the set Z stands for happen, the probability will be computed by the following Eq:

$$P(X_{i1}, X_{i2} \dots X_{ir} \mid Z) = \prod_{r=1}^{r} P(X_{ir} \mid Z)$$

Let $O=\{o_1,o_2...o_l\}$ and $A=\{a_1,a_2...a_j\}$be the nodesets of Bayesian network ζ. We assume O is independent when events which the set A stands for happen. Thus we will logically obtain Eq. (1) based on **Proposition 1**.

$$P(o_1, o_2 \dots o_l \mid a_1, a_2 \dots aj) = \prod_{l=1}^{l} P(o_l \mid a_1, a_2 \dots aj) \qquad (1)$$

More unconditionally, for any attack evidence $o_l \in O$, if $A=\{A_1,A_2...A_l\}$is a different form of the set A, which A_l is also one subset of A, and the set A_l keeps attack evidence o_l and all subsets within $\{A_1,A_2...A_{l-1}\}$ mutual independence, then $P(o_1,o_2...o_l|a_1,a_2...a_j)$ can be further computed from Eq. (1), using the multiply:

$$\begin{aligned} P(o_1, o_2 \dots o_l \mid a_1, a_2 \dots aj) &= \prod_{l=1}^{l} P(o_l \mid a_1, a_2 \dots aj) \\ &= P(o_l \mid Al) \end{aligned} \qquad (2)$$

However, as is usually the condition of Eq. (2), it isn't easy to compute the value of $P(o_l\ |A_l)$, since only the independence of both two is really complex. Thus we will define the proportion of n_l (the number of elementary operations within o_l) and k_s (the number of elementary operations within a_j equals to n_l) to represent $P(o_l\ |A_l)$. Here, we assume that the set of k_s corresponding to $\{A_1,A_2...A_l\}$ is $\{k_1\ k_2...k_s\}$, then we obtain Eq. (3).

$$P\left(o_l \mid A_l\right) = \prod_{s=1}^{s} \frac{n_l}{k_s} \tag{3}$$

Defining the probability $P(o_1,o_2...o_l|a_1,a_2...a_j)$ as Eq. (4):

$$P(o_1,o_2 \ldots o_l \mid a_1,a_2 \ldots aj) = \prod_{l=1}^{l} \prod_{s=1}^{s} \frac{n_l}{k_s} \tag{4}$$

In Fig. (**2**), we assume that elementary operations of a_1,a_2,o_1 and o_2 as follows: $a_1=\{m_1,m_2,m_3,m_4\}$ and $a_2=\{m_5,m_6,m_7,m_8\}$, $o_1=\{m_1,m_2,m_4\}$,$o_2=\{m_1,m_2,m_7\}$.From Eq.(4), we finally know that $P(o_1|a_1)$=1,$P(o_1|a_2)$=0, $P(o_2|a_1)$=2/3, $P(o_2|\ a_2)$=1/3.

Definition 4 For atomic attacks and attack evidence, the vector $\delta_l=(\alpha_{l1},\ \alpha_{l2},\ ...\alpha_{lj})$ is employed to describe their causation. In it, α_{lj} actually represents the causation between a_j and o_l, and RDS(α_{lj}) eventually conveys the result of α_{lj}. If there exists the causation of two, then the value of RDS(α_{lj}) will equal to 1. Else it equals to 0. Further more, we utilize $\phi_{lj}=(\lambda_{l1},\lambda_{l2},...\lambda_{lj})$ to describe the posterior probability of atomic attacks corresponding to δ_l, and we use λ_{lj} to convey above posterior probability. By Eq. (3), we know $P(o_l|a_j)=n/s$(n,s, the number of elementary operations corresponding to o_l). So, the Eq. (5) is as follows based on the Bayes' theorem:

$$l_{lj} = p(a_j \mid o_l) = \frac{p(a_j).p(o_l \mid a_j)}{p(o_l)} = \frac{p(a_j)}{p(o_l)}.\frac{k_s}{n_l} \tag{5}$$

To calculate λ_{lj}, we must know the prior probability of a_j and $P(o_l)$. In fact, it has already happened when we observed o_l from monitoring system. So $P(o_l)$=1. In this case, if the value of max(ϕ) equals to λ_{lm}, there will be the result that a_m produces o_l. Here we give Algorithm **1** (causal relation detection algorithm, CRDA) to determine their causation.

The uppermost essence of procedure determining the causation is described as follows: During the inner loop FOR….DO (05-10), it begins to calculate the posterior probability of a_j which there exists causation between o_l and a_j; then the maximum value among above posterior probability is taken for the result of detection causation (11); in 13 line, the causation denoted by $R(o_l\text{-}a_j)$ is pushed intoΨ. By running Algorithm **1**, we will obtain the atomic attacks from Ψ based on observed attack evidence.

