Compositional Approach to Quantify the Vulnerability of Computer Systems

Hossein Homaei, Hamid Reza Shahriari

Department of Computer Engineering and Information Technology,
Amirkabir University of Technology

Abstract

Although analyzing the complex systems could be a complicated process, current approaches to quantify system security or vulnerabilty usually consider the whole system as a single component. In this paper, we propose a new compositional method to evaluate the vulnerability measure of complex systems. Composablity means that we can compute the vulnerability measure of a complex system using precaclulated vulnerability measures of its components. It would be useful to analyze complex systems which may have very complicated models, specially when the off-the-shelf components are used and detailed information are not available. This approach reduces the state space and complexity of computation. On the other hand, if a component is replaced by another one, the vulnerability measures for other components do not change and are reused in new computation. Thus, the calculation of vulnerability measure for new system is well reduced. Our method uses a state machine to model the system. The model considers unauthorized states and attacker capabilities. In this model both probability of attack and delay time to reach the target state are considered.

Keywords: compositional evaluation, security quantification, vulnerability analysis

Email addresses: homayi@aut.ac.ir (Hossein Homaei), shahriari@aut.ac.ir (Hamid Reza Shahriari)

1. Introduction

To compare the security of two same systems, or assess the security improvement of a system, we need security evaluation methods. If we have an accurate estimation of the system security, we can evaluate the effectiveness of security countermeasures on the system and compare system's security.

Although some methods have been proposed to quantify vulnerability, these methods have some drawbacks. For example, some methods [8, 10] can just represent the abstract model of the system; thus, it seems difficult to assign specific quantities to the model parameters like transition probabilities [12, 19]. Moreover, some methods such as [8, 23] just model one kind of attack or attacker, and can not model all kinds of attacks such as internal attacks. Another drawback to some existing models for example [24] is that they are based on some assumptions such as exponential distribution for transition probability which has not been proved yet [7].

It is worthy to note that the vulnerability quantity of a system is not an absolute value but rather depends on the threatening adversary. For example, a system might be secure against an amateur attacker, but vulnerable against a skilled one.

On the other hand, it seems that the vulnerability measure depends on attack duration. The more attack duration takes, the more system is secure. Accordingly, any proposed vulnerability measurement approach should consider attack duration.

A computer system might be violated in a specific state, which is not an unauthorized state in other system with different security policy. Thus, considering the security policies is valuable in security quantification.

Another important issue is that calculating the security quantity for large and complicated systems is very difficult. For example, if we use state machine to model a complex system, it will be very large and complicated especially if we want to model low level system details.

In this paper we propose a general model to quantify the vulnerability of computer system. We believe that the model could surmount above mentioned drawbacks. The major contributions of the model are providing a composable apprach to anlayze vulnerabiltiy of the systems and considering attacker capabilities.

The method can compute the vulnerability quantity of systems using the vulnerability measures of constitutive components. In other words, to quantify the vulnerability of a large system, we can compute the vulnerability measure of its components and then compose these quantities to measure the vulnerability of the whole system.

On the other hand, it can model computer systems against diverse attacks and attackers considering pertinent unauthorized states and appropriate attacker's capabilities. Moreover, The model uses the delay time parameter in the evaluation process and thus it can discriminate between two systems have the same probability to reach unauthorized states. Also, there is no assumption for transition probability such as exponential distribution.

The rest of this paper is organized as follows: section two surveys related work on vulnerability quantification. Section three describes a formal model of computer systems. The vulnerability measurement process is presented in section four. The fifth section explains the compositional approach to quantify security based on the model which is represented in section three. Section six specifies a practical case study to clarify the vulnerability measurement process. Finally, section seven concludes this paper and discusses our future work.

2. Related work

Littlewood et al. [9] have discussed similarities between reliability and security. They have suggested using reliability and dependability measurement methods [17, 2] to predict the amount of security. The major intention of [9] is to invite discussion about the feasibility of this approach. Thus, some open questions have been identified need to be answered.

Jonsson and Olovsson [6] believe that the attacking process can be split into three phases. They have indicated that the times between breaches in the standard attack phase are exponentially distributed. Thus, they have suggested using traditional reliability methods to model security. The results of experiment performed in [6] have been achieved in a special environment and could not be extended to all environments.

Kaâniche et al. [7] have represented empirical analyses based on the data collected from honeypot platforms deployed on the Internet. They found out that mixture of a Pareto and an exponential distribution is the best fit for the observed times between attacks. Thus, the traditional assumption considered in hardware reliability evaluation does not seem to be satisfactory for security evaluation.

Madan et al. [10] have considered security as a Quality of Service attribute. They have used stochastic modeling techniques to quantify security attributes of intrusion-tolerant software systems. They use Mean time (or effort) to security failure metric to quantify security. The method could be used as a framework and needs more development.

Sallhammar et al. [19] have proposed a new approach to integrated security and dependability evaluation, which is based on stochastic game. The state machine used in this method is unable to model system's details. Thus, quantifying "accumulated failure intensity" would be difficult.

Manadhata and Wing [11] have proposed a metric to determine whether one version of a system is relatively more secure than another with respect to the system's attack surface. This method could not compare different systems which have different resources.

Singh et al. [21] have divided the security quantification methods into penetration test and analysis of formal models of system. They have proposed a new approach to quantify security by performing repeated penetration testing of detailed system models. This approach will utilize the accuracy of formal modeling and fastness of penetration test. But the security metric, number of attack paths, is not accurate; Because the time and probability of attack paths are not considered.

Pamula et al. [15] have presented a quantitative metric to assess the amount of network security based on attack graphs analysis. The metric measures the security strength of a network in terms of the strength of the weakest adversary who can successfully penetrate the network. The capabilities of the weakest adversaries are not always comparable for all systems.

Frigault et al. [3, 4] have used attack graphs as a special Bayesian Network to measure the security of a system. However, they have not proposed a new metric to quantify the system security; but rather, they have used the Common Vulnerability Scoring System (CVSS) [14] to measure the exploit probability of a vulnerability based on dependency relations between vulnerabilities.

Mehta et al. [13] have represented two algorithms to rank states of an attack graph. The first algorithm is similar to the PageRank algorithm used by Google. The second algorithm ranks individual states based on the reachability probability of an attacker in a random simulation. They have used the total rank of all error states as a system security measurement. But the method does not consider overlaps between paths.

Pham et al. [16] have constructed a graph based on open ports of network hosts and also the access rights between hosts. They have used a metric called attackability metric to compute the assurance level of the system. The metric uses just the number of access points on each host to measure the assurance level. Thus, it could not be used as a security or vulnerability metric.

Other methods which use attack graph analysis for security quantification could be divided into two categories: reliability methods and minimum cost hardening methods. Reliability methods like the method proposed in [5] use the probability of a successful attack as a security metric. On the other hand, minimum cost hardening methods like [18, 22] use the cost of obviation of attack paths as a security metric. The reliability methods do not consider the cost of attack path. The main challenge of minimum cost hardening methods, On the other hand, is that they use just the cost of vulnerability obviation to quantify security; nevertheless, we could increase the security of system by other methods like changing the configuration.

Wang et al. [23] have proposed a two-steps framework to measure various aspects of network security. In the first step, the individual security components are evaluated. The second stage focuses on the composition of individual measures.

