Journal of Network and Systems Management
https://doi.org/10.1007/s10922-020-09558-5

Bayesian Decision Network-Based Security Risk Management Framework

Masoud Khosravi-Farmad[1] · **Abbas Ghaemi-Bafghi**[1]

Received: 19 August 2019 / Revised: 30 May 2020 / Accepted: 17 July 2020

Abstract
Network security risk management is comprised of several essential processes, namely risk assessment, risk mitigation and risk validation and monitoring, which should be done accurately to maintain the overall security level of a network in an acceptable level. In this paper, an integrated framework for network security risk management is presented which is based on a probabilistic graphical model called Bayesian decision network (BDN). Using BDN, we model the information needed for managing security risks, such as information about vulnerabilities, risk-reducing countermeasures and the effects of implementing them on vulnerabilities, with the minimum need for expert's knowledge. In order to increase the accuracy of the proposed risk assessment process, vulnerabilities exploitation probability and impact of vulnerabilities exploitation on network assets are calculated using inherent, temporal and environmental factors. In the risk mitigation process, a cost-benefit analysis is efficiently done using modified Bayesian inference algorithms even in case of budget limitation. The experimental results show that network security level enhances significantly due to precise assessment and appropriate mitigation of risks.

Keywords Risk assessment · Risk mitigation · Risk management framework · Cost-benefit analysis · Decision making · Bayesian decision network

1 Introduction

In today's complex computer networks, one of the main challenges of the network security administrators is to identify, assess and prioritize the security risks to their network assets and also to determine appropriate mitigation strategies to address

✉ Abbas Ghaemi-Bafghi
ghaemib@um.ac.ir

Masoud Khosravi-Farmad
m.khosravi@mail.um.ac.ir

[1] Data and Communication Security Lab., Computer Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran

Published online: 03 August 2020

these risks. The net negative impact of the exploitation of a security vulnerability is called security risk which considers both the probability and the impact of vulnerability exploitation. Security risk management is the process of risk assessment, risk mitigation and risk validation and monitoring with the aim of minimizing or eliminating the potential risks in the systems [1–3]. Risk assessment process refers to identification and evaluation of risks, and also recommendation of risk-reducing countermeasures. In this classification, the risk assessment process includes risk analysis, which is the process of measuring the probability of vulnerabilities exploitation and their expected impact. Risk mitigation process includes prioritizing, implementing and maintaining the appropriate countermeasures recommended from the risk assessment process. Risk validation and monitoring is a continual process which determines whether the residual risk is at an acceptable level or whether there is a need to implement additional countermeasures to further reduce or eliminate the residual risk.

Several techniques for identifying and measuring the characteristics of individual vulnerabilities are available, such as the Common Vulnerability Scoring System (CVSS) [4]. While these techniques are helpful to assess vulnerabilities, they have a major limitation, which is that they only focus on individual vulnerabilities and do not consider the interactions between them. This limitation is serious, because in order to compromise network assets, attackers generally exploit sequences of related vulnerabilities. Such attacks are called multi-step attacks which can be clearly demonstrated using security models such as attack graphs (AGs) [5, 6].

One of the main drawbacks of AGs is that they give no information about the probability of exploiting vulnerabilities, nor their severity level [7]. These two parameters are essential factors for doing risk assessment. So, it is difficult to assess the risks caused by multi-step attacks on the network assets using only AGs.

Bayesian networks are powerful tools that can represent information about causal relationships between vulnerabilities. They also provide a more compact representation of AGs, yet still keep necessary information about vulnerabilities such as their probability of exploitation. Moreover, these networks provide a formalism for reasoning about partial beliefs under conditions of uncertainty [8]. To take advantage of the benefits of Bayesian network concept, we can convert AGs into Bayesian Attack Graphs (BAGs) [9], so possible multi-step attacks can be demonstrated and the uncertainties about probabilities of attacker actions can be captured in the model.

The main shortcoming of BAG is that it doesn't provide any information about essential characteristics of possible security countermeasures such as their coverage, implementation cost and expected outcome, which are needed for performing risk mitigation. In the proposed framework, using BDN, the BAG model is modified and augmented to make the risk mitigation possible in an integrated manner.

Exploitation probability of vulnerabilities may change during network lifetime. For instance, public availability of tools and techniques for vulnerability exploitation increases the number of potential attackers by including those who are unskilled; thereby, the exploitation probability of the vulnerability increases. Also, the current available options for vulnerability remediation, the confidence level about vulnerability existence, the technical knowledge about exploitation available to the public

and their credibility are important factors which influence the number of potential attackers and thereby, the exploitability of vulnerabilities changes [4]. In the proposed framework, these changes are handled considering temporal characteristics of vulnerabilities. So the result is more accurate and closer to the reality.

There are also various environmental factors which can affect the impact that a vulnerability imposes to an organization assets [4]. Therefore, the security administrator may want to modify and adjust the impact of vulnerability exploitation according to the importance of confidentiality, integrity and/or availability requirements of the affected IT assets. In the proposed framework, the impact of vulnerability exploitation is calculated considering the environmental factors affecting the security requirements of the assets.

In this paper, BDN is used to model the security properties of computer networks, therefore, in addition to demonstrating potential multi-step attacks, it is also possible to model the security countermeasures and their characteristics for performing risk mitigation. Also by employing Bayesian network concept in the model, it is possible to capture uncertainties about attacker actions. Moreover, by using BDNs, security administrators are able to define risk-reducing countermeasures covering vulnerabilities in the network, the cost of implementing these countermeasures and their expected outcome. Finally, using the BDN model employed in the proposed framework, a cost-benefit analysis is conducted which enables the network security administrators to identify the optimal subset(s) of security countermeasures even if the allocated budget for network hardening is limited.

Briefly, the main contributions of this paper are as follows:

- A BDN-based integrated framework for security risk management is proposed which encompasses all the information required for managing risks within it.
- The uncertainty in countermeasures coverage level is modeled as a probabilistic value, rather than just a Boolean value.
- An algorithm for converting BAG model into the BDN model, along with a formal definition for utility tables in BDN model and a method for filling their entries are proposed.
- A variable elimination-based algorithm for conducting cost-benefit analysis is also presented on the proposed BDN model which identifies the optimal subset(s) of countermeasures even in case of budget limitation.
- Finally, a feasibility analysis is conducted using a case study.

The rest of the paper is organized as follows: The next section presents a brief review on related work. Section 3 presents the proposed framework for security risk management. Results of applying the proposed method on a test network are presented in Sect. 4. A discussion is presented in Sect. 5. Finally, in the last section we conclude the paper and point out our future research.

2 Related Work

Our related work includes two streams of literature: (1) security risk assessment and (2) security risk mitigation. These are presented in Sect. 2.1 and 2.2, respectively.

2.1 Security Risk Assessment

A variety of cybersecurity risk management methodologies have been developed for assessing risks in IT systems, thereby, enabling systems security administrators to make correct decisions towards mitigating the most important risks in the operational environments.

AG-based security assessment models are becoming an important part in both qualitative and quantitative risk management activities since they can model multi-step attacks by representing possible paths attackers can use to penetrate into IT systems [10–14]. Most of quantitative AG-based methods use CVSS scores [4] as the probability of successful exploitation of vulnerabilities [9, 15–17]. Among them, some methods propagate these probabilities (CVSS scores) through the AG according to its conjunctive and disjunctive dependencies, such as [18–20], and some others use the probabilistic approaches like Bayesian networks.

Liu and Man are one of the pioneers in applying Bayesian networks for network vulnerability assessment [8]. They model potential attack paths using Bayesian networks and represent the security of the network by a quantitative value. Frigault and Wang [21] use Bayesian networks with AGs to calculate general security metrics regarding information system networks. The resulting model is called Bayesian attack graph (BAG) which contains all nodes of the original AG. These nodes are populated based on CVSS Base Scores as the probability values encoded in the conditional probability tables. There are some other works that use Bayesian network concept for assessing network security risks and capturing uncertainties in unknown attacker behaviours, such as [22, 23]. Later, in 2012, Poolsappasit et al. [9] revisited the BAG model and extended it by assigning a disjunctive or a conjunctive identifier to nodes with at least two incoming edges. They also proposed a method to assess the security risk of vulnerability exploitations based on CVSS Base metrics. Moreover, they provide a platform for static and dynamic analysis of risks in networked systems. In [24], authors propose a security risk analysis model based on Bayesian networks which determines the attack paths with the highest probability and the largest estimated risk value using ant colony optimization algorithms. Bayesian networks are also used in [25, 26] to implement Factor Analysis of Information Risk (FAIR) which is one of the popular models for security risk assessment.