4. MODELING ATTACK MODEL WITH BAYESIAN NETWORK AND CVSS

To efficiently and synthetically identify potential vulnerabilities, we built the system architecture of attack model and model *BAG* by analyzing transitions of system states and improved an algorithm generating attack graphs.

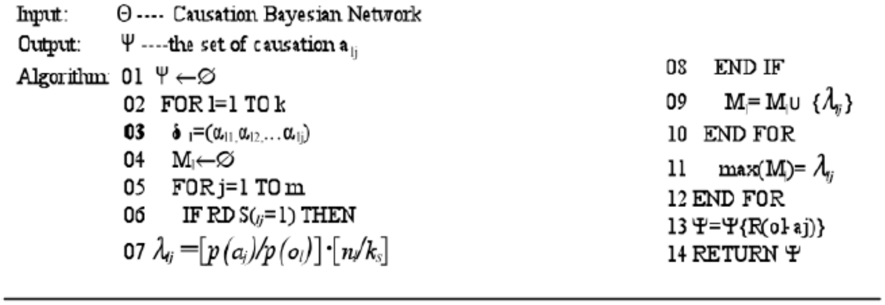
4.1. System Architecture of Attack Model

The result carrying out an atomic attack successfully is one transition of system states. So we keep the process exploiting vulnerabilities regard as transitions of system states. Under the prerequisite of network attacks, only if attackers carry out attack action and obtain privileges, can system states be transited. And there will be attack evidence with it. Hence, Let STS=(*precondition, a_j, postcondition, v_i, o_l*) be transitions of system states. In STS, the *precondition* denotes preconditions of atomic attacks. And it consists of permissions for source host *dst_access* and destination host. The *sr_l_dst* is the connection of source host and destination host. The *postcondition* denotes results of states transitions, and it means the ascension of attackers' ability. In this section, the ability will be represented by attackers' privileges which includes *None, User, Root.* Fig. (**3**) is the model of System State Transition.

The process attacking networks commonly includes a series of complex changes of system states carried out in the order loop of staging attacks and accessing to the host privileges consecutively. And it will not stop until the attacker access to the most important privileges. During it, there will be simultaneously attack evidence. So we can build the network model based on observed attack evidence and the primary steps are as follows:

First, it begins a collection of attack evidence, atomic attacks and vulnerabilities existing in networks and establishes the corresponding lists of them. Second, it must find out all possible atomic attacks yet obtained privileges by attackers

Algorithm 1. Causal relation detection algorithm (θ).



```
Input:      Θ ---- Causation Bayesian Network
Output:     Ψ ----the set of causation $a_{lj}$
Algorithm:  01 Ψ ←∅
            02 FOR l=1 TO k
            03    $\delta_l=(\alpha_{l1},\alpha_{l2},\ldots\alpha_{lj})$
            04    $M_l$←∅
            05    FOR j=1 TO m
            06       IF RDS($_{lj}$=1) THEN
            07 $\lambda_{lj}=[p(a_j)/p(o_l)]\cdot[n_l/k_s]$
            08    END IF
            09       $M_l=M_l\cup\{\lambda_{lj}\}$
            10  END FOR
            11     max(M)= $\lambda_{lj}$
            12 END FOR
            13 Ψ=Ψ{R(ol-aj)}
            14 RETURN Ψ
```

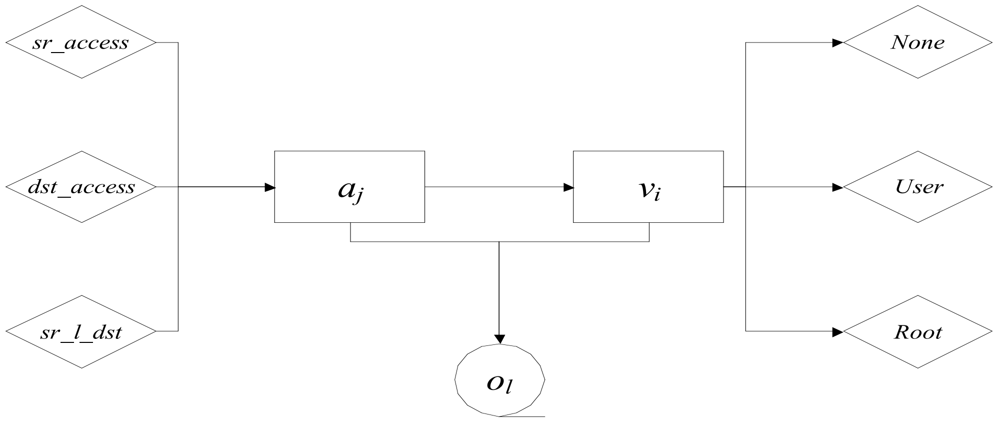


Fig. (3). System state transition.