Shahriari [20] has proposed a new method for vulnerability analysis using state machine. In this method, the vulnerability of a state, S, is defined as the probability of reaching an unauthorized state, starting from S. The system vulnerability is defined as the vulnerability of the initial state. Thus, the vulnerability of any system which has not any absorbing state will be equal to one. Using this method, there could be two systems with the same quantity of vulnerability but it takes different time to reach an unauthorized state. Thus, the vulnerabilities of these systems are not equal practically.

3. Modeling

3.1. System model

In this subsection we model computer systems. The model consists of a set of states connected to each other by some transitions. Each transition is related to an event. Events might be triggered by the system, another system or person and cause system to travel between states. Each transition occurs with specific probability and a delay function specifies its completion time. An intruder could trigger some events in the system and conduct system to an unauthorized state where at least one of the security policies is violated. The following definition specifies a formal definition of the system.

Definition 1. A computer system say CS, is a 7-tuple $CS = (Q, q^i, E, \Delta, \rho, \delta, Q_{UA})$ where, Q is a finite set of all states, $q^i \in Q$ is the initial state, E is the finite set of all possible events, $\Delta \subseteq Q \times E \times Q$ is the transition relation, $\rho : \Delta \rightarrow (0, 1]$ is the transition probability function, $\delta : E \rightarrow [0, \infty)$ is the delay function, and $Q_{UA} \subseteq Q$ is the set of all unauthorized states.

Each part of the model is described as follows:

- Q is a finite set of all states in the system. Each state is specified by a collection of system variables which can be changed when a system action occurred. These variables are called state variables. For example, user privilege is a state variable. Q should represent all possible states in the system. State variables could be chosen by evaluator according to the modeling purpose.

- $q^i \in Q$ is the initial state in which the system starts working. If there is more than one initial state, we can use a virtual state as the initial state and connect it to the real initial states by virtual transitions.

- E is the set of events. The set is partitioned into three finite subsets E^{in}, $E^{int}$and E^{out} which show input, internal and output events respectively.

  - An input event occurs in the environment and effects on the system. The system has no control on the generation of input events. The probability of occurring and the delay time of input event can be changed in different environments. But all possible transitions are known during system modeling. In other words, we model just deterministic systems.

    Input events are partitioned into two sets $E^{in_{nor}}$ and $E^{in_{att}}$ represent normal and attacker input events respectively. An attacker input event occurs when an attacker enforce system to do an action which the system does not expect to receive it normally. Other input events are called normal input events.

  - Internal event is produced by the system and has no effect on the environment. In other words, internal events are hidden from external view.

  - Output event is generated by the system and has an external effect on the environment. System services are represented by output events.

The union of internal and output events are called local events and denoted by E^{loc}; because they are produced by the system locally.

- $\Delta \subseteq Q \times E \times Q$ is the transition relation. The relation is not input-enabled because in practice not all states can receive all input events. In contrast, some specific input events can occur in some states. However, we can provide input-enabled property considering that esch input event which has no effect on the system will not change the state.

- $\rho : \Delta \longrightarrow (0, 1]$ is the transition probability function which satisfies following properties:

  1. $\rho(q, e, r) > 0$ iff $(q, e, r) \in \Delta$
  2. $\forall q, r \in Q \; \forall e \in E^{in} \; [(q, e, r) \in \Delta \rightarrow \rho(q, e, r) = 1]$
  3. $\forall q \in Q \sum\limits_{r \in Q} \sum\limits_{e \in E^{loc}} \rho(q, e, r) = 1$

  In contrast to the Markov model, it is not necessary to have the exit probability of one for all transitions. Because, the system has control just on the local events and we can not predict the probabilities of input events. In other words, the transition probability function does not represent the probability of occurring input events; in contrast it shows the transition probability. As we mentioned before, the model is deterministic. Thus, the transition probability of any input event is equal to one.

- $\delta : E \longrightarrow [0, \infty)$ is the delay function. It takes $\delta(e)$ for event e to complete. $\delta(e)$ is known for local events; but to specify the delay time for input events, we should consider the kind of attacker and the environment. If the delay time of an input event is not known in the modeling step, we assign zero to $\delta(e)$.

- $Q_{UA} \subseteq Q$ is the set of all unauthorized states. An unauthorized state is the state in which at least one of the security policies is violated. For example, if an attacker gains root access on the database server, the security policy is violated and thus the system will reach an unauthorized state.

The model introduced in definition 1 is a formal model that specifies security policies in addition to the system properties. The final goal of this kind of modeling is not just to determine a specific system but it provides a general framework to model various kinds of systems. For example, if we consider just attacker input events, the model is well reduced to the attack graph or scenario graph.

A path from q_1 to q_2 is a sequence of transitions from state q_1 to state q_2 and denoted by $path_{q_1q_2}$. If $path_{q_1q_2}$ contains some transitions of normal input events, E^{in}, we call it conditional path and denote it by $(path_{q_1q_2}|E^{in})$.

There could be no path or either many paths between two states in the model. It is obvious that if there is a path from initial state to an unauthorized state, the system have a vulnerability which can be exploited by attackers. But notice that different attackers can reach the unauthorized state by different efforts. Thus, we need to model the attacker and then measure the vulnerability of system against particular attacker. In the following subsection we will define the attacker model.

3.2. Attacker model

A computer system could be secure against an attacker but vulnerable against another one; because different kinds of attackers have various capabilities and skills and they can do diverse actions. Moreover, an intruder with more capabilities is able to accomplish an attack faster. For example, Blaze et al. [1] have compared the needed time to find a 56-bits key of cryptographic algorithm by different types of attackers. A single hacker needs more time to attack a system than an intelligence agency needs. Thus, we need an attacker model to evaluate the security of a computer system. If we can not identify an especial intruder, the strongest attacker is considered.

Definition 2. An intruder say I, is a triple $I = (E, SL, ED)$ where E is a set of all possible attacker events, $SL : E \rightarrow (0, 1]$ is the skill level function, and $ED : E \rightarrow (0, \infty]$ is the event delay function.

More skill level of the intruder means more probability of having exploit codes and successful attacks. Thus, the attacker skill level represents the probability of executing attacker actions. On the other hand, the required time to accomplish an attack step is represented by event delay function.

3.3. The system under attack model

In this subsection we specify the composition of attacker and system models. The combination of these two models is called the under-attack system and defined as follows.

Definition 3. The composition of computer system $CS = (Q_{cs}, q^i_{cs}, E_{cs}, \Delta_{cs}, \rho_{cs}, \delta_{cs}, Q_{UA_{cs}})$ and intruder $I = (E_{att}, SL, ED)$ is called the under-attack system and denoted by $\mathcal{CS} \leftarrowtail I$. The under-attack system is a system like $\mathcal{CS} \leftarrowtail I = (Q, q^i, E, \Delta, \rho, \delta, Q_{UA})$ where

- $Q = Q_{cs}$;
- $q^i = q^i_{cs}$;
- $E = E_{cs}$;
- $\Delta = \Delta_{cs}$;
- $\forall q, r \in Q \forall e \in E_{att}[\rho(q, e, r) = SL(e)]$,
  $\forall q, r \in Q \forall e \in E \backslash E_{att}[\rho(q, e, r) = \rho_{cs}(q, e, r)]$;
- $\forall e \in E_{att}[\delta(e) = ED(e)]$, $\forall e \in E \backslash E_{att}[\delta(e) = \delta_{cs}(e)]$; and
- $Q_{UA} = Q_{UA_{cs}}$.

