Application of Bayesian network in risk assessment for website deployment scenarios

Vu Thi Huong Giang, Nguyen Manh Tuan

***Abstract*—The rapid development of web-based systems in the digital transformation era has led to a dramatic increase in the number and the severity of cyber-attacks. Current attack prevention solutions such as system monitoring, security testing and assessment are installed after the system has been deployed, thus requiring more cost and manpower. In that context, the need to assess cyber security risks before the deployment of web-based systems becomes increasingly urgent. This paper introduces a cyber security risk assessment mechanism for web-based systems before deployment. We use the Bayesian network to analyze and quantify the cyber security risks posed by threats to the deployment components of a website. First, the deployment components of potential website deployment scenarios are considered assets, so that their properties are mapped to specific vulnerabilities or threats. Next, the vulnerabilities or threats of each deployment component will be assessed according to the considered risk criteria in specific steps of a deployment process. The risk assessment results for deployment components are aggregated into the risk assessment results for their composed deployment scenario. Based on these results, administrators can compare and choose the least risky deployment scenario.**

***Tóm tắt*—Sự phát triển mạnh mẽ của các hệ thống trên nền tảng web trong công cuộc chuyển đổi số kéo theo sự gia tăng nhanh chóng về số lượng và mức độ nguy hiểm của các cuộc tấn công mạng. Các giải pháp phòng chống tấn công hiện nay như theo dõi hoạt động hệ thống, kiểm tra và đánh giá an toàn thông tin mạng được thực hiện khi hệ thống đã được triển khai, do đó đòi hỏi chi phí và nhân lực thực hiện lớn. Trong bối cảnh đó, nhu cầu đánh giá rủi ro an toàn thông tin mạng cho các hệ thống website trước khi triển khai thực tế trở nên cấp thiết. Bài báo này giới thiệu một cơ chế đánh giá rủi ro an toàn thông tin mạng cho các hệ thống website trước khi triển khai thực tế. Chúng tôi sử dụng mạng Bayes để phân tích và định lượng rủi ro về an toàn thông tin do các nguồn đe dọa khác nhau gây ra trên các thành phần triển khai của một website. Đầu tiên, các thành phần triển khai của các kịch bản triển khai website tiềm năng được mô hình hoá dưới dạng các tài sản, sao cho các thuộc tính của chúng đều được ánh xạ với các điểm yếu hoặc nguy cơ cụ thể. Tiếp đó, các điểm yếu, nguy cơ của từng thành phần triển khai sẽ được đánh giá theo các tiêu chí rủi ro đang xét tại mỗi thời điểm cụ thể trong quy trình triển khai. Kết quả đánh giá của các thành phần triển khai được tập hợp lại thành kết quả đánh giá hệ thống trong một kịch bản cụ thể. Căn cứ vào kết quả đánh giá rủi ro, người quản trị có thể so sánh các kịch bản triển khai tiềm năng với nhau để lựa chọn kịch bản triển khai ít rủi ro nhất.**

Keywords—deployment scenario; risk assessment; CVE; Bayesian network; scenario-based risk assessment.

Từ khoá—kịch bản triển khai; đánh giá rủi ro; CVE; mạng Bayes; đánh giá rủi ro dựa trên kịch bản.

I. INTRODUCTION

Web-based applications and systems today come for a nearly endless variety of purposes, including e-commerce, business, education, social media, entertainment, and so on. Individuals and organizations that own such systems are increasingly interested in securing the web operation. Most websites are fully packaged by their providers, and their quality is assured by compliance with standards as cited in the service-level agreement. In this case, the system deployment plays an important role in securing the next stages of web operation such as operation, maintenance, tuning, and repair of these systems.

The system deployment scenario summarizes deployment activities, and it is the basis for designing the deployment architecture. It includes the logical architecture combined with the quality requirements of the system to be deployed. Logical architecture describes

This manuscript is received on July 23, 2021. It is commented on July 28, 2021 and accepted on September 27, 2021 by the first reviewer. It is commented on November 01, 2021 and accepted on November 15, 2021 by the second reviewer.

software components that provide the services necessary to meet the business goals of the system. System quality requirements include availability, latent capacity, performance, scalability, security, and serviceability. The deployment architecture maps software components in the logical architecture to computing nodes on the Internet or Intranet such as CPUs, storage devices, hardware components, and other network devices of the physical environment. The software components of the logical architecture or the hardware components of the physical environment are called deployment components. Activities aimed to realize this deployment architecture are called deployment activities; these are performed according to the deployment process.

Problems arising during the deployment process such as the selection of older versions of software, errors in the configuration of components or default user accounts will pose high security risks to the system. Web-based systems that are maintained in an infrequently updated environment also face security risks. The study [1] of 5.6 million websites over an 18-month period found that most of the pages analyzed used outdated software. The results showed that 95% of these sites used at least one product containing weaknesses.

Therefore, before setting into operation a web-based system, the manager often needs to assess the potential cyber security risks that occur with its components. The purpose of the assessment is to prevent and control risks and provide a scientific basis for ensuring cyber security [2]. Currently, qualitative and quantitative risk analysis are two common techniques used to assess information security risks [3]. These techniques are integrated into different assessment processes or frameworks that can be used in different contexts. Besides, the expert opinion and the specific requirements and features of the organization operating the web-based system are important factors in finding the most compatible model for the risk assessment process. In addition, this process brings many difficulties such as uncertainties, complexity of quantitative calculations or lack of data on security breaches, incidents, and threats. These challenges hinder the development of risk assessment models. The Bayesian network and probability theory - described as an expert system of uncertain facts - are powerful tools to cope with these challenges [4].

In this paper, we propose a mechanism to assess cyber security risks before deploying a web-based system. The risk assessment process is carried out according to the possible deployment scenarios of this system. The deployment components are considered assets to be protected. An asset consists of a finite number of states. The asset can change from one state to another state in response to a deployment activity. Relationships between assets are expressed through deployment activities that cause state transition of an asset, while also triggering the state transition of another asset. Deployment activities are associated with specific risk criteria. The attributes of deployment components are mapped with a known set of vulnerabilities and threats. The latter will be evaluated according to the risk criteria under consideration. The likelihood of risks is calculated based on the Bayesian network, which provides quantitative calculations for the assessment process.

The paper is organized as follows. The theoretical foundations and related works are reviewed in section 2. Section 3 describes the data model representing the deployment components and the deployment activities. The proposed mechanism for applying Bayesian networks in security risk assessment for website deployment scenarios is described in section 4. Some experimental results are provided in section 5. Final conclusions are discussed in the last section.

