Quantitative Security Analysis of a Dynamic Network System under Lateral Movement-based Attacks

Yu Shi[a], Xiaolin Chang[a], Ricardo J. Rodríguez[b], Zhenjiang Zhang[c], Kishor S. Trivedi[d]
[a]Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, P. R. China
[b]Centro Universitario de la Defensa, Academia General Militar, Zaragoza, Spain
[c]School of Electronics and Information Engineering, Beijing Jiaotong University, P. R. China
[d]Duke University, Durham, NC 27708, USA

Abstract—Malicious lateral movement-based attacks have become a potential risk for many systems, bringing highly likely threats to critical infrastructures and national security. Through launching this kind of attacks, adversaries first compromise a fraction of the targeted system within the infrastructure and then move laterally to the rest of the system until the whole system is infected. Various approaches have been proposed to study and/or defend against the behaviors of lateral movement-based attacks. However, few of them studied the transient behaviors of dynamic attacking and dynamic targeted systems. This paper aims to analyze the transient security of a dynamic network system under lateral movement-based attacks from the time that attack-related abnormity in the system is detected until mechanisms are designed and deployed to defend against attacks. We explore state-space modeling techniques to construct a survivability model for quantitative analysis. A phased piecewise constant approximation (P^2CA) approach is also proposed to derive the model state transient probabilities, with which we derive the formulae for calculating metrics of interest. The proposed approach allows both model state transition rates and the number of model states to be time-varying during the system recovery. Numerical analysis is finally carried out for investigating the impact of various dynamic system parameters on system security.

Keywords—Lateral Movement-based Attack; Dynamic Transient Analysis; Non-homogeneous Continuous-Time Markov Chain; Piecewise Constant Approximation;

I. INTRODUCTION

Lateral movement-based attacks are a set of stealthy and continuous computer hacking processes often orchestrated by individuals or organizations targeting a specific entity (usually private organizations, states or both) for business or political motivations [1][2]. When launching these attacks, adversaries (attackers) will first infect one or more vulnerable access points of the system to get a success login in them. Then they will compromise other access points through these access points persistently, thus expanding the control over other computers, servers, and infrastructure devices within the whole system. Finally, the whole system will be controlled. Lateral movement-based attacks are in fact a form of horizontal privilege escalation and have been exploited in different kinds of malicious software. Fig.1 illustrates the concept map of lateral movement-based attacks.

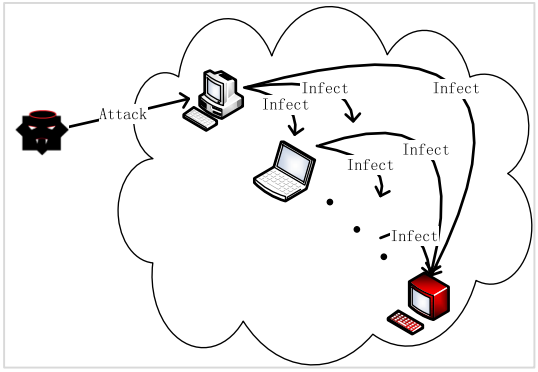

Fig.1 Concept map of lateral movement-based attack.

Over the past years lateral movement-based attacks were wildly used in many scenarios and brought various threats because of its covert property. One typical example is that it's used as one phase of advanced persistent threat (APT) attacks, which are viewed as one of the most dangerous threats to many systems, critical infrastructures, and even national security. An APT attack is mostly directed at enterprise targets for financial gains and at political and military targets for ideological motivations [3]. The security reports of Kaspersky Lab [4] claimed that there was a sharp rise in the sophistication of nation-state sponsored attackers and a merger of tactics, techniques, and procedures (TTPs) between APT actors and financially motivated cyber criminals. Typically, an APT attack consists of six phases including reconnaissance and weaponization, delivery, initial intrusion, command and control, lateral movement, and data exfiltration. Lateral movement usually happens in conjunction with command and control communications to gather internal system structure information, guide the expansion process, and ultimately cause data exfiltration. In the phase of lateral movement, adversaries will first perform internal reconnaissance and acquire intelligence of the network to invade some vulnerable access points in the system. Then they will compromise additional parts of the system to obtain escalated privileges. During the compromising process, adversaries can identify and collect various classified information they need (for instance, intellectual property, passwords, or other login data) [7]. The lateral movement may last for a long period and cause various damages to the targeted systems before the attack is effectively defended. This situation becomes even more critical when cybercriminals attempt to attack critical systems that provide essential services to the society, such as nuclear facilities or water treatment plants.