An under-attack system represents real quantities for probabilities and delays of local and attacker events. But, we do not have any information about delays and probabilities of normal input events. Thus, we should use default quantities, one for probability and zero for delay time, for normal input events.

4. Vulnerability measurement

After constructing the under-attack system, we propose a metric to quantify the vulnerability. Before that, we define some other required parameters.

Definition 4. The cost of transition (q, e, r) is denoted by $cost(q, e, r)$ and defined as event delay per transition probability ratio. In other words, $cost(q, e, r) = \frac{\delta(e)}{\rho(q,e,r)}$

It means that the cost of each step of attack is related to the delay time and the reverse of transition probability. For example, if the delay time of event e is 1 hour and the probability of transition (q, e, r) is 0.5, the cost of transition (q, e, r) will be equal to 2. The probability of transition is 0.5; Thus, in every two attempts to do the transition (q, e, r), one of them is successful, statistically. In other words, the delay time of transition (q, e, r) approximates to two instead of one hour.

When we talk about the cost of normal input transitions, it suffices to know the delay time and probability of event occurrence. Because, the transition probability is equal to one and we can specify the cost of these transitions using the probability and delay of events. The cost of normal input event, e, is denoted by $DPR(e)$ and defined by $DPR(e) = \frac{\delta(e)}{P(e)}$, where $P(e)$ is the probability of occurring event e.

The cost of path is defined as follows.

Definition 5. the cost of conditional path $(path_{q_1q_2}|E^{in_{nor}}_{path_{q_1q_2}})$ in an under-attack system $\mathcal{CS} \leftarrowtail I$ is denoted by $cost(path_{q_1q_2}|E^{in_{nor}}_{path_{q_1q_2}})$ and calculated by the following equation:

$$cost(path_{q_1q_2}|E^{in_{nor}}_{path_{q_1q_2}}) = \sum_{\forall(q,e,r)\in path_{q_1q_2}} cost(q,e,r) \qquad (1)$$

If the parameters of normal input events $E^{in_{nor}}_{path_{q_1q_2}} = \{e^{in}_1, e^{in}_2, ..., e^{in}_k\}$ are available, the cost of $Path_{q_1q_2}$ is calculated as follows:

$$cost(path_{q_1q_2}) = cost(path_{q_1q_2}|E^{in_{nor}}_{path_{q_1q_2}}) + \sum_{i=1}^{k} DPR(e^{in}_i) \qquad (2)$$

Thus, the shortest path from initial state to an unauthorized state in the system represents the most interesting path from an intruder point of view. Because, the attacker wants to reach unauthorized states by the minimum effort.

The shortest conditional path from initial state, q^i, to the state q under the condition of occurring the set of normal input events $E^{in_{nor}}_{pre}$, denoted by $(path^{Min}_{q^iq}|E^{in_{nor}}_{pre})$, is the path with minimum cost within all possible conditional paths like $(path_{q^iq}|E^{in_{nor}}_{pre})$. In other words

$$cost(path^{Min}_{q^iq}|E^{in_{nor}}_{pre}) = \underset{\forall(path_{q^iq}|E^{in_{nor}}_{pre})\in Paths_{q^iq}}{Min} \{cost(path_{q^iq}|E^{in_{nor}}_{pre})\} \qquad (3)$$

Using this concept, we can define the vulnerability measure of an unauthorized state as follows:

Definition 6. The conditional vulnerability measure of an unauthorized state $q \in Q_{UA}$ under the condition of occurrence of the set of normal input events $E_{pre}^{in_{nor}}$ in the under-attack system $\mathcal{CS} \leftarrowtail I$ is denoted by $\nu(q|E_{pre}^{in_{nor}})$ and defined as reverse of the shortest path cost. In other words,

$$\nu(q|E_{pre}^{in_{nor}}) = 1/cost(path_{q^i q}^{Min}|E_{pre}^{in_{nor}}) \tag{4}$$

If $E_{pre}^{in_{nor}}$ is null, $\nu(q|\emptyset)$ is called unconditional vulnerability measure of state q.

The less it costs to reach an unauthorized state, the more vulnerable the state is. Thus, the definition matches the concept of vulnerability practically.

Any possible path could be chosen by intruders to compromise the system. We could not predict which path will be chosen by the attacker. But we know that the attacker tries to choose the convenient path, the path with lower cost. Thus, to measure the vulnerability of system we use the shortest path. In other words, we find an upper bound for the quantity of vulnerability. The interest in using the shortest path arises of the attack concept. The hard intruder of today becomes the most probable intruder of tomorrow. Thus, the system will be exploited by stronger attacker in the future [12].

After computing the quantity of vulnerability of each state, we can evaluate the vulnerability of the whole system. As we want to find an upper bound for vulnerability measure, we use the following definition to assess the vulnerability quantity.

Definition 7. conditional vulnerability measure of an under-attack system $CS \leftarrowtail I$ under the condition of occurrence of a set of normal input events $E_{pre}^{in_{nor}}$ is denoted by $\nu(CS \leftarrowtail I|E_{pre}^{in_{nor}})$ and calculated by the following equation:

$$\nu(CS \leftarrowtail I|E_{pre}^{in_{nor}}) = \underset{\forall q \in Q_{UA}}{MAX}\{\nu(q|E_{pre}^{in_{nor}})\} \tag{5}$$

To determine unconditional vulnerability measure, we should have real parameters of normal input events. Thus, the unconditional vulnerability could be measured by the following equation:

$$\nu(CS \leftarrowtail I) = \underset{1 \leqslant i \leqslant n}{MAX}\{\frac{1}{\frac{1}{\nu(CS \leftarrowtail I \left| E_{pre_i}^{in_{nor}} \right.)} + \sum\limits_{\forall e \in E_{pre_i}^{in_{nor}}} DPR(e)}\} \tag{6}$$

Where n is the number of conditional vulnerabilities and $\nu(CS \leftarrowtail I \left| E_{pre_i}^{in_{nor}} \right.)$ is the ith conditional vulnerability measure for the system.

5. Compositional evaluation method

In previous section we proposed a model of computer systems and a method to quantify system vulnerability. But it seems difficult to measure the vulnerability of complex systems. In this section we use the same model introduced in definition 1. But we apply some compositional methods to measure the vulnerability of complex systems.

5.1. Compatible systems

For compositional evaluation of vulnerability, we should illustrate which systems can combine. Thus, we define compatible unauthorized states and also compatible systems.

Definition 8. Assume that two systems CS_1 and CS_2 are modeled as $CS_1 = (Q_1, q_1^i, E_1, \Delta_1, \rho_1, \delta_1, Q_{UA_1})$ and $CS_2 = (Q_2, q_2^i, E_2, \Delta_2, \rho_2, \delta_2, Q_{UA_2})$. Each unauthorized state $q \in Q_{UA_1}$ is compatible with system CS_2, if and only if $(q \notin Q_2)$ or $(q \in Q_2 \rightarrow q \in Q_{UA_2})$.

It means that the security policies should be compatible. In other words, an unauthorized state should be an unauthorized state in both systems.