II. Related works

In this section, the research results are summarized and introduced in three main groups: the website deployment process, the relationship between assets, threats and security risks, and the usage of Bayesian networks in security risk assessment.

A. Deployment process

There are many types of deployments: deploying a new website system (for the first time), replacing or upgrading new functionalities for an existing website system.

Starting just after the completion of the product testing phase, the deployment process consists of 4 stages as shown in Fig. 1, in which the security objectives will be expressed in different ways in each stage.

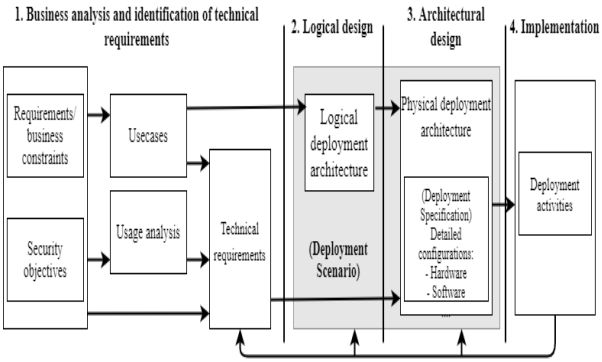


Fig. 1. Stages in the deployment process [5].

1. Business analysis and identification of technical requirements

This stage aims at determining the business goals of a deployment, the business constraints that could limit the achievement of this deployment, and the security objectives of this deployment. For example, a business goal could be "Deploy a content management website providing access to enterprise services". A business constraint could be "Using existing hardware and network resources". Security objectives could be expressed as non-functional requirements of this process, such as "Make secure communication between deployed servers" or "Make secure transactions with requested third-party services." Next, the identified business constraints are converted into technical requirements of the deployment process such as system availability, performance, scalability, and security (authentication, authorization, identity management, etc.), considering the identified security objectives.

2. Logical design

The outputs of the first stage are used to define the logical deployment architecture, which is composed of components providing the software services needed to meet the business goals of a deployment. The logical deployment architecture is then combined with the specification of technical requirements to constitute a *deployment scenario*. At this stage, the *security objectives* are decomposed into specific security objectives of the *assets* to be protected such as servers, applications, network infrastructure, and data. Based on the logical architecture of the assets, managers can choose appropriate security controls and countermeasures (such as firewalls, proxy servers, IPS/IDS, network monitoring systems, etc.) to minimize security risks, while considering their costs.

3. Architectural design

This stage aims at identifying the physical resources required to set into operation the web-based system according to the logical architecture defined in the previous step. The result of this phase is a *detailed design specification*. This specification could be combined with a database of vulnerabilities (CVE - Common Vulnerabilities and Exposures) [6] or lists of types of hardware and software weaknesses (CWE - Common Weakness Enumeration) [7] to identify potential vulnerabilities in assets and select appropriate countermeasures. For example, CVE-2006-3747 describes the vulnerability of Apache web server software version 2.0.36, allowing an attacker to exploit and gain user privileges on the web server.

4. Implementation

The detailed design specification produced in the previous stage is a starting point for implementation of the deployment. During the implementation phase, the deployment architecture is built. The steps taken can vary depending on the type of assets, in an order determined by the manager. For each asset, the deployment activities change its state while triggering the state transitions of related assets. For example, given an Apache web server to be deployed, the deployment activities for this asset include: (i) Setting up the platform; (ii) Installing

the system environment; (iii) Installing services; (iv) Configuring system. Website systems also include some specific deployment activities such as putting the website's executable code on the web server, initializing data into the database server, or managing the website's domains.

In this way, deployment scenarios and deployment activities play an important role to identify the assets to be protected and the threats of exploiting potential vulnerabilities that pose security risks on the property. Using them, the manager may select countermeasures to eliminate risks or reduce risks to an acceptable level, primarily based on his own experiences. Meanwhile, current studies/solutions have not focused on using system deployment scenario information as input for the risk assessment process. The reason is that the scenarios do not have a common pattern and the relationship and detailed configuration of assets varies depending on the problem, while there is no data model flexible enough to describe this information.

B. Relationship between assets, threats, and security risks

According to ISO/IEC 27005 [8], information security risks are typically expressed as a combination of the consequences of information security events and their likelihood of occurrence. The risk assessment process begins with the identification of the asset's security objectives: *confidentiality, integrity, and availability* (CIA) of information processed, transmitted, and stored [2]. This is followed by identifying vulnerabilities, asset threats, and assessing the negative impact of attacks. ISO 15408 provides definitions of system assets, threats, vulnerabilities, and risks [9].

Threats are the potential for harmful and undesired events to system assets. Threats can arise from objective or subjective reasons, from within or outside the organization. Some threats can simultaneously affect multiple components and cause different effects depending on the affected component. Threats of an information system are classified into 6 groups: operating environment, people, operational errors, unauthorized behavior, violation of security policy, malicious code [21].

Vulnerability is potential flaws inside assets that can be exploited by threats and increases the risk of assets being attacked. Vulnerabilities are not self-destructive but require the threat of exploitation by attackers. The classification of vulnerabilities is related to the properties of the asset under consideration [10]. Detailed information about potential vulnerabilities within an asset are provided in databases such as CVE [6] / CWE [7] and NVD (National Vulnerability Database) [19]. According to [13], the exploitability of each vulnerability is pre-calculated as a basis for experts to manually assess the impact of the vulnerability on assets, as illustrated in Table 1.

TABLE 1. EXPLOITABILITY OF VULNERABILITY

Element	Value	Score
Access Vector (AV)	Local (L)	0.395
	Adjacent Network (A)	0.646
	Network (N)	1.0
Access Complexity (AC)	High (H)	0.35
	Medium (M)	0.61
	Low (L)	0.71
Authentication (AU)	Multiple (M)	0.45
	Single (S)	0.56
	None (N)	0.704

Risk is the possibility of threats causing loss or damage to assets when their vulnerabilities are exploited by these threats. Typical security breaches include the loss of confidentiality, integrity, and availability of assets [2]. Risks can be classified according to various criteria, for example based on the severity of the risk. The OWASP Top 10 [11] provides a classification of the security risks associated with Web applications. Based on this classification, many research have been conducted for identifying and analyzing vulnerabilities of web-based systems. For example, in [24], an interactive application security testing (IAST) approach has been proposed, resulting in 249 identified vulnerabilities of government Web applications.

