

Software Vulnerability Patch Management with Semi-Markov Decision Process

Chien-Cheng Huang^{1,*}, Kwo-Jean Farn², Feng-Yu Lin³ and Frank Yeong-Sung Lin¹

¹ Department of Information Management, National Taiwan University, Taipei 10617, Taiwan

² Institute of Information Management, National Chiao Tung University, Hsinchu 30010, Taiwan

³ Department of Criminal Investigation, Central Police University, Taoyuan 33304, Taiwan

Received: 4 Apr. 2013, Revised: 8 Aug. 2013, Accepted: 10 Aug. 2013

Published online: 1 Nov. 2013

Abstract: Information security incidents frequency has been increasing dramatically, the aim of this study is to analyze the state-space reachability problems through the transition of vulnerable status after the informative system vulnerability exposure. In this research we took into consideration the time factor to analyze the arrival time to reachable states problem discussed in stochastic Petri nets. The mean arrival time and variance of the process between starting from an initial state and arriving at reachable states. We will therefore elaborate a novel model based on the semi-Markov stochastic Petri nets model for analyzing the period between the exposure of the vulnerability and the completion of its patch. We use the semi-Markov process to analyze the state-space reachability problems of the stochastic Petri nets, resulting in a novel model for software vulnerability patch management. Moreover, we include also the concept of discounted multi-objective semi-Markov decision process to obtain the total of the efficient extreme point set.

Keywords: Software vulnerability, patch management, arrival time, stochastic Petri nets, semi-Markov decision process.

1 Introduction

Along with the growing importance of information systems all over the globe, a new technology was born and pushed human life style toward the cyber-society of knowledge-based economy. The usage of information technology changed people's life, environment and even their value. A new era therefore started. However, even if we enjoy today a more and more convenient life, we face at the same time the new threats of information security. Due to the significant importance of internet and system software, the frequency of network attacks increased also. The countless categories or sources of these dangers have a very negative impact on information security, even affecting the economy [1, 12]. These attacks will not only result in information security incidents but also damage the informative system itself or at the extreme edge paralyze the entire network. In general, most of the internet assaults are using the vulnerabilities caused by misconfiguration or non-patch of the informative system [2, 3, 5, 7, 17].

In the actual word, assuming that a multinational enterprise has more than 30,000 employees all over the globe, the users of its system including office personnel

and tele-worker, and that the information assets include more than 30,000 devices. Usually there is not enough time to deploy and update the patch of vulnerability, the attack may occur before.

In recent years, the frequency of zero-day attack increased a lot, the vulnerabilities amount of the application already surpassed the vulnerabilities amount of the operating system itself [26]. In the studies related to the field of optimal patching policies, [15] proposed a mathematical model for trade-off between the confidentiality and availability when considering the release of the vulnerability patch. Through this model, they calculated the optimal frequencies of regular and irregular patching.

In our study, we use the semi-Markov process to analyze the state-space reachability problems of the stochastic Petri nets, modeling a software vulnerability patch management system. In the concept of Petri nets, it is assumed that all transitions occur instantaneously. On our side we enlarge the concept to stochastic Petri nets to neutralize the limitation of this assumption in the real world. At the same time, we analyze the state-space reachability problems of the stochastic Petri nets, and

* Corresponding author e-mail: d97725002@ntu.edu.tw

explain individually the reachable states of the Petri nets: serial, parallel, self-loop. Plus also the general case in which the three states combine together. Moreover, we also take into consideration the discounted multi-objective semi-Markov decision process, applying the policy iteration to solve the equation to obtain the total set of the efficient extreme point [10, 11, 14, 16].

The structure of this paper is briefly explained as below: Section 1 is the introduction. Section 2 is the related works presentation, including the semi-Markov process and stochastic Petri nets. Section 3 analyzes the application of the semi-Markov process on Petri nets. Section 4 is the study of discounted multi-objective semi-Markov decision process. Finally, the research will be concluded in Section 5.

2 Related works

2.1 Semi-Markov process

The Markov decision process has already been widely used in the field of management science [13]. As an example the solved problems include: queueing theory, inventory theory, maintenance of equipment and dynamic programming etc... all of them are applicable for the Markov decision process model, through which we can obtain an optimal policy to manage all these complex problems. However, the common continuous-time Markov process assumes that its continuous time does not violate the exponential distribution. Therefore, there are some limitations on usage in the real world. In other words, the common continuous-time Markov process is only one special case of the semi-Markov process [6, 20, 21, 24, 25, 27, 28]. The application range of the semi-Markov process on decision making is much bigger than the continuous-time Markov process.

Usually the semi-Markov process is for achieving a single objective, while in reality the decision maker needs a multi-objective system which can optimize the decision process toward several targets. But, contradiction may occur between the various objectives, as an example: pursue profit maximization and cost minimization, or the objectives don't have a uniform conversion unit, all these leading to the problem of multi-objective programming. [18] already mentioned this concept in his studies, however it was [19] whom proposed it via a mathematical model listing the problems of vector maximization and solved efficiently the maximum constraint. After this point, studies started to flourish in this field of research [4, 8].

2.2 Stochastic Petri nets

The primary design and application of Petri nets was for systems modeling, especially on mutually independent

components. Through application on a system with cooperative control problems, Petri nets could elaborate a model describing its internal structure and let us realize which nodes were the key of the system itself [23]. The importance of the time factor was usually significant on system research therefore in this study we included the time factor when analyzing the arrival time problem of the reachable states of the Petri nets. From the mean arrival time and variance between the initial state and the arrival at a reachable state. This move provides us with a better analytic tool when building the dynamical system from Petri nets.

In the concept of Petri nets, the transition is accomplished instantaneously, meaning that the symbolized event or action is finished immediately with a zero need of time and implying that two events or actions cannot happen at the same time. But in a real system, all kinds of transition need time, the difference being the duration [9, 22]. Therefore, in this study we assume that the time needed for each transition is a random variable and provide it with probability distribution function, and that the probability that two events or actions occur at the same time is zero, the result being the stochastic Petri nets. Through the process mentioned above, we are then able to solve the limitation on Petri nets.