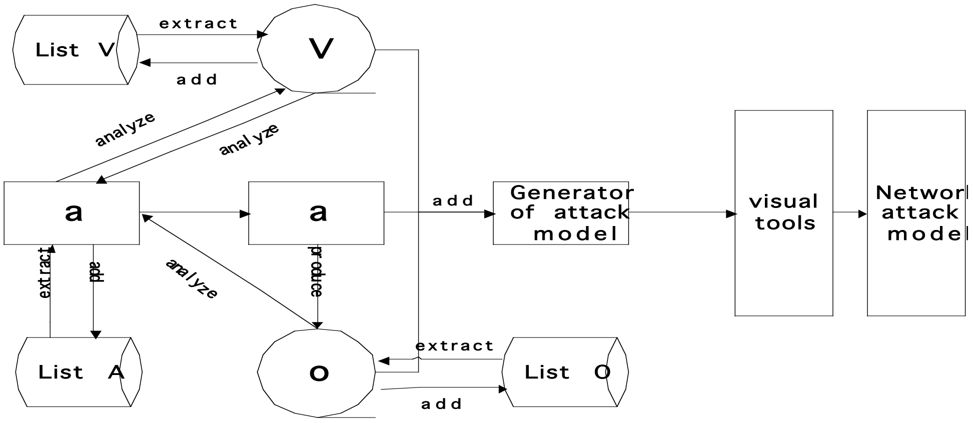


Fig. (4). System architecture of attack model.

and attack evidence so that we can identify vulnerabilities as much as possible by analyzing vulnerabilities. Finally, it should be added to the attack model. The attack model is as Fig. (**4**).

4.2. Bayesian Attack Graph

Definition 5 Bayesian Attack Graph A Bayesian Attack Graph is a tuple $BAG=(T,W,\Pi,E,C)$, where

(1) $T=V\cup A\cup O$. T is the set of different types of nodes consisted of *BAG*. $V=V_0\cup V_d$. V_0 denotes the set of initial states of nodes and V_d denotes the set of accessible states of nodes within *BAG*. For any $t\in V\cup A$, either the value of it is *True* or *False*. Here, we let $Pre(t)$ be the parent set of node t and $Post(t)$ be the subset of node t.

(2) W is the probability of o_l which is produced by a_j. For $\forall w\in W$, it's a tuple of (n,m). n denotes the number of elementary operations in o_l which is observed by monitoring system and m is the number of elementary operations within a_j equals to n corresponding to o_l. In Fig. (**2**), for example, $n(o_2)=3$ and $m(a_1,o_2)$=2, so the value of $w(a_1,o_2)$ equals to $m(a_1,o_2)/n(o_2)$=2/3.

(3) $\Pi(BAG)\subset\{P(t)\cup P(t|Pre(t))\}$, is the set of probability distribution on nodes within *BAG*.

(4) $E\subset([(V_0\cup V_d)\times A]\cup[A\times(V_0\cup V_d)]\cup[A\times O])$, denotes the set of directed edges associated with nodes in *BAG*.

(5) C denotes the constraints that must be satisfied for the *BAG*, where

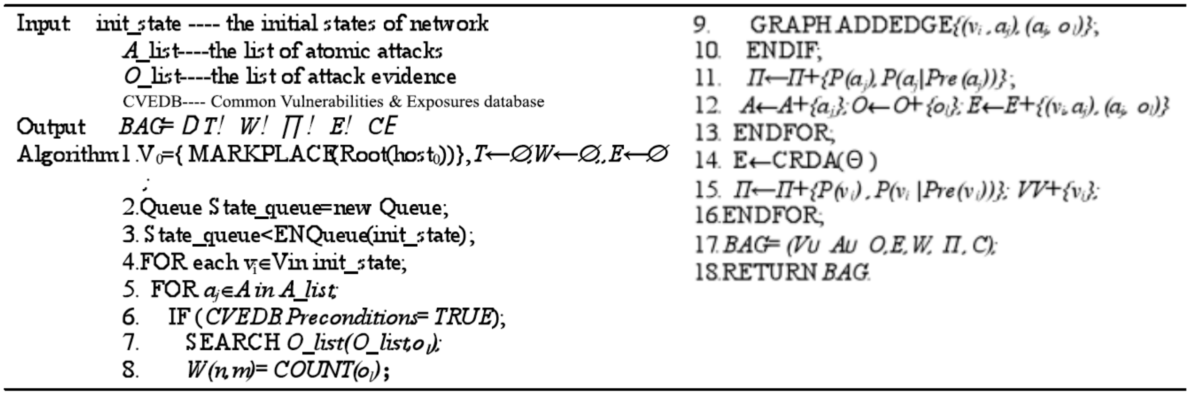
a) For any node $t\in A$, their parent nodes represent within *BAG* as logical *OR-nodes* and satisfy with the constraint $(Pre(t)\subset V_0)\wedge(Post(t)\subset V_d)$