To reduce the risk of assets, managers select suitable countermeasures from technical

solutions (e.g., using firewalls, monitoring systems, intrusion detection systems, applying a patch on the server, etc.) or other management controls such as developing security policies and procedures. The selection criteria rely on an analysis of different aspects of countermeasures: the effectiveness and the cost when applied to protect assets, the affection to the system's performance, feasibility, or user acceptance.

Risk assessment methods are usually classified into two approaches of qualitative and quantitative analysis. The qualitative approach that is based on the analysis of the relationship between threats, vulnerabilities and countermeasures related to protected assets is the most popular method [12]. By classifying risk levels (low, medium, high) and considering relevant factors, the qualitative method with its flexible characteristics and quick application allows managers to focus on high-priority risks. In addition, qualitative analysis also provides a basis for further quantitative risk assessments.

Quantitative assessment includes calculations related to the probability of risk and values that represent the impact of the risk [22]. These measurable assessment results provide more detailed information about cost-benefit analysis when developing risk response plans. However, quantitative risk assessment is a complex approach requiring expert knowledge with large amounts of input data. Many quantitative risk assessment frameworks have been proposed [3], some based on CVSS (Common Vulnerability Scoring System) [13] and CWSS (Common Weakness Scoring System) [14] which quantitatively measures the severity of the risks to potential vulnerabilities in assets. In [25], a fuzzy logic machine consisting of 27 "if-then" rules and linguistic variables allows scanning vulnerabilities of web-based systems that have been put into practice. While these frameworks have provided experts with quantitative parameters of the exploitability or impact indicators of specific asset's vulnerabilities, they did not include information on the relationship and impact of different vulnerabilities on the same asset. Therefore, the accuracy of risk classification and assessment are mainly based on expert experience and knowledge. The study [15] is one of the pioneering studies on modeling the relationship between the asset's vulnerabilities through the construction of an attack graph (AG - attack graph) representing multi-step attacks that exploit vulnerabilities. Also in this study, the authors combined Bayesian network with attack graph to form Bayesian attack graph (BAG) that provided quantitative calculations for security risks. However, this study focused on showing the relationship between the asset's vulnerabilities, but the parameters such as vulnerability exploitability and vulnerability impact were not analyzed in detail. The results have covered only the probabilistic analysis of exploiting each vulnerability on the asset over the lifetime of the system.

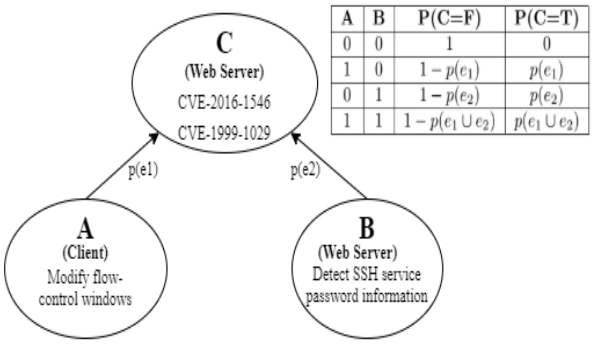
C. Bayesian network in information security risk assessment

The Bayesian network is built out of the Bayesian conditional probability formula. Each Bayesian network consists of two parts [23]. The first part is a graphical representation consisting of nodes and edges of the graph. Each node represents a random variable, and each edge between the node represents the conditional dependence between the corresponding random variables. The second part is the joint probability distribution of the variables determined by the graph structure of the network. Each node includes probabilistic information of certain states. Edges are directed from parent node to child node, each node is attached to a conditional probability table (CPT) based on the values of the parent node. The Bayesian formula allows calculating the probability of hypothesis A when event B has occurred [23].

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

Bayesian networks have been widely applied in building security risk assessment models and frameworks [15], [16], [17], [18]. In these studies, the determination of Bayesian network structure (dependencies between nodes), the priori probability and conditional probability of nodes were established by experts. In [16], the

Bayesian network is used to dynamically evaluate and quantify the level of security risk in an SCADA network. In [15], the Bayesian network is used to model the attack graph, i.e., the paths of an attacker through the system by exploiting successive vulnerabilities, called Bayesian attack graph (BAG). From the results of [15], many other researches like [17], [18] have applied BAG as a security risk assessment framework.



A	B	P(C=F)	P(C=T)
0	0	1	0
1	0	$1-p(e_1)$	$p(e_1)$
0	1	$1-p(e_2)$	$p(e_2)$
1	1	$1-p(e_1 \cup e_2)$	$p(e_1 \cup e_2)$

Fig. 2. A simple BAG.

In a BAG, nodes represent the different security states that an attacker can acquire, and edges to a node from its parent nodes represent activities that compromise its state. The conditional probability table (CPT) at each node provides information about the likelihood of the current state based on different combinations of parent nodes. Fig. 2 illustrates a simple BAG. It consists of 3 nodes A, B, and C, each representing a potential security breach state for a component in the network. Nodes also represent the same component as different states. Node C represents a state of a Web server in which two vulnerabilities CVE-1999-1029, CVE-2016-1546 could be exploited. Node A represents the state of a client in the network connecting to this Web server and exploiting the vulnerability CVE-2016-1546. The edge from A to C represents an exploit of CVE-2016-1456 causing server service outages; the probability of success of this activity is *p(e1)*. Similarly, the edge from B to C describes an exploit of vulnerability CVE-1999-1029, in which the attacker attempts to login and guest the SSH service's password; the probability of success of this activity is *p(e2)*. Then, the formula of AND type (2) or the formula of OR type (3) is used to calculate the CPT's values of node *i*, depending on the relationship between edges to *i* from one of its parent nodes *j* [18]:

$$P(s_j \,|Pa[s_j]) = \begin{cases} 0 \;\exists\; s_i \in Pa[s_j] |\; s_i = 0 \\ \prod_{s_i=1} p(e_i) \end{cases} \quad (2)$$

$$P(s_j \,|Pa[s_j]) = \begin{cases} 0 \;\forall\; s_i \in Pa[s_j] | s_i = 0 \\ 1 - \prod_{s_i=1}(1 - p(e_i)) \end{cases} \quad (3)$$

where $Pa[s_j]$ is the set of parent nodes' states of node i, $p(e_i)$ is the probability of success of activity that changes the state of the network from s_i to s_j.