In this study, t symbolizes time, T means transition. There are two parameters for each transition T , in GERT (Graphical Evaluation and Review Technique) they are (p_i, t_i) , in the semi-Markov process they are (p_{ij}, τ_{ij}) . p_i and p_{ij} both represent the probability of transition, while t_i and τ_{ij} both represents the time needed for this transition. Of course, $p_i, p_{ij} \geq 0$, and starting from a random state $\sum p_i = \sum p_{ij} = 1$.

As a summary, in this study we classify the reachable states of stochastic Petri nets as serial, parallel, self-loop. After this, we convert them into GERT and semi-Markov process flowchart, analyzing and solving them by their respective manner.

3 An approach to semi-Markov stochastic Petri nets

This study focuses on the arrival states problem of the stochastic Petri nets, in other word in it we analyze the mean arrival time and variance during the process between the starting point of a state toward another state we arrive after several transitions. We discuss briefly and separately about the four different situations: serial, parallel, self-loop and general case.

3.1 Serial

The state on Petri net is serial, as shown in Fig. 1. Fig. 1 (a) is the reachability tree. After spreading Fig. 1 (b) we obtain the token machine as shown in Fig. 1 (c). p_{ij} represents the

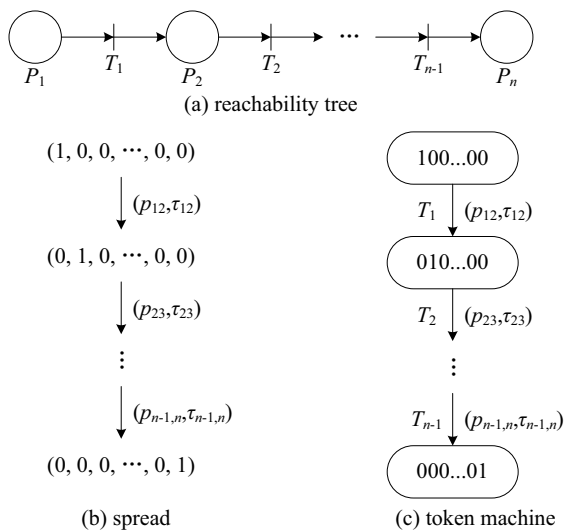


Fig. 1 Serial.

probability of transition from state i to the next state j . τ_{ij} represents the time needed to transit from the state i to the next state j . $h_{ij}(\cdot)$ represents the probability distribution function of τ_{ij} . We therefore develop Theorem 1 as below:

Theorem 1. The status of Petri nets is serial and assuming that the time needed for each transition are mutually independent, and that when transiting from one of the state i to another state j , the state transition departs from state i and arrives at state j . Thus, the mean arrival time: $\bar{t} = \sum_{m=1}^{j-1} \bar{t}_{m,m+1}$, the variance: $\check{t} = \sum_{m=1}^{j-1} \check{t}_{m,m+1}$.

3.2 Parallel

The state on the Petri nets is parallel, as shown in Fig. 2 (a)(b) and converted into the semi-Markov process flow chart as shown in Fig. 2 (c)(d), making state 1 and 2 respectively representing state (1,0) and (0,1).

3.3 Self-loop

The status of Petri nets is self-loop, as shown in Fig. 3 (a)(b), the status 1 and 2 respectively represent the status of (1,0) and (0,1). The conversion process to semi-Markov is as described in Fig. 3 (c)(d). Based on this we developed the Theorem 2 and 3.

Theorem 2. If one of the Petri nets status shows self-loop, whatever the time needed for transition τ_{ij} , its probability distribution function $h_{ij}(\cdot)$ is disparate or continuous. Then from the initial state 1 to the final state 2, the mean arrival time: $\bar{t} = \bar{\tau}_1 / (1 - p_{11}) = \bar{k} \cdot \bar{\tau}_1$, the variance: $\check{t} = [(1 - p_{11})\check{\tau}_1 + p_{11}\bar{\tau}_1(2\bar{\tau}_{11} - \bar{\tau}_1)] / (1 - p_{11})^2 = \bar{k}_1\bar{\tau}_1 + \check{k} \cdot \bar{\tau}_1 \cdot (2\bar{\tau}_{11} - \bar{\tau}_1)$. On the other hand, the mean transition

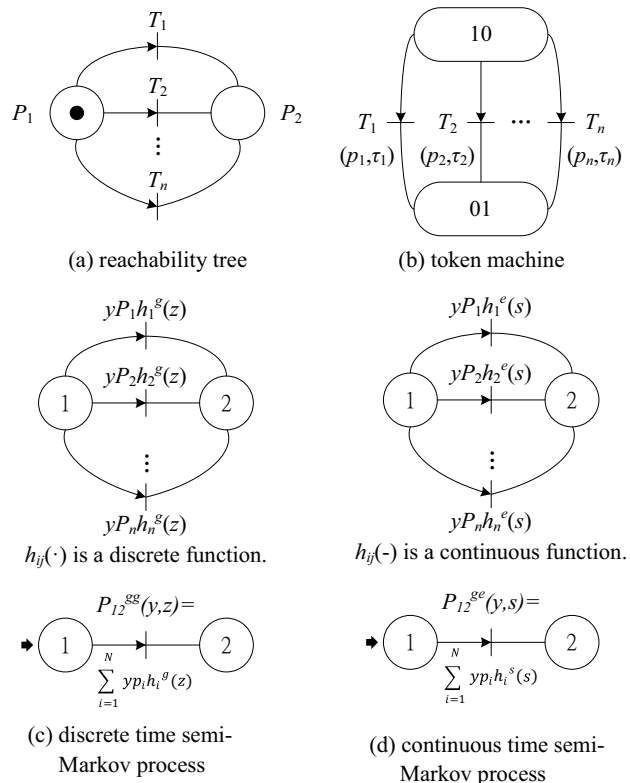


Fig. 2 Parallel.

frequency from the initial state 1 to the final state 2: $\bar{k} = 1 / (1 - p_{11})$, the variance: $\check{k} = p_{11} / (1 - p_{11})^2$.