Most of recent works on risk assessment use the Base Score of CVSS as the vulnerability exploitation probability, such as [7, 15, 16, 27, 28]. According to the CVSS's documentation [4], the CVSS's Base Score, like its Temporal and Environmental Scores are risk values. The risk of a vulnerability exploitation can be calculated by multiplication of its probability in its impact [29, 30]. A Bayesian network is a probabilistic graphical model in which its nodes are assigned Conditional Probability Tables (CPTs)[31, 32]. These tables list the probability values for each joint

assignment to nodes and their parents. In fact, each CPT is a table that has one probability value for every possible combination of parent and child states. Therefore Bayesian networks should be applied only on the probability values.

The probability of vulnerabilities exploitation may change over time. Different environments can also have an influence on the impact of vulnerability exploitations. Using only the Base Scores of CVSS is not enough to evaluate these changes, since the Base metrics of CVSS only capture the inherent properties of vulnerabilities. Hence, we need to take into account temporal and environmental characteristics of vulnerabilities.

The proposed risk assessment method decomposes the risk into probability and impact and propagates only the vulnerability exploitation probability through the model. Then the risk can be calculated by multiplying the propagated probability in the exploitation impact. The vulnerability exploitation probability is calculated using appropriate Base and Temporal metrics of CVSS and the result is then propagated throughout the model. The impact on organization's assets caused by exploiting vulnerabilities is computed considering the appropriate Environmental metrics of CVSS which adjusts the impact of exploitation on the security requirements, such as confidentiality, integrity and availability.

2.2 Security Risk Mitigation

Several approaches have been presented for addressing the risk assessment process, such as [33–38], while ignoring the risk mitigation process, which is a crucial step in security risk management. In order to successfully manage the identified risks, security administrator needs a thorough understanding of the magnitude of the risks in the network and also their covering countermeasures. Only in this case he can conduct a cost-benefit analysis to find appropriate countermeasures to mitigate the most risky threats.

In the field of security risk mitigation, several methods have been proposed to determine safeguards and countermeasures to improve the security level of networks, such as [39–41]. In [42], minimum-cost hardening measures are identified using exploit dependency graphs. In [43], the minimal subset of attacks that are necessary for reaching a goal is determined. After that, the minimal subset of countermeasures that covers the subset of attacks is identified.

The mentioned techniques are useful, but they miss out one major issue. Most of the times, the allocated budget for network security hardening is limited, which may preclude the security administrators from implementing all possible countermeasures or even certain measures that cover all of the vulnerabilities. Therefore, there is a need to find a trade-off between the implementation cost of a subset of countermeasures and the residual damage after the security decisions have been made and these countermeasures are implemented. This problem is first formulated by Dewri et al. [44] as a multi-objective optimization problem on the attack tree model of the networked system. After that, an evolutionary algorithm is used to solve the problem. The main shortcoming about this work is that the modeling of the problem is

static. The probability of vulnerability exploitation may change during the lifetime of a network. Therefore, the model should be dynamic and be able to consider these changes and their effects. This problem is solved in [9] by revisiting the framework of BAG. The authors augment BAG with additional nodes and values representing defenses. After that, they solved the problem of finding the set of optimal defenses using a multi-objective optimization problem.

The main shortcoming of the aforementioned methods is that they don't provide an integrated framework for network security risk management which contains all the information needed for risk assessment and risk mitigation within itself. Therefore, there is a necessity for expert's knowledge for doing risk mitigation. Moreover, these methods need special algorithms, i.e. heuristic algorithms, for inferring the best set of countermeasures.

A BDN model [45] is an extension of BAG model [9] which allows administrators to model security countermeasures and their properties using the nodes and their assigned data structures provided by the model. We use this model as part of the proposed security risk management framework. Therefore, we can take advantage of standard Bayesian inference algorithms in the proposed framework.

In this paper, an integrated security risk management framework is presented which utilizes the BDN model for efficiently doing both risk assessment and risk mitigation processes together. Formal definition of BDN model along with an algorithm for converting BAG models into their corresponding BDN models is proposed. Moreover, an algorithm for conducting Cost-Benefit Analysis over the BDN models is presented. In fact, the proposed framework covers most of the aforementioned drawbacks and can be used independently to conduct a complete risk management without the need for extra information from experts.

3 Proposed Risk Management Framework

The proposed risk management framework is made up of three main steps, namely, risk assessment, risk mitigation and risk validation and monitoring. The risk assessment phase starts with modeling network attacks using AG. Then, the temporal probability of successful vulnerability exploitation is calculated for each vulnerability in the network. After that, the AG model is converted into BAG to change it into a quantitative model. Finally, the adjusted environmental impact of vulnerabilities exploitation is calculated. The risk mitigation phase starts with identifying and assessing possible countermeasures to mitigate the exploitability of vulnerabilities. Then, using this information, the BAG model can be converted into the BDN model by adding countermeasures to it and filling its utility tables. Finally a cost-benefit analysis is conducted to recommend the optimal subsets of countermeasures even if the allocated budget for network hardening is limited. In the risk validation and monitoring phase, the changes in network states are continually tracked to make sure that the residual risk matches the desired risk level. The data flow diagram of the main phases of the proposed security risk management framework is depicted in Fig. 1.

In the following sections, each step shown in Fig. 1 is explained in detail.

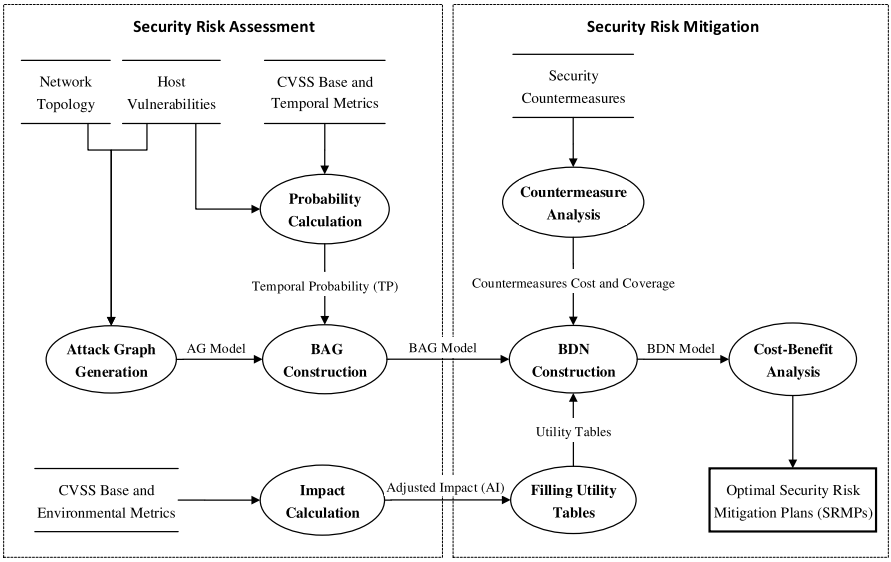

Fig. 1 Proposed security risk management framework

3.1 Security Risk Assessment

The proposed security risk assessment process consists of four steps, namely, modeling network attacks using AG, probability calculation, BAG construction and impact calculation which are described in the following.

3.1.1 Modelling Network Attacks

AGs are useful tools for modeling network security vulnerabilities and their interactions which form multi-step attacks. These tools are widely used in several areas of network security, including security risk assessment, such as [35, 46], because they can depict paths in which an attacker can exploit vulnerabilities to compromise a security policy in a network. Each of these paths consists of one or more vulnerabilities in the form of a chain in which some of them are prerequisites of some others to exploit.