Thus, the structure of the Bayesian network or the BAG graph depends on two main factors: (i) the attack data and (ii) the detailed parameters of the system deployment configuration. However, the mechanisms and processes are used to combine these factors to define dependencies between nodes, priori, and conditional probabilities of nodes to fit the deployment scenario have not been clarified.

The above remaining issues are the motivation for our research. In this study, we propose a data model for deployment scenarios and an associated security risk assessment mechanism, that (i) clarify the relationships between assets and vulnerabilities in each deployment scenario, (ii) consider quantitative parameters of exploitability or the impact of a particular vulnerability on each asset, (iii) include expert knowledge of the vulnerabilities' exploitability under potential attack scenarios.

III. DATA MODEL

A. Asset

The deployment components are modeled as assets to be protected. Important attributes of assets include importance index, configuration parameters and security properties. The state of the asset corresponds to a set of values of these properties at a time. In other words, a change in the value of these properties leads to a change in the state of the assets.

1. Asset types

We identify four types of assets including servers, network infrastructure, applications, and data. Each asset of the above categories has a

specific value for the *asset_value* attribute, representing the quantitative value for assets established by the manager. The value of the asset can be calculated based on the total cost of purchasing assets such as the cost of hardware, software, or services from third parties. In addition, each asset class also includes properties that describe information such as its identifier, its brief description, its creation date, and so on.

2. Configuration parameters

For each asset type, the configuration parameters will be defined in terms of tuples *(attribute, value)*. For example, Table 2 shows a description of the Web server component:

TABLE 2. CONFIGURATION PARAMETERS OF WEB SERVER

Asset	Configuration parameters
Webserver	**Hardware configuration information** Properties: - Device classification - Supplier - Configuration: {IP Address, Processor, RAM, Storage Drive} **Software configuration information** - Operating System: {Name, Version, Authentication, Logging} - Service: {Service Name, Service Port, Protocol, State, Logging} **Web service configuration information** - Web Services: {Service Port, Protocol, Vendor Name, Open Source: (yes/no), Technology, Version, Configuration File, Authentication, Logging}

As analyzed in section II.A, when assets are set to specific values at stage 3, the CVE and CWE databases are used to identify the CVE-IDs corresponding to the configuration parameters. Assess vulnerabilities are represented by the mapping of the configuration parameter to a set of CVE-IDs.

The CVE_ID values and the NVD and CVSS databases enable the calculation of the vulnerability's exploitability and its impact metrics. CVSS provides groups of metrics (Base, Temporal, and Environmental) to quantitatively assess the severity of existing IT weaknesses. Each of these groups generates a *score* from 0 to 10, where 10 is the severest.

3. Importance index

The value of the *weight_of_asset* attribute represents the importance index of an asset in a deployment scenario. This value depends on the asset type and will be approved by the owner or manager. The *asset_value* attribute is calculated from the cost of acquiring or deploying the asset, which may be impaired when risks arise. Thus, the overall value of an asset is determined through the formula:

$$TotalAssetValue = AssetValue \times WeightOfAsset \quad (4)$$

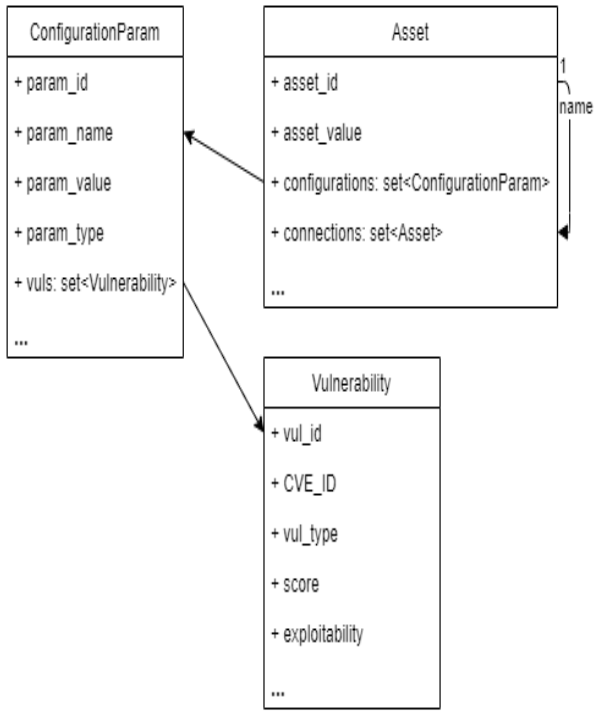


Fig. 3. Vulnerabilities of assets mapped with configuration parameters.

4. Security attributes

Each asset is associated with a set of security attributes. A security attribute is represented as a tuple *(name, value, security objective)*.

B. Deployment Scenario

The deployment scenario includes a set of security objectives to be met, a set of assets to be used and the relationship between those assets.

1. Security objectives

The security objectives that the deployment scenario must achieve and not be compromised in any way when security risks arise. These goals include confidentiality, integrity, and availability (CIA). Security objectives are reflected through attributes that represent the importance of the goal and the security attributes of the assets that meet that goal. We assign the values of the security objectives to 3 levels *{Low, Medium, High}* with the corresponding score values of *{1, 2, 3}*. For the convenience of managers in establishing the appropriate value, descriptions of the effects or rates of loss of value are constructed for each level.

2. Relationship between assets

At the logical design stage, the relationships between properties are determined. These relationships are represented by the transition of one asset that can simultaneously trigger the transition of another asset. For example, the relationship between a website's module and a web server is *hosted-on*. In the case of Web server and database server assets, the relationship between them is *connect-to*, representing the data flow between the two.

The relationship between the assets to be protected is closely related to the security objectives of the deployment scenario. For example, in the case of a website and a web server with a *hosted-on* relationship, the security objectives are also interdependent: the availability of the website's module depends on the availability of the Web server.

3. Risk countermeasures

Risks can be minimized by implementing various risk prevention measures. A common technique to eliminate a weakness in software is regularly updating patches from the vendor. Control mechanisms are used to limit, regulate, or minimize vulnerabilities; they can detect or prevent risks. The countermeasure description includes: *{Identifier, Description, Cost, Set of CVE-IDs covered by this measure}*.

The implementation of a security control for the asset is described by a mapping:

$$Asset\ X\ Countermeasure \rightarrow \{0,1\}$$

where value 1 shows that this countermeasure has been applied; 0 is the opposite.

C. Deployment process

At each stage of the deployment process, the deployment scenario information is updated through the deployment process activities.