b) For any node $t\in V$, their parent nodes represent within *BAG* as logical *AND-nodes* and satisfy with the constraint $(\exists t\subset A)\vee(Post(t)\subset A);$

c) For any node $t\in T$, it always satisfies with the normative constraint of probability measure, namely $\Pi(BAG)\subset\{P(t)\cup P(t|Pre(t))\to[0,1]$.

Algorithm **2** is used to generate Bayesian Attack Graph automatically. Its' procedure is described as follows: First, it provides a process initializing all nodes and parameters within *BAG* (1-3). The 1^{st} line sets up a list of privileges by calling the MARKPLACE process and gives a blank state to *BAG*. It also gives the attackers access to Root. In line 2~3, it creates a queue in network state and adds the *init_state* to the queue to initialize the *BAG*. *Pseudocodes*4-16 is the second stage with loops. The 4^{th} line provides an outer loop to extract nodes from *init_state*. *Pseudocodes*5-13 is an inner loop. For any nodes in the *init_state*, we traverse all atomic attacks associated with it in the *A_list*(5). If the atomic attack attacks the node according to CVEDB(6), it will find out the attack evidence produced by it from the *O_list*(7~8) and

Algorithm 2. Bayesian attack graph generation algorithm BAGA.



```
Input     init_state ---- the initial states of network
          A_list---the list of atomic attacks
          O_list---the list of attack evidence
          CVEDB---- Common Vulnerabilities & Exposures database
Output    BAG= D T! W! ∏! E! CE
Algorithm 1.V_0={ MARKPLACE(Root(host_0))},T←∅,W←∅,E←∅
          ;
          2.Queue State_queue=new Queue;
          3. State_queue<ENQueue(init_state);
          4.FOR each v_i∈V in init_state;
          5.  FOR a_j∈A in A_list;
          6.    IF (CVEDB.Preconditions= TRUE);
          7.      SEARCH O_list(O_list,o_l);
          8.      W(n,m)= COUNT(o_l);
9.        GRAPH ADDEDGE{(v_i , a_j), (a_j, o_l)};
10.     ENDIF;
11.     Π←Π+{P(a_j), P(a_j|Pre (a_j))};
12.    A←A+{a_j}; O←O+{o_l}; E←E+{(v_i, a_j), (a_j, o_l)}
13.  ENDFOR;
14.  E←CRDA(Θ )
15.  Π←Π+{P(v_i) , P(v_i |Pre(v_i))}; VV+{v_i};
16.ENDFOR;
17.BAG= (V∪ A∪ O,E,W, Π, C);
18.RETURN BAG.
```

make them the node *T*(11~12)and their edges *E*(9). The loop will continue until the *init_state* is null. Finally, it will add above nodes *T* and edges *E* to *BAG* and return the *BAG* eventually (17~18).

According to the real-time observed attack evidence, Algorithm **2** can automatically generate Bayesian network consisted of nodes of atomic attacks, attack evidence and vulnerabilities and show vulnerabilities existing in networks, which makes it effectively identify potential vulnerabilities.

When generating *BAG* by running Algorithm **2**, there will be two issues: 1) Privileges, such as *User*, are the fallout exploiting vulnerabilities, but will not be a type of nodes in its own right; 2) There are relationships of one-to-one correspondence among vulnerabilities, atomic attacks and attack evidence each other, but it can't display the relationship what reflects their causation in *BAG*. So, we make up rules alleviating above drawbacks.

① We name nodes of vulnerabilities as "vulnerability number(v_i)". For example, CVE575(v_1) means the vulnerability v_1 within networks is *Buffer Overflow.*

② We name privilege as "*access level(Host)*". For example, R(H0) means that attackers have obtained the Root privilege from Host0.

③ We name nodes of privileges as "*access level(Host)_ a_j_vulnerability(v_i)*". For example, *R(H0)_a_2_ CVE575(v_1)* means attackers will attack v_l using a_2 when he has obtained the Root privilege from Host0 by exploiting *Buffer Overflow Vulnerability*.