Theorem 3. If one of the Petri nets shows self-loop status, whatever the time needed for transition τ_{ij} , its probability distribution function $h_{ij}(\cdot)$ is disparate or continuous. Then from the initial state 1 to the final state 2, the mean arrival time: $\bar{t} = \bar{\tau}_1 / (1 - p_{11})$, the second moment: $\bar{t}^2 = (\bar{\tau}_1^2 + 2p_{11}\bar{\tau}_1\bar{\tau}_{11}) / (1 - p_{11})$, then variance: $t = \bar{t}^2 - \bar{t}^2$.

3.4 General Case

For the general case of Petri nets, the token machine of the reachable states can finally be simplified into the three types mentioned above, the theorem is the same as the simplified logic of the signal flow graph. As the example of Petri nets in Fig. 4 (a), the token machine of its reachable states is as in Fig. 4 (b).

From Theorem 4 and 5 we calculate $\bar{k}, \check{k}, \bar{t}$ and \check{t} , we can convert Fig. 4 (b) into the flowchart of semi-Markov process, as shown in Fig. 4 (c). Through even more simplification we solve the equation. The process only includes the disparate state of $h_{ij}(\cdot)$ while the state in which $h_{ij}(\cdot)$ is continuous can accept the same logic and deduction process.

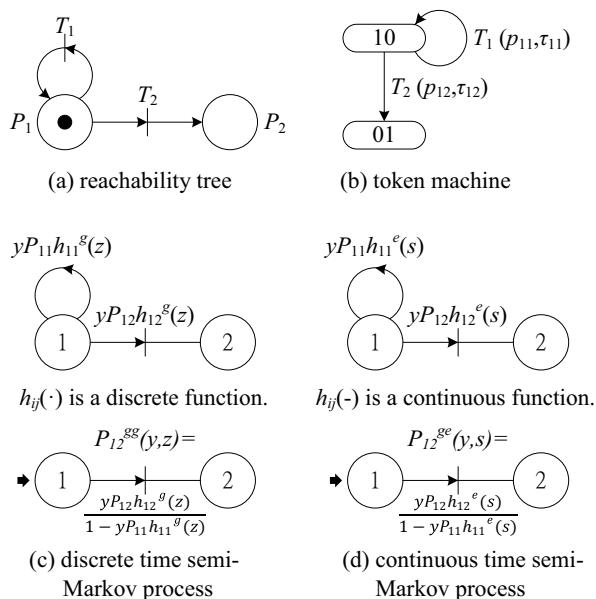


Fig. 3 Self-loop.

Theorem 4. If one of the Petri nets show parallel phenomenon status, for each of the time needed for transition τ_{ij} , its probability distribution function $h_{ij}(\cdot)$ is disparate, then from the initial state 1 to the final state 2, the mean arrival time: $\bar{t} = \frac{d}{dz} p_{12}^{gg}(1, z)|_{z=1}$, second moment: $\bar{t}^2 = \frac{d^2}{dz^2} p_{12}^{gg}(1, z)|_{z=1} + \bar{t}$, the variance: $\check{t} = \bar{t}^2 - \bar{t}^2$. On the other hand, the mean transition frequency from the initial state 1 to the final state 2, $\bar{k} = \frac{d}{dy} p_{12}^{gg}(y, 1)|_{y=1}$, the second moment: $\bar{k}^2 = \frac{d^2}{dy^2} p_{12}^{gg}(1, z)|_{z=1} + \bar{k}$, the variance: $\check{k} = \bar{k}^2 - \bar{k}^2$.

Theorem 5. If one of the Petri nets shows parallel phenomenon status, and for each transition the time needed is τ_{ij} and its probability function $h_{ij}(\cdot)$ is continuous, then from initial state 1 to final state 2, the mean arrival time: $\bar{t} = -\frac{d}{ds} p_{12}^{ge}(1, s)|_{s=0}$, second moment: $\bar{t}^2 = \frac{d^2}{ds^2} p_{12}^{ge}(1, s)|_{s=0} + \bar{t}$, variance: $\check{t} = \bar{t}^2 - \bar{t}^2$. And the mean transition frequency from initial state 1 to final state 2 as $\bar{k} = \frac{d}{dy} p_{12}^{ge}(y, 0)|_{y=1}$, second moment: $\bar{k}^2 = \frac{d^2}{dy^2} p_{12}^{ge}(y, 0)|_{y=1} + \bar{k}$, variance: $\check{k} = \bar{k}^2 - \bar{k}^2$.

3.5 Parallel phenomenon

Due to the fact that Petri nets can be used in systems modeling of cooperative processing problems, before analyzing the problems of nets with the time factor, we view every place as an individual state and therefore parallel phenomenon can happen. Regarding the fact that the previous theory is not adequate anymore, we assume

that from initial state 1 to final state 2 exist n simultaneous paths. If we want to arrive at final state 2 by starting from state 1 and passing at the same time by these n paths and their reachable state 2. The process of semi-Markov is a below in Fig. 5.

If we wish to obtain the mean arrival time \bar{t} and variance \check{t} from state 1 to state 2, we must first take $p_{12}^{(i)gg}(y, z)$ or $p_{12}^{(i)ge}(y, s)$, $i = 1, 2, \dots, n$ and do separate anti-geometric conversion or anti-exponential conversion, so that we can deduce the probability function \tilde{t}_i of each separate path time variable and its probability density function. After this we take $\tilde{t} = \max(\tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_n)$ and calculate \bar{t} and \check{t} from the probability function or probability density function of \tilde{t} . Thanks to the above analysis we can summarize Theorem 6.