1. Stages

The four identified stages are: business analysis and identification of technical requirements, logical design, architectural design, and implementation.

2. Deployment activities

Deployment activities aim to realize the deployment architecture. When a deployment activity has been performed, the state of the related assets is changed. The deployment activity is modeled by a tuple of value representing the activity identifier, the activity description, a set of related assets along with their source and target states.

IV. Security risk assessment mechanism for deployment scenarios

The main activities in the security risk assessment for website deployment scenarios include: 1- Identifying security objectives, 2- Analyzing the relationship between vulnerabilities, threats, and countermeasures and 3 - Security risk assessment

A. Identifying security objectives

The deployment scenario, including information about assets and relationships between assets, are updated at each stage of the deployment process as follows:

Stage 1: Define security objectives and technical requirements for deployment.

Stage 2: Identify the assets to be protected, their types, and their relationships. Set the security attributes for assets based on the identified security objectives.

Stage 3: Set the configuration parameters for assets and selected countermeasures to minimize risks.

Stage 4: Identify activities to realize the deployment scenario.

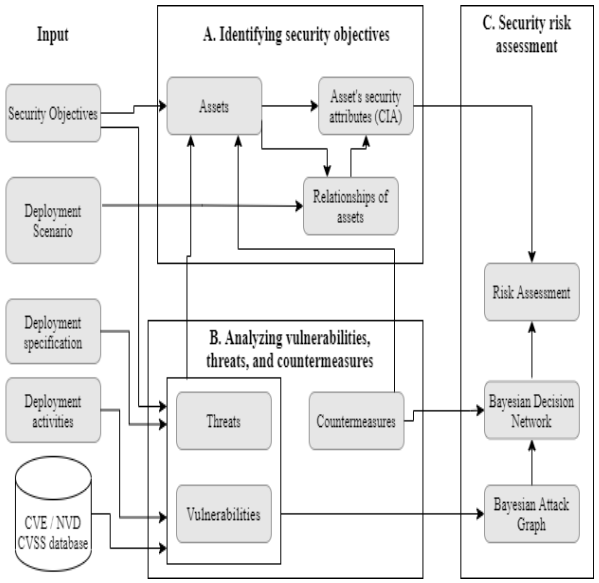


Fig. 4. Activities in security risk assessment for website deployment.

B. Analyzing vulnerabilities, threats, and countermeasures

This step aims at identifying the security risks for the protected assets by using the model depicted in Fig. 5.

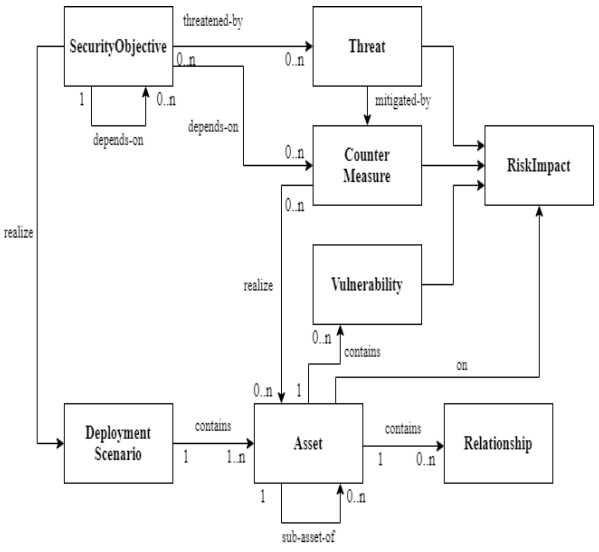


Fig. 5. Cyber security risk analysis model for deployment scenarios.

Security risks do not appear on their own, they are made up of threats that exploit vulnerabilities on assets. The severity of threats varies depending on their impact on security objectives. For example, attack threats from within an organization have serious consequences by exposing sensitive data. Vulnerabilities are identified on assets through detailed configuration information. Each vulnerability also has values representing its severity and exploitability. The analysis of the relationship between threats and vulnerabilities is the basis to define security risks along with estimates of probability/likelihood of impact.

Once the risk is identified, it can be assessed through its impact on the assets. If the risk is acceptable, then the manager only needs to communicate and monitor the risk within their organization. If the risk is unacceptable, it must be controlled through appropriate countermeasures. In the development process, the impact of risk is quantified using the following formula:

$$RiskImpact = TotalAssetValue * SeverityOfVulnerability * SeverityOfThreat * Probability \quad (5)$$

In formula (5), the risk impact shows the depreciation of the protected assets when potential security breaches are realized, making the security objectives unachievable. The total value of asset is calculated according to the formula (4). Other parameters are the severity of the vulnerability and the threat, and the probability of exploiting the vulnerability from that threat. To calculate this probability with better reliability, instead of using expert opinion, we use a Bayesian network to model the relationship between vulnerabilities on assets by a particular exploitable threat.

Countermeasures are applied to mitigate the impact of the risk. An asset can have many countermeasures. The list of possible countermeasures could be established manually by managers or could be collected from security vendors. Conversely, a control option can cover one or more of the asset's vulnerabilities. Analyzing the cost and the effectiveness of these measures helps the manager prepare an appropriate risk reduction plan.

C. Security risk assessment

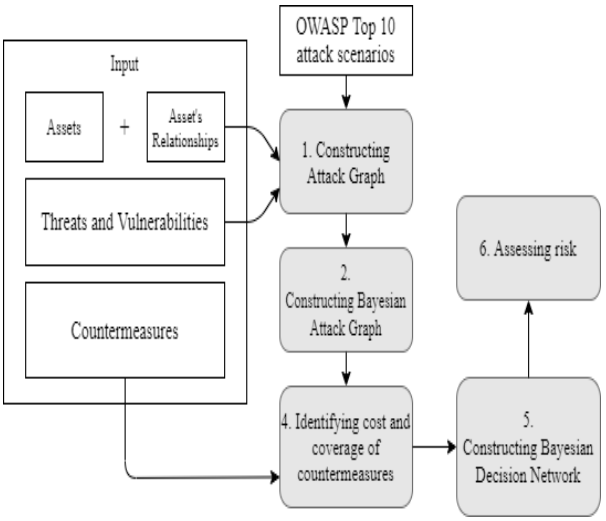


Fig 6. Security risk assessment for deployment scenario.