This paper aims to make a model-based study of the transient security of a dynamic network system under lateral movement-based attacks from the time that attack-related abnormity is detected in the system (e.g., by intrusion detection systems), until defense mechanisms are deployed. Generally

speaking, the system is composed of multiple access points, which are supposed to be computers in this paper just for convenience of description. To minimize the power of attackers, we assume that the engineers of the targeted system make a timely and shortly incident response by quickly designing defensive mechanisms once they are aware of the attack-related abnormity existing in the system. Before defensive mechanisms are well deployed, more computers in the system might be compromised and more data exfiltration may occur. The system will be secure against those attacks once the designed defensive mechanisms are deployed.

Note that system parameter values may vary during the system recovery from the attacks. For example, the rates for attackers to gather sensitive information from compromised computers may increase. In addition, the number of active computers may decrease due to damaged computer or increase due to the expansion of the network system. Here, an active computer is defined as a computer running in the system and then it is susceptible to attacks. These variations may lead to that state transition rates and/or the number of model states are time-varying.

In this paper, we exploit state-space modeling techniques to construct a survivability model to quantitatively analyze the transient security. A phased piecewise constant approximation (denoted as P^2CA) approach is proposed for deriving state transient probabilities in the scenario where both model state transition rates and model state number may vary. To the best of our knowledge, it is the first time to investigate the scenario with the varying state number during the system recovery. The key idea of P^2CA is to divide the recovery process into phases according to when there is change in the number of model states. Namely, the state numbers are different in two adjoining phases. Furthermore, in each phase the number of model states is constant while the transition rates may vary. Therefore, the survivability model in each phase is a non-homogeneous continuous time Markov chain and piecewise constant approximation (PCA) [5][6] method is adopted to derive formulae for calculating model state transient probabilities. More details of P^2CA including how to calculate state transient probabilities are given in Section III. Numerical analyses are later carried out for evaluating the impact of various system parameters on system security dynamically.

The rest of the paper is organized as follows. Section II gives related work. Section III describes the system considered in this paper and the model. The concrete solutions of our model are also provided in this section. Numerical analysis and discussions are presented in Section IV. Section V concludes the paper and states future work.

II. Related Work

This section first presents related works on lateral movement attacks and then discusses the existing transient models, highlighting the difference of our modeling work from the existing survivability models.

Lateral movement is a kind of common and dangerous network attack methods, usually causing stealthy and continuous damage to the targeted system. Efforts have been put to study this kind of attacks. Chen *et al.* [7] studied APT attack including the lateral movement, characterizing its distinguishing characteristics and attack model, and analyzing its common techniques. Some recent attacking methods used by APTs and attack patterns were first analyzed, and then effective countermeasures were proposed for preventing and handling the APT in [8]. In [9] Ussath *et al.* analyzed 22 different APT reports and gave an overview of the used techniques and methods. Sanders *et al*. [10][11] proposed methods of detecting lateral movement-based attacks and they also proposed a game-theoretic approach for automatic network response to an attacker that was moving laterally in an enterprise network [12]. Greco *et al*. [13][14] demonstrated how to effectively use extended finite state machine patterns to defend against lateral movement attacks. Apruzzese *et al*. [15] proposed an innovative method for detection and threat prioritization of pivoting attacks. Different from these works, our paper aims to quantitatively analyze the security of a network system under lateral movement-based attacks. Our modeling in this paper is complementary to the above studies.

Recently Hasan *et al.* [16] proposed game-theoretic modeling approach to quantitatively study the interaction between attacks and the system. Different from [16], our paper exploits state-space modeling techniques to make the transient analysis of a dynamic network system. These two kinds of modeling methods could complement each other for better analysis of system security. Table 1 presents the comparison between the existing researches of lateral movement-based attacks and our paper.

Table 1. Related works of lateral movement attacks.