To generate an AG for a given network, the information about existing vulnerabilities and network topology and hosts connectivity is required. Vulnerabilities existing on hosts can be discovered using available network vulnerability scanners such as Nessus [47], OpenVAS [48] and Retina [49] or searching the online vulnerability repositories such as the US National Vulnerability Database (NVD) [50] and MITRE's Common Vulnerabilities and Exposures (CVE) [51]. Network hosts connectivity and topology can be determined according to either the network security administrator's knowledge or by using available network discovery tools, such as Nmap [52]. With this information available, AGs can be automatically generated using tools such as MulVAL [53].

Definition 1 *Attack Graph*. An attack graph (AG) is a tuple $AG = (S, s_0, s_g, \tau)$, where

- S is a set of states in the network. Each state is the result of exploiting one or more vulnerabilities. Exploitation of each vulnerability changes the state of the AG.
- $s_0 \subseteq S$ is the leaf node of the AG. It denotes the attacker's entry points to the network and hence is the initial state.
- $s_g \subseteq S$ is the set of root nodes in the AG. Each root node represents a goal for attackers. In fact attackers start from leaf node and exploit sequences of vulnerabilities to reach their goals. Therefore each AG can have multiple root nodes depending on the potential attractive targets for attackers.
- $\tau \subseteq S \times S$ is the set of transition relations. Each transition between two states represents a vulnerability exploitation which changes the state of the network from $S_{precondition}$ to $S_{postcondition}$.

3.1.2 Probability Calculation

AGs are powerful tools for modeling potential attacks targeting network assets, but they are unable to measure the security level of networks quantitatively. To make AGs quantitative, security administrator should add the exploitation probability of each vulnerability to its nodes. Because of today's vast networks consisting of many hosts connected together, each of which containing several vulnerabilities, number of existing vulnerabilities in networked systems is very high. Therefore extracting the probability of successful exploitation of each vulnerability from expert's knowledge is an error prone, tedious and time consuming task. To overcome this problem, we use CVSS's metrics to calculate the probability of successful vulnerabilities exploitation.

CVSS provides sets of metric groups, namely, Base, Temporal and Environmental, to quantitatively assess the severity level of existing IT security vulnerabilities [4]. Regarding various attributes, each of these groups produce a numeric score ranging from 0 to 10. The Base Score is used to describe the intrinsic characteristics of vulnerabilities using two subscores: (1) the exploitability subscore and (2) the impact subscore. The Temporal Score quantifies the characteristics of vulnerabilities that change over time. The Environmental Score captures the characteristics of vulnerabilities that are associated with a specific IT environment. Each CVSS metric can be assigned with several values which are listed and defined in [4]. In this paper, metrics from Base and Temporal groups of CVSS are selected to calculate the probability of exploiting vulnerabilities. Therefore, the calculated probabilities are more accurate and closer to the reality in the time of assessment.

Exploitability of a vulnerability is calculated using the metrics of the Base group of CVSS as bellow:

$$Exploitability = 2 \times AV \times AC \times AU \tag{1}$$

where AV is access vector, AC is access complexity and AU is authentication instances [4].

Exploitability only reflects the inherent properties of vulnerabilities, i.e. the properties that are constant with time and across different environments. Therefore it cannot represent the current state of vulnerabilities exploitation. This is important, because for doing a precise and accurate risk assessment, the characteristics of vulnerabilities at the time of risk assessment must be considered. Vulnerabilities have versatile nature and their exploitability changes over time depending on factors such as the available tools for exploiting vulnerabilities, current remediation level of vulnerabilities, the degree of confidence in the existence of vulnerabilities and the credibility of the known technical details.

To take into account the properties of vulnerabilities which change over time, Temporal metrics of CVSS are also used [54]. These metrics adjust the value of Exploitability (Equation 1) to the time of assessment as below:

$$TemporalProbability(TP) = (E \times RL \times RC) \times Exploitability \tag{2}$$

where TP stands for temporal probability and is the probability of exploiting a vulnerability at the time of risk assessment. E measures the current state of exploitable tools and techniques, RL is the remediation status of the vulnerability and RC is the report confidence [4].

3.1.3 BAG Construction

In this step, AG is converted to BAG by adding CPTs to each of its nodes. The only constraint for this conversion is that the AG should be a directed acyclic graph. This can be guaranteed by monotonicity assumption which states that the attacker never needs to backtrack [5].

Definition 2 *Conditional Probability Table.* A Conditional Probability Table (CPT) is a tabular form of a conditional probability distribution representing the values of $Pr(s_i|Pa[s_i])$. Where s_i denotes a state in the BAG and $Pa[s_i]$ denotes its parents nodes. Each node in a BAG has an associated CPT which specifies the chances of compromising a network state given different combination of states of its parents. The entries of a CPT are filled with the conditional probabilities of vulnerabilities exploitations, i.e. TP values.

Definition 3 *Bayesian Attack Graph.* Let AG be an attack graph. A Bayesian attack graph associated with AG, denoted by BAG, is a Bayesian network over the same set of nodes, S, such that there exists a CPT for each state node. Formally, a BAG is defined as a tuple $BAG = (S, \tau, \varepsilon, P)$ where S denotes the set of nodes (i.e. the set of states in the network). The edges connecting the nodes in the graph are reflected with a set of ordered pair τ. The conjunctive or disjunctive relations between multiple edges pointing to a node are represented by ε with possible values of {AND, OR}. P represents the set of CPTs associated with the BAGs nodes.

In a BAG, each node has an associated CPT which shows the probability of the node given the states of its parents. The CPT of each node is generated as follows.

The exploit which changes the state of the network from s_i to s_j for $s_i \in Pa[s_j]$ is called e_i. The conditional probability distribution function of s_j is $Pr(s_j|Pa[s_j])$ which is defined as below.

In the case when s_j has more than one parent, if the relation between incoming edges to node s_j is AND ($\varepsilon = AND$), the product rule is used.

$$Pr\left(\mathrm{s_j}|\, Pa[\mathrm{s_j}]\right) = \begin{cases} 0, & \exists\, \mathrm{s_i} \in Pa[\mathrm{s_j}], \mathrm{s_i} = 0 \\ Pr\left(\underset{\mathrm{s_i}=1}{\cap} \mathrm{e_i}\right) = \prod\limits_{\mathrm{s_i}=1} TP(\mathrm{e_i}), & otherwise \end{cases} \tag{3}$$

If the relation between incoming edges to node s_j is OR ($\varepsilon = OR$), the noisy OR operator is used [8].

$$Pr\left(\mathrm{s_j}|\, Pa[\mathrm{s_j}]\right) = \begin{cases} 0, & \forall\, \mathrm{s_i} \in Pa[\mathrm{s_j}], \mathrm{s_i} = 0 \\ Pr\left(\underset{\mathrm{s_i}=1}{\cup} \mathrm{e_i}\right) = 1 - \prod\limits_{\mathrm{s_i}=1} \left[1 - TP(\mathrm{e_i})\right], & otherwise \end{cases} \tag{4}$$

3.1.4 Impact Calculation

To calculate the impact of vulnerabilities in the environment under assessment, first, the Base metrics of the impact subscore (i.e. C, I and A) for each vulnerability are measured (available at the existing vulnerability databases like national vulnerability database (NVD) [50]). After that, in order to consider the environmental impact of vulnerabilities, CVSS's Environmental metrics are measured by security administrators. These metrics are confidentiality requirement (CR), integrity requirement (IR) and availability requirement (AR) metrics, which are used to customize the CVSS Base impact subscore. More details about CVSS metrics can be found in [4].

Finally, using the Eq. 5, the adjusted impact of exploiting vulnerabilities in the network under assessment will be calculated [54].

$$AdjustedImpact(AI) = 1 - (1 - C \times CR) \times (1 - I \times IR) \times (1 - A \times AR) \tag{5}$$

Considering environmental impact of vulnerabilities in the network under assessment, more precise results will be produced which are more compatible with current situation.

3.2 Security Risk Mitigation

The proposed security risk mitigation process consists of several steps, namely, countermeasure analysis, BDN construction, filling utility tables and cost-benefit analysis which are described in the following.