Theorem 6. If the existence of a reachable state of Petri nets occurs along with parallel phenomenon (as shown in Fig. 5), then the mean arrival time \bar{t} and variance \check{t} from initial state 1 to final state 2 are as below:

- (i) If \tilde{t}_i are mutually independent and share the same cumulative distribution function $F(t)$ and probability density function $f(t)$, plus the fact that $f(t)$ is with positive value and continuous on $0 \leq t \leq \infty$ and zero in other places, then $\tilde{t} = \tilde{t}_{(n)}$, the probability density function of \tilde{t} is:

$$g(t) = \begin{cases} n[F(t)]^{n-1}f(t) & 0 \leq t \leq \infty \\ 0 & \text{otherwise} \end{cases},$$

$$\bar{t} = \int_0^{+\infty} t \cdot g(t)dt,$$

$$\bar{t}^2 = \int_0^{+\infty} t^2 g(t)dt,$$

$$t = \bar{t}^2 - \bar{t}^2$$

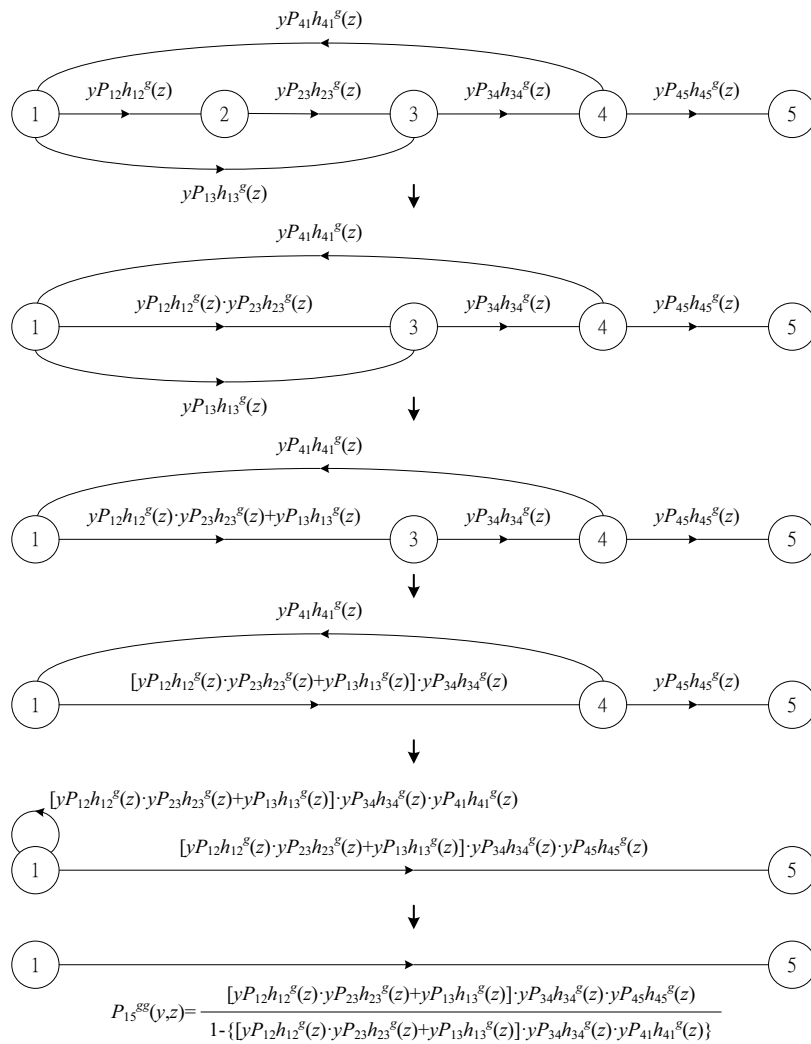
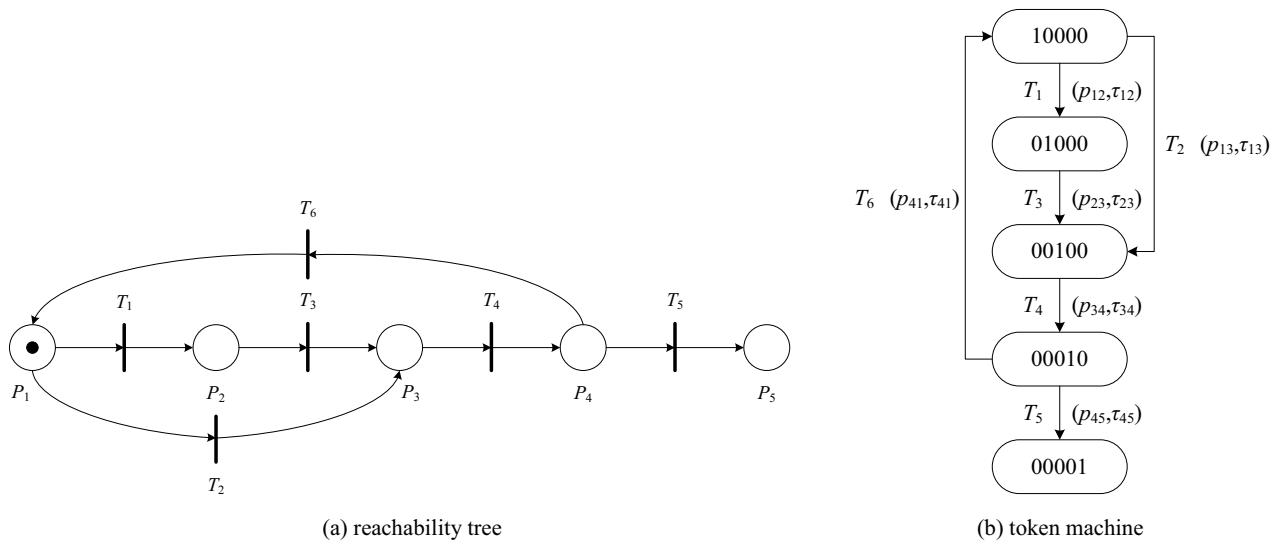
- (ii) If \tilde{t}_i are mutually independent and do not share the same fractional function, letting $f_i(t)$ and $F_i(t)$ respectively represent the probability density function and cumulative distribution function of \tilde{t}_i , plus the fact that $f_i(t) = 0, t < 0$. Then, $\tilde{t} = \tilde{t}_{(n)}$ and the probability density function of \tilde{t} is:

$$g(t) = \begin{cases} \prod_{i=1}^n [F_i(t)] \sum_{i=1}^n \left[\frac{f_i(t)}{F_i(t)} \right] & t \geq 0 \\ 0 & t < 0 \end{cases},$$

$$\bar{t} = \int_0^{+\infty} t \cdot g(t)dt,$$

$$\bar{t}^2 = \int_0^{+\infty} t^2 g(t)dt,$$

$$t = \bar{t}^2 - \bar{t}^2$$



(c) semi-Markov process

Fig. 4 General case.

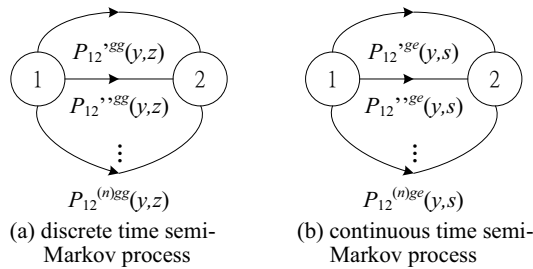


Fig. 5 Parallel phenomenon.

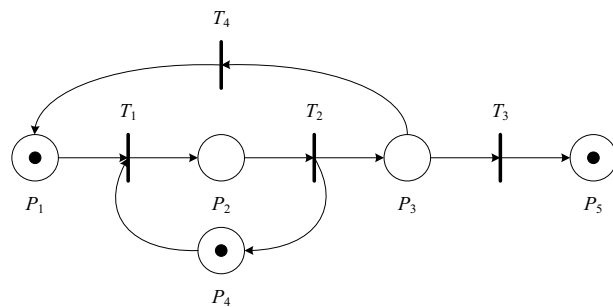


Fig. 6 Modeling software vulnerability patch management.

3.6 Case study of software vulnerability patch management

Regarding information systems products and services, from receiving a vulnerability notice until its update, the process is modeled by Petri nets as in Fig. 6. P_i are the i st places. T_j are the j st transitions. P_1 represents the arrival to vulnerability and awaiting for update. P_2 represents that the vulnerability is being updated. P_3 means that the update is finished and needs verification. P_4 means that the system is without loading. P_5 means that the update process is complete. T_1 means the update starts. T_2 means the update is finished and the verification already started. T_3 means that the update is complete and received certification. T_4 means that the update failed.

Assuming that the qualified rate of the update is 0.9, the probability distribution function of the transition time needed for T_1, T_2, T_3, T_4 is $h_{ij}(m)$ and respectively are $1/4 (3/4)^{m-1}, 1/5 (4/5)^{m-1}, 1/2 (1/2)^{m-1}, 1/3 (2/3)^{m-1}, m = 1, 2, 3, \dots$ helping us to calculate the mean time and variance of a vulnerability from its appearance until the update is complete. The token machine of this Petri nets is as shown in Fig. 7. Making the states 1, 2, 3, 4 respectively representing the states $(1,0,0,1,0), (0,1,0,0,0), (0,0,1,0,0),$ and $(0,0,0,1,1)$. This is converted into the semi-Markov process as the flow chart in Fig. 8 shows it.

Using $f(n) = c \cdot a^n$ and converting it into $f^g(z) = c/(1-a)$ and $f(n-k)$ through geometric, the result is

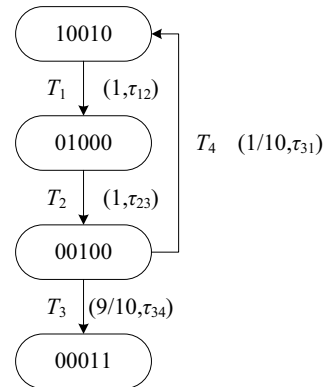


Fig. 7 Token machine of Petri nets for software vulnerability patch management.

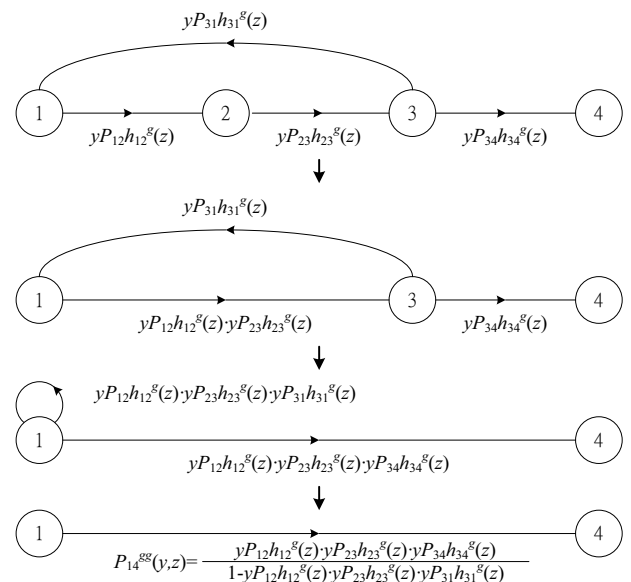


Fig. 8 Semi-Markov process for software vulnerability patch management.