Research	Quantitative Analysis	Qualitative Analysis
[7]	-	√
[8]	-	√
[9]	-	√
[10]	-	√
[11]	-	√
[12]	-	√
[13]	-	√
[14]	-	√
[15]	-	√
[16]	Steady-state	-
Our research	Transient & Dynamic	-

Note that quantitative studies have been carried out for other types of attacks. Feedback control approaches were explored to analyze the transient analysis of the storage [17] and power grid [18] under attacks. In [19] quantitative security assessment was performed by incorporating attack-defense trees and continuous-time Markov chain (CTMC) to represent attacks, defenses, and their interaction. Wagner *et al.* explored non-state space modeling techniques to quantitatively analyze the effectiveness of various host-level and network-level defensive mitigations in [20] and [21], respectively.

In this paper, we pay more attention to the transient analysis of the lateral movement-based attack. There exist researches on transient analysis of the system security. See [22]-[27], and reference therein. As in [22]-[27], our paper also explores a survivability model for transient analysis. Survivability, a transient measure, is defined to describe the ability of the system to recover a predefined service in a timely manner after the occurrence of undesired events [29]. Its quantitative analysis can help to improve the systems' capability in critical service provision when damage occurs to part of the system or when the whole system get damaged.

Let us remark that the studies in [22]-[27] only considered the case where all system parameters are constant. In [28] non-homogeneous CTMC was used to analyze transient performance of power distribution network under time-varying

system parameters. However, they only allowed the variation in model state transition rates during the system recovery. Unlike [28], our paper allows both model state number and transition rates are time-varying.

III. System Description and Model

This section first describes the network system of interest and the survivability model. Then model solutions for dynamic scenarios with time-varying transition rates are presented. Finally, the P^{2}CA approach for solving the model under both varying rates and state number is introduced.

Table 2. Notations and default settings.

Notation	Definition	Default Value
I_i	Denote that i computers are at state *INTRUDED* $(1 \le i \le m)$	-
C_i	Denote that i computers are at state *COMPROMISED* $(1 \le i \le m)$	-
$1/\delta$	Decision time	Very small
m	The number of computers in the system	6
β	Intruding rate per day	0.5
λ	Compromising rate per day	0.143
μ	Recovery rate per day	0.5
γ	Fixing rate per day	0.5
q_i	Probability that there are i *INTRUDED* computers $(1 \le i \le m)$	$1/m$

A. System Description

The network system is assumed to consist of m computers in total. Each computer is at one of four different states during the system recovery. The first state is *Non-INTRUDED*, denoting that this computer is not intruded by attackers. The second state is *INTRUDED*, denoting that this computer is intruded by attackers but there is no security damage to this computer currently. The third state is *COMPROMISED*, denoting that information exfiltration occurs in this computer after this computer is at state *INTRUDED*. The last state is *FIX*, denoting that the defensive mechanisms are deployed and the attack is defended.

Table 2 describes the variables to be used in the rest of the paper. The values of λ, μ and γ are set according to [22], respectively. Other values are set in order to highlight the effectiveness of our survivability model. We assume that when attackers begin intruding the network system, some abnormity appears in the system. Whenever this abnormity is detected, e.g. by intrusion detection systems, i computers (i=1, 2,..., m) may be at state *INTRUDED* and defensive mechanisms must start to be designed. During the process that engineers design and deploy defensive mechanisms with rate γ in order to stop the attack, attackers move laterally to intrude the left j (j=m-i, m-i-1,...,1, 0) *Non-INTRUDED* computers with rate β. Meanwhile, attackers compromise those *INTRUDED* computers (e.g., to obtain classified information in these computers) with rate λ. For an *INTRUDED* computer, if some of its secrecy is detected by the attacker, it will turn into state *COMPROMISED*. After attackers complete the handling of the detected secrecy with rate μ, the computer turns from state *COMPROMISED* to *INTRUDED*. Whenever defensive mechanisms are ready for deployment, all the computers are at state *FIX* immediately and the attack is stopped. Namely, the system is at *FIX* state. The metrics of interest in this paper are as follows:

m1) Probability that there are i computers at state *INTRUDED*/*COMPROMISED* at time instant τ when λ, β, γ, μ and/or the number of active computers are time-varying .

m2) Accumulated time that there are i computers at state *INTRUDED*/*COMPROMISED* in time interval [0, τ] when λ, β, γ, μ and/or the number of active computers are time-varying.

m3) Probability that the system is at state *FIX* at time instant τ when λ, β, γ, μ and/or the number of active computers are time-varying.

m4) Accumulated time that the system is at state *FIX* in time interval [0, τ] when λ, β, γ, μ and/or the number of active computers are time-varying.