3.2.1 Countermeasure Analysis

A security countermeasure or a security control (SC) is a risk-reducing measure which can be implemented on vulnerabilities to further reduce or eliminate the residual risk by reducing the exploitability of the affected vulnerabilities [2]. Therefore, SCs prevent attackers from reaching their goals, i.e. compromising the IT assets.

The status of SC_i is defined as a Boolean variable with the possible states of {True, False}. The True state implies that the SC is implemented ($Status_{(SC_i)} = True$) and the False state implies that the SC is not implemented as part of the security risk mitigation plan ($Status_{(SC_i)} = False$). Each SC_i also has an associated cost of implementation ($Cost_{(SC_i)}$). Cost of each SC is an organization dependent factor which should be determined by security administrators.

By implementing a SC on a vulnerability, the amount of effort an attacker needs to do for exploiting that vulnerability will increase, resulting the reduction of the exploitability of that vulnerability. The ability of SC in reducing the exploitation probability of covered vulnerabilities is measured by $Coverage_{(SC_i)}$.

The coverage of each SC can either be acquired from available security reports and documents about the effectiveness of security patches, updates, workarounds and etc. or the security administrator can determine it manually.

Definition 4 *Security Countermeasure.* A security countermeasure is a tuple $SC = (Status, Cost, Coverage)$ where, *Status* is a Boolean value representing whether the SC is implemented or not, *Cost* represents the implementation cost of SC and *Coverage* is the reduction percentage of exploitation probability of covered vulnerabilities.

3.2.2 BDN Construction

BAG itself is not a complete model for doing risk mitigation, because basically it does not consider SCs and their properties. Therefore we need a more comprehensive and general model. In this paper, BDN model is used which is compatible to other graph based models like AG and BAG and is suitable for doing the processes needed in the risk management framework, especially risk mitigation process.

Definition 5 *Bayesian Decision Network.* Let BAG be a Bayesian attack graph. A Bayesian decision network associated with BAG, denoted by BDN, is an extension of a BAG which has three types of nodes:

- **Chance nodes**, representing the same states (S) existing in BAG. Each chance node is associated with a CPT, representing $Pr(s_i|Pa[s_i])$. These node types are represented as ovals.
- **Decision nodes**, representing the security countermeasures (SC) covering the states (S). The security administrator can choose to whether select these nodes (SCs) as part of the security risk mitigation plan or not, with values of {True, False}. These node types are represented as rectangles.

– **Utility nodes**, representing the effects of each SC (decision node) on its affected states (chance nodes) in the form of a utility table. These node types are represented as diamonds.

A BDN is a directed acyclic graph over these nodes, such that the utility nodes have no children. An example of a BDN model is depicted in Fig. 3. Each utility node in a BDN is associated with a utility table, which gathers the preferences for all decision choices about SC implementation. The parents of a utility node represent the set of nodes on which the utility node depends.

In order to convert BAG model of a network into its corresponding BDN model, it is just needed to add decision and utility nodes to it. The procedure of converting BAG to BDN is described below.

Algorithm 1- Converting BAG to BDN

Input:

-*BAG* model of the network under assessment

Output:

-*BDN* model of the network under assessment

Step 1. For each SC covering a vulnerability, a decision node must be added to the *BAG* model of the network under assessment. A directed arc should be pointed from this decision node to the state, s, representing the covered state.

Step 2. By implementing each SC on its covered vulnerability, the vulnerabilities exploitation probability will be reduced. Therefore implementing each SC has a specific effect on its covered vulnerabilities. The amount of this effect is modeled using utility nodes. Each utility node has an associated table called utility table. This table quantifies the effect of implementing each SC on its covered vulnerabilities. Therefore, a utility node must be added to the network wherever there is a SC covering vulnerabilities. Two directed arcs should be placed pointing at the utility node, one from covered state (i.e. chance node) and the other from the SC (i.e. decision node) covering that vulnerability.

Step 3. Fill each utility table associated with existing utility nodes in the *BDN* according to Sect. 3.2.3.

3.2.3 Filling Utility Tables

Network security administrator as a decision maker has to choose between two possible actions: implement or not to implement a given SC_j. There are also two possible events: either vulnerability V_i is exploited by attackers or not.

– If V_i is exploited by attackers, the targeted network assets will suffer from attack damage, which is proportional to the AI of V_i. And if V_i is not exploited by attackers, no adverse effects will occur.
– If security administrator implements covering SC_j on V_i, he will reduce the attack damage of V_i, which is relative to the benefit of SC_j, but he incurs the cost of implementing SC_j. And if security administrator does not implement SC_j, then he incurs no cost of implementation.

As mentioned before, each utility node has a utility table which quantifies the effect of implementing each SC on its covered vulnerabilities. Therefore the entries of these tables must be filled according to the effectiveness of the SCs and the impact level of the covered vulnerabilities. The proposed definition of utility tables is shown in Table 1. Note that, it is assumed that the attack damage is a negative value.

Table 1 demonstrates that if vulnerability V_i is exploited while security countermeasure SC_j is implemented on it, the damage of V_i exploitation ($V_i.AttackDamage$) is reduced based on benefit of SC_j ($SC_j.Benefit$). Therefore, the total utility is equal to the residual damage of V_i, minus implementation cost of SC_j.

In the case where vulnerability V_i is exploited while no countermeasures is implemented on it, the total utility is equal to the damage incurred because of V_i exploitation, which is considered as a negative value.

In the case where V_i is not exploited but SC_j is implemented on it, the utility is equal to the negative value of SC_j implementation cost.

In the case where no security countermeasure is implemented and no vulnerability is exploited, the utility is equal to zero.

Providing these values for each combination of SCs and vulnerabilities requires the judgment of experts (i.e. security administrators) which indeed is a tedious and time-consuming process. To reduce the need of expert's knowledge, the AI value (a value in the interval of [0, 1] calculated using Eq. 5) is used to calculate attack damage by multiplying it in Damage Criterion which is assigned by experts. The attack damage value represents the damage caused by exploiting a vulnerability on affected assets. Therefore the attack damage can be calculated using Eq. 6.

$$AttackDamage = DamageCriterion \times AdjustedImpact(AI) \tag{6}$$

Similarly, to determine the benefit of implementing a security countermeasure (SC_i) on a vulnerability, the coverage percentage of SC_i, i.e. $Coverage_{(SC_i)}$ is multiplied in a Benefit Criterion which is assigned by experts using Eq. 7.

$$SCBenefit = BenefitCriterion \times SecurityControlCoverage \tag{7}$$

As the result, regardless of the size of the network under assessment and the number of vulnerabilities and countermeasures, the experts only need to determine two values of Damage Criterion and Benefit Criterion.

Table 1 Utility table definition

Security control \ exploitation	V_i is exploited	V_i is not exploited
SC_j is implemented	$((V_i.AttackDamage) + (SC_j.Benefit)) - SC_j.Cost$	$-SC_j.Cost$
SC_j is not implemented	$V_i.AttackDamage$	0

3.2.4 Cost-Benefit Analysis

The goal of this phase is to find the optimal subset(s) of SCs in which by implementing them, the security level of the network maintains in an acceptable level. Each subset of SCs is called a security risk mitigation plan (SRMP). A SRMP is defined as a Boolean vector ($\overrightarrow{SRMP}$) that represents which SCs are chosen to be implemented and which are not. The cost of implementing a SRMP ($Cost_{\overrightarrow{(SRMP)}}$) is calculated by summing up the implementation costs of the selected SCs, as shown in Eq. 8.

$$Cost_{\overrightarrow{(SRMP)}} = \sum_{i} SC_i \times Cost_{(SC_i)} \tag{8}$$

By implementing a SRMP, the exploitation probability of a subset of existing vulnerabilities on network assets will be reduced. Therefore, we need to combine the utility values of different outcomes; hence, we ascribe an expected utility (EU) value to each SRMP. Thus, it is possible to compare different SRMPs using their EU values. As the result, the network security administrator's objective is to identify the most effective SRMPs, i.e. the plans with the highest EU. However, sometimes, due to the budget limitation, it is not possible to implement all of the SCs; therefore the plans with the highest EU and the implementation cost lower than the limited budget should be identified.