$z^k f^g(z)$, and:

$$\begin{aligned}
 h_{12}(m) &= 1/4(3/4)^{m-1} \rightarrow h_{12}^g(z) = ((1/4)z)/(1-3/4z) \\
 h_{23}(m) &= 1/5(4/5)^{m-1} \rightarrow h_{23}^g(z) = ((1/5)z)/(1-4/5z) \\
 h_{34}(m) &= 1/2(1/2)^{m-1} \rightarrow h_{34}^g(z) = ((1/2)z)/(1-1/2z) \\
 h_{31}(m) &= 1/3(2/3)^{m-1} \rightarrow h_{31}^g(z) = ((1/3)z)/(1-2/3z)
 \end{aligned}$$

$$P_{14}^{sg}(y, z) = \frac{\frac{1}{4}z \cdot \frac{1}{5}z \cdot \frac{9}{10}z}{1 - \frac{3}{4}z \cdot 1 - \frac{4}{5}z \cdot 1 - \frac{1}{2}z}$$

$$\therefore P_{14}^{sg}(1, z) = \frac{\frac{9}{400}z^3 - \frac{6}{400}z^4}{1 - \frac{163}{60}z + \frac{329}{120}z^2 - \frac{713}{600}z^3 + \frac{241}{1200}z^4}$$

$$= \frac{27z^3 - 18z^4}{1200 - 3260z + 3290z^2 - 1462z^3 + 241z^4}$$

Therefore, we calculate the expected mean time needed from the arrival of vulnerability until its update is completed.

$$\bar{t} = \left. \frac{d}{dz} P_{14}^{sg}(1, z) \right|_{z=1}$$

$$= \left. \frac{d}{dz} \frac{27z^3 - 18z^4}{1200 - 3260z + 3290z^2 - 1462z^3 + 241z^4} \right|_{z=1}$$

$$= \left. \frac{9(10800z^2 - 29160z^3 + 29430z^4 - 13160z^5 + 2201z^6)}{(1200 - 3260z + 3290z^2 - 1462z^3 + 241z^4)^2} \right|_{z=1}$$

$$\cong 12.3$$

$$\bar{t}^2 = \left. \frac{d^2}{dz^2} P_{14}^{sg}(1, z) \right|_{z=1} + \bar{t} \cong 208$$

The variance: $t = \bar{t}^2 - \bar{t}^2 \cong 56.7$

4 Study of discounted multi-objective semi-Markov decision process

In the field of software vulnerability patch management, two targets need to be achieved: confidentiality and availability. The decision makers wish to reach an optimal status for both of them, even if they are contradictory. We calculate the result through discounted multi-objective semi-Markov decision process and apply the policy iteration concept, obtaining the total set of efficient extreme point.

Assuming that in a problem of discounted multi-objective semi-Markov decision process, there are two states, $S = \{1, 2\}$, respectively being “normally” and “to be patched”. For each state we have three possible options, $K_i = \{1, 2, 3\}, \forall i \in S$, being: no action, apply the update, using alternative devices. We have here two goals to achieve: confidentiality and availability. We set the discount factor $\alpha = 0.9, 0 \leq \alpha < 1$. Letting F being the function set of from state space S to policy space K . F is a finite set. F represents the decision vector under each state.

$$F = \{f_1, f_2, f_3, \dots, f_9\}$$

$$f_j = \{1, j\}, j = 1, 2, 3$$

$$f_{3+j} = \{2, j\}, j = 1, 2, 3$$

$$f_{6+j} = \{3, j\}, j = 1, 2, 3.$$

Assuming that $q_{ij}^k(0.9) = \int_0^\infty e^{-0.9t} dQ_{ij}^k(t), i \in S, j \in S, k \in K_i$ is already known, we set:

$$q(0.9, f_1) = \begin{bmatrix} 0.6 & 0.12 \\ 0.208 & 0.3 \end{bmatrix}$$

$$q(0.9, f_5) = \begin{bmatrix} 0 & 0.6 \\ 0.692 & 0 \end{bmatrix}$$

$$q(0.9, f_9) = \begin{bmatrix} 0.43 & 0.21 \\ 0.51 & 0.1 \end{bmatrix}$$

The respective vectors of return are:

$$\vec{\rho}(1, 1) = \begin{bmatrix} 25.2 \\ 28 \end{bmatrix}, \vec{\rho}(1, 2) = \begin{bmatrix} 42 \\ 30.4 \end{bmatrix}, \vec{\rho}(1, 3) = \begin{bmatrix} 16 \\ 15.2 \end{bmatrix}$$

$$\vec{\rho}(2, 1) = \begin{bmatrix} 20.1 \\ 18 \end{bmatrix}, \vec{\rho}(2, 2) = \begin{bmatrix} 10.7 \\ 12.5 \end{bmatrix}, \vec{\rho}(2, 3) = \begin{bmatrix} 18.4 \\ 20.2 \end{bmatrix}$$

I. Assuming that $\lambda = (0.4, 0.6)$, and using the policy iteration concept to calculate, taking randomly a policy $f_4 = \{2, 1\}$:

$$\vec{W}_{0.9}(1, f_4) = \begin{bmatrix} V_1(1, f_4) \\ V_2(1, f_4) \end{bmatrix}$$

$$= \begin{bmatrix} 42 + 0V_1(1, f_4) + 0.6V_1(1, f_4) \\ 28 + 0V_2(1, f_4) + 0.6V_2(1, f_4) \end{bmatrix}$$

$$\vec{W}_{0.9}(2, f_4) = \begin{bmatrix} 20.1 + 0.208V_1(1, f_4) + 0.3V_1(1, f_4) \\ 18 + 0.208V_2(1, f_4) + 0.3V_2(1, f_4) \end{bmatrix}$$

So, we obtain the two sets of simultaneous equations:

$$\begin{cases} V_1(1, f_4) - 0.6V_1(2, f_4) = 42 \\ -0.208V_1(1, f_4) + 0.7V_1(2, f_4) = 20.1 \end{cases}$$

$$\begin{cases} V_2(1, f_4) - 0.6V_2(2, f_4) = 28 \\ -0.208V_2(1, f_4) + 0.7V_2(2, f_4) = 18 \end{cases}$$

$$V_1(1, f_4) = 72.08, V_1(2, f_4) = 50.13$$

$$V_2(1, f_4) = 52.85, V_2(2, f_4) = 41.42$$

$$\vec{\lambda} \vec{W}_{0.9}(i, \pi^*) = \vec{\lambda} \vec{W}_{0.9}(i, \pi^*)$$

$$\pi^* = \{f_3\}$$

Making $E = \{\pi^*\}$, at this time B is an empty set. Using the efficiently testing criteria, we test all the policies adjacent to f_3 , its neighboring policy being $f_1(1, 1), f_2(1, 2), f_6(2, 3), f_9(3, 3)$.