B. Survivability Model of the Time-varying Network System

This section presents the general survivability model of the dynamic network system. Let I_i and C_i (i=1, 2, ..., m) denote the number of *INTRUDED* computers and the number *COMPROMISED* computers, respectively. $\lambda(t)$, $\beta(t)$, $\mu(t)$ and $\gamma(t)$ denote the values of λ, β, μ and γ at time instant t, respectively. All the states except the absorbing state are transient, namely, without a steady state probability. We assume that there is no security loss before detecting out the abnormal behaviors. This assumption just makes the following model formulae easy to understand. The following methods for deriving model solutions could be easily applied for the scenario without this assumption. Without loss of generality, let $q_i = 1/m$. The mean time to start the design of defensive mechanisms after the alarm of attack-related abnormity in the system is $1/\delta$, which is assumed much smaller than other values. Therefore, the initial states of the model are I_1, ..., I_m. Fig.2 illustrates the general survivability model.

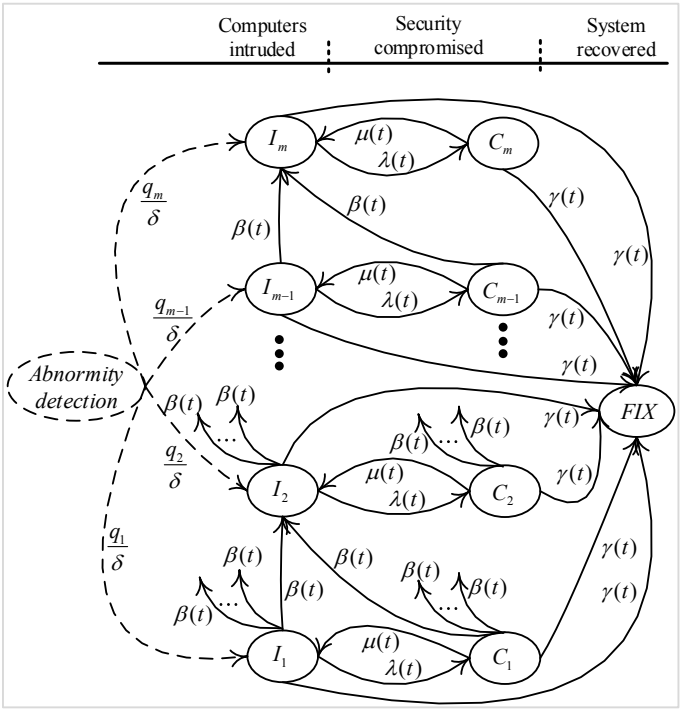

Fig.2 General survivability model.

C. *Security Analysis under Time-Varying Transition Rates*

This section derives the formulae for calculating state transient probabilities under time-varying transition rates. When the model state number is constant, the model in Fig.2 is a non-homogeneous continuous-time Markov (NHCTMC) model with an absorbing state. So we exploit PCA method [6] to derive the formulae for calculating transient probabilities, shown in the following.

Although one or more of λ, β, μ and γ are time-varying, these values are constant in a very small sub-interval, denoted $(t_{n-1}, t_n]$ and illustrated in Fig.3. t_0 denotes the time instant of starting recovery and n is non-negative integer. Then the non-homogeneous model becomes a homogeneous model in each $(t_{n-1}, t_n]$. λ_n, β_n, μ_n and γ_n are defined to denote λ, β, μ and γ in $(t_{n-1}, t_n]$, respectively. The infinitesimal generator matrix of the NHCTMC is defined as matrix Q in Fig.4, which is a $(2m+1)$-order matrix where m is the number of active computers in the network system. In this matrix, elements from the 1st column of the 2nd line to the 1st column of the $(2m+1)^{th}$ line represents fixing rates. Elements from the 2nd line to the $(m+1)^{th}$ line represent rates from state *INTRUDED* to other states. Elements from the $(m+2)^{th}$ line to the $(2m+1)^{th}$ line represent rates from state *COMPROMIZED* to other states. The key idea of our approach is as follows. Based on initial states and infinitesimal generator matrix, the state probability vector of the Markov chain at the end of the first sub-interval could be calculated. These state probabilities then form the initial probability vector for the next time sub-interval and so on. More details are given in the following.