To conduct a cost-benefit analysis over the BDN model, a simple algorithm (Algorithm 2) is proposed which slightly changes the variable elimination (VE) algorithm. VE algorithm is of the simplest and the most general inference algorithms used in probabilistic models such as Bayesian networks [32]. This algorithm is computationally much more efficient than the basic Bayesian inference algorithms, because it uses the caching of dynamic programming to save redundant computation. Therefore, VE algorithm can perform inference even for large and complex networks in a very reasonable time. For more information about VE algorithm and its operators refer to [32].

In the proposed algorithm (Algorithm 2), we slightly modified the VE algorithm to further improve its efficiency by:

- Restricting the calculations only over the subsets of security controls which their implementation costs are lower than the allocated budget.
- Using Max-Product operator instead of standard VE's simple Product operator.

Algorithm 2- Cost-Benefit Analysis

Inputs:

- *BDN* model of the network under assessment

SecurityControls

$\{(SC_1, Cost_{(SC_1)}), (SC_2, Cost_{(SC_2)}), ..., (SC_n, Cost_{(SC_n)})\}$ // Set of security controls and their implementation costs

- *Budget*; // allocated budget for network hardening

Algorithm 2- Cost-Benefit Analysis

Output:

- Subsets of SCs with total implementation costs lower than the allocated *Budget* for network hardening

Step 1. Find all subsets of the set *SecurityControls* while the condition bellow holds and store the results in a table named *Combinations*.

$$\sum_i Cost_{(SC_i)} \leq Budget \qquad (9)$$

Step 2. Use VE algorithm on the *BDN* while:

1. Calculations are restricted to the *Combinations* entries.
2. Max-Product operator is used instead of Product operator.

As seen in the algorithm above, the Max-Product operator is substituted for the simple Product operator used in standard VE algorithm [32]. When multiplying two factors, this operator multiplies only the factors which have the greatest value and discards other factors. Hence space, time and computational complexity are reduced.

The output of Algorithm 2 is the SRMPs with the highest EU and the implementation costs lower than the allocated budget.

3.3 Security Risk Validation and Monitoring

Risk validation and monitoring is a continual process which determines whether the residual risk is at an acceptable level or whether there is a need to implement additional countermeasures to further reduce or eliminate the residual risk. Since this is a continual process during the lifetime of the network (once the BDN model is constructed), it is not depicted in the framework shown in Fig. 1.

In this phase, the BDN model is continually updated based on the observations of changes in network states. We have two types of observations: observing weather a vulnerability is exploited or not, and observing weather a SC is implemented or not. In case where a vulnerability is exploited by attackers and therefore a BDN state is reached, the TP of that state in its CPT is changed to 1. In case where SC_i is implemented by security administrator ($SC_i = True$), the entries of relevant utility table with $SC_i = False$ will not be participated in calculations. After each observation, the algorithm 2 is applied on the modified BDN model to find the best SRMPs regarding the current situation.

4 Experimental Results

In this section, the results of applying the proposed framework on a test network used in [9] is presented.

4.1 The Test Network

The proposed framework is applied to a test network which is shown in Fig. 2 [9]. There are several hosts in this network which are located within two zones: (1)

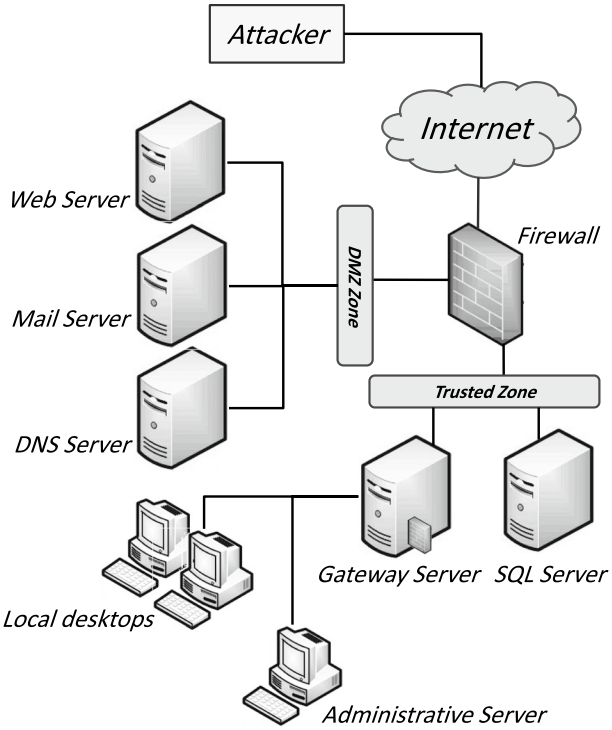

Fig. 2 Topology of the test network [9]

the DMZ zone which contains the mail server, DNS server and web server and is accessible to the public through a firewall; (2) the trusted zone which contains SQL server, administrative server, gateway server and several local desktops. All accesses from external sources to the trusted zone are restricted and also all communications to external parties are delivered through the gateway server. A DMZ tri-homed firewall is installed with preset policies to ensure that the DMZ zone is separated from the trusted zone. According to the policies, the web server is allowed to send SQL queries to the SQL server. Also, the remote desktop service of all local desktops, including the administrative server, is enabled to make employees able to communicate from remote sites via wired or wireless mediums. The remote connections are monitored by SSHD protocol which is installed in the gateway server.

Network hosts connectivity and topology are determined by using Nmap network discovery tool [52] and hosts vulnerabilities are discovered by using Nessus network vulnerability scanner [47]. Information about existing vulnerabilities in this network including their CVE IDs [51] is listed in Table 2.

Table 2 List of vulnerabilities in the test network

Host	Vulnerability	CVE ID
Administrative server	MS SMV service Stack BOF	CVE 2008-4050
DNS server	DNS Cache Poisoning	CVE 2008-1447
	Heap corruption in OpenSSH	CVE 2003-0693
Gateway server	Improper cookies handler in OpenSSH	CVE 2007-4752
	Open SSL uses predictable random	CVE 2008-0166
	Remote login	CA 1996-83
Local desktops	LICQ Buffer Overflow (BOF)	CVE 2001-0439
	MS Video ActiveX Stack BOF	CVE 2008-0015
	Squid port scan vulnerability	CVE 2001-1030
Mail server	Remote code execution in SMTP	CVE 2004-0840
	Error message information leakage	CVE 2008-3060
SQL server	SQL Injection	CVE 2008-5416
Web server	IIS vulnerability in WebDAV service	CVE 2009-1535

4.2 The Risk Assessment Results

Having network topology and information about existing vulnerabilities, we model network attacks by generating AG model of the network using MulVAL network security analyzer [53] (as mentioned in Sect. 3.1.1).

In order to convert the AG model into BAG model, we need to populate its CPT tables with Temporal Probability (TP) of vulnerability exploitations. The TP of each vulnerability existing in the test network with its relevant CVSS metrics values are calculated using Eq. 2 and are listed in Table 3. The CVSS metrics values (which are described in detail in [4]) are obtained from available online databases, such as NVD database [50]. With these information available, the BAG model of the network can be constructed (as explained in subsection 3.1.3).We have implemented BAG model using GeNIe Modeler [55].

When a temporal characteristic of a vulnerability changes during system lifetime, CVSS Temporal metrics (E, RL and RC metrics) are changed accordingly. Therefore, based on Eq. 2, the TP of the affected vulnerability is recalculated. As the result, the probability of corresponding nodes in the model should be recalculated using Eq. 3 and 4. Finally, using Bayesian inference algorithm, the probability of all nodes in the model are updated.

The adjusted impact (AI) of vulnerabilities existing in the test network with their relevant impact metrics of CVSS are listed in Table 4. The AI values of these vulnerabilities are calculated using Eq. 5. The CVSS metrics values are obtained from available online databases, such as NVD database [50]. Note that the Environmental metrics are organization dependent and should be determined by the network security administrator itself. Therefore, here, the values of CR, IR and AR metrics are hypothetically assigned.