Definition 9. A finite set of under-attack systems $\{(CS_i \leftarrowtail I) : i \in \mathrm{I}\}$ is compatible if and only if each unauthorized state in each system is compatible with other systems and for all $i, j \in \mathrm{I}$ $(i \neq j)$ following equations are satisfied:

1. $E_i^{out} \cap E_j^{out} = \emptyset$
2. $E_i^{int} \cap E_j = \emptyset$

It is obvious that the compatibility is defined just for systems with the same attacker model; because compositional analysis is possible if all components are analyzed against the same intruder.

On the other hand, compatible systems do not have same outputs; because each system has its own tasks and services. Moreover, each system should have its own internal events; because an external observer can not view these events.

5.2. Compositional operators

In this subsection we define some operators to combine components and construct composite systems. We can determine four operators as follows:

1. Just one of the components is chosen and executed.

2. Components execute sequentially.
3. Each component executes independently. In other words, executing one component does not depend on the other components.
4. Components execute simultaneously until they need to execute a common event. One component might stop working until the common event happens. This kind of composition is called parallel synchronous composition. Based on definition 9, the common event might be an output of one component and input event of other one or it could be a common input event for both components. Thus, we have two kinds of synchronous composition: output-input synchronization and input-input synchronization.
   Synchronization on the input event has no practical meaning; because if an input event is available, it can effect on the system. In other words, the system does not stop working until another component proceed to the point in which it can receive the common input. Thus, just output-input synchronization is considered in this article.

Notice that in the parallel composition, all components can execute but when we use the choice operator, just one of the components is selected and executed.

5.3. Service parameters

Each output event of the system is a service. Before talking about the compositional methods, we should define some other parameters about the system services.

For any output event like $e \in E^{out}$, the set of all states where event e is occurred is called exit states of e and denoted by Q_e. To give a service like e, the system should go to one of the exit states $q \in Q_e$ and then execute the output event. There could be several paths to reach state q. Thus, we should determine which path is chosen by the system. To have a proportional definition to the vulnerability measure, we use the concept of shortest path. Service path of e denoted by $path_s^e$ is the shortest path from initial state to any state q^e, where q^e is the next state of executing e in the state $q \in Q_e$.

Thus, the service time is defined as follows.

Definition 10. Service time for an output event $e \in E^{out}$ denoted by ST^e is defined as the cost of service path $path_s^e$.

Another parameter that we need in compositional analysis is the completion time.

Definition 11. Completion time of a system denoted by ST is the cost of shortest path from initial state to the final state. In other words, $ST = cost(path^{Min}_{q^iq^f})$.

If there are more than one final state in the system, the shortest path from initial state to any final state is calculated and the path which has the lowest cost is used to calculate the completion time.

5.4. Proposed compositional method

In this subsection we give some theorem to measure the vulnerability of complex systems. In each theorem, one of the compositional operators is studied.

Theorem 1. *The vulnerability measure of the system $C = C_1 + C_2 + C_3 + ... + C_n$ against an intruder I is computed by the following equation:*

$$\nu(C \leftarrowtail I) = \underset{1 \leqslant i \leqslant n}{MAX}\{\nu(C_i \leftarrowtail I)\} \tag{7}$$

Proof. If components combined by choice operator, the composed system might behave as each of constitutive components. In the worst case, the component with the highest vulnerability is chosen. Thus, the vulnerability of the system will be equal to the vulnerability of that component.

Theorem 2. *The vulnerability measure of composite system $C = C_1.C_2.C_3. ....C_n$ against an intruder I is calculated by the following equation:*

$$\nu(C \leftarrowtail I) = MAX\{\nu(C_1 \leftarrowtail I), \underset{2 \leqslant k \leqslant n}{MAX}\{\frac{1}{\sum\limits_{i=1}^{k-1} ST_{C_i} + \frac{1}{\nu(C_k \leftarrowtail I)}}\}\} \tag{8}$$

Proof. All possible unauthorized states in the composite system should be an unauthorized state in one of the constitutive components. Assume that there are two components C_1 and C_2 which combine and construct $C = C_1.C_2$. Thus, if an unauthorized state of component C_1 causes the unauthorized state of C, the vulnerability of composite system will be equal to the vulnerability of component C_1; because the shortest path from initial state to the unauthorized state is start from initial state of C_1 and reach the unauthorized state of the same component.

But the shortest path to reach the unauthorized state caused by an unauthorized state in C_2 is the path which starts from initial state in C_1, reaches the final state of C_1 by the shrtest path, and is continued to the unauthorized state in C_2 through the shortest vulnerable path. Thus, the cost of this path is the sum of completion time of C_1 and the cost of vulnerable path of C_2.

If there is a shorter path than this path, it should be a shorter path for completion of C_1 or a shorter path to reach the unauthorized state in C_2 which are in direct contradiction to the definitions of completion time and vulnerability measure.

We can extent the proof to more components and achieve equation 8 using inductive reasoning. Assume that equation 8 is true for n component. We want to show that this equation is true for $n+1$ components. If there are $n+1$ components in the system, the vulnerability of the system $C = C_1.C_2.C_3..C_n.C_{n+1}$ will be equal to the maximum of inverse cost of the shortest path to reach the unauthorized state in C_1, C_2, , C_n or C_{n+1}. This quantity for components C_1, C_2, , and C_n is calculated by equation 8 using the induction assumption. If an unauthorized state occurs in component C_{n+1}, the related vulnerability of the system will be computed as follows:

$$\nu(C^{n+1} \leftarrowtail I) = \frac{1}{\sum\limits_{i=1}^{n} ST_{C_i} + \frac{1}{\nu(C_{n+1} \leftarrowtail I)}},$$

where $\nu(C^{n+1} \leftarrowtail I)$ means the vulnerability of system C if the unauthorized state occurs in component C_{n+1}. Because to reach the unauthorized state in C_{n+1}, all components from C_1 to C_n should complete their work and then the vulnerable path of component $n+1$ should be traversed.

Thus, the vulnerability of system C is calculated as follows:

$$\nu(C \leftarrowtail I) = MAX\{\nu(C_1 \leftarrowtail I), \underset{2\leqslant k\leqslant n}{MAX}\{\frac{1}{\sum\limits_{i=1}^{k-1} ST_{C_i} + \frac{1}{\nu(C_k \leftarrowtail I)}}\},$$

$$\frac{1}{\sum\limits_{i=1}^{n} ST_{C_i} + \frac{1}{\nu(C_{n+1} \leftarrowtail I)}}\} = MAX\{\nu(C_1 \leftarrowtail I), \underset{2\leqslant k\leqslant n+1}{MAX}\{\frac{1}{\sum\limits_{i=1}^{k-1} ST_{C_i} + \frac{1}{\nu(C_k \leftarrowtail I)}}\}\}$$

In other words, equation 8 is true for $n+1$ components as it is true for n components.

Example: Assume that two systems C1 and C2 are modeled as shown in figure 1. The unauthorized states are depicted by gray color. The label of each transition specifies the transition cost. State S3 is the final state of components C_1.