Fig.3 System recovery timeline.

$$Q = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\gamma_n & -(m-1)\beta_n - \lambda_n - \gamma_n & \beta_n & \beta_n & \beta_n & \lambda_n & 0 & 0 & 0 \\
\gamma_n & 0 & -(m-2)\beta_n - \lambda_n - \gamma_n & \beta_n & \beta_n & 0 & \lambda_n & 0 & 0 \\
\gamma_n & 0 & 0 & -\beta_n - \lambda_n - \gamma_n & \beta_n & 0 & 0 & \lambda_n & 0 \\
\gamma_n & 0 & 0 & 0 & -\lambda_n - \gamma_n & 0 & 0 & 0 & \lambda_n \\
\gamma_n & \mu_n & \beta_n & \beta_n & \beta_n & -(m-1)\beta_n - \mu_n - \gamma_n & 0 & 0 & 0 \\
\gamma_n & 0 & \mu_n & \beta_n & \beta_n & 0 & -(m-2)\beta_n - \mu_n - \gamma_n & 0 & 0 \\
\gamma_n & 0 & 0 & \mu_n & \beta_n & 0 & 0 & -\beta_n - \mu_n - \gamma_n & 0 \\
\gamma_n & 0 & 0 & 0 & \mu_n & 0 & 0 & 0 & -\mu_n - \gamma_n
\end{bmatrix}$$

Fig.4 Infinitesimal generator matrix.

Let $\pi_{I_i}(t)$, $\pi_{C_i}(t)$ and $\pi_{FIX}(t)$ be the transient probability of model state I_i, C_i and *FIX* at time instant t, respectively. Thus, $\sum_{i=1}^{m}\left[\pi_{I_i}(t)+\pi_{C_i}(t)\right]+\pi_{FIX}(t)=1$. The initial probabilities of each state are set as $\pi_{I_i}(0)=q_i$ for each $i\in[1,m]$, $\pi_{C_i}(0)=0$ for each $i\in[1,m]$ and $\pi_{FIX}(0)=0$. Eqs.(1)-(3) describe the formulae for calculating $\pi_{I_i}(\tau)$, $\pi_{C_i}(\tau)$ and $\pi_{FIX}(\tau)$ for each τ, $t_{n-1}<\tau\le t_n$. Let $L_{I_i}(\tau)$, $L_{C_i}(\tau)$ and, $L_{FIX}(\tau)$ denote the mean accumulated time of state I_i, C_i, and *FIX* by time τ. Then, $L_{state}(\tau)=L_{state}(t_{n-1})+\int_{t_{n-1}}^{\tau}\pi_{state}(x)dx$ where $state=\{I_i, C_i, FIX\}$. Thus, we can obtain Eqs.(4)-(6) for the mean accumulated time over $(0,\tau]$ of each state.

$$\pi_{I_i}(\tau)=e^{-[(m-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}[\pi_{I_i}(t_{n-1})+\frac{\mu_n\sum_{j=1}^{i}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{\lambda_n+\mu_n}(e^{(\lambda_n+\mu_n)(\tau-t_{n-1})}-1)+\frac{(\beta_n-\mu_n)\sum_{j=1}^{i-1}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{\lambda_n+\mu_n-\beta_n}(e^{(\lambda_n+\mu_n-\beta_n)(\tau-t_{n-1})}-1)] \quad (1)$$

$$\pi_{C_i}(\tau)=e^{-[(m-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}[\pi_{C_i}(t_{n-1})+\frac{\lambda_n\sum_{j=1}^{i}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{\lambda_n+\mu_n}(e^{(\lambda_n+\mu_n)(\tau-t_{n-1})}-1)-\frac{\lambda_n\sum_{j=1}^{i-1}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{\lambda_n+\mu_n-\beta_n}(e^{(\lambda_n+\mu_n-\beta_n)(\tau-t_{n-1})}-1)] \quad (2)$$

$$\pi_{FIX}(\tau) = 1 - \sum_{i=1}^{m} \pi_{I_i}(\tau) - \sum_{i=1}^{m} \pi_{C_i}(\tau) \tag{3}$$