Table 3 List of TP values of the test network vulnerabilities

CVE ID	AV	AC	AU	Exploitability	E	RL	RC	TP
CA 1996-83	Network	High	Single	0.39	Functional	Temporary Fix	Confirmed	0.3335
CVE 2001-0439	Network	Low	None	1.0	Proof of Concept	Unavailable	Uncorroborated	0.855
CVE 2008-0015	Network	Medium	None	0.86	Functional	Unavailable	Uncorroborated	0.7761
CVE 2008-4050	Network	Medium	None	0.86	Proof of Concept	Unavailable	Uncorroborated	0.7353
CVE 2008-0166	Network	Low	None	1.0	Functional	Official Fix	Confirmed	0.8265
CVE 2003-0693	Network	Low	None	1.0	Unproven	Official Fix	Confirmed	0.7395
CVE 2007-4752	Network	Low	None	1.0	Unproven	Official Fix	Confirmed	0.7438
CVE 2008-5416	Network	Low	Single	0.8	Proof of Concept	Official Fix	Confirmed	0.6264
CVE 2004-0840	Network	Low	None	1.0	Functional	Official Fix	Confirmed	0.8265
CVE 2008-3060	Network	Low	None	1.0	High	Unavailable	Uncorroborated	0.95
CVE 2001-1030	Network	Low	None	1.0	Proof of Concept	Official Fix	Confirmed	0.783
CVE 2008-1447	Network	Low	None	1.0	Functional	Official Fix	Confirmed	0.8265
CVE 2009-1535	Network	High	None	0.49	Functional	Workaround	Confirmed	0.4422

Table 4 List of AI values of the test network vulnerabilities

CVE ID	C	I	A	CR	IR	AR	Adjusted impact
CA 1996-83	Complete	Complete	Partial	Low	Low	Low	0.6128
CVE 2001-0439	Partial	Partial	Partial	Medium	Low	Low	0.4607
CVE 2008-0015	Complete	Complete	Complete	High	High	High	1.0
CVE 2008-4050	Complete	Complete	Complete	High	Medium	Medium	0.9996
CVE 2008-0166	Complete	None	None	Medium	Low	Low	0.66
CVE 2003-0693	Complete	Complete	Complete	Medium	Medium	Medium	0.9607
CVE 2007-4752	Partial	Partial	Partial	Medium	Low	Low	0.4607
CVE 2008-5416	Complete	Complete	Complete	High	High	Medium	1.0
CVE 2004-0840	Complete	Complete	Complete	High	High	Medium	1.0
CVE 2008-3060	Partial	None	None	Low	Low	Low	0.1375
CVE 2001-1030	Partial	Partial	Partial	Medium	Medium	Medium	0.6189
CVE 2008-1447	None	Partial	Partial	High	High	Low	0.4957
CVE 2009-1535	Complete	Complete	Complete	High	Medium	High	1.0

Table 5 List of SCs covering the test network vulnerabilities

Security countermeasure (SC)	Covered vulnerability(ies)	Implementation cost	Coverage
	CA 1996-83		
SC_0-filtering external traffics	CVE 2004-0840	70	0.62
	CVE 2009-1535		
SC_1-apply MS workaround	CVE 2008-0015	14	0.65
SC_2-disable WebDAV	CVE 2009-1535	250	0.44
SC_3-patch OpenSSH	CVE 2003-0693	63	0.75
	CVE 2007-4752		
SC_4-disable port scan	CVE 2001-1030	11	0.45
	CVE 2001-1030		
SC_5-add network IDS	CVE 2008-3060	102	0.38
	CVE 2009-0568		
SC_6-gateway firewall	CVE 2001-0439	205	0.33
SC_7-query restriction	CVE 2008-5416	84	0.28
SC_8-apply MS09-004 work around	CVE 2008-5416	31	0.43
SC_9-encryption	CVE 2008-1447	34	0.31
SC_{10}-limit access to DNS server	CVE 2008-1447	53	0.5
SC_{11}-digital signature	CVE 2008-3060	33	0.3
SC_{12}-use POP3	CVE 2008-3060	153	0.25

4.3 The Risk Mitigation Results

After generating the BAG model of the network under assessment and calculating the impact of vulnerabilities exploitation, we should identify and assess the available

security countermeasures (SCs) covering network vulnerabilities. The SCs covering network vulnerabilities with their implementation costs and coverage percentages are listed in Table 5.

Some countermeasures cannot completely eliminate the exploitability of vulnerabilities; i.e. there is uncertainty in the coverage level of SCs. Therefore, the proposed method, allows the security administrator to define the coverage percentage of the SC with uncertainty. The coverage level of SCs can be directly determined by the security administrator or can be inferred using the information available in the security databases and reports.

The output of Algorithm 2 for the test network is depicted in Fig. 3. In this BDN model, the ovals represent attacker exploits which are the states of the network, the edges represent their pre-conditions and post-conditions, SCs which prevent the exploits are shown as rectangles and diamonds represent utility nodes. Note that we have implemented BDN model using GeNIe Modeler [55].

Equations 3 and 4 are used to fill the CPTs of the chance nodes in the BDN model. To fill the utility tables of utility nodes, the definition presented in Table 1 is used. The value of attack damage is determined using Eq. 6. In this equation, the damage criterion value is considered as 100. Therefore the damage to the network assets is ranged between 0 and 1000.

By applying Algorithm 2 on the BDN model, we can run multiple inferences with different goals. Three scenarios are considered here as examples. Scenario number 1 runs inference algorithm with the aim of finding the optimal SRMP(s) with the highest EU regardless of any implementation cost limitations. Scenario number 2 infers

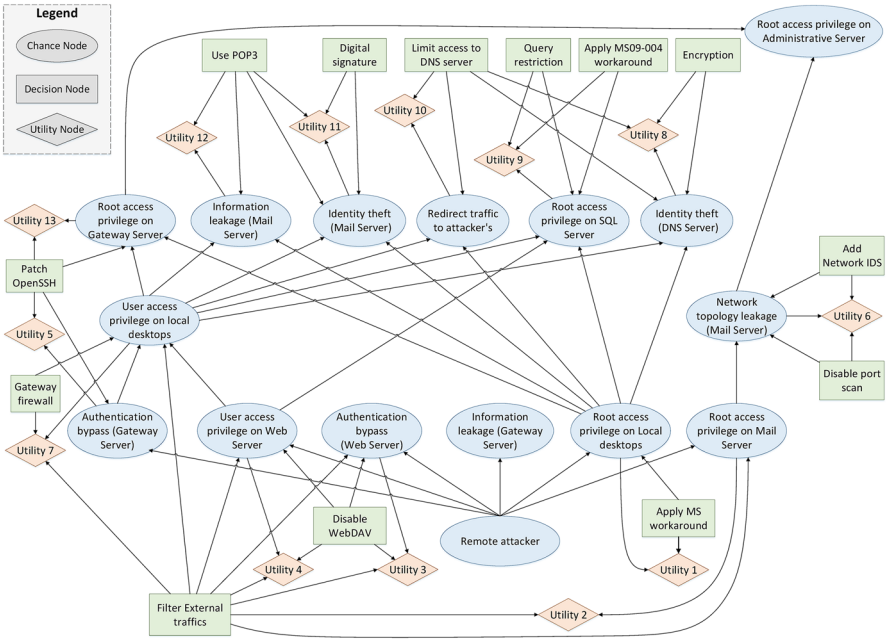

Fig. 3 BDN model of the test network (Fig. 2)

the optimal SRMP(s) with the highest EU with the implementation cost lower than the limited budget. Scenario number 3 tries to identify the SRMP(s) with implementation costs exactly equal to a pre-specified value and determines the best one which has the highest EU.

Scenario 1: In this scenario, we have no limitations for budget. Therefore, the output of algorithm 2 is a SRMP with the highest EU. The results show that the highest expected utility ($EU_{highest}$ = 4986.198 units) is accessible by implementing $\overrightarrow{SRMP_{highest}} = \{SC_0, SC_1, SC_3, SC_4, SC_8, SC_9, SC_{10}\}$ with the overall cost of MPC($\overrightarrow{SRMP_{highest}}$) = 276 units.