Sequential composition of these two components is shown in figure 2. There are two unauthorized paths in the composite system. The cost of path which reaches the unauthorized state (S4, S1') is X+Z and the cost of second path which reaches state (S3, S3') is X+Y+S. On the other hand, the cost of

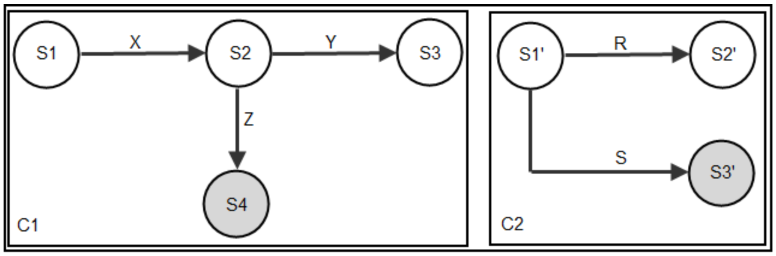

Figure 1: Two sample systems used for sequential composition

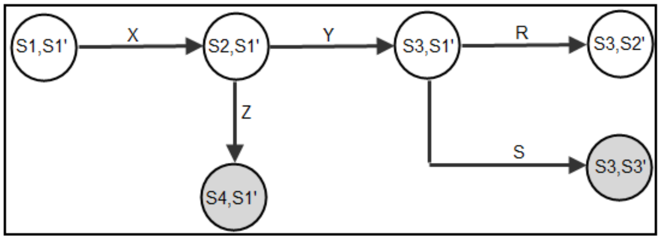

Figure 2: Sequential composition

reaching state S4 in component C1 and reaching state S3' in component C2 are equal to $X + Z$ and S respectively. The completion time of component C1 is equal to $X + Y$. Thus, the vulnerability measure of each unauthorized state in the composite system is calculated as follows.

$\nu(S4, S1') = \frac{1}{X+Z} = \nu(C_1 \leftarrowtail I)$

$\nu(S3, S3') = \frac{1}{X+Y+S} = \frac{1}{ST_{C_1} + \frac{1}{\nu(C_2 \leftarrowtail I)}}$

Thus, the vulnerability measure of the composite system will be equal to the maximum of these two measures. This example clarifies the use of equation 8.

Theorem 3. *The vulnerability measure of composite system $C = C_1||C_2||C_3 ||...||C_n$ against an intruder I is calculated by the following equation:*

$$\nu(C \leftarrowtail I) = \underset{1 \leqslant i \leqslant n}{MAX}\{\nu(C_i \leftarrowtail I)\} \tag{9}$$

Proof. We can use the same reasoning that is used to proof theorem 1 because in the parallel composition, the component which reaches its unauthorized state earlier, with less cost, can determine the vulnerability of the composite system.

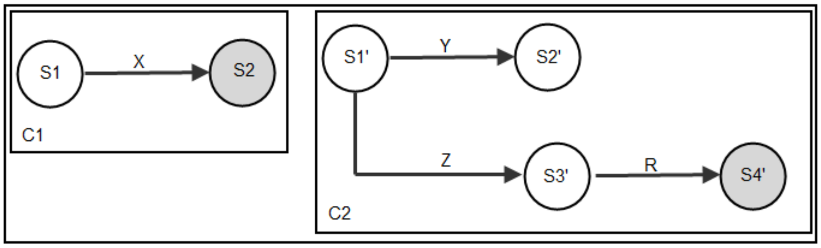

Figure 3: Two sample systems used for parallel composition

The only difference between the choice and parallel composition operators is that after occurrence of an unauthorized state in an arbitrary component, other unauthorized states of other components can be reached.

But accessing these unauthorized states just possible by going through more edges in the state machine. Thus, the cost of reaching these states will be more than the cost of reaching just one unauthorized state of one component. Therefore, these paths will not effect on the vulnerability measure of the composite system.

Example: Figure 3 depicts two systems C1 and C2. Each system has one unauthorized state. The cost of reaching state S2 and S4' are X and Z+R respectively. The vulnerability measures of the components are equal to the reverse of these quantities.

Figure 4 shows the parallel composition of these components. It is obvious that each unauthorized state is an unauthorized state in the composite system. In other words, each state in which one of the states S2 or S4' is occurred will be an unauthorized state. Thus, the vulnerability measure of each unauthorized state in the composite system is calculated as follows.

$\nu(S2, S1') = \frac{1}{X} = \nu(C1 \leftarrowtail I)$
$\nu(S2, S2') = \frac{1}{X+Y}$
$\nu(S2, S3') = \frac{1}{X+Z}$
$\nu(S2, S4') = \frac{1}{X+Z+R}$
$\nu(S1, S4') = \frac{1}{Z+R} = \nu(C2 \leftarrowtail I)$

But, $X + Y$, $X + Z$ and $X + Z + R$ are greater than X. Thus, the vulnerability of related states are less than the vulnerability of state (S2, S1') and they have no effect on the vulnerability of the composite system. In other words, the maximum quantity of $\nu(C_1 \leftarrowtail I)$ and $\nu(C_2 \leftarrowtail I)$ will be equal to $\nu(C \leftarrowtail I)$.

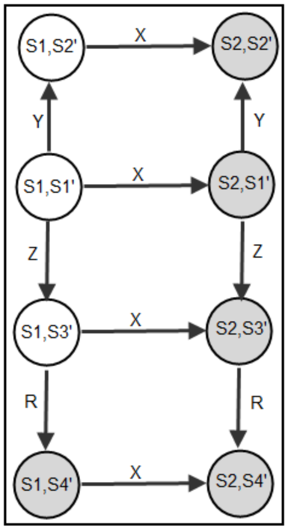

Figure 4: Parallel composition

To evaluate the vulnerability measure of a synchronous parallel system, we should determine the cost of each vulnerable path before and after the input event occurrence in the second component. The following definition satisfies this requirement.

Definition 12. The cost of vulnerable conditional path $(path_{q^iq}|\{e\})$ under the condition of occurring input normal event e in the under-attack system $CS \leftarrowtail I$ is denoted by $cost(path_{q^iq}|\{e\})$ and defined as $cost(path_{q^iq}|\{e\}) =< cost_{q^e}, cost_q >=< cost(path_{q^iq^e}), cost(path_{q^eq}) >$; where q^e is the state in which the input event e is occurred.

It is obvious that there could be more than one path has the form $(Path_{q^iq}|\{e\})$ in the system; because the states between the initial state and final state could be different.

Theorem 4. *The vulnerability measure of a composite system* $C = C_1 \underset{e}{\Updownarrow} C_2$ *against an intruder* I *is calculated by the following equation.*

$$\nu(C \leftarrowtail I) = MAX\{\nu(C_1 \leftarrowtail I), \nu(C_2 \leftarrowtail I), \frac{1}{\underset{\forall(path_{q^iq}|\{e\})\in paths^{C_2}_{q^iq}}{MIN\{}\quad MAX\{ST^e_{C_1}, cost^{C_2}_{q^e}\} + cost^{C_2}_q\}}\} \tag{10}$$

Where $cost_{q^e}^{C_2}$ and $cost_{q}^{C_2}$ are the first and second part of $cost(path_{q^iq}|\{e\})$ for component C_2 respectively.

Proof. Any unauthorized state in C can be an unauthorized state in C_1 or an unauthorized state in C_2. The vulnerability measure of the first case is equal to $\nu(C_1 \leftarrowtail I)$. The unauthorized state in C_2 might happen without occurring the common event e, or it might happen after occurring the common event.