$$\begin{aligned} L_{I_i}(\tau) = L_{I_i}(t_{n-1}) &- \frac{\pi_{I_i}(t_{n-1})}{[(m-i)\beta_n + \lambda_n + \mu_n + \gamma_n]}[e^{-[(m-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})} - 1] - \\ &\frac{\mu_n \sum_{j=1}^{i}[\pi_{I_j}(t_{n-1}) + \pi_{C_j}(t_{n-1})]}{(\lambda_n+\mu_n)[(m-i)\beta_n+\gamma_n]}[e^{(-[(m-i)\beta_n+\gamma_n])(\tau-t_{n-1})} - 1] + \\ &\frac{\mu_n \sum_{j=1}^{i}[\pi_{I_j}(t_{n-1}) + \pi_{C_j}(t_{n-1})]}{(\lambda_n+\mu_n)[(m-i)\beta_n+\lambda_n+\mu_n+\gamma_n]}[e^{-[(m-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})} - 1] - \\ &\frac{(\beta_n-\mu_n)\sum_{j=1}^{i-1}[\pi_{I_j}(t_{n-1}) + \pi_{C_j}(t_{n-1})]}{[(m-i+1)\beta_n+\gamma_n](\lambda_n+\mu_n-\beta_n)}[e^{-[(m-i+1)\beta_n+\gamma_n](\tau-t_{n-1})} - 1] + \\ &\frac{(\beta_n-\mu_n)\sum_{j=1}^{i-1}[\pi_{I_j}(t_{n-1}) + \pi_{C_j}(t_{n-1})]}{[(m-i+1)\beta_n+\lambda_n+\mu_n+\gamma_n](\lambda_n+\mu_n-\beta_n)}[e^{-[(m-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})} - 1] \end{aligned} \tag{4}$$

$$\begin{aligned} L_{C_i}(\tau) = L_{C_i}(t_{n-1}) &- \frac{\pi_{C_i}(t_{n-1})}{[(m-i)\beta_n + \lambda_n + \mu_n + \gamma_n]}(e^{-[(m-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})} - 1) - \\ &\frac{\lambda_n \sum_{j=1}^{i}[\pi_{I_j}(t_{n-1}) + \pi_{C_j}(t_{n-1})]}{(\lambda_n+\mu_n)[(m-i)\beta_n+\gamma_n]}(e^{-[(m-i)\beta_n+\gamma_n](\tau-t_{n-1})} - 1) + \\ &\frac{\lambda_n \sum_{j=1}^{i}[\pi_{I_j}(t_{n-1}) + \pi_{C_j}(t_{n-1})]}{(\lambda_n+\mu_n)[(m-i)\beta_n+\lambda_n+\mu_n+\gamma_n]}(e^{-[(m-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})} - 1) - \\ &\frac{\lambda_n \sum_{j=1}^{i-1}[\pi_{I_j}(t_{n-1}) + \pi_{C_j}(t_{n-1})]}{(\lambda_n+\mu_n-\beta_n)[(m-i)\beta_n+\lambda_n+\mu_n+\gamma_n]}(e^{-[(m-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})} - 1) + \\ &\frac{\lambda_n \sum_{j=1}^{i-1}[\pi_{I_j}(t_{n-1}) + \pi_{C_j}(t_{n-1})]}{(\lambda_n+\mu_n-\beta_n)[(m-i+1)\beta_n+\gamma_n]}(e^{-[(m-i+1)\beta_n+\gamma_n](\tau-t_{n-1})} - 1) \end{aligned} \tag{5}$$

$$L_{FIX}(\tau) = L_{FIX}(t_{n-1}) + \tau - t_{n-1} - \sum_{i=1}^{m}[L_{I_i}(\tau) - L_{I_i}(t_{n-1})] - \sum_{i=1}^{m}[L_{C_i}(\tau) - L_{C_i}(t_{n-1})] \tag{6}$$

D. P²CA for Transient Security Analysis under Time-varying State Number and Transition Rates

Section III.C assumes that the number of model states is constant during the system recovery. However, this assumption may not be true because of the variation in the number of active computers. This section presents the P²CA approach for deriving the formulae when both transition rates and the model state number are not constant by extending the formulae previously introduced.

P²CA divides the recovery process into phases. The state number is different in two adjoining phases. Each phase has the constant number of model states and consists of one or more time slots, in each of which all system parameters are constant. See Fig.5 where Phase k comprises two time slots, $(t_{n-1}, t_n]$ and $(t_n, t_{n+1}]$. Fig.6 is an example illustrating the model variation in two adjoining phases. Phase k denotes the model of the system with four active computers. Later, two computers are damaged and thus state I_4, C_4, I_3 and C_3 are deleted. Phase k+1 denotes the new model of the system with two active computers.