The results of the previous analysis step provided the main components of the security risks that were the input for the assessment at this step. Three inputs of this process are (i) a list of assets and their relationships; (ii) a list of identified vulnerabilities, identified threats, and selected countermeasures; and (iii) a list of possible attack scenarios as described by OWASP Top 10.

As illustrated in Fig. 6, the security risk assessment process begins with the construction of an attack graph. For that, the attack scenario is classified into different security risk types. Table 3 illustrates an example of an attack scenario posing sensitive data exposure risk.

TABLE 3. ATTACK SCENARIOS FOR A3:2017

Category	Attack scenarios
A3:2017-Sensitive Data Exposure	***Scenario #1***: A data that is encrypted in the database but is automatically decrypted on access -> allowing an SQL injection vulnerability to access this data in plaintext. ***Scenario #2***: Use a weak hash or salt to encrypt the stored password. A file upload vulnerability allows an attacker to obtain a password database.

Next, to analyze the cost and effectiveness of countermeasures, the Bayesian decision network (BDN) is built by extending the BAG with the addition of two more types of nodes: the *decision nodes* and the *utility nodes*.

For example, consider a BDN illustrated in Fig. 7. The nodes of BAG (representing assets' vulnerabilities) are called opportunity nodes (eclipse). The decision nodes (rectangle) represent the selected countermeasures to cover up assets' vulnerabilities. In this case, the countermeasure applied to the Web server is the mod_http2 module patch of the Apache Server 2.4.20.

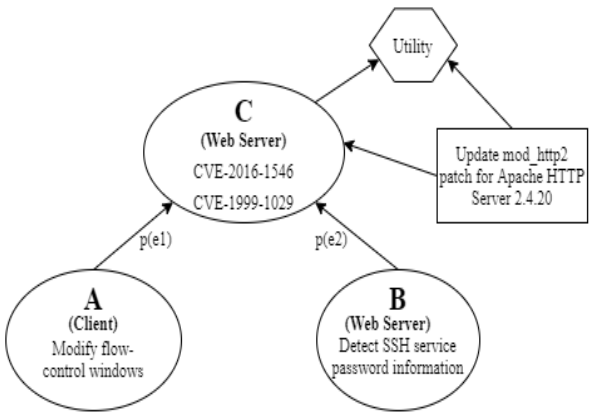


Fig. 7. Bayesian decision network.

To evaluate the effectiveness of the selected countermeasures, decision nodes are linked to a utility node (hexagon). The edge from an opportunity node to a utility node represents a vulnerability exploiting step. The edge from a decision node to an opportunity node describes the selected countermeasure to cover up a vulnerability. When a countermeasure is applied, it alters the attacker's ability to exploit vulnerabilities. The effectiveness of the applied countermeasure is quantified by using a utility table predefined by the manager. As illustrated in Table 4, the utility table consists of values representing the performance of jointly applied countermeasures.

TABLE 4. THE UTILITY TABLE STRUCTURE

	e_i vulnerability exploited	**e_i vulnerability not exploited**
Countermeasures applied	*value 1*	*value 2*
Countermeasures not applied	*value 3*	*value 4*

V. EXPERIMENT

A. Experiment context

The proposed approach is applied to evaluate the potential deployment scenarios of a website deployment architecture illustrated in Fig. 8. The security objectives of deployment architecture include "Ensure secure communication between web servers and database servers" and "Make data available for queries from web applications". Security breaches such as service interruptions on database servers or inappropriate information displayed on the website can compromise these objectives. In this architecture, the components of the system are deployed in two subnet zones. The DMZ zone contains a Web server that receives user requests from the Internet. The second subnet includes application servers and database servers, this is the trusted zone so outside access will be restricted. The access control policies set up on the firewall will help separate the Web server from the trusted zone. For security purposes, queries from the Web application are sent to the database server through the specified port and the appropriate authentication account. In addition, the remote access service (RLS - *remote login service*) is also enabled on the servers to help administrators easily access to administer and configure services. Remote connections are made over the SSH protocol.

The experiment deployment scenario is described as follows: the critical assets to be protected are three physical servers containing the website components. Each server has detailed configurations about the operating system, the software and services set up on it. The Web server contains the front-end components that receive requests via port 80 or 443. The application server contains the back-end components written in PHP language. The data of the website is managed on a separate server with the SQL Server 2005 SP2. The database of assets to be protected, deployment scenarios and deployment procedures are built according to the data model described in part III.

B. Order of execution

The sequence of the 3-step security risk assessment mechanism for the deployment scenario described above is performed as follows:

1. Step 1 - Define protection requirements

Based on the CVE database and assets configuration, the table below describes the list of vulnerabilities on the servers along with the corresponding CVE-ID numbers. Each vulnerability is assigned an exploit probability value, which varies depending on the deployment scenario.

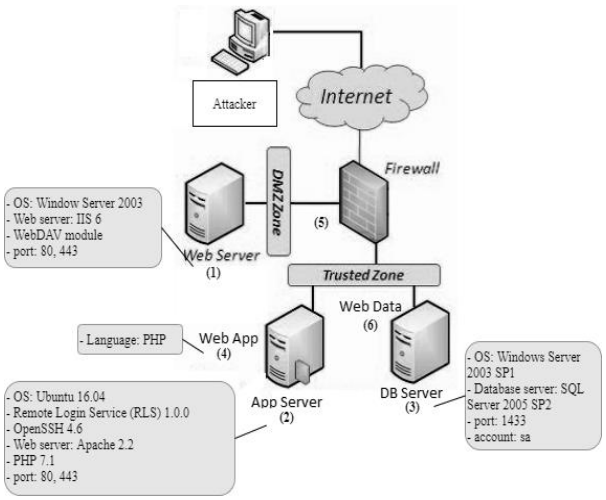


Fig. 8. Experiment deployment architecture.

TABLE 5. LIST OF ASSETS AND VULNERABILITIES

Asset	Vulnerability	CVE-ID	Exploit-ability
Web server	IIS 5.1 and 6.0 WebDAV Authentication Bypass Vulnerability	CVE-2009-1535	0.49
App server	Remote Login Vulnerability	CVE-2012-0959	0.39
	Improper cookies handler in OpenSSH	CVE 2007-4752	1
	Open SSL uses predictable random	CVE 2008-0166	1
DB server	SQL Server sp_replwritetovarbin Limited Memory Overwrite Vulnerability	CVE 2008-5416	0.8

2. Step 2 - Analyze vulnerabilities, threats, and countermeasures

The list of vulnerabilities listed in step 1 shows that the threat in this scenario comes from remote attackers, aiming to affect the operation of the website system. The threat of attack through asset vulnerabilities affecting web application's data access can be described as follows: older versions of the WebDAV module on IIS6 Web server enable attackers to upload to the server a WebShell. The RLS service vulnerability allowed the collection of valid user credentials on the application server. The OpenSSH version 4.6 software improperly handles cookie data that causes an attacker to gain privileges to execute a client deemed trusted. Then executing queries to the data server with SQL injection can call the *sp_replwritetovarbin* extended stored procedure with a set of invalid parameters that trigger a memory overwrite that causes a buffer overflow in SQL. Server. The result is a service interruption on the database server.

