Journal of Computing and Security

2020, Volume 7, Number 1 (pp. 19–33)
http://www.jcomsec.org

A Risk Estimation Framework for Security Threats in Computer Networks

Razieh Rezaee [a] , Abbas Ghaemi Bafghi [a,*]
[a] *Data and Communication Security Lab., Computer Dept., Ferdowsi University of Mashhad, Iran.*

ARTICLE INFO.

Article history:
Received: 4 January 2020
Revised: 1 May 2020
Accepted: 17 May 2020
Published Online: 19 June 2020

Keywords:
Security Threat, Analysis Model, Computer Networks, Risk Estimation, Attack Graph, Bayesian Network.

ABSTRACT

In security risk management of computer networks, some challenges are more serious in large networks. Specifying and estimating risks is largely dependent on the knowledge of security experts. In this paper, a framework for security risk estimation is proposed to address this issue. It represents the security knowledge required for security risk estimation and utilizes current security metrics and vulnerability databases. This framework is a major step towards automating the process of security risk estimation so that a network administrator can estimate the risk of the network with less expertise and effort. As a case study, the proposed framework is applied to a sample network to show its applicability and usability in operational environments. The comparison of results with two existing methods showed the validity of the estimations given by the proposed framework.

1 Introduction

Security vulnerabilities in computer networks make security threats for system assets. Attackers can exploit the vulnerabilities to penetrate the system and get access to its assets. One of the main activities in network security is security risk management, which includes the process of identification, estimation, and mitigation of security risks [1].

There are several methods [2–6] and various tools to estimate the risk of exploiting a vulnerability. For brevity, we refer to "the risk of vulnerability exploitation" as "vulnerability risk". These methods are not accurate as they estimate the risk of each vulnerability separately without considering the effect of related vulnerabilities. Intruders usually exploit a sequence of related vulnerabilities to reach their goal in the network. In these multi-stage attacks, the probability of exploiting vulnerability depends on the exploitation probability of other related vulnerabilities [7].

For multi-stage attacks, interdependent security models are employed for modeling the dependency between single vulnerabilities. Game theory, Petrinets, and attack graphs are the main models used for modeling the multi-stage attacks. Network administrators should have security expertise to consider the dependency between reported vulnerabilities and obtain more realistic results.

In this paper, a framework for estimating the risk of security threats in computer networks has been proposed. The security threats are identified by the existing vulnerabilities in the network, and the risk of vulnerabilities are estimated based on their interrelations. In the proposed framework, Bayesian network [8–13] combined with attack graph [14] has been used to model the multi-stage attacks and probability dependencies of vulnerabilities. In this framework, the

* Corresponding author.
Email addresses: rezaee.ra@mail.um.ac.ir (R. Rezaee), ghaemib@um.ac.ir (A. Bafghi)
https://dx.doi.org/10.22108/jcs.2020.120412.1038
ISSN: 2322-4460

expertise requirement is reduced and the network administrator will be able to use this framework with less security expertise to create the security model of the network and determine the security threats, their risk values, and other useful information for the selection of security controls.

This paper extends the conference version [15] significantly by proposing a framework for the previous approach, which precisely specifies the relationships and communications of its components. Furthermore, by applying the previously proposed models on various real-world networks, some limitations and inconsistencies were discovered and improved. The implementation also enhanced to calculate the risk of vulnerabilities by adding their exploit impact and to produce more useful results such as the most probable path and the most important vulnerability.

The remainder of the paper is organized as follows. In Section 2, a brief description and review of related work are presented. Section 3 introduces the background material, and Section 4 elaborates on the proposed framework. The experimental results of applying the proposed framework to the test networks are presented in Section 5. In Section 6, conclusions are drawn, and directions for future works are discussed.

2 RELATED WORK

The most widely-used approach for quantitative estimation of network security risk involves estimating the risk of threats separately and then aggregating them to achieve the total risk of the network. A variety of methods have been proposed for separate estimation of threat risks, which assess the risk of single vulnerability exploitation [2–5, 16]. For example, CVSS [4] and CWSS [5] are two systems used for scoring the risk of vulnerabilities and weaknesses, respectively. These methods [2, 3, 16] estimate the risk of vulnerabilities separately, without considering the interconnections and dependencies of related vulnerabilities, although the majority of network attacks are made in multiple stages. Several tools such as Petrinets [17–19], game theory [20–24], and attack graphs/trees [8, 13, 25, 26] have been used for the modeling of multi-stage attacks. These models represent the relations between network vulnerabilities and determine the attack paths to a certain goal in the network.

Game theory is also used for modeling of interdependency between network defenders as the players of the game [27]. In [28] authors have chosen interdependent security games to model the effect of the neighbors of a node on its attack probability. In [29–31] the interdependency is perceived to find an efficient solution for defenders' security investment on risk reduction.

Among the models proposed for multi-stage attacks, attack graphs/trees are the most popular due to their understandability for users and conformity to the graph theory. For example, in [9] the authors have combined attack trees with the graph and algebraic theory to evaluate the risk of attack scenarios probabilistically. The attack graph is also joined with the fuzzy set theory to deal with uncertainty difficulties in [32].

In [33] multi-stage attacks on Supervisory Control and Data Acquisition (SCADA) systems are modeled using attack graphs. The attack graph is generated by employing a model-checker which finds attack sequences to a target as counterexamples. Such methods [33, 34] only identify and represent the attack path and do not specify the risk value of attacks.

Attack graphs are not only used to specify and analyze the attack paths but also can be used to represent the dependencies of vulnerabilities and calculate their probability [9][32–37]. Attack graphs have also been combined with a Bayesian network [8, 11, 35] to represent the dependencies between probabilities of vulnerabilities. Furthermore, the Bayesian decision network has been used to find a set of the riskiest threats and choose the optimal set of security controls to obtain the most secure network with a limited budget [10]. Although these works are so efficient in calculating the risks and costs of security threats, they do not address the construction of the attack graph and its challenges such as the security expertise need, complexity of the security model, and its flexibility in the face of the system changes.

To reduce security expertise need, Sommestad et al. [36] used Probabilistic Relational Model to assess the security risks of the network. In their work, Bayesian inference was used to compute the attack probability. They extended their model in [12, 13] and proposed a language for constructing their model. However, this model was designed for SCADA systems with few attacks in this area. For other networks and attacks, the models need to be redesigned by security experts.

In this paper, we proposed a framework for specifying attack paths in networks and estimating their security risks. By representing the necessary information and knowledge in an abstract model, the framework reduces the need for security expertise for network administrators. It is also designed to be flexible to network changes. These changes are applied to the network model and their costs will be lower than that of other risk estimation approaches that reconstruct the network model in the case of network changes. Furthermore, the proposed framework is not limited to a specific type of network or attack.

3 BACKGROUND

Modeling the multi-stage attacks is an indispensable step in the security risk estimation of networks. Attack graphs are dominantly used for representing the attack paths in multi-stage attacks. Bayesian networks combined with attack graphs are also used for estimating the probability of multi-stage attack exploitations. To measure the probability and impact of each vulnerability, CVSS metrics have been used in the proposed framework. In this section, the background material for our framework is described.

3.1 Bayesian Networks

Bayesian network G is a Directed Acyclic Graph (DAG) in which, each node has a discrete random variable, and each edge $X \leftarrow Y$ indicates the relation between parent Y and child X. Each node has a Conditional Probability Table (CPT), which specifies the probability of the node for every given state of its parents. In the Bayesian network, it is assumed that a node is conditionally independent of nodes that are not connected to its edges [37].

In the Bayesian networks, the joint probability of all variables is calculated by the chain rule (Eq. (1)).

$$P\left(x_1, \ldots, x_n\right) = \prod_{i=1}^{n} P\left(x_i | Pa\left(x_i\right)\right) \qquad (1)$$

Where X=$\{x_1, \ldots, x_n\}$ is a set of random variables, and Pa(x_i) denotes the specific values of variables in the parent nodes of x_i.

3.2 Bayesian Attack Graph

Bayesian Attack Graph (BAG) is a combination of attack graph and Bayesian network and formally is defined as follows.

Let S be a set of network states as graph nodes and A be the set of atomic attacks defined on S. An atomic attack is associated with vulnerability exploitation which takes the intruder from a network state (S_{pre}) to another (S_{post}) [8]. A BAG is defined as a tuple $BAG = (S, \tau, \varepsilon, \mathcal{P})$, where

- $S = N_{\text{internal}} \cup N_{\text{external}} \cup N_{\text{terminal}}$, where
  $S_i \in N_{\text{internal}}$, if $\exists a_1, a_2 \in A | [S_j = pre\left(a_1\right)$ and $S_j = post\left(a_2\right)$
  $S_i \in N_{\text{external}}$, if $\nexists a \in A | S_i = post(a)$
  $S_i \in N_{\text{terminal}}$, if $\nexists a \in A | S_k = pre(a)$
- τ is a set of ordered pairs of nodes.
  $(S_{pre}, S_{post}) \in \tau$, if $S_{pre} \mapsto S_{post} \in A$
  Further, for $S_i \in S$, parent set of S_i is defined as $Pa\left[S_i\right] = \{S_j \in S | \left(S_j, S_i\right) \in \tau\}$
- ε is a set of decomposition tuples that specifies $\{AND, OR\}$ relations for all $S_j \in N_{\text{internal}} \cup N_{\text{terminal}}$ internal or terminal nodes. For tuple $\langle S_j, d_j \rangle$,
  d_j is AND, if $S_j = 1 \Rightarrow \forall S_i \in Pa\left[S_j\right], S_i = 1$,
  d_j is OR, if $S_j = 1 \Rightarrow \exists S_i \in Pa\left[S_j\right], S_i = 1$
- $\mathcal{P}$ is a set of discrete conditional probability distribution functions. For each internal and terminal node, there is a Local Conditional Probability Distribution (LPCD) indicting the values of $\Pr\left(\mathrm{S_j} | \mathrm{Pa}\left[\mathrm{S_j}\right]\right)$

The AND relation between parents of n_i indicates that all of its parents should be compromised until the attacker reach n_i. The OR relation between parents of n_i suggests that the attacker can reach n_i if any of its parents is compromised. Therefore, the value of $\Pr\left(n_i | Parent\left[n_i\right]\right)$ is calculated based on joint probability distribution and Noisy-OR, as defined below [38].

$$\text{if } d_i = AND, \Pr\left(n_i | Parent\left[n_i\right]\right) = \begin{cases} 0 & \exists n_i \in Parent\left[n_i\right], n_j = 0 \\ \prod_{n_j=1} \Pr(e_i) & otherwise \end{cases} \qquad (2)$$

$$\text{if } d_i = OR, \Pr\left(n_i | Parent\left[n_i\right]\right) = \begin{cases} 0 & \forall n_i \in Parent\left[n_i\right], n_j = 0 \\ \prod_{n_j=1} [1 - \Pr(e_i)] & otherwise \end{cases} \qquad (3)$$

Where e_i is the vulnerability exploitation, which exhibits the transition of the attacker from state n_j to n_i, and $\Pr\left(e_i\right)$ is the single exploitation probability of e_i.

In the proposed model, when the probability of a node is changed, the probability of its successors is updated. This propagation of changes is implemented by the Bayes' formula (Eq. (4)) and the updated probabilities are the posterior probabilities of successor nodes.

$$P(X|Y) = \frac{P(X)P(Y|X)}{P(Y)} \qquad (4)$$

3.3 CVSS Framework

In the Common Vulnerability Scoring System [4] a set of metrics has been presented for assessing the risk score of a vulnerability. These metrics are employed to assess the probability and impact of vulnerability exploitation. The proposed models are not dependent on a certain probability and impact assessment method, but a subset of base metrics in CVSS was chosen for this purpose.

To estimate the probability of single vulnerability exploitation, the base metrics of CVSS for exploitability of vulnerability were used. These metrics included

Table 1. Values for CVSS Exploit Probability Metrics.

Metric	Metric Value	Numerical Value
Access vector	Local access	0.395
	Adjacent network	0.646
	Network	1.0
Access complexity	High	0.35
	Medium	0.61
	Low	0.71
Authentication	Multiple authentications	0.45
	Single authentication	0.56
	No authentication	0.704

Access Vector, *Access Complexity* and *Authentication*. The probability of single vulnerability exploitation is assessed by Eq. (5).

$$\begin{aligned} ExploitationProbability \\ = 2 \times AccessVector \\ \times AccessComplexity \\ \times Authentication \end{aligned} \tag{5}$$

The values of the metrics are specified in Table 1 [4].

To estimate the impact of vulnerability exploitation, *Confidentiality Impact*, *Integrity Impact*, and *Availability Impact* were used. The *Impact* value is assessed as follows [4].

$$\begin{aligned} Impact = 1 - [(1 - Impact_{conf}) \\ \times (1 - Impact_{Integ}) \\ \times (1 - Impact_{Avail})] \end{aligned} \tag{6}$$

According to the CVSS standard, the values of impact metrics are shown in Table 2.

4 THE PROPOSED FRAMEWORK

In this paper, a framework for security risk estimation is proposed, which is intended to automate this activity by reducing the need for a security expert. The proposed framework is designed to be general enough and not restricted to a specific type of network or a certain set of attacks. The contributions of the proposed framework are summarized below.

- This paper proposes a framework to reduce the need for security experts by representing essential information and knowledge required for security risk estimation in an abstract model. Expertise reduction is a major step towards automating the process of security risk management.
- Based on an exhaustive review of literature in the area, we designed an abstract and a concrete model for the estimation of security risks. Moreover, a process to construct the concrete model from the abstract model is presented.
- This framework defines a process for risk estimation, which is characterized by the flexibility of the security model to the network changes. By adding or removing a host, connection, or vulnerability, the effect of this change is propagated throughout the model and the risk values are updated accordingly. Unlike works such as [8, 12, 13, 39], in the proposed framework, a model reconstruction is not required to apply changes to the model.
- In the proposed models, central parts are network nodes, which represent Data Terminal Equipment (DTE). Therefore, there is a one-to-one correspondence between network nodes in the models and DTEs in the network. As a result, the logical network security model conforms with the network topology (the physical model of the network) and represents the reachability of the hosts. This capability helps network administrators to apply countermeasures more efficiently in the risk mitigation phase [40].

In the following subsections, an outline of the framework and its components are presented.

4.1 Risk Estimation Framework for Security Threats

The outline of the proposed framework is depicted in Figure 1. This framework comprises two mod-

Table 2. Values for CVSS Impact Metrics.

Metric	Metric Value	Numerical Value
Confidentiality, Integrity, and Availability Impact	Complete Impact	0.660
	Partial Impact	0.275
	None	0

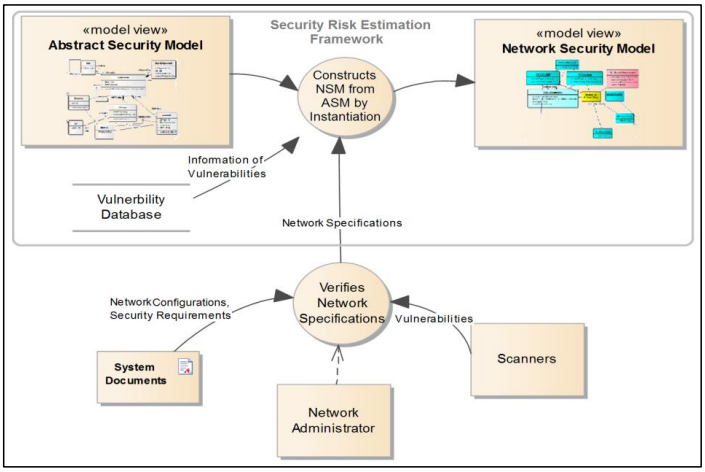

Figure 1. The Outline of the Proposed Framework.

els, the Abstract Security Model (ASM) and the Network Security Model (NSM). The ASM is a high-level model used to represent the security information necessary for risk estimation. This model describes network nodes and their connections, security vulnerabilities and their characteristics, and the relations between security vulnerabilities, among other things.

Since the ASM is general, it is created only once and applied to every given Network Under Analysis (NUA) to construct its NSM. The framework also has a process that constructs the NSM from ASM. This process gives the specifications of an NUA as inputs, and according to them, instantiates the required objects from relevant classes defined in the ASM. These objects together form the NSM of a given NUA. The attack paths and the risk value of each vulnerability are specified in the NSM.

4.2 Abstract Security Model

The Abstract Security Model is a high-level model of computer networks. It lays the foundation for probabilistic reasoning to identify and estimate security risks. In the proposed framework, ASM is represented by a UML class diagram. The design of ASM is based on an exhaustive review of literature in this area such as NIST SP 800-30 [41], NIST SP 800-39 [42], and Common Criteria [43].

As shown in the ASM diagram (Figure 2), a network node is represented by the *NetworkNode* class, which represents a network device such as a host or server. Every network node has a set of privileges represented by the *Privilege* class. It indicates the access type of a *User* to a *NetworkNode*. The *Attacker* is also a kind of *User*, which may have a level of access to some *NetworkNodes*. A *Connection* class connects two *NetworkNodes*.

Each class has a set of attributes and operations, which are essential for the elicitation of attack paths and the calculation of their threat probabilities. With their relationships and attributes, the classes represent a logical security model of the network. The operations of ASM classes implement the process of identifying and estimating security risks. Risk identification is performed by the reasoning of attack paths based on pre/post conditions of vulnerabilities, and the risk value is estimated according to the vulnerabilities relations specified in the attack graph.

According to the characteristics of the risk analysis problem, the Bayesian network modeling approach is used to estimate the probability of multi-stage attacks. The probability of each step in attack paths is calculated by the BAG described in Section 3.

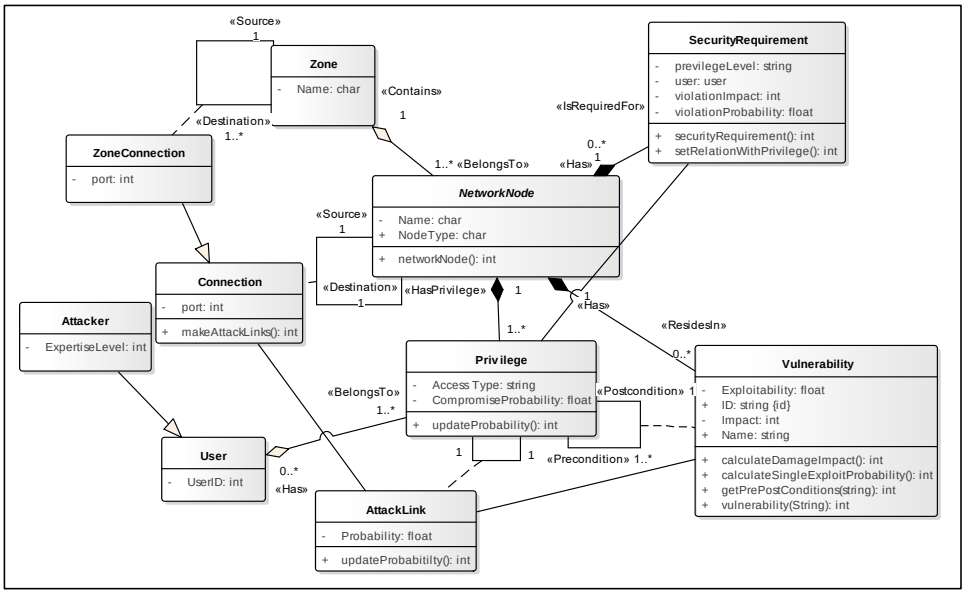

Figure 2. The Abstract Security Model for Network Security Risk Analysis.

Algorithm 1 Constructor Method of Vulnerability Class.
INPUT: CVE_ID of Vulnerability

```
1: call getProbabilityMetrics(CVE_ID);
2: call calculateSingleExploitProbability(); //according to Eq.(5)
3: call getImpactMetrics(CVE_ID);
4: call calculateDamageImpact(); //according to Eq.(6)
5: call getPrePostConditions(CVE_ID); //gets the pre/post-conditions and the type of their relation
6: for i ← 1 to number_Of_Preconditions do
7:     if pre_Post_condition[i]==null then //if such privilege does not exists
8:         instantiate precondition[i] from Privilege class
9:     end if
10:    if relation_Type==AND then
11:        call SetConditionalProbabilityTable(); //according to Eq.(2)
12:    else
13:        if relation_type==OR then
14:            call SetConditionalProbabilityTable(); //according to Eq.(3)
15:            postCondition.probability = calculateUnconditionalProbability(); //–according to Eq.(4)
16:            call self.postCondition.updateProbability();
17:            call setRelations(); //sets the relation of vulnerability with security requirements
18:        end if
19:    end if
20: end for
```

The constructor operation of the Vulnerability class among other object operations is explained here. This operation gets the information of the vulnerability from the vulnerability database and estimates the probability and damage impact of its exploitation. Algorithm 1 exhibits this process. First, the values of impact and exploitation probability metrics of the vulnerability are queried from the vulnerability database. Then, the damage impact of the vulnerability and its single exploitation probability are calculated. The pre/post-conditions of vulnerabilities and the relation type between pre-conditions are also queried from the vulnerability database. After setting the conditional probability table based on Eq. (2) and (3), the unconditional probability of vulnerability exploitation in a BAG is assessed according to Eq. (4). Moreover, the effects of vulnerability exploitation on security requirements are considered by setting the relationships between vulnerability and affected requirements.

The security risk analysis of vulnerability is a two-

Algorithm 2 Construction of NSM From ASM.

INPUT: The ASM, Network Topology, Network Vulnerabilities, Security Requirements

```
for i ← 1 to number_Of_Network_Nodes do
    instantiate n_i from NetworkNode class
    for j ← 1 to number_Of_Security_requirement of n_i do
        instantiate sr_ij from SecurityRequirement class
        call sr_ij.setRelations(); //sets the relations between requirement and privileges
    end for
    for l ← 1 to number_Of_connection of n_i do
        instantiate c_il from Connection class
    end for
    for k ← 1 to number_Of_Vulnerability of n_i do
        instantiate V_ik from Vulnerability class
    end for
end for
```

step process:

- Specifying the security impact of vulnerability exploitation
- Estimating the probability of vulnerability exploitation

The risk value is calculated by multiplying the probability and impact of vulnerability exploitation.

$$Risk = Probability \times Impact \tag{7}$$

The security impact of vulnerability exploitation heavily depends on the context the vulnerability exists in. For example, it depends on the IT asset compromised by the vulnerability exploitation, the security requirements of that asset, and the significance of the asset in that organization. The proposed framework is not dependent on a specific set of metrics, but the base metrics of CVSS for vulnerability exploitability and impact are used for estimating the probability and impact of vulnerability exploitation.

4.3 Network Security Model

The NSM represents the security model of a given network. To construct the NSM for a given NUA, the required objects are instantiated from the appropriate classes of ASM. For each node of an NUA, the needed objects are instantiated to represent the node and its requirements, vulnerabilities, etc. The NSM is constructed iteratively by adding new objects. After instantiating an object, its constructor is called to initiate its properties and apply the effect of adding this object to the NSM. It may affect the probability attribute of previous objects or may cause to instantiate another object. Moreover, in the case of a probability change, it is propagated to the related objects in the model. Probabilities are calculated using the Bayesian rule and any change in probabilities is disseminated through forwarding propagation in BAG [8].

For example, consider a step of generating NSM when vulnerability V is created for node N and added to the current NSM. The post-condition of V is a privilege $N.P_i$. With the addition of V, the probability of $N.P_i$ may be increased. By changing the probability of $N.P_i$, the *updateProbability()* function in all *Privilege* objects *linked* to N will be called. This function calculates the new value for the probability of the object.

The final NSM contains risk values of each vulnerability and security requirement violation. This information is used as input in the risk mitigation phase. The vulnerabilities with the highest risk and the path with the highest probability to a certain goal of the attacker are other outputs of the framework, which allow administrators to prioritize the threats. Given the limited budget for applying security controls, it is indispensable for the administrator to decide which vulnerabilities need to be remediated and which paths should be blocked with higher priority.

4.4 The Process of Constructing NSM From ASM

To construct NSM from ASM for an NUA, the specifications of the NUA are required as the input of the process. The output of this process is the NSM of the network with its security risks. This NSM can be used as the input of the risk mitigation phase in the risk management process.

Algorithm 2 describes the process of constructing NSM. For each node in the network, an object from *NetworkNode* class is instantiated. Then, for each security requirement of the node, an object from *SecurityRequirement* is instantiated. It is also specified whether the attacker access to privilege violates the security requirement. Furthermore, the connections

Table 3. List of the Vulnerabilities of the Network Under Analysis.

Host	CVE ID	Vulnerability
Gateway server	CVE 2007-4752	Improper cookie handler in OpenSSH
Local desktops	CA 1996-83	Remote login
	CVE 2001-0439	LICQ Buffer Overflow (BOF)
	CVE 2008-0015	MS Video ActiveX Stack BOF
SQL server	CVE 2008-5416	SQL Injection
Web server	CVE 2009-1535	IIS vulnerability in WebDAV service

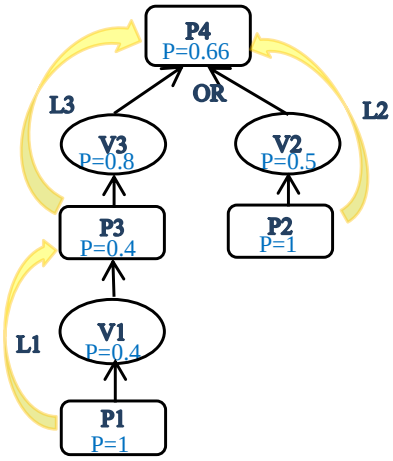

Figure 3. A BAG Deduced From a Sample NSM.

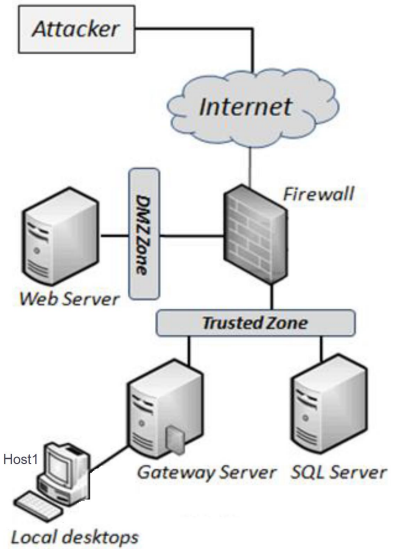

Figure 4. The Topology of the Network Under Analysis [8, 10].

of the node are represented in the NSM by objects from *Connection* class, and for each reported vulnerability of the node, an object from *Vulnerability* class is instantiated.

The framework has an object-oriented design, which makes the building blocks of its models standalone; therefore, the model is flexible to modification. When an object is added to or removed from the NSM, the relevant values are updated and necessary objects are instantiated and added to the NSM automatically. It may change the probabilities and risks of other objects, or contribute to their construction or destruction. The dynamic characteristic of the framework is necessary both in the risk estimation phase and the risk mitigation phase.

When the administrator applies a countermeasure against a vulnerability, the value of its exploitation probability changes. The administrator can apply this change to NSM of the network by changing its exploitability value. This change is propagated through the NSM so that all affected risk values are updated and the riskiest vulnerabilities and the most probable path are selected again. For mitigation, the administrator can apply countermeasures iteratively, make the relevant changes in the NSM, and obtain the new most risky vulnerability and path.

4.5 Deducing the Underlying BAG

The underlying BAG of an NSM can be constructed by tracking the *AttackLinks* beginning from *Attacker* node, using forward chaining method, or from the target asset, using backward chaining. In the proposed framework the tracking of *AttackLinks* is performed iteratively from *Attacker* to any accessible targets.

For example, if the *Attacker* has *Privileges* P1 and P2, and there are *AttackLinks* as follows

- *AttackLink* L1 from P1 to P3 by *Vulnerability* V1 with probability 0.4,
- *AttackLink* L2 from P2 to P4 by textitVulnerability V2 with probability 0.5,
- *AttackLink* L3 from P3 to P4 by *Vulnerability* V3 with probability 0.8.

The BAG will be as Figure 3 shows and the proba-

Table 4. Connections Between Network Nodes.

Source and Destination of Connection	Allowed Protocol
Local desktops, Gateway Server	Basic network protocols
Gateway server, SQL Server	SQL
Gateway Server, Web Server	HTTP
Web Server, SQL Server	SQL

bilities are the same as specified in the NSM.

4.6 Network Specifications

Network specifications of an NUA serve as the input of the framework for constructing NSM. Network specifications include network configurations, security requirements of each node, and the list of vulnerabilities that exist in the NUA. This list is provided by vulnerability scanners. The configuration of the network and its security requirements are extracted from network system documents and network mapper tools. The network administrator also verifies this information to correct the faulty and/or missed data.

4.7 Vulnerability Database

The information required to estimate the risk of existing vulnerabilities is retrieved from the vulnerability database. This database contains information about vulnerabilities such as their pre-conditions and post-conditions and the value of their CVSS metrics.

5 RESULTS AND DISCUSSION

To investigate the applicability and usability of the proposed framework, it is applied to multiple networks. In this section, the results of applying the proposed framework to a test network (Figure 4) are shown. The test network is comprised of a DMZ zone with a web server, and a trusted zone that contains a SQL server, a gateway server, and two local desktops. The vulnerabilities reported by the scanner for each network node are shown in Table 3. The information on identified vulnerabilities is queried from the database of vulnerabilities. This information is extracted from vulnerability databases such as NVD [1] [44] or OSVDB [2] [45]. Furthermore, Table 4 specifies the connections between network nodes.

By using ASM proposed in the previous section (Figure 2) and the specifications of the test network, the NSM of this network is constructed. The resulted NSM is shown in Figure 5. For each node in the network, the required objects are instantiated to represent that node and its specifications and connections as described in Algorithm 2.

As an object is instantiated and added to the NSM model, the constructor of the object is called. The constructor applies the effects of this addition on the objects that already exist in the NSM.

For example, to represent the connection between *Host1* and *SQLServer*, an object named *H-SQLS-Con* is instantiated from *Connection* class. This connection means that *Host1* has network access to *SQLServer* on port 1433. By adding this object to NSM, an attack link *(H-SQL-Lnk)* is established which links privileges *H-LP* and *H-RP* to privilege *SQLS-NP*. This link means the local user and root user on *Host* have network access on port 1433 to *SQLServer*. The probability of this access is 1, because the firewall allows this access. Therefore, an attacker with local or root access on *Host1* has sufficient privilege to exploit *SQL-Vul*. *SQL-Vul* is a SQL injection vulnerability, which its pre-condition privilege is network access to its hosted machine.

In Table 5, the threats of exploiting vulnerabilities are listed. The value of exploitability and impact metrics are obtained from the vulnerability database. For each vulnerability, the probability of its exploitation is estimated when it is considered as single vulnerability and its relations to other vulnerabilities are neglected (Eq. (6)). Then, the probability of vulnerability exploitation is estimated concerning other nodes of the network. This probability is calculated through the Bayesian Attack Graph in the NSM. The visual representation of the BAG that is created from the NSM for the test network is illustrated in Figure 6.

Attack paths and the unconditional probability of the privileges and vulnerabilities are depicted in this figure. For example, there are four ways for an attacker to gain root access to *SQLServer*, and the probability of this threat is 0.78. If the relation of the network nodes and their vulnerabilities were neglected, the

[1] National Vulnerability Database
[2] Open Source Vulnerability Database

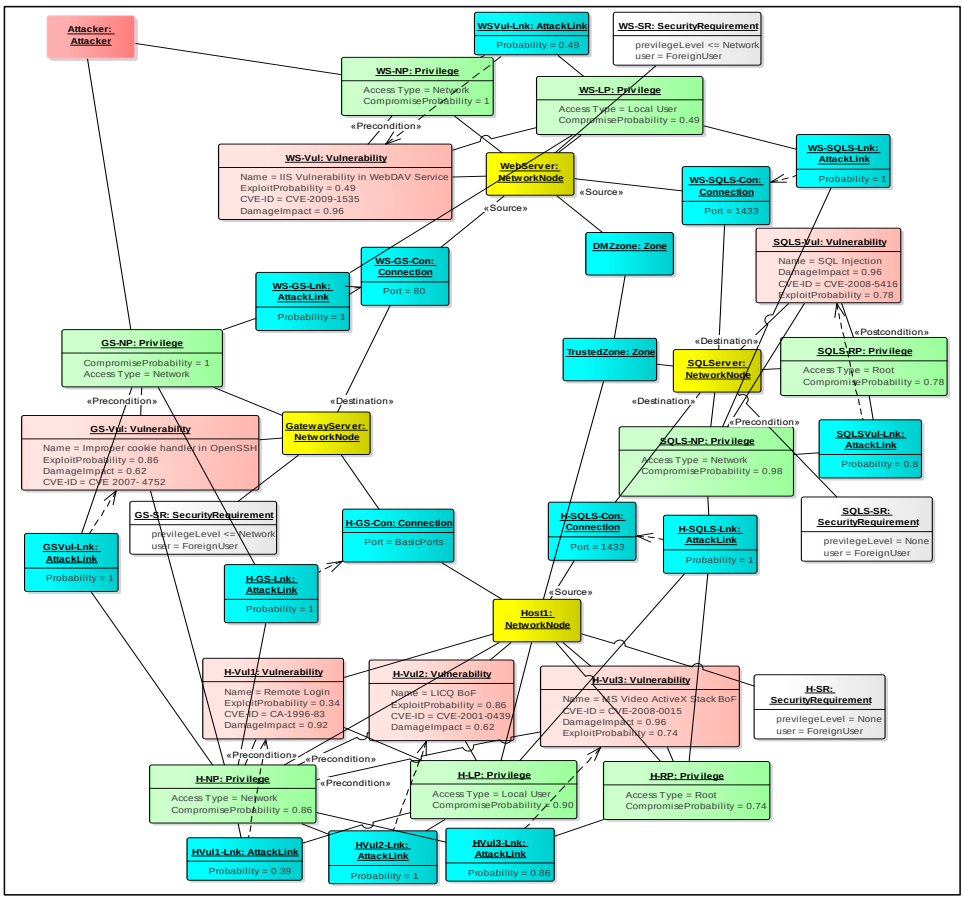

Figure 5. NSM for the Network Under Analysis.

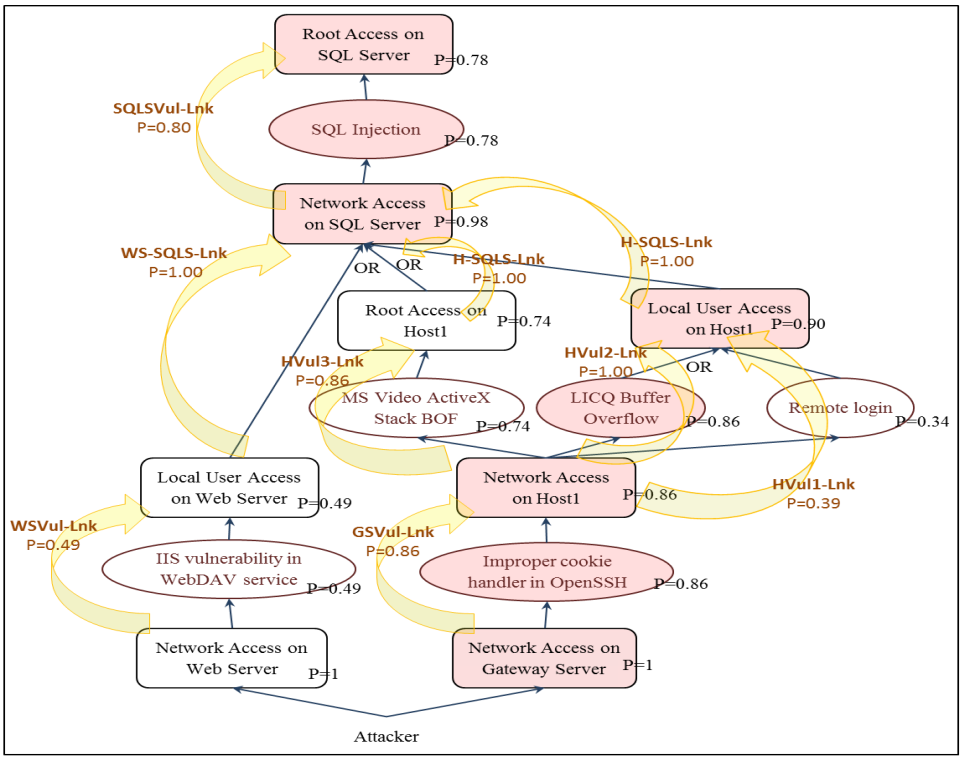

Figure 6. The Bayesian Attack Graph Produced by the NSM of the Network Under Analysis.

Table 5. Multi-Column Table

Security Threat	CVE ID	Probability of Threat			Damage Impact of Threat		Risk of Security Threat
		Exploitability Metrics	Exploit Probability (Single Vulnerability)	Exploit Probability (Unconditional Probability in BAG)	Impact Metrics	Impact Value	
Exploit of Improper cookie handler in OpenSSH	CVE 2007-4752	AV=N AC=M AU=N	0.86	0.86	$Impac_{Conf} = P$ $Impact_{Integ} = P$ $Impact_{Avail} = P$	0.62	0.53
Exploit of Remote login	CA 1996-83	AV= N AC=H AU=S	0.39	0.34	$Impac_{Conf} = C$ $Impact_{Integ} = C$ $Impact_{Avail} = P$	0.92	0.31
Exploit of LICQ Buffer Overflow (BOF)	CVE 2001-0439	AV= N AC=L AU=N	1	0.86	$Impac_{Conf} = P$ $Impact_{Integ} = P$ $Impact_{Avail} = P$	0.62	0.53
Exploit of MS Video ActiveX Stack BOF	CVE 2008-0015	AV= N AC=M AU=N	0.86	0.74	$Impac_{Conf} = C$ $Impact_{Integ} = C$ $Impact_{Avail} = C$	0.96	0.71
Exploit of SQL Injection	CVE 2008-5416	AV= N AC=L AU=S	0.8	0.78	$Impac_{Conf} = C$ $Impact_{Integ} = C$ $Impact_{Avail} = C$	0.96	0.74
Exploit of IIS vulnerability in WebDAV service	CVE 2009-1535	AV= N AC=H AU=N	0.49	0.49	$Impac_{Conf} = C$ $Impact_{Integ} = C$ $Impact_{Avail} = C$	0.96	0.74

probability of root access for the attacker would be 0, because the attacker does not direct network access to *SQLServer*.

The structure and values of resulted BAG are compared with the BAG and Bayesian decision network produced by applying [8] and [10] to the test network, respectively. The conformance of the BAG of our framework with these two models shows the validity of the models and processes of the proposed framework.

The Damage impact of each vulnerability exploit is also specified in Table 5. The damage impact is calculated according to Table 2 and Eq. (5). The final risk values in the latter column are obtained by Eq. (7). The fifth row of Table 5 is shaded for the riskiest vulnerability according to the resulted risk values. The most probable path of the attacker to gain root access to the SQL server is also highlighted in Figure 6 with a red shade.

From another risk management perspective, the effect of vulnerability exploitation on security requirements is studied. The probability of security requirements violations for the test network is presented in Table 6. In the first column, specified security requirements are listed, and the privileges of the attacker which violates those requirements are also showed in the second column.

Table 6. Security Requirements With Their Probability of Violation.

Security Requirement	Violations of Security Requirement	Probability of violation
All access of foreign users to SQL Server is denied.	CA 1996-83Network Access of attacker to SQL Server	0.98
	Local User Access of attacker to SQL Server	0.78
	Root Access of attacker to SQL server	0.78
Network access of foreign users to Web Server is allowed.	Local User Access of attacker to Web Server	0.49
	Root Access of attacker to Web Server	0
Network access of foreign users to Gateway Server is allowed.	Local User Access of attacker to Gateway Server	0
	Root Access of attacker to Gateway Server	0
All access of foreign users to Host1 is denied.	Network Access of attacker to Host1	0.86
	Local User Access of attacker to Host1	0.90
	Root Access of attacker to Host1	0.74

6 CONCLUSIONS AND FUTURE WORK

To reduce the need for security experts in network security risk estimation, this paper proposed a framework to represent experts' security knowledge. Abstract Security Model presents an abstract view to the network components which participate in security risks. ASM is designed once and can be frequently used to construct the Network Security Model for any network. Using the provided ASM, the process iteratively completes the NSM according to the specifications of the given NUA and calculates the risk values simultaneously. In the final NSM, the security threats of the network and their risk values are obtained. The information needed for constructing NSM from ASM can be provided by system documents, automatic scanners, or network administrator. In all three cases, there is a minimum need for security expertise. In this paper, the specifications of NUA are given manually, although it can be automated.

The proposed framework is dynamic and flexible to network changes. By adding an object to the NSM, its effects on the other objects are applied automatically by the operations of ASM classes. Another advantage of our analysis model is conformity with the network's physical model that enables the administrator to have a better assignment of security controls. For each Data Terminal Units in the network, an object is instantiated in the analysis model, and the effect of each Data Communication Units are represented by connection objects.

For future work, we are going to fully automate the process of constructing NSM. To this end, the process of constructing NSM should be formally defined. This framework can also be extended to cover the risk mitigation phase and recommend the best set of countermeasures according to estimated risks. Providing a UML profile for network security risk analysis from the proposed models is also considered for our future plans.

References

[1] X. Li, M. Li, and H. Wang. Research on Network Security Risk Assessment Method Based on Bayesian Reasoning. In *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 1–7. IEEE, 2019. ISBN 978-1-7281-1190-2. doi:10.1109/ICEIEC.2019.8784470.

[2] B. Karabacak and I. Sogukpinar. ISRAM: information security risk analysis method. *Computers & Security*, 24(2):147–159, 2005. doi:10.1016/j.cose.2004.07.004.

[3] P. Saripalli and B. Walters. Quirc: A quantitative impact and risk assessment framework for cloud security. In *2010 IEEE 3rd inter-*

national conference on cloud computing, pages 280–288. IEEE, 2010. ISBN 978-1-4244-8207-8. doi:10.1109/CLOUD.2010.22.

[4] CWE. Forum of Incident Response and Security, Common Vulnerability Scoring System v3.1. `http://cwe.mitre.org/cwss/cwss_v1.0.1.html`, Date Accessed: December 10, 2019.

[5] CWE. Forum of Incident Response and Security, Common Weakness Scoring System v1.0.1. `https://www.first.org/cvss/v3-1`, Date Accessed: December 10, 2019.

[6] A. Mukhopadhyay, S. Chatterjee, K. K. Bagchi, P. J. Kirs, and G. K. Shukla. Cyber risk assessment and mitigation (CRAM) framework using logit and probit models for cyber insurance. *Information Systems Frontiers*, 21(5):997–1018, 2019. doi:10.1007/s10796-017-9808-5.

[7] S. Kabir and Y. Papadopoulos. Applications of Bayesian networks and Petri nets in safety, reliability, and risk assessments: A review. *Safety science*, 115:154–175, 2019. doi:10.1016/j.ssci.2019.02.009.

[8] N. Poolsappasit, R. Dewri, and I. Ray. Dynamic security risk management using bayesian attack graphs. *IEEE Transactions on Dependable and Secure Computing*, 9(1):61 – 74, 2011. doi:10.1109/TDSC.2011.34.

[9] B. Kordy, M. Pouly, and P. Schweitzer. Probabilistic reasoning with graphical security models. *Information sciences*, 342:111–131, 2016. doi:10.1016/j.ins.2016.01.010.

[10] M. Khosravi-Farmad, R. Rezaee, A. Harati, and A. G. Bafghi. Network security risk mitigation using Bayesian decision networks. In *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 267–272. IEEE, 2014. ISBN 978-1-4799-5487-2. doi:10.1109/ICCKE.2014.6993444.

[11] M. Khosravi-Farmad, R. Rezaee, and A. G. Bafghi. Considering temporal and environmental characteristics of vulnerabilities in network security risk assessment. In *2014 11th International ISC Conference on Information Security and Cryptology*, pages 186–191. IEEE, 2014. ISBN 978-1-4799-5383-7. doi:10.1109/ISCISC.2014.6994045.

[12] T. Sommestad, M. Ekstedt, and H. Holm. The cyber security modeling language: A tool for assessing the vulnerability of enterprise system architectures. *IEEE Systems Journal*, 7(3):363–373, 2012. doi:10.1109/JSYST.2012.2221853.

[13] H. Holm, K. Shahzad, M. Buschle, and M. Ekstedt. P2CySeMoL: Predictive, Probabilistic Cyber Security Modeling Language. *IEEE Transactions on Dependable and Secure Computing*, 12(6):626 – 639, 2014. doi:10.1109/TDSC.2014.2382574.

[14] P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, graph-based network vulnerability analysis. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 217–224, 2014. doi:10.1145/586110.586140.

[15] R. Rezaee, A. G. Bafghi, and M. Khosravi-Farmad. A threat risk estimation model for computer network security. In *2016 6th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 223–228. IEEE, 2016. ISBN 978-1-5090-3586-1. doi:10.1109/ICCKE.2016.7802144.

[16] S. H. Houmb, V. N. Franqueira, and E. A. Engum. Quantifying security risk level from CVSS estimates of frequency and impact. *Journal of Systems and Software*, 83(9):1622–1634, 2010. doi:10.1016/j.jss.2009.08.023.

[17] J. Zhou, G. Reniers, and L. Zhang. Petri-net based attack time analysis in the context of chemical process security. *Computers & Chemical Engineering*, 130:106546, 2019. doi:10.1016/j.compchemeng.2019.106546.

[18] X. Zhang and D. Zhang. Quantitative Risk Assessment of Cyber Physical Power System Using Bayesian Based on Petri Net. In *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pages 988–992. IEEE, 2018. ISBN 978-1-5386-6005-8. doi:10.1109/CCIS.2018.8691249.

[19] D. Pramod and S. V. Bharathi. Developing an Information Security Risk Taxonomy and an Assessment Model using Fuzzy Petri Nets. *Journal of Cases on Information Technology (JCIT)*, 20 (3):48–69, 2018. doi:10.4018/JCIT.2018070104.

[20] S. Lee, S. Kim, K. Choi, and T. Shon. Game theory-based security vulnerability quantification for social internet of things. *Future Generation Computer Systems*, 82:752–760, 2018. doi:10.1016/j.future.2017.09.032.

[21] S. Musman and A. Turner. A game theoretic approach to cyber security risk management. *The Journal of Defense Modeling and Simulation*, 15(2):127–146, 2018. doi:10.1177/1548512917699724.

[22] S. Yang, Y. Zhang, and C. Wu. Attack-Defense Quantification Based On Game-Theory. *arXiv preprint arXiv:1902.10439*, 2019. doi:10.1177/1548512917699724.

[23] Y. Yang, B. Che, Y. Zeng, Y. Cheng, and C. Li. MAIAD: a multistage asymmetric information attack and defense model based on evolutionary game theory. *Symmetry*, 11(2):215, 2019. doi:10.3390/sym11020215.

[24] K. Zhang. Analysis method based on rough attack-defense Bayes game model. *International Journal of Security and Its Applications*, 9(1):

109–118, 2015. doi:10.14257/ijsia.2015.9.1.12.

[25] T. Sommestad, M. Ekstedt, and P. Johnson. Combining defense graphs and enterprise architecture models for security analysis. In *2008 12th International IEEE Enterprise Distributed Object Computing Conference*, pages 349–355. IEEE, 2008. ISBN 978-0-7695-3373-5. doi:10.1109/EDOC.2008.37.

[26] V. Lisỳ and R. Píbil. Computing optimal attack strategies using unconstrained influence diagrams. In *Pacific-Asia Workshop on Intelligence and Security Informatics*, pages 38–46. Springer, 2008. ISBN 978-3-642-39692-2. doi:10.1007/978-3-642-39693-9_5.

[27] A. Laszka, M. Felegyhazi, and L. Buttyan. A survey of interdependent information security games. *ACM Computing Surveys (CSUR)*, 47(2): 1–38, 2014. doi:10.1145/2635673.

[28] A. R. Hota and S. Sundaram. Interdependent security games under behavioral probability weighting. In *International Conference on Decision and Game Theory for Security*, pages 150–169. Springer, 2015. ISBN 978-3-319-25593-4. doi:10.1007/978-3-319-25594-1_9.

[29] S. Amin, G. Schwartz A, and S. S. Sastry. Security of interdependent and identical networked control systems. *Automatica*, 49(1):186–192, 2013. doi:10.1016/j.automatica.2012.09.007.

[30] M. Abdallah, P. Naghizadeh, A. R. Hota, T. Cason, S. Bagchi, and S. Sundaram. Behavioral and Game-Theoretic Security Investments in Interdependent Systems Modeled by Attack Graphs. *arXiv preprint arXiv:2001.03213*, 2020. doi:10.1109/tcns.2020.2988007.

[31] S. A. Hasheminasab, B. Tork Ladani, and T. Alpcan. Interdependent Security Game Design over Constrained Linear Influence Networks. *ISeCure-The ISC International Journal of Information Security*, 11(2):95–111, 2019. doi:10.22042/isecure.2019.186923.467.

[32] W. Shang, T. Gong, C. Chen, J. Hou, and P. Zeng. Information Security Risk Assessment Method for Ship Control System Based on Fuzzy Sets and Attack Trees. *Security and Communication Networks*, 2019. doi:10.1155/2019/3574675.

[33] A. T. Al Ghazo, M. Ibrahim, H. Ren, and R. Kumar. A2G2V: Automatic Attack Graph Generation and Visualization and Its Applications to Computer and SCADA Networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1 – 11, 2019. doi:10.1109/TSMC.2019.2915940.

[34] M. Albanese, S. Jajodia, and S. Noel. Time-efficient and cost-effective network hardening using attack graphs. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, pages 1–12. IEEE, 2012. ISBN 978-1-4673-1624-8. doi:10.1109/DSN.2012.6263942.

[35] S. Zhang and S. Song. A novel attack graph posterior inference model based on bayesian network. *Journal of Information Security*, 2(1):8–27, 2011. doi:10.4236/jis.2011.21002.

[36] T. Sommestad, M. Ekstedt, and P. Johnson. A probabilistic relational model for security risk analysis. *Computers & security*, 29(6):659–679, 2010. doi:10.1016/j.cose.2010.02.002.

[37] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. rentice Hall Press, 2009. ISBN 978-0136042594.

[38] K. Zhou, A. Martin, and Q. Pan. The belief noisy-or model applied to network reliability analysis. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 24(06):937–960, 2016. doi:10.1142/S0218488516500434.

[39] J. Sembiring, M. Ramadhan, Y. S. Gondokaryono, and A. A. Arman. Network security risk analysis using improved MulVAL Bayesian attack graphs. *International Journal on Electrical Engineering and Informatics*, 7(4), 2015. doi:10.15676/ijeei.2015.7.4.15.

[40] G. S. Bopche and B. M. Mehtre. Attack graph generation, visualization and analysis: issues and challenges. In *International Symposium on Security in Computing and Communication*, pages 379–390. Springer, 2014. ISBN 978-3-662-44965-3. doi:10.1007/978-3-662-44966-0_37.

[41] G. Stoneburner, A. Goguen, and A. Feringa. Risk management guide for information technology systems. *Nist special publication*, 2002.

[42] R. S. Ross. Managing information security risk. *Nist special publication*, 2011.

[43] CCMB. Common Criteria for Information Technology Security Evaluation, ISO/IEC 15408, Version 3.1. 2017. *Common Criteria*, 2017.

[44] NIST. National Vulnerability Database. `https://nvd.nist.gov/`, Date Accessed: June 9, 2020.

[45] CVE. CVE Reference Map for Source OSVDB. `https://cve.mitre.org/data/refs/refmap/source-OSVDB.html`, Date Accessed: June 9, 2020.

Razieh Rezaee received her B.S. and M.S. degrees in computer engineering from Ferdowsi University, Mashhad, Iran, in 2007 and 2009, respectively. She is currently pursuing a Ph.D. degree in computer engineering with the Department of Computer Engineering, Ferdowsi University, Mashhad, Iran. She was a researcher in Web Technology Lab, Mashhad, Iran, from 2007 to 2009. She is a researcher in Data and Communication Security Lab, Mashhad, Iran, since 2012. Her current research interests include security risk management, security vulnerabilities, and security models.

Abbas Ghaemi-Bafghi received his B.S. degree in Applied Mathematics in Computer from Ferdowsi University of Mashhad, Mashhad, Iran, in 1995. He received his M.S. and Ph.D. degrees in Computer engineering from Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 1997 and 2004, respectively. He is a member of the Computer Society of Iran (CSI) and the Iranian Society of Cryptology (ISC). He is an associate professor in the computer engineering department, Ferdowsi University of Mashhad, Mashhad, Iran. His research interests are cryptology and security.