When the system moves from a time slot to the next, there are two cases of variation in the model state number: increase or decrease. We present P²CA in Phase k for each case. Without loss of generality, $(t_{n-1}, t_n]$ is assumed to be the first time slot of Phase k. As in Section III.C, λ_n, β_n, μ_n and γ_n denote the corresponding values in $(t_{n-1}, t_n]$.

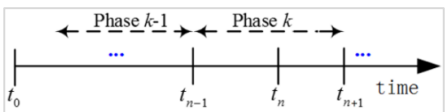

Fig.5 System recovery timeline.

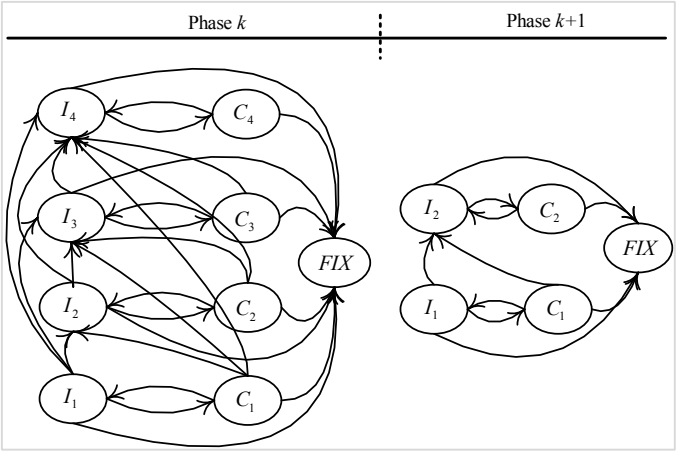

Fig.6 An example for two phases under P²CA

The **first case** is when the model state number increases from $2m+1$ to $2(m+x)+1$ because new x computers are added in the system, compared to that in the last time slot. Let $\pi_{Ia_i}(\tau)/L_{Ia_i}(\tau), \pi_{Ca_i}(\tau)/L_{Ca_i}(\tau),$ and $\pi_{FIXa}(\tau)/L_{FIXa}(\tau)$ be defined as the probability/accumulated time at/by time τ that there are i computers at state *INTRUDED, COMPROMISED* and the system is at state *FIX*, respectively, after x computers are added in the system. All $\pi_{I_i}(t_{n-1})$ and $\pi_{C_i}(t_{n-1}), (i=m+1,...,m+x)$ are set to zero. With these definitions, we obtain Eqs.(7)-(12) for calculating the transient probabilities and accumulated time for τ in the first time slot of this phase. Eqs.(1)-(6) are used for the subsequent time slots of this phase.

The **second case** is when the number of model states decreases from $2m+1$ to $2(m-y)+1$. Here, y denotes the number of damaged computers, compared to that in the last time slot. Again, let $\pi_{Ib_i}(\tau)$ / $L_{Ib_i}(\tau)$, $\pi_{Cb_i}(\tau)$ / $L_{Cb_i}(\tau)$, and $\pi_{FIXb}(\tau)$ / $L_{FIXb}(\tau)$ be defined as the probability/accumulated time at/by time τ that there are i computers at state *INTRUDED, COMPROMISED*, and the system is at state *FIX*, respectively, after y computers are damaged in the system. Under P²CA , if y computers are damaged ($y<m$), the probability $\sum_{i=m-y+1}^{m}\pi_{Ib_i}(\tau)$ will be added into $\pi_{Ib_{(m-y)}}(\tau)$ and then it becomes the initial probability of this model state in the second phase. Similarly, the probability $\sum_{i=m-y+1}^{m}\pi_{Cb_i}(\tau)$ will be added into $\pi_{Cb_{(m-y)}}(\tau)$. Thus, in Fig.6 the probability summation of I_4 and I_3 is added to that of state I_2, and the probability summation of C_4 and C_3 is added to that of state C_2. With these definitions, we obtain Eqs.(13)-(18) for calculating the transient probabilities and accumulated time for τ in the first time slot of this phase. Eqs.(1)-(6) are used for the subsequent time slots of this phase. Note that in Eqs.(7)-(12) or Eqs.(13)-(18), when $i \le m$ or $i < m-y$, the value of m in corresponding $\pi_{I_i}(\tau)$ or $\pi_{C_i}(\tau)$ are $m+x$ or $m-y$, respectively.