Scenario 2: In this scenario, the allocated budget for network hardening is considered as 240 units. In this case, the algorithm will not calculate the EU of the plans with the MPC more than 240 units. Therefore, the highest expected utility ($EU_{highest}$ = 4910.54 units) is accessible by implementing $\overrightarrow{SRMP_{highest(240)}} = \{SC_0, SC_1, SC_3, SC_4, SC_8, SC_9\}$ with the overall cost of MPC ($\overrightarrow{SRMP_{highest(240)}}$)=223 units. Comparing to the previous scenario, the SC_{10} is not selected here. This means that SC_{10} is not as important as countermeasures in $\overrightarrow{SRMP_{highest(240)}}$, so, in order to keep the overall implementation cost lower than the predefined value, the algorithm chooses not to select this countermeasure.

Scenario 3: In this scenario, we try to identify the SRMP(s) with implementation costs exactly equal to 220 units and determine which one of them is the best and has the highest EU. To do so, we modify the Equation 9 to $\sum_i Cost_{(SC_i)} = 220$. Therefore, all subsets of the set *SecurityControls* which their total implementation costs are exactly equal to 220 are identified and are stored in Combinations table. The results show that there are six SRMPs with implementation costs equal to 220 units. These SRMPs are listed in Table 6.

The Table 6 shows that there are six SRMPs which their implementation costs are equal to 220 units, but their EU are different from each other. $SRMP_1$ has the highest EU among others, while $SRMP_6$ has the lowest EU which is a negative value. Negative EU for a SRMP implies that the SRMPs benefit is lower than its implementation cost. Therefore, implementing these SRMPs is not economically reasonable. The reason about SRMP's negative EU is that the SRMP contains SCs with low coverage implemented on vulnerabilities with high exploitation probabilities and hence they cannot cover the vulnerabilities efficiently. Another reason is that SCs are implemented on less important vulnerabilities, so, the allocated budget is not

Table 6 SRMPs with implementation costs equal to 220 units

Number	SRMP	Expected utility (EU)
1	$\{SC_0, SC_1, SC_5, SC_9\}$	3931.66
2	$\{SC_0, SC_3, SC_9, SC_{10}\}$	3592.84
3	$\{SC_1, SC_{10}, SC_{12}\}$	1137.00
4	$\{SC_5, SC_7, SC_9\}$	65.68
5	$\{SC_5, SC_8, SC_9, SC_{10}\}$	643.48
6	$\{SC_9, SC_{11}, SC_{12}\}$	− 341.64

efficiently used and therefore, more important vulnerabilities with higher exploitation probabilities are not covered.

5 Discussions

This paper presents an integrated framework for security risk management of computer networks which utilizes a powerful probabilistic model, i.e. Bayesian decision network (BDN), as the core model for doing risk management. The proposed framework can be applied to organizations caring for their assets security. It helps security administrators to assess and prioritize existing vulnerabilities on networks hosts and guides them to determine appropriate sets of security countermeasures to combat potential cyber attacks. The approach used in this study brings several advantages and needs several improvements which are discussed in this section.

First of all, we have proposed an integrated framework for network security risk management which encompasses all the information needed for managing identified risks within it.

The model is based on effective well known model called attack graph used widely in recent researches and therefore it utilizes its benefits, such as its ability to represent scenarios of multistep attacks targeting attackers' goals. Therefore, it can be converted to BAG and BDN models using the proposed algorithms with almost no cost of incompatibility.

Understandability of the BDN model is achieved by using standard representation concepts used in Bayesian theory, i.e. the representation of various nodes are identified as the standard shapes presented in Bayesian theory. The data structures definitions like CPTs and utility tables are well defined which increases the comprehensibility of the model.

The data required for feeding the model can be acquired automatically from the existing data sources and repositories. Moreover, by using the proposed equations, the need for expert's knowledge is minimized for doing both security risk assessment and risk mitigation.

The proposed BDN model is generated once and can be reused several times for managing identified risks when needed without the need for reconstructing the model. Moreover, the exploitation probabilities of vulnerabilities can be updated as new information become available and can be propagated through the BDN model during network lifetime using the standard Bayesian forward and backward propagation algorithms [32]. Therefore, the BDN model is dynamic and is able to consider changes in exploitation probability of vulnerabilities.

In the risk assessment phase, the exploitation probability and impact values of vulnerabilities are precisely calculated considering temporal characteristics of vulnerabilities and properties of environment under assessment. Therefore, the results are more accurate and closer to the reality in the time of assessment.

In the risk mitigation phase, we assumed that the SCs do not have just Boolean effect on their covered vulnerabilities. This assumption is valid; because one cannot guarantee that a SC can fully cover and mitigate all aspects of a vulnerability. Hence, we modeled this uncertainty by considering SC's coverage as a probabilistic value. Moreover, by

using standard Bayesian inference algorithms and slightly modifying them to be able to consider budget limitation, the optimal subsets of SCs are inferred as part of the cost-benefit analysis.

We slightly modified the VE algorithm to further reduce its time and computational complexity.

Although the proposed inference algorithm for finding the optimal subsets of SCs is simple and practical, but it is not efficient for large networks with huge number of SCs. Therefore it is desired to propose more scalable algorithms for doing cost-benefit analysis over the large BDN models.

The proposed framework cannot deal with zero-day attacks, because unknown vulnerabilities and zero-day exploits cannot be modeled using the proposed BDN model.

6 Conclusions and Future Work

In this paper, an integrated framework for network security risk management is proposed. BDN model is used for modeling the information needed for network security risk management. The proposed framework can be adapted according to the current time conditions and the specific network security requirements, which produces results closer to the reality. For the risk mitigation process, a cost-benefit analysis is conducted to identify the optimal subsets of risk-reducing countermeasures. To do so, VE algorithm is slightly changed to take into account budget limitation. In future, we try to further improve the accuracy of the risk assessment process by considering attacker capabilities in estimating vulnerability exploitation probabilities using metrics presented in [56]. Also, we could use the revisions to the CVSS Base metrics presented in [57] to increase the accuracy of the proposed security risk assessment process. In order to conduct a cost-benefit analysis over the BDN model in the risk mitigation process, a more scalable and efficient algorithm needs to be proposed in future studies. Moreover, we intend to evaluate the applicability of the proposed framework on preventing targeted attacks like advanced persistent threats (APTs).

Compliance with Ethical Standards

Conflict of Interest Authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with animals performed by any of the authors.

Informed Consent Informed consent was obtained from all individual participants included in the study.