If the unauthorized state happens without occurring the common event, the vulnerability of system will be equal to $\nu(C_2 \leftarrowtail I)$. But in the second case we should calculate the cost of shortest path from initial state in C_1 to the unauthorized state in C_2.

The cost of path from initial state in C_2 to the state in which the common event is occurred is equal to the maximum of service time of e in C_1 and the cost of reaching the state in which input event occurs in component C_2; because one component can operate simultaneously with the other ones. Thus, the components can execute events independently until the second component arrived to the common event. In this situation, the second component might wait for e. Thus, if $ST_{C_1}^{e}$ is greater than $cost_{q^e}^{C_2}$, the system should wait until e happens in C_1.

After occurring e, component C_2 can continue its work and reach the unauthorized state. Thus, in this case, the cost of vulnerable path of the system will be equal to the sum of $cost_{q}^{C_2}$ and the maximum of $cost_{q^e}^{C_2}$ and $ST_{C_1}^{e}$.

But we should repeat this process for all possible vulnerable conditional paths. Thus, the vulnerability measure of the system is achieved by equation 10.

Example: Figure 5 shows an example of a parallel synchronous composite system. The numbers depict transition costs. Thus, the service time of common event e is equal to 4. There are two vulnerable paths to reach the unauthorized state 5 in component C2. The cost of vulnerable paths {1, 2, 3, 4, 5} and {1, 6, 7, 8, 5} without considering the common event cost are equal to 6 and 7 respectively. Thus, the cost of first path is less than the second one.

To reach the unauthorized state in the composite system the cost of common event should be considered. The cost of reaching state 3 in component C2 is equal to 2. But, the transition e could not occur until component C1

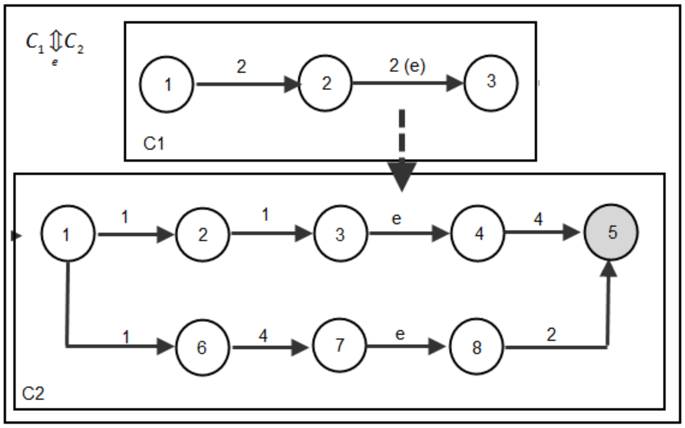

Figure 5: Synchronous parallel systems

produces the common event. Thus, the cost of reaching state 4 will be equal to 4 instead of 2. But, the cost of reaching state 8 in the second path is equal to 5; because the common event is produced before component C2 reaches state 7. In other words, the vulnerability measure of the system is calculated as follows.

$\nu(C \leftarrowtail I) = \frac{1}{MIN\{4+4,5+2\}} = \frac{1}{7}$

Therefore, the cost of second vulnerable path will be less than the first one in the composite system. We can simply calculate this quantity using equation 10 as follows:

$\nu(C \leftarrowtail I) = MAX\{0, 0, \frac{1}{MIN\{\{MAX\{4,2\}+4\},\{MAX\{4,5\}+2\}\}}\} = \frac{1}{7}$

6. Case study

In this section we present a case study to illustrate the compositional vulnerability measurement process in a practical network. This example also specifies how different attackers cause different quantity of vulnerability in the same system.

Figure 6 depicts the configuration of a practical network. We want to measure the vulnerability of this network using compositional approach specified in section 5.

The network consists of two subnets. The first one is DMZ and contains web server and some workstations. The second one is internal subnet which contains database, files, application servers and workstations. There are

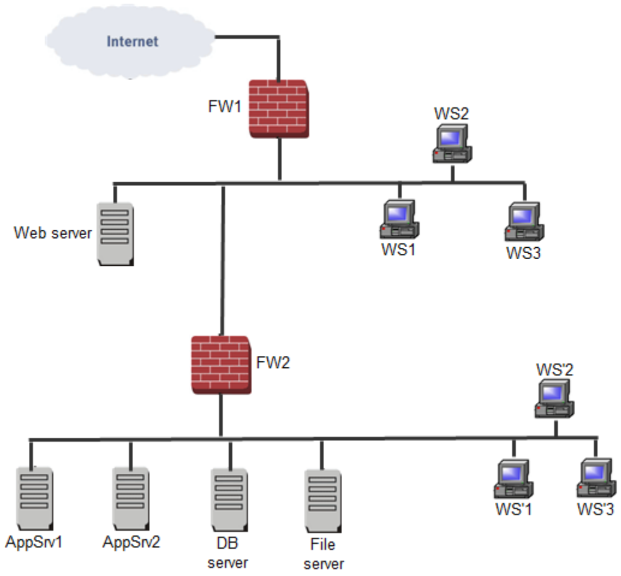

Figure 6: Configuration of a sample network

two firewalls in the network which separate DMZ from Internet and internal subnet from DMZ.

The attacker is an external user who wants to access the internal subnet. The security policies are as follows:

1. Root access to the application server is restricted to a specific user called admin.
2. Just the admin has root access on database.
3. Just the admin user has root access on file server.

Thus, access to the firewalls, workstations or DMZ's instruments does not violate the security policies. In other words, these components have no unauthorized state in their model and only their service time will effect on vulnerability measurement.

The data flow of the system is specified in figure 7.

Each request should pass the firewalls and DMZ before arriving to the internal network. To access to the database or file server the user should be authenticated first. Authentication result returns from application servers to the user and then other requests could be sent to the application servers. These servers pass requests to the database or file server. We have assumed

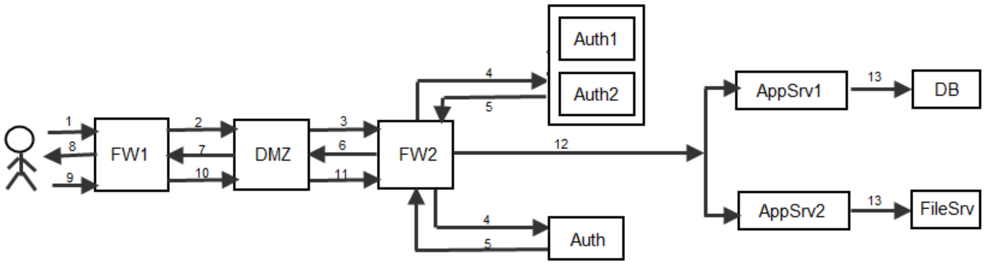

Figure 7: Configuration of a sample network

that there are two authentication mechanisms in the first application server. Considering figure 7, we can specify following relations between components.

1. Authentication mechanisms in AppSrv1 are connected to each other by the choice operator.
2. AppSrv1 is connected to the database by the synchronous parallel operator. In other words, the output of application server is used as database input. The same relationship exists between AppSrv2 and file server.
3. The set of AppSrv1 and database with the set of AppSrv2 and file server operate in the parallel manner.