④ We name nodes of attack evidence as *"a_i_v_l_o_l"*. For example, a_2_ v_1_o_1 means that it will produce o_1 when *Buffer Overflow (v_1)* within networks is exploited by a_2.

5. ASSESSMENT OF THREAT WITH BAG

We must targeted give priority to dealing with vulnerabilities posing a worst threat to systems considerably in different cost and severity. Here, we define DVT and formulas to quantifying the severity of vulnerabilities.

5.1. Probability of nodes in BAG with CVSS

Definition 6 The degree of vulnerability threat (DVT) is a function mapping each vulnerability in *BAG* to probability interval *[0,1],* described mathematically *p: V→[0,1].* It can be used to measure the level of privilege attackers will obtain by exploiting vulnerabilities. The larger of its value, the higher of the level of privilege is and the more severe of vulnerabilities are.

The *BAG* could be effectually utilized to assess the severity of vulnerabilities within networks. To accurately measure DVT, it is prioritized and necessary to assign self-probability of atomic attack node *s(a_j)* and vulnerability node *s(v_i),* which their self-probability are from the attribute "*Access-Complexity*" of CVSS. For any vulnerability, the value of its self-probability is commonly steadfast. So we make the rule that *s(v_i)* equals to 1.0. The values of *s(a_j)* in our network are as Table **1**.

Table 1. Basic score of CVSS.

Attribute of CVSS	Level	S(a_j)
Access Complexity	Low	0.71
	Medium	0.61
	High	0.35

5.2. Hierarchical threats of vulnerabilities

More generally, the probability of nodes in Bayesian network associate with the attribute of its parents and itself. In this section, for any node t within Bayesian network, we assume that the number of nodes that are independent of each other is *K* in the set *pre(t)* and the array *d[i](1≤i≤k)* refers to the node number. Based on Bayes' theorem:

$$P(t) = (t \mid pr(t)) = p(t \mid pr\left(t\right), pr\left(t\right) \to t) \tag{6}$$

Considering the logical relation d_t among its parents, the probability of node *t*is defined as Eq. (7):

$$p(t) = \begin{cases} P(\bigcap_{i=1}^{k} d[i]\ ) = \prod_{i=1}^{k} P(d[i]) & dt = AND \\ p(\bigcup_{i=1}^{k} d[i]\ ) = \prod_{i=1}^{k} \left\{1 - P(d[i])\right\} & dt = OR \end{cases} \tag{7}$$

Definition 7 For any *t∈T(BAG)* within *BAG,* the absolute probability *p(t)* is defined as follows:

$$P(t) = \begin{cases} 1.0 & \forall t \in root; \\ \dfrac{s(t)\bullet \prod_{i=1}^{k} p[d(i)].p(o_l \mid t)}{p(o_l)} & \forall t \in A; \\ s(t)\bullet[1 - \prod_{i=1}^{k}(1 - p(d(i)))] & \forall t \in V_0 \cup V_d \, AND \\ & \forall t \in root \end{cases} \tag{8}$$

To accurately calculate DVT considerately in attack evidence, we discuss the probability distribution of 3 kinds of nodes. (1) root nodes. Root nodes represent one kind of nodes that have been successfully exploited in the beginning state of networks. So the actual probability $p(t)=1.0$.(2) $\forall t\in A$. The probability of atomic attacks associates with its parents, attacks evidence and itself. So the actual probability is $p(t)= s(t)\cdot \prod_{i=1}^{k} p[d(i)]\cdot p(o_l \mid t) / p(o_l)$. (3) $\forall t\in V_0\cup V_d\, AND\ \forall t\notin root$. The probability of vulnerabilities exploited successfully associates with its parents, and itself. So the actual probability is $s(t)\cdot[1 - \prod_{i=1}^{k}(1 - p(d(i)))]$.

Because of the risk and crime cost, only the path maximizing the success rate exploiting vulnerabilities may be selected to reach destination nodes by attackers eventually. Hence, we define the function of DVT based on Definition4 and Definition6 as Definition8:

Definition 8 Given $BAG=(T, W, \Pi, E, C)$, for any node $t\in V_0\cup V_d$, p_{max} represents its DVT and $p_{max}=s(t)\cdot MAX\{\dfrac{P[d(i)]}{p(t)} \mid d(i)$ is a parent of node t and $i=1,2,3,\cdots k\}$.

In fact, $p[d(i)]/p(t)$ is the largest success rate exploiting vulnerabilities among all nodes of $p[d(i)]$. So p_{max} can be used to measure the most possible threats of vulnerabilities exploited successfully.