References

1. Thomas, PR.: Information security risk analysis, 3rd edition, Auerbach publications, Boco Raton (2010)

2. Ross, R.S.: Guide for conducting risk assessments, Special Publication (NIST SP)-800-30 Rev. 1, (2012)
3. Evan, W.: Security risk management: building an information security risk management program from the ground up, 1st edn. Elsevier, Burlington (2011)
4. Mell, P., et al.: A complete guide to the common vulnerability scoring system version 2.0, Published by FIRST-Forum of Incident Response and Security Teams, vol. 1, (2007)
5. Ammann, P., et al.: Scalable, graph-based network vulnerability analysis, Proceedings of the 9th ACM Conference on Computer and Communications Security, ACM (2002)
6. Sheyner, O., et al.: Automated generation and analysis of attack graphs, In Proceedings 2002 IEEE Symposium on Security and Privacy. IEEE, New York (2002)
7. Gallon, L., Bascou, J. J.: Cvss attack graphs, In 2011 Seventh International Conference on Signal Image Technology & Internet-Based Systems, pp. 24–31. IEEE, New York (2011)
8. Liu, Y., Man, H.: Network vulnerability assessment using Bayesian networks, In Data mining, intrusion detection, information assurance, and data networks security, vol. 5812, pp. 61–71, International Society for Optics and Photonics, Bellingham (2005)
9. Poolsappasit, N., et al.: Dynamic security risk management using bayesian attack graphs. IEEE Trans. Dependable Secure Comput. **9**(1), 61–74 (2012)
10. Hong, J.B., et al.: A survey on the usability and practical applications of graphical security models. Comput. Sci. Rev. **26**, 1–16 (2017)
11. Lippmann, R.P., Ingols, K.W.: An annotated review of past papers on attack graphs, No. PR-IA-1, Massachusetts Inst of Tech Lexington Lincoln Lab (2005)
12. Garg, U., et al.: Empirical analysis of attack graphs for mitigating critical paths and vulnerabilities. Comput. Security **77**, 349–359 (2018)
13. Kaynar, K.: A taxonomy for attack graph generation and usage in network security. J. Inform. Security Appl. **29**, 27–56 (2016)
14. He, W., et al.: Unknown vulnerability risk assessment based on directed graph models: a survey. IEEE Access **7**, 168201–168225 (2019)
15. Cheng, P., et al.: Aggregating CVSS base scores for semantics-rich network security metrics, In 2012 IEEE 31st Symposium on Reliable Distributed Systems, IEEE, New York (2012)
16. Wang, C., et al.: A novel comprehensive network security assessment approach, In 2011 IEEE International Conference on Communications (ICC), IEEE, New York (2011)
17. Wang, S., et al.: Exploring attack graph for cost-benefit security hardening: a probabilistic approach. Comput. Security **32**, 158–169 (2013)
18. Wang, L., et al.: An attack graph-based probabilistic security metric, In IFIP Annual Conference on Data and Applications Security and Privacy, pp. 283–296. Springer, Berlin, Heidelberg (2008)
19. Ghosh, N., Ghosh, S.K.: An approach for security assessment of network configurations using attack graph, In 2009 First International Conference on Networks Communications, pp. 283–288. IEEE, New York (2009)
20. Noel, S., et al.: Measuring security risk of networks using attack graphs. Int. J. Next Gen. Comput. **1**(1), 135–147 (2010)
21. Frigault, M., Wang, L.: Measuring network security using Bayesian network-based attack graphs, In 2008 32nd Annual IEEE International Computer Software and Applications Conference, pp. 698–703. IEEE, New York (2008)
22. Kondakci, S.: Network security risk assessment using Bayesian belief networks, In 2010 IEEE Second International Conference on Social Computing, pp. 952–960. IEEE, New York(2010)
23. Xie, P., et al.: Using Bayesian networks for cyber security analysis, In 2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN), pp. 211–220. IEEE, New York (2010)
24. Feng, N., et al.: A security risk analysis model for information systems: causal relationships of risk factors and vulnerability propagation analysis. Inform. Sci. **256**, 57–73 (2014)
25. Le, A., et al.: Incorporating FAIR into bayesian network for numerical assessment of loss event frequencies of smart grid cyber threats. Mobile Networks Appl.**24**(5), 1713–1721 (2019)
26. Wang, J., et al.: A Bayesian network approach for cybersecurity risk assessment implementing and extending the FAIR model, Computers Security 89, 101659
27. Frigault, M., et al.: Measuring the overall network security by combining cvss scores based on attack graphs and Bayesian networks, in Network Security Metrics, pp. 1–23. Springer, Cham (2017)
28. Noel, S., Jajodia, S.: A suite of metrics for network attack graph analytics, in network security metrics, pp. 141–176. Springer, Cham (2017)

29. Norman, T.L.: Risk analysis and security countermeasure selection, 2nd edn. CRC Press, Cleveland (2015)
30. Wheeler, E.: Security risk management: building an information security risk management program from the Ground Up, 1st edn. Elsevier, Amsterdam (2011)
31. Russell, S.J., Norvig, P.: Artificial intelligence: a modern approach, 4th edn. Pearson Education Limited, Malaysia (2020)
32. Koller, D., Friedman, N., Bach, F.: Probabilistic graphical models: principles and techniques, 1st edition, MIT press, Cambridge (2009)
33. Ahmed, M.S., et al.: Objective risk evaluation for automated security management. J. Network Syst. Manag. **19**(3), 343–366 (2011)
34. Alali, M., et al.: Improving risk assessment model of cyber security using fuzzy logic inference system. Comput. Security **74**, 323–339 (2018)
35. Dai, F., et al.: Exploring risk flow attack graph for security risk assessment. IET Infor. Security **9**(6), 344–353 (2015)
36. Wangen, G., et al.: A framework for estimating information security risk assessment method completeness. Int. J. Inform. Security **17**(6), 681–699 (2018)
37. Rusek, K., et al.: Effective risk assessment in resilient communication networks. J. Network Syst. Manag. **24**(3), 491–515 (2016)
38. Awan, M.S.K., et al.: Identifying cyber risk hotspots: a framework for measuring temporal variance in computer network risk. Comput. Security **57**, 31–46 (2016)
39. Nespoli, P., et al.: Optimal countermeasures selection against cyber attacks: a comprehensive survey on reaction frameworks. IEEE Commun. Surveys Tutorials **20**(2), 1361–1396 (2018)
40. Gehani, A., Kedem, G.: Rheostat Real Time Risk Manag. In: international workshop on recent advances in intrusion detection, pp. 296–314. Springer, Berlin, Heidelberg (2004)
41. Dabbebi, O., et al.: An online risk management strategy for VoIP enterprise infrastructures. J. Network Syst. Manag. **23**(1), 137–162 (2015)
42. Noel, S., et al.: Efficient minimum-cost network hardening via exploit dependency graphs. In 19th Annual Computer Security Applications Conference Proceedings, IEEE, New York. pp. 86–95 (2003)
43. Jha, S., et al.: Two formal analyses of attack graphs. In Proceedings 15th IEEE Computer Security Foundations Workshop, CSFW-15, IEEE, New York. pp. 49–63 (2002)
44. Dewri, R., et al.: Optimal security hardening using multi-objective optimization on attack tree models of networks, In Proceedings of the 14th ACM conference on computer and communications security, ACM. pp. 204–213, (2007)
45. Khosravi-Farmad, M., et al.: Network security risk mitigation using Bayesian decision networks, In 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE), IEEE. pp. 267–272 (2014)
46. Liu, S. C., Liu, Y.: Network security risk assessment method based on HMM and attack graph model, In 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), IEEE, New York. pp. 517–522 (2016)
47. Nessus Vulnerability Scanner. http://www.tenable.com/products/nessus-vulnerability-scanner
48. OpenVAS, Open Vulnerability Assessment System. http://www.openvas.org/
49. Retina Network Security Vulnerability Scanner. https://www.beyondtrust.com/products/retina-network-security-scanner/
50. NIST. US National vulnerability database (NVD). https://nvd.nist.gov/
51. Common Vulnerabilities and Exposures (CVE). https://cve.mitre.org/
52. Nmap, The Network Mapper. https://nmap.org/
53. Ou, X., et al., MulVAL: A Logic-based Network Security Analyzer, In USENIX Security Symposium, pp. 113–128 2005
54. Khosravi-Farmad, M., et al.: Considering temporal and environmental characteristics of vulnerabilities in network security risk assessment, In 2014 11th International ISC Conference on Information Security and Cryptology, IEEE. pp. 186–191 (2014)
55. GeNIe Modeler, BayesFusion, LLC. https://www.bayesfusion.com/
56. ben Othmane, L., et al.: Incorporating attacker capabilities in risk estimation and mitigation., Computers Security 51, pp. 41–61 (2015)
57. Holm, H., et al.: An expert-based investigation of the common vulnerability scoring system. Comput. Security **53**, 18–30 (2015)

Masoud Khosravi-Farmad is a Ph.D. candidate of computer software engineering at Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran. He received his M.Sc. degrees in computer software engineering from Ferdowsi University of Mashhad. His research interests are computer and network security, security risk management and intrusion detection and prevention system.

Abbas Ghaemi-Bafghi received his B.S. degree in Applied Mathematics in Computer from Ferdowsi University of Mashhad, Iran. He received his M.Sc. and Ph.D. degrees in Computer engineering from Amirkabir (Tehran Polytechnique) University of Technology, Iran, in 1997 and 2004 respectively. He is member of Computer Society of Iran (CSI) and Iranian Society of Cryptology (ISC). He is an associated professor in Department of Computer Engineering, Ferdowsi University of Mashhad, Iran. His research interests are in cryptology and security, and he has published more than 100 conference and journal papers.