AppSrv1 has two different authentication mechanisms. An attacker can violate each mechanism and get user or root access on the application server. Assume that the costs of reaching user and root access on first authentication mechanism of AppSrv1 by an attacker are c_1 and c'_1 respectively. These costs are equal to c_2 and c'_2 for the second authentication mechanism of AppSrv1. Reaching root access on AppSrv1 is a violation of security policies. Considering the choice operator between the Auth1 and Auth2, the vulnerability of AppSrv1 is calculated as follows:

$$\nu(AppSrv1) = MAX\{\nu(Auth1), \nu(Auth2)\} = MAX\{\frac{1}{ST_1 + c'_1}, \frac{1}{ST_1 + c'_2}\} \tag{11}$$

Where ST_1 is the sum of service time of FW1, DMZ and FW2.

$$ST_1 = ST_F(FW1) + ST_F(DMZ) + ST_F(FW2) \tag{12}$$

ST_F depicts the service time of the forward transition from outside to the internal network. If $c'_1 < c'_2$, the vulnerability measure of AppSrv1 will be equal to $\frac{1}{ST_1+c'_1}$.

The vulnerability measure of AppSrv2 could be computed by the same way. But, AppSrv2 has just one authentication mechanism. Assume that the costs of reaching user and root access on AppSrv2 are c_3 and c'_3 respectively. Thus, the vulnerability measure of AppSrv2 will be equal to $\frac{1}{ST_1+c'_3}$.

The abstract model of AppSrv1 is shown in figure 8. If an attacker has user access to the application server, he can get guest access to the database. And if he has root access on the application server, he will get user access to the database. Moreover, the intruder can attack the application server to get user access on database using his primitive user access on AppSrv1.

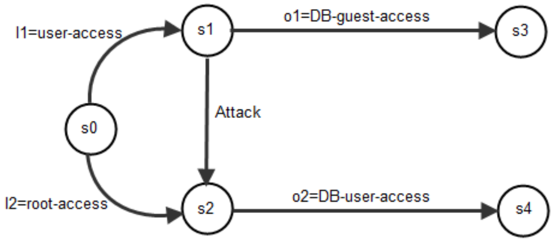

Figure 8: The abstract model of application server

After receiving the authentication result by the user, he can send other requests to the application servers. The service time of sending authentication request, receiving authentication result and sending second response is calculated as follows:

$$ST_1 = 2 \times ST_1 + ST_B(FW1) + ST_B(DMZ) + ST_B(FW2) \qquad (13)$$

Where ST_B depicts the backward service time from internal network to outside.

The service time of reaching user and root access on each application server is calculated as follows. $ST_{AppSrv1}^{user-access} = ST_2 + MIN\{C_1, C_2\} = ST_2 + C_2$

$ST_{AppSrv1}^{root-access} = ST_2 + MIN\{C'_1, C'_2\} = ST_2 + C'_1$
$ST_{AppSrv2}^{user-access} = ST_2 + C_3$
$ST_{AppSrv2}^{root-access} = ST_2 + C'_3$

Considering figure 8, the service time of each output event in AppSrv1 is calculated by following equations. $ST_{AppSrv1}^{o_1} = (ST^{o_1}|I_1) + ST_{AppSrv1}^{user-access} = \cos t(DB - guest - access) + ST_{AppSrv1}^{user-access}$

$ST^{o_2}_{AppSrv1} = MIN\{(ST^{o_2}|I_1) + ST^{user-access}_{AppSrv1}, (ST^{o_2}|I_2) + ST^{root-access}_{AppSrv1}\} = MIN\{\cos t(attack) + \cos t(DB - user - access) + ST^{user-access}_{AppSrv1},$
$\cos t(DB - user - access) + ST^{root-access}_{AppSrv1}\}$

Assume that AppSrv2 has the same construction as AppSrv1. Thus, we can use same equations for ApSrv2. Notice that AppSrv2 can access to the file server instead of database. Thus, we should replace $cost(DB - guest - access)$ and $cost(DB - user - access)$ by $cost(FileSrv - guest - access)$ and $cost(FileSrv - user - access)$ respectively.

An intruder can crack database password if he has guest access to it. He can also get user access and then use buffer overflow attack to reach the root access. The abstract model of database is specified in figure 9.

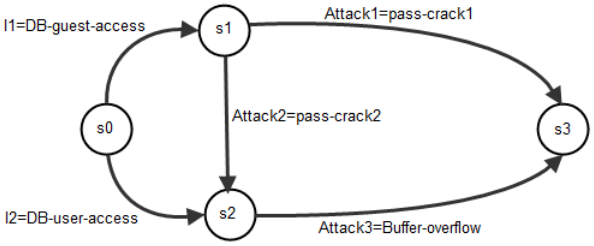

Figure 9: The abstract model of database server

According to the model, conditional vulnerability measure of the database are calculated as follows. $\nu(DB|I_1) = MAX\{\frac{1}{\cos t(attack_1)}, \frac{1}{\cos t(attack_2) + \cos t(attack_3)}\}$
$\nu(DB|I_2) = \frac{1}{\cos t(attack_3)}$

Considering the synchronous parallel composition of AppSrv1 and database and theorem 4, we can use the following simplified equation to measure the vulnerability of composed system.

$$\nu(C \leftarrow\!\prec I) = MAX\{\nu(C_1 \leftarrow\!\prec I), \nu(C_2 \leftarrow\!\prec I), \frac{1}{\underset{\forall paths \in C_2}{MIN}\{ST^{output}_{c_1} + \frac{1}{\nu(C_2 \leftarrow\!\prec I|Input)}\}}\} \tag{14}$$

Thus, the vulnerability measure of the composed system will be calculated as follows.

$\nu_1 = MAX\{\nu(AppSrv1), \nu(DB), \frac{1}{MIN\{ST^{o_1}_{AppSrv1} + \frac{1}{\nu(DB|I_1)}, ST^{o_2}_{AppSrv1} + \frac{1}{\nu(DB|I_2)}\}}\}$

The file server model is same as database. Thus, the vulnerability measure of the composition of AppSrv2 and FileSrv is calculated by the following equation.

Table 1: Sample values of service time and event costs for two intruders

Parameters	Intruder1	Intruder2
$ST_F(FW_1) = ST_B(FW_1)$	1	2
$ST_F(FW_2) = ST_B(FW_2)$	1	2
$ST_F(DMZ) = ST_B(DMZ)$	2	4
c_1	10	12
c_2	7	10
c_3	8	11
$c_1^{'}$	12	15
$c_2^{'}$	20	25
$c_3^{'}$	17	20
$cost(DB-guest-access)$	1	1
$cost(DB-user-access)$	1	1
$cost(attack)$	5	7
$cost(FileSrv-guest-access)$	1	1
$cost(FileSrv-user-access)$	1	1
$cost(attack_1)$	20	24
$cost(attack_2)$	10	15
$cost(attack_3)$	5	6

$$\nu_2 = MAX\{\nu(AppSrv2), \nu(FileSrv), \frac{1}{MIN\{ST_{AppSrv2}^{o1} + \frac{1}{\nu(FileSrv|I_1)}, ST_{AppSrv2}^{o2} + \frac{1}{\nu(FileSrv|I_2)}\}}\}$$

Now, we can use theorem 3 to calculate the vulnerability measure of the whole system.

$$\nu = MAX\{\nu_1, \nu_2\}$$

In the remaining of this section we use some arbitrary values for the parameters of the former equations and compare the results for two different intruders who have different capabilities. Assume that the first attacker is stronger than the second one. The cost of transitions and service time are shown in table 1.