(i)

$$\vec{W}_{0.9}(1, f_1) = \begin{bmatrix} 25.2 + 0.6V_1(1, f_1) + 0.12V_1(2, f_1) \\ 28 + 0.6V_2(1, f_1) + 0.12V_2(2, f_1) \end{bmatrix}$$

$$\vec{W}_{0.9}(2, f_1) = \begin{bmatrix} 20.1 + 0.208V_1(1, f_1) + 0.3V_1(2, f_1) \\ 18 + 0.208V_2(1, f_1) + 0.3V_2(2, f_1) \end{bmatrix}$$

So, we obtain two sets of simultaneous equations:

$$V_1(1, f_1) = 77.12, V_1(2, f_1) = 47$$

$$V_2(1, f_1) = 83.69, V_2(2, f_1) = 50.09$$

Because $\vec{e}_2(2, f_1) < 0$.

Therefore, f_1 does not support the efficiently testing criteria.

(ii)

$$\vec{W}_{0.9}(1, f_2) = \begin{bmatrix} 25.2 + 0.6V_1(1, f_2) + 0.12V_1(2, f_2) \\ 28 + 0.6V_2(1, f_2) + 0.12V_2(2, f_2) \end{bmatrix}$$

$$\vec{W}_{0.9}(2, f_2) = \begin{bmatrix} 10.7 + 0.692V_1(1, f_2) + 0V_1(2, f_2) \\ 12.5 + 0.692V_2(1, f_2) + 0V_2(2, f_2) \end{bmatrix}$$

So, we obtain two sets of simultaneous equations:

$$V_1(1, f_2) = 82.76, V_1(2, f_2) = 67.87$$

$$V_2(1, f_2) = 92.19, V_2(2, f_2) = 76.18$$

Because $\vec{e}_k(i, f_2) > 0, \forall k \in K_i, i \in S$.

Therefore, f_2 does support the efficiently testing criteria.

(iii)

$$\vec{W}_{0.9}(1, f_6) = \begin{bmatrix} 42 + 0.6V_1(2, f_6) \\ 30.4 + 0.6V_2(2, f_6) \end{bmatrix}$$

$$\vec{W}_{0.9}(2, f_6) = \begin{bmatrix} 18.4 + 0.51V_1(1, f_6) + 0.1V_1(2, f_6) \\ 20.2 + 0.51V_2(1, f_6) + 0.1V_2(2, f_6) \end{bmatrix}$$

So, we obtain two sets of simultaneous equations.

$$V_1(1, f_6) = 82.78, V_1(2, f_6) = 67.49$$

$$V_2(1, f_6) = 66.92, V_2(2, f_6) = 60.52$$

Because $\vec{e}_1(1, f_6) < 0$.

Therefore, f_6 does not support the efficiently testing criteria.

(iv)

$$\vec{W}_{0.9}(1, f_9) = \begin{bmatrix} 16 + 0.43V_1(1, f_9) + 0.21V_1(2, f_9) \\ 15.2 + 0.43V_2(1, f_9) + 0.21V_2(2, f_9) \end{bmatrix}$$

$$\vec{W}_{0.9}(2, f_9) = \begin{bmatrix} 18.4 + 0.51V_1(1, f_9) + 0.1V_1(2, f_9) \\ 20.2 + 0.51V_2(1, f_9) + 0.1V_2(2, f_9) \end{bmatrix}$$

So, we obtain two sets of simultaneous equations.

$$V_1(1, f_9) = 44.55, V_1(2, f_9) = 45.48$$

$$V_2(1, f_9) = 43.71, V_2(2, f_9) = 46.99$$

Because $\vec{e}_1(1, f_9) < 0$.

Therefore, f_9 does not support the efficiently testing criteria.

Accordingly, $E = \{f_2, f_3\}$, $B = \{f_3\}$.

II. Using the efficiently testing criteria to test the policies adjacent to f_2 , from the result we can know that they cannot support the efficiently testing criteria. Consequently, $E = \{f_2, f_3\}$, $B = \{f_2, f_3\}$.

III. Because, $E = B$, the sets of efficient stationary strategies $E_s(\alpha) = \{f_2, f_3\}$ and its respective total mean target vector are:

$$\vec{W}_{0.9}(1, f_2) = \begin{bmatrix} 82.76 \\ 92.19 \end{bmatrix}$$

$$\vec{W}_{0.9}(2, f_2) = \begin{bmatrix} 67.87 \\ 76.18 \end{bmatrix}$$

$$\vec{W}_{0.9}(1, f_3) = \begin{bmatrix} 82.96 \\ 92.08 \end{bmatrix}$$

$$\vec{W}_{0.9}(2, f_3) = \begin{bmatrix} 67.37 \\ 74.53 \end{bmatrix}.$$

5 Conclusion

In this study, we analyze the mean arrival time and variance of stochastic Petri nets with the semi-Markov process. This approach rely on the flow chart theory, each part on the flow chart has two parameters (P_a, \tilde{t}_a) , P_a represents the probability of this path, while \tilde{t}_a represents the time variable. Regarding the reachable states of Petri nets, we take the three basic internet types: serial, parallel and self-loop as basis and assume that the time variables of each path are mutually independent. Finally, we take into account the software vulnerability patch management concept of the discounted multi-objective semi-Markov decision process to obtain the total set of efficient extreme points.