6. IMPLEMENTATION AND EXPERIMENTAL RESULTS

6.1. Implementation and Network Configuration

In this section, the validity of our method is verified by a simulation. Fig. (**5**) shows a sample network, which will be used to demonstrate the implementation of our method.

Network Configuration: Experimental network is separated internal network and external network by Firewall1. The internal network includes the secured area of **M1, M2,**

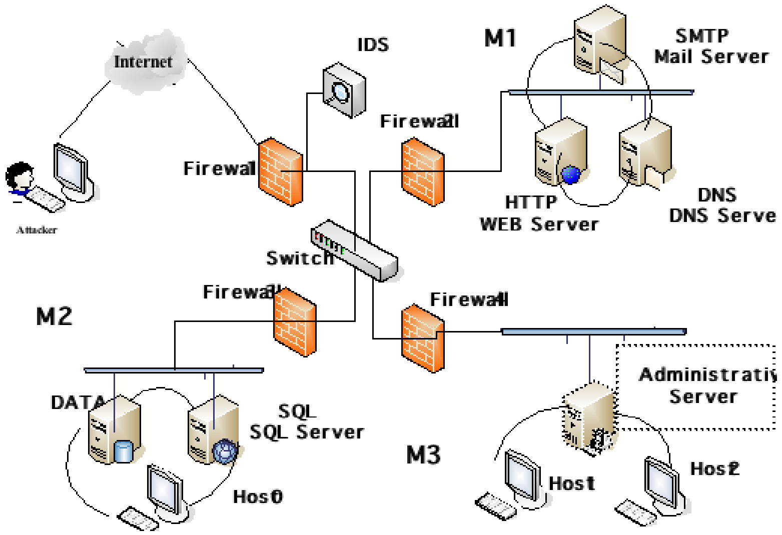


Fig. (**5**). Example experimental network.

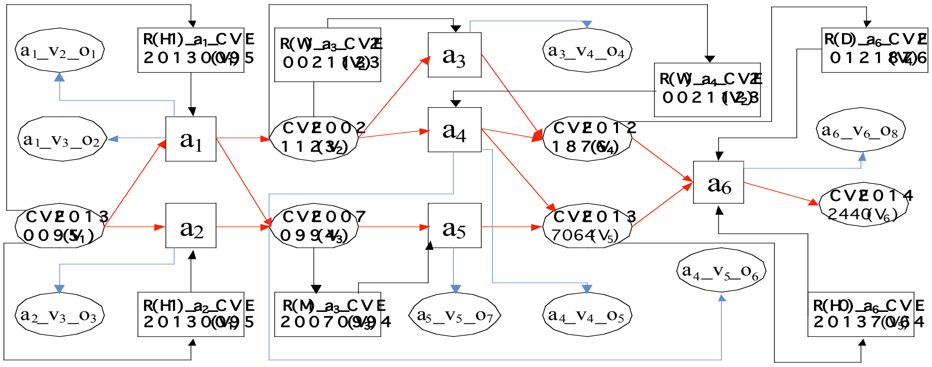


Fig. (**6**). The bayesian attack graph.

Table 2. Vulnerability information of host.

Secured Area	Node	Host	Vulnerability	Service	CVE	Type of Privilege	Score of Access Conexity
MB	V_1	Host 1/ Host 2	Unintended Content Loading Vulnerability	RSH/ HTTP/SQL	CVE-2013-0095	Non/User	0.71
M1	V_2	Web Server	Buffer overflow in the authentication function	HTTP	CVE-2002-1123	User	0.71
	V_3	Mail Server	Remote code execution in SMTP	SMTP	CVE-2007-0994	User/Root	0.61
	V_4	DNS Server	Col Element Remote Code Execution Vulnerability	DNS	CVE-2012-1876	Non/User/Root	0.61
M2	V_5	Host 0/ DATA	Cross-site scripting (XSS) Vulnerability	HHS/FTP	CVE-2013-7064	Root	0.35
	V_6	SQL Server	Unspecified vulnerability in the MySQ	SQL	CVE-2014-2440	User/Root	0.35

M3, which are separated by firewall each other. The external network includes the Internet and several hosts which provide services as Fig. (**5**), and the attack host0 is the destination host.

6.2. Experimental Results

Vulnerabilities within networks will enable an attacker to gain control of hosts. In this section, the vulnerabilities and servers within the sample network are shown in Table **2** and the *BAG* generated by running Algorithm **2** is shown in Fig. (**6**).

With the growing of intrusion for networks, the ability of attackers obtaining higher permissions is continuously improved. Within networks, the IDS deployed on the mirror port can detect potential intrusion behavior and analysis host permissions gained by attackers by collecting and analyzing various network data. To hierarchically assess the severity of vulnerabilities, we will introduce several parameters in the experimental process as Table **3**.