Using these values, the vulnerability measure of the network against attacker I_1 and I_2 will be equal to 0.0625 and 0.0434, respectively. In other words, a stronger attacker can violate security policies easier. Thus, the network is more vulnerable against first intruder than second one.

7. Conclusion and future work

In this paper we introduce a new metric for vulnerability quantification and propose a method to evaluate the security. The method uses system properties and potential attacks and attacker's capabilities to model the system. We can use just attacker's events to construct an abstract level of the system model. Thus, our method is flexible and could be used to construct systems with different detail levels.

Another advantage of the method is that the amount of security is calculated considering the penetration delay time in addition to the probability of accessing unauthorized states. Thus, we can compare the vulnerability of systems with the same probability to reach unauthorized states. Therefore, in contrast to some previous work, it could be a better estimation of security.

The method has compositional property and we can calculate the vulnerability measure of a complex system using the vulnerability quantities of constitutive components.

We have assumed that all unauthorized states have the same importance and impact levels. But it seems that this assumption is not always true in practice. For example, if an intruder gain a root access, she can make more damage on the system in contrast to the situation where she has only a user access. Thus, we suggest to consider this parameter in the model as a future work.

References

[1] Blaze, M., Diffie, W., Rivest, R., Schneier, B., Shimomura, T., Thompson, E., & Wiener, M. (1996). *Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security*. Technical Report Ad Hoc Group of Cryptographers and Computer Scientists.

[2] Eusgeld, I., Fechner, B., Salfner, F., Walter, M., Limbourg, P., & Zhan, L. (2008). Dependability metrics. Lecture Notes in Computer Science (LNCS) 4909 chapter 9 Hardware reliability. (pp. 59–103). Berlin: Springer-verlag.

[3] Frigault, M., & Wang, L. (2008). Measuring network security using bayesian network-based attack graphs. In *COMPSAC '08: Proceedings of the 32nd Annual IEEE International Computer Software and*

Applications Conference (pp. 698–703). Washington, DC, USA: IEEE Computer Society.

[4] Frigault, M., Wang, L., Singhal, A., & Jajodia, S. (2008). Measuring network security using dynamic bayesian network. In *QoP '08: Proceedings of the 4th ACM workshop on Quality of protection* (pp. 23–30). Alexandria, Virginia, USA: ACM.

[5] Jha, S., Sheyner, O., & M., W. J. (2002). *Minimization and reliability analysis of attack graphs*. Technical Report CMU-CS-02-109 School of Computer Science, Carnegie Mellon University.

[6] Jonsson, E., & Olovsson, T. (1997). A quantitative model of the security intrusion process based on attacker behavior. *IEEE Transactions on Software Engineering*, *23*, 235–245.

[7] Kaâniche, M., Alata, E., Nicomette, V., Deswarte, Y., & Dacier, M. (2006). Empirical analysis and statistical modelling of attack processes based on honeypots. In *Supplemental Proceedings of the 2006 International Conference on Dependable Systems and Networks (DSN)* (pp. 119–124). Philadelphia.

[8] Leversage, D. J., & Byres, E. J. (2008). Estimating a system's mean time-to-compromise. *IEEE Security & Privacy*, *8*, 52–60.

[9] Littlewood, B., Brocklehurst, S., Fenton, N., Mellor, P., Page, S., Wright, D., Dobson, J., McDermid, J., & Gollmann, D. (1993). Towards operational measures of computer security. *Journal of Computer Security*, *2*, 211–230.

[10] Madan, B., Goseva-Popstojanova, K., Vaidyanathan, K., & Trivedi, K. (2002). Modeling and quantification of security attributes of software systems. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN 02)* (pp. 505–514).

[11] Manadhata, P., & Wing, J. (2004). *Measuring a system's attack surface*. Technical Report CMU-CS-04-102 CMU School of Computer Science.

[12] McDermott, J. (2005). Attack-potential-based survivability modeling for high-consequence systems. In *Proceedings of the Third IEEE International Workshop on Information Assurance (IWIA '05)* (pp. 119–130). Washington, DC, USA.

[13] Mehta, V., Bartzis, C., Zhu, H., Clarke, E., & Wing, J. (2006). Ranking attack graphs. In *Recent Advances in Intrusion Detection* chapter Ranking Attack Graphs. (pp. 127–144). Springer Berlin / Heidelberg volume 4219 of *Lecture Notes in Computer Science (LNCS).*

[14] Mell, P., Scarfone, K., & Romanosky, S. (2006). Common vulnerability scoring system. *IEEE Security & Privacy*, *4*, 85–89.

[15] Pamula, J., Jajodia, S., Ammann, P., & Swarup, V. (2006). A weakest-adversary security metric for network configuration security analysis. In *Proceedings of the 2nd ACM workshop on Quality of protection* (pp. 31–38).

[16] Pham, N., Baud, L., Bellot, P., & Riguidel, M. (2008). A near real-time system for security assurance assessment. In *The Third International Conference on Internet Monitoring and Protection (ICIMP 08)* (pp. 152–160). Bucharest: IEEE Computer Society Press.

[17] Pradhan, D. K. (1996). *Fault-tolerant computer system design*. New jersey: Prentice hall.

[18] S., N., S., J., B., O., & M., J. (2003). Efficient minimum-cost network hardening via exploit dependency graphs. In *The 19th Annual Computer Security Applications Conference (ACSAC 03)* (p. 8695).

[19] Sallhammar, K., Helvik, B., & Knapskog, S. (2006). Towards a stochastic model for integrated security and dependability evaluation. In *Proceedings of the First International Comference on Availability, Reliability and Security (AReS).*

[20] Shahriari, H. R. (2007). *Modeling and analysis of computer system vulnerabilities*. Ph.D. thesis Sharif University of Technology.

[21] Singh, S., Lyons, J., & Nicol, D. M. (2004). Fast model-based penetration testing. In *The 2004 Winter Simulation Conference, Vols 1 and 2* (pp. 309–317).

[22] Wang, L., Noel, S., & Jajodia, S. (2006). Minimum-cost network hardening using attack graphs. *Computer Communications*, *29*, 3812–3824.

[23] Wang, L., Singhal, A., & Jajodia, S. (2007). Toward measuring network security using attack graphs. In *Proceedings of the 2007 ACM workshop on Quality of protection* (pp. 49–54).

[24] Xiang, Z., Chen, Y., Jian, W., & Yan, F. (2005). A jackson network-based model for quantitative analysis of network security. *Intelligence and Security Informatics*, (pp. 517–522).

Hossein Homaei received his B.Sc in Information technology in 2007 and received his M.Sc in Information security in 2010 from AmirKabir University of Technology, Tehran, Iran. His Msc thesis is about quantifying the vulnerability of computer systems. His research interests are security analysis, and Formal Methods in Security.

Hamid Reza Shahriari received his Ph.D. in Computer Science in Sharif University of Technology in 2007. He received his M.Sc. in Computer Science from Amir-Kabir University of Technology, Tehran, Iran, in 2000. His Ph.D. thesis is about vulnerability analysis of computer networks. His research interests are Information Security and Formal Methods in Security.