$$\pi_{Ia_i}(\tau)=\begin{cases}\pi_{I_i}(\tau) & ,i\le m\\ e^{-[(m+x-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}[\dfrac{\mu_n\sum_{j=1}^{m}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{\lambda_n+\mu_n}(e^{(\lambda_n+\mu_n)(\tau-t_{n-1})}-1)+ & \\ \qquad\dfrac{(\beta_n-\mu_n)\sum_{j=1}^{m}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{\lambda_n+\mu_n-\beta_n}(e^{(\lambda_n+\mu_n-\beta_n)(\tau-t_{n-1})}-1)] & ,m<i\le m+x\end{cases} \quad (7)$$

$$\pi_{Ca_i}(\tau)=\begin{cases}\pi_{C_i}(\tau) & ,i\le m\\ e^{-[(m+x-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}[\dfrac{\lambda_n\sum_{j=1}^{m}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{\lambda_n+\mu_n}(e^{(\lambda_n+\mu_n)(\tau-t_{n-1})}-1)- & \\ \qquad\dfrac{\lambda_n\sum_{j=1}^{m}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{\lambda_n+\mu_n-\beta_n}(e^{(\lambda_n+\mu_n-\beta_n)(\tau-t_{n-1})}-1)] & ,m<i\le m+x\end{cases} \quad (8)$$

$$\pi_{FIXa}(\tau)=1-\sum_{i=1}^{m+x}\pi_{Ia_i}(\tau)-\sum_{i=1}^{m+x}\pi_{Ca_i}(\tau) \quad (9)$$

$$L_{Ia_i}(\tau)=\begin{cases} L_{I_i}(\tau) & ,i\le m \\ L_{Ia_i}(t_{n-1})-\dfrac{\mu_n\sum_{j=1}^{m}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{(\lambda_n+\mu_n)[(m+x-i)\beta_n+\gamma_n]}[e^{-[(m+x-i)\beta_n+\gamma_n](\tau-t_{n-1})}-1]+ \\ \dfrac{\mu_n\sum_{j=1}^{m}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{(\lambda_n+\mu_n)[(m+x-i)\beta_n+\lambda_n+\mu_n+\gamma_n]}[e^{-[(m+x-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}-1]- \\ \dfrac{(\beta_n-\mu_n)\sum_{j=1}^{m}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{(\lambda_n+\mu_n-\beta_n)[(m+x-i+1)\beta_n+\gamma_n]}[e^{(-[(m+x-i+1)\beta_n+\gamma_n])(\tau-t_{n-1})}-1]+ \\ \dfrac{(\beta_n-\mu_n)\sum_{j=1}^{m}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{(\lambda_n+\mu_n-\beta_n)[(m+x-i)\beta_n+\lambda_n+\mu_n+\gamma_n]}[e^{-[(m+x-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}-1] & ,m<i\le m+x \end{cases} \tag{10}$$

$$L_{Ca_i}(\tau)=\begin{cases} L_{C_i}(\tau) & ,i\le m \\ L_{Ca_i}(t_{n-1})-\dfrac{\lambda_n\sum_{j=1}^{m}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{(\lambda_n+\mu_n)[(m+x-i)\beta_n+\gamma_n]}(e^{-[(m+x-i)\beta_n+\gamma_n](\tau-t_{n-1})}-1)+ \\ \dfrac{\lambda_n\sum_{j=1}^{m}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{(\lambda_n+\mu_n)[(m+x-i)\beta_n+\lambda_n+\mu_n+\gamma_n]}(e^{-[(m+x-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}-1)+ \\ \dfrac{\lambda_n\sum_{j=1}^{m}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{(\lambda_n+\mu_n-\beta_n)[(m+x-i)\beta_n+\lambda_n+\mu_n+\gamma_n]}(e^{-[(m+x-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}-1)- \\ \dfrac{\lambda_n\sum_{j=1}^{m}[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})]}{(\lambda_n+\mu_n-\beta_n)[(m+x-i+1)\beta_n+\gamma_n]}(e^{-[(m+x-i+1)\beta_n+\gamma_n](\tau-t_{n-1})}-1) & ,m<i\le m+x \end{cases} \tag{11}$$

$$L_{FIXa}(\tau)=L_{FIXa}(t_{n-1})+\tau-t_{n-1}-\sum_{i=1}^{m+x}[L_{Ia_i}(\tau)-L_{Ia_i}(t_{n-1})]-\sum_{i=1}^{m+x}[L_{Ca_i}(