Table 3. Parameters of running experiment.

Test	*N*	q_R	q_U	q_N
1	100	0.4	0.4	0.2
2	200	0.5	0.3	0.2
3	300	0.6	0.3	0.1

The number of atomic attacks N: It represents the number of intrusion behavior which has recorded by the IDS in an experiment.

Attack rate q: It represents the percentage of atomic attacks used to obtain a level of privilege. It will include three kinds of attack rate: Root rate (disproportionately, set q_R), User rate (disproportionately, set q_u) and None rate (disproportionately, set q_N). Root rate means that the percentage of atomic attacks used to obtain Root privilege is q_R and q_R*=n(Root)/[n(Root)+ n(User)+ n(None)]*. Like that, q_u*=n(User)/[n(Root)+ n(User)+ n(None)]* and q_N *=n(None)/[n(Root)+ n(User)+ n(None)]* . Considering the relation of Attack rate and the level of privilege, we will assume that only if the value of $p_{max}(v_i)$ is between q_U+q_N and $q_U+q_{N+}q_R$ (intervally, $p_{max}(v_i)\in[q_U+q_N, q_U+q_{N+}q_R]$), can attackers enable to obtain the Root privilege. And likewise, we know $p_{max}(v_i)\in[q_U+q_N, q_N]$ for User privilege and $p_{max}(v_i)\in[q_N, 0]$ for None privilege.

Fig. (**7**) graphically shows DVT within networks. The blue line represents DVT. The minimum of Root privilege is represented by q_R with purple line and the ceiling is 1.0. The minimum of User privilege is represented by q_N with red line and the ceiling is represented by q_U with black line. The ceiling of None privilege is represented by q_N and the minimum is 0. For example, if $p_{max}(v_i)\in[q_R,1.0]$, we think that attackers will obtain Root privilege by exploiting v_i.

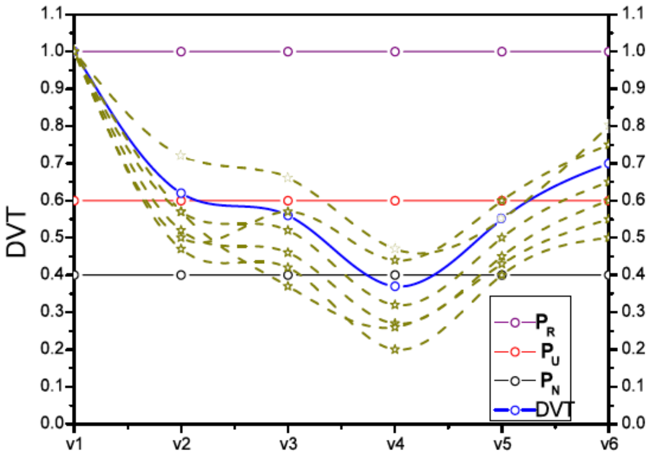


Fig. (7). The severity of vulnerabilities.

In Fig. (**7**), vulnerabilities are hierarchically divided into 3 groups. (1) v_1, v_2, v_6. Their values of DVT are in the interval *[0.6,1.0]* and they will be exploited by attackers to obtain Root privilege in networks. (2) v_3, v_5. Their values of DVT are in the interval *[0.4,0.6]* and they will be exploited by attackers to obtain User privilege in networks. (3) v_4. Its value of DVT is in the interval *[0,0.4]* and attackers will not obtain privilege by exploiting it in networks. In crosswise comparison, if we sort by severity of vulnerabilities, the order of v_1,v_2,v_6 should be v_1,v_6,v_2. And likewise, the order of

v_3,v_5 should be v_3,v_5. Further more, in longitudinal comparison, if we sort by severity of vulnerabilities, the order of them should be v_1,v_6,v_2, v_3,v_5, v_4. So we can draw a conclusion from Fig. (**7**) that the priority to dealing with vulnerabilities is v_1,v_6,v_2, v_3,v_5, v_4.

Fig. (**8**) graphically shows the result of assessment to the overall security of networks. There are overall three lines: black line represents the threshold threatening to destination host; red line A represents the trend that the DVT of v_6 has with the growing of atomic attacks when Host0 is taken for the destination host; blue line B represents the trend that the DVT of v_6 has with the growing of atomic attacks when Host0 is not taken for the destination host.