To mitigate this security risk, countermeasures can be incorporated in the deployment plan. These measures may cover one or more vulnerabilities; the cost of each measure is extracted from the historical data or established by the expert. Table 6 describes the results obtained in terms of seven envisaged countermeasures from C1 to C7.

TABLE 6. LIST OF COUNTERMEASURES FOR DEPLOYMENT SCENARIO

Countermeasure	Covered CVE-ID	Related assets	Cost
C1 – Apply security patches to OpenSSH	CVE 2007-4752	App server	63
C2 – Set firewall rules to filter outbound traffic	CVE-2009-1535 CVE-2012-0959	Web server, App server	70
C3 – Applying security patch MS09-004 on SQL Server blocks remote code execution	CVE 2008-5416	DB server	31
C4 – Disable WebDAV module on Web Server	CVE-2009-1535	Web server	250
C5 – Install a Web application firewall (WAF) to filter requests	CVE-2009-1535	Web server	205
C6 – Apply security patches to OpenSSL	CVE 2008-0166	App server	34
C7 – Limit data queries	CVE 2008-5416	DB server	84

3. Step 3 - Security risk assessment

The threat that exploits vulnerabilities on assets to be protected in this deployment scenario comes from remote attackers. An attack can be initiated from insecure services on the App server or Web server. From there, the attacker can obtain execution permission on the App server, then sends queries that interrupt the data service. Based on this analysis, the attack graph is built as shown in Fig. 9. In this case, the BAG consists of 4 nodes and 6 edges. Each node represents a state of the asset when the attacker performed the corresponding exploiting activity. The relationship between the nodes is determined according to the attack steps leading to the risk of service interruption on the database server.

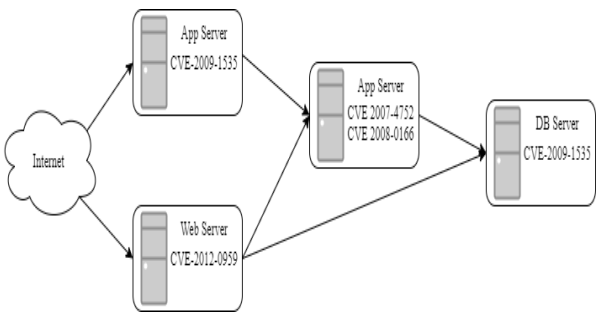


Fig. 9. Asset vulnerability attack graph.

The countermeasures obtained in step 2 should be evaluated for effectiveness by constructing a Bayesian decision network. An opportunity node (node in the BAG) is associated with multiple decision nodes, each representing the application of a countermeasure to prevent the exploiting activity of the correspondent asset. A utility node contains values showing the effectiveness while combining a set of countermeasures. The result networks for the experiment deployment scenario are built on the GeNIe Academic tool [20] in Fig. 10.

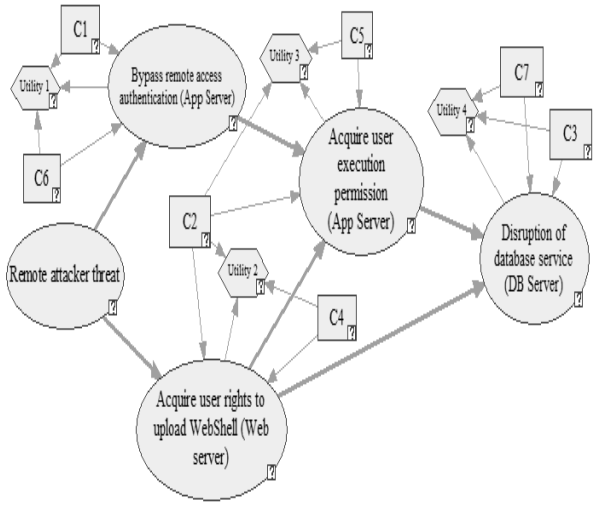


Fig. 10. Bayesian decision network for experiment deployment scenario.

From this result, to limit the exploitation of user rights on the Web server, measures C2 and C4 can be implemented. The manager can set a probability value for the state of the node representing the Web server's vulnerability and observe the change in the value of other nodes in the network. The resulting probability value represents the risk likelihood of service interruption on the database server. The effectiveness of the prevention measures is aggregated from the utility values at each node. The calculation results are shown in Table 7.

TABLE 7. EFFECTIVE VALUE OF COUNTERMEASURES

Countermeasures	Effective values
C1	865
C2	646
C3	263
C4	521
C5	285
C6	242
C7	65

Managers always need to consider budget constraints in the deployment plan. If the costs allocated for securing the network are limited, only the deployment scenario with the overall cost below this threshold is feasible. For example, with a total budget devoted to securing assets of 600, the highest effective value is obtained by 2317 units through the application of the group of measures {C1, C2, C4, C5} with a total cost of 588 not exceeding the limit. Another alternative with {C1, C2, C3, C6, C7} has a total cost of 282, but it is not optimal because the resulting efficiency values is only 2081.

Based on the results obtained, countermeasures can be considered in relation to the expected cost and effect as shown in Fig. 11. This chart shows that measures C1 and C2 should be selected because they are highly effective and low cost. The effects of C3 and C6 are quite similar. Measures C4 and C5 need more consideration because of high implementation cost, whereas C7 can be eliminated because of low efficiency.

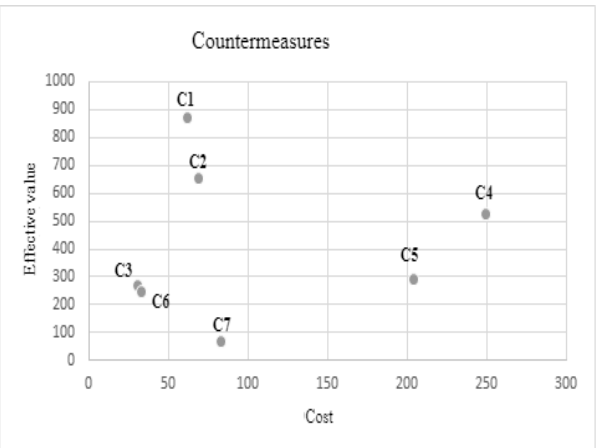


Fig. 11. Correlation of cost and effectiveness of countermeasures.