References

- [1] Anderson, R., Moore, T.: The economics of information security. *Science*, **314**, 610–613 (2006).
- [2] Arbaugh, W.A., Fithen, W.L., McHugh, J.: Windows of vulnerability: a case study analysis. *Computer*, **33**, 52–59 (2000).
- [3] Arora, A., Caulkins, J.P., Telang, R.: Research note-sell first, fix later: impact of patching on software quality. *Management Science*, **52**, 465–471 (2006).
- [4] Arrow, K.J., Barankin, E.W., Blackell, D.: Admissible points of convex sets. *Contributions to the Theory of Games*, **2**, 87–91 (1953).
- [5] August, T., Tunca, T.: Network software security and user incentives. *Management Science*, **52**, 1703–1720 (2006).
- [6] Barlow, R.E., Proschan, F.: *Mathematical theory of reliability*. Wiley (1965).
- [7] Cavusoglu, H., Cavusoglu, H., Raghunathan, S.: Efficiency of vulnerability disclosure mechanisms to disseminate vulnerability knowledge. *IEEE Transactions on Software Engineering*, **33**, 171–185 (2007).
- [8] Cohon, J.L.: *Multiobjective programming and planning*. Academic Press. (1978).
- [9] Coolahan, JR., J.E., Roussopoulos, N.: Timing requirements for time-driven systems using augmented Petri nets. *IEEE Transactions on Software Engineering*, **SE-9**, 603–616 (1983).

- [10] de Cani, J.S.: A dynamic programming algorithm for embedded Markov chains when the planning horizon is at infinity. *Management Science*, **10**, 716–733 (1964).
- [11] Fox, B.: Markov renewal programming by linear fractional programming. *SIAM Journal on Applied Mathematics*, **14**, 1418–1432 (1966).
- [12] Gordon, L. A., Loeb, M.P.: The economics of information security investment. *ACM Transactions on Information and System Security*, **5**, 438–457 (2002).
- [13] Hillier, F. S., Lieberman, G.J.: *Introduction to operations research*. 9th edition, McGraw-Hill, (2010)
- [14] Howard, R.A.: Research in semi-Markov decision structures. *Journal of the Operational Research Society of Japan*, **6**, 163–199 (1964).
- [15] Ioannidis, C., Pym, D., Williams, J.: Information security trade-offs and optimal patching policies. *European Journal of Operational Research*, **216**, 434–444 (2012).
- [16] Jewell, W.S.: Markov-renewal programming. I: formulation, finite return models. *Operations Research*, **11**, 938–948 (1963).
- [17] Kannan, K., Telang, R.: Market for software vulnerabilities think again. *Management Science*, **51**, 726–740 (2005).
- [18] Koopmans, T. C.: *Activity analysis of production and allocation*. Cowles Commission Monograph, Wiley, **13**, (1951).
- [19] Kuhn, H.W., Tucker A.W.: *Nonlinear programming*. Proceedings Second Berkeley Symposium on Mathematical Statistics and Probability, 481–492 (1951).
- [20] Lévy, P.: Systems semi-Markoviens ayant au plus une infinie denombrable d'Etats possibles. *Porc. Intern. Congr. Math.*, **2**, 294 (1954a).
- [21] Lévy, P.: Processus semi-markoviens. *Porc. Intern. Congr. Math.*, **3**, 416–426 (1954b).
- [22] Lewis, T.G.: *Software engineering: analysis and verification*. Reston Publishing Company, (1982).
- [23] Petri, C.A.: *Kommunikation mit automaten*. Doctoral Dissertation, Darmstadt University of Technology, (1962).
- [24] Pyke, R.: Markov renewal processes: definitions and preliminary properties. *Annals of Mathematical Statistics*, **32**, 1231–1242 (1961a).
- [25] Pyke, R.: Markov renewal processes with finitely many states. *Annals of Mathematical Statistics*, **32**, 1243–1259 (1961b).
- [26] SANS Institute: *Top cyber security risks - vulnerability exploitation trends*, (2009).
- [27] Smith, W. L.: Regenerative stochastic processes. *Proceedings of the Royal Society A*, **232**, 6–31 (1955).
- [28] Takács, L.: Some investigations concerning recurrent stochastic processes of a certain type. *Magyar Tud. Akad. Mat. Kutató Int. Kozl.*, **3**, 115–128 (1954).



Chien-Cheng Huang received his MS degree in information management from the National Chiao Tung University, Taiwan, Republic of China, in 2008. He is currently a Ph.D. student in information management at the National Taiwan University. His research interests include information systems, database systems, data mining, and information security.



Kwo-Jean Farn is an adjunct associate professor with the National Chiao Tung University in Taiwan. He received his Ph.D. degree in 1982. He has had a 20-year career at Information Technology and about 10-year career at Information Security. He was the chair of the Implementation National Critical Information Infrastructure Protection Project at the Computer & Communications Research Laboratories/Industrial Technology Research Institute (CCL/ITRI) from 1999 to 2000. He had worked at the ITRI for more than 18 years until the summer of 2001. He has 9 patents in the information security area. He also received the National Standardization Award in 2009.



Feng-Yu Lin received his Ph.D. degree in traffic and transportation from the National Chiao Tung University, Taiwan, Republic of China, in 2004. He is an adjunct assistant professor with the Department of Criminal Investigation, Central Police University in Taiwan. His research interests include communication/network forensics, data mining, and information security.

**Frank Yeong-Sung**

Lin received his BS degree in electrical engineering from the National Taiwan University in 1983, and his Ph.D. degree in electrical engineering from the University of Southern California in 1991. After graduating from the USC,

he joined Telcordia Technologies (formerly Bell Communications Research, abbreviated as Bellcore) in New Jersey, U.S.A. Since 1996, he has been with the faculty of the Information Management Department, National Taiwan University. His research interests include network optimization, network planning, network survivability, performance evaluation, high-speed networks, distributed algorithms, content-based information retrieval/filtering, biometrics and network/information security.