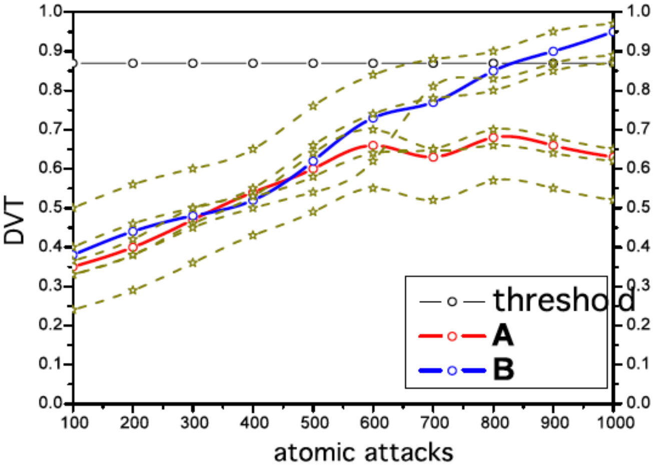


Fig. (8). The overall security of target network.

The line A shows that the DVT of v_6 first becomes larger and then tends to stability. It demonstrates that for the non-destination host, attackers don't try to obtain privilege with higher level. So, the trend of overall security of networks in the situation will be the same as the line A. The B shows that the DVT of v_6 has been growing until it is more than threshold. It suggests that the attacker will try to obtain the highest authority of destination host. Therefore, our method can support the network early warning for managers so that they will make a clear decision.

CONCLUSION

The paper proposes a method to evaluate overall security risk of networks and severity of vulnerabilities simultaneously by using Bayesian network and CVSS. The method is practical in the assessment of them, as well as determination of the causation between atomic attacks and attack evidence. By running it, we can give the different priority to dealing with vulnerabilities and support the network early warning for managers so that they will make a clear decision.

For future work, we will further study on the uncertainty within networks and verify the effectiveness of our method in more complex networks.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work was financially supported by the National Natural Science Foundation of China (No. 61300216), supported by Research Fund for the Doctoral Program of Higher Education of China (No. 20124116120004), and supported by Educational Commission of Henan Province of China (No. 13A510325).

REFERENCES

[1] B. Mahmoud, N. Larrieu, and A. Pirovano, "A Risk Propagation Based Quantitative Assessment Methodology for Network Security Aeronautical Network Case Study", In: *Proceedings of the Integrated Communications, Navigation and (ICNS)*, 2011, pp. 1- 9.

[2] M. Frigault, and L.Y. Wang, "Measuring network security using Bayesian network-based attack graphs", In: *Proceedings of the 3rd IEEE International workshop on Security, Trust, and Privacy for Software Applications*, 2008, pp. 698-703.

[3] A.M. Xie, W.P. Wen, L. Zhang, J.B. Hu, and Z. Chen, "Applying Attack Graphs to Network Security Metric", In: *Proceedings of the Multimedia Information Networking and Security (MINES)*, 2009, pp. 427-431.

[4] M.Salim, E. Al-Shaer, and L. Khan, "A novel quantitative approach for measuring network Security", In: *Proceedings of the 27th Conference on Computer Communications (INFOCOM)*, 2008, pp. 1957-1965.

[5] J. Homer, X.M Ou, and D. Schmidt, "*A Sound and Practical Approach to Quantifying Security Risk in Enterprise Networks*", *Kansas State University Technical Report*, 2009, pp 1-15.

[6] P. Mell, K. Scarfone, and S. Romanosky, "Common Vulnerability Scoring System", *Security & Privacy, IEEE*, vol. 4, no. 6, pp. 85-89, 2006.

[7] R. Sawilla, and X.M. Ou, "Identifying critical attack assets in dependency attack graphs", In: *Proceedings of the 13th European Symposium on Research in Computer Security*, 2008, pp. 18-34.

[8] V. Mehta, C. Bartzis, H. Zhu, E. Clarke, and J. Wing, "Ranking attack graphs", In: *Proceedings of Recent Advances in Intrusion Detection*, 2006, pp. 127-144.

[9] P. Nayot, D. Rinku, and R. Indrajit, "Dynamic Security Risk Management Using Bayesian Attack Graphs", *Dependable and Secure Computing, IEEE Transactions*, vol. 34, no. 1, pp. 61-74, 2011.

[10] A. Singhal, and X.M. Ou, "*Security Risk Analysis of Enterprise Networks Using Attack Graphs*", SpringerBriefs in Computer Science, Springer, New York, 2011, pp. 13-23.

[11] H. Holm, M. Ekstedt, and D. Andersson, "Empirical Analysis of System-Level Vulnerability Metrics through Actual Attacks", *Dependable and Secure Computing, IEEE Transactions*, vol. 9, no. 6, pp. 825-837, 2012.

Received: June 10, 2015 Revised: July 29, 2015 Accepted: August 15, 2015