CONCLUSION

In this study, we have proposed and developed a quantitative mechanism for assessing security risks in the deployment scenario of website systems. Through the analysis of the stages in the deployment process, the assessment inputs are determined including security objectives, deployment scenarios with detailed configuration parameters for the assets to be protected, the relationship between assets, and deployment activities to realize the scenario. These inputs are combined with data from trusted sources such as CVE, CVSS to help us describe in detail the relationship between vulnerabilities, threats, and countermeasures and security risks of assets. This mechanism uses Bayesian networks with analysis of potential attack scenarios that provide managers with

quantitative assessments. To select suitable risk mitigation options, the attack graph is expanded to Bayesian decision networks with cost and effectiveness information of the selected control measures applied.

For further studies, we plan to extend the model with deployment scenarios in the DevOps development workflow with the help of tools to automate deployment activities. Another direction is to expand the risk assessment mechanism for the entire website system deployment process. Besides, the consideration of temporal and environmental indexes in CVSS for asset vulnerabilities is also a research direction that gives our research a comprehensive cyber security risk evaluation.

ACKNOWLEDGMENT

The authors would like to express their sincere thanks to the valuable comments of the reviewers, the suggestions, and the support from the VNIST JSC. and Hoang Trung Hieu – student at Hanoi University of Science and Technology.

REFERENCES

[1] Nurullah Demir, Tobias Urban, Kevin Wittek and Norbert Pohlmann, "Our (in)Secure Web: Understanding Update Behavior of websites and Its Impact on Security," PAM 2021: Passive and Active Measurement pp 76-92, 2021.

[2] Gary Stoneburner, Alice Goguen, and Alexis Feringa, “Risk Management Guide for Information Technology Systems,” NIST Special Publication 800-30, 2002.

[3] Ines Meriah, Latifa Ben Arfa Rabai, " A Survey of Quantitative Security Risk Analysis Models for Computer Systems," Association for Computing Machinery, ICAAI, October 2018.

[4] Huang Jiwen, Deng Zhilong, "A Bayesian Assessment Method of Network Risk," Applied Mechanics and Materials Vols 513-517 pp 1684-1687, Trans Tech Publications, Switzerland 2014.

[5] Sun Java™ System, "Sun Java Enterprise System Deployment Planning White Paper," Sun Microsystems, Inc., 2004.

[6] The MITRE Corporation. CVE® List https://cve.mitre.org/.

[7] The MITRE Corporation. Common Weakness Enumeration – https://cwe.mitre.org/.

[8] ISO/IEC 27005:2011 Information technology - Security techniques - Information security risk management.

[9] ISO/IEC 15408-1:2009 Information technology — Security techniques — Evaluation criteria for IT security.

[10] Serkan Ozkan. CVE details – the ultimate security vulnerability data source. https://www.cvedetails.com/.

[11] OWASP Foundation. OWASP Top Ten. https://owasp.org/www-project-top-ten/2017/.

[12] Armaghan Behnia, Rafhana Abd Rashid, and Junaid Ahsenali Chaudhry, "A Survey of Information Security Risk Analysis Methods," Smart Computing Review, vol. 2, no. 1, 2012.

[13] Forum of Incident Response and Security Teams, Inc. Common vulnerability scoring system. https://www.first.org/cvss/.

[14] The MITRE Corporation. Common weakness scoring system. https://cwe.mitre.org/.

[15] Yu Liu and Hong Man, "Network Vulnerability Assessment using Bayesian Networks," Proceedings of SPIE Vol. 5812, 2005.

[16] Kaixing Huang, Chunjie Zhou, Yu-Chu Tian, Weixun Tu, Yuan Peng, "Application of Bayesian Network to Data-Driven Cyber-Security Risk Assessment in SCADA Networks," 27th International Telecommunication Networks and Applications Conference (ITNAC), 2017.

[17] S. Zhang and S. Song. "A novel attack graph posterior inference model based on bayesian network," Journal of Information Security, 2011.

[18] Razieh Rezaee and Abbas Ghaemi Bafghi, "A Risk Estimation Framework for Security Threats in Computer Networks," Journal of Computing and Security, Volume 7, Number 1 (pp. 19-33), 2020.

[19] NIST. National vulnerability database. https://nvd.nist.gov/.

[20] Bayes Fusion LLC. GeNIe Modeler. https://www.bayesfusion.com/genie/.

[21] Mouna Jouinia, Latifa Ben Arfa Rabaia and Anis Ben Aissab, "Classification of security threats in information systems," 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014).

[22] Ines Meriah and Latifa Ben Arfa Rabai, "A Survey of Quantitative Security Risk Analysis Models for Computer Systems," ICAAI 2018.

[23] Irad Ben-Gal, "Bayesian Networks," Encyclopedia of Statistics in Quality and Reliability, John Wiley & Sons, Ltd. 2008.

[24] Hermawan Setiawan, Lytio Enggar Erlangga, Ido Baskoro, "Vulnerability Analysis Using The Interactive Application Security Testing (IAST) Approach For Government X Website Applications," 3rd International Conference on Infomation and Communications Technology (ICOIACT), 2020.

[25] Pavel B. Khorev, Maxim I. Zheltov, "Assessing Information Risks When Using Web Applications Using Fuzzy Logic," International Conference on Information Technologies in Engineering Education, 2020.

ABOUT THE AUTHORS

Vu Thi Huong Giang

Workplace: Hanoi University of Science and Technology.

Email: giangvth@soict.hust.edu.vn

Education: She received her Informatics Engineering degree in 2001, and her master's degree in information technology in 2003 at Hanoi University of Science and Technology. She received her PhD in Systems and Software at Grenoble INP, France in 2008.

Current research direction: security, advanced software engineering methods and applications.

Nguyen Manh Tuan

Workplace: Hanoi University of Science and Technology.

Email: tuannm@soict.hust.edu.vn

Education: He received his bachelor of engineering degree in 2007 and his master's degree in information technology in 2016.

Current research direction: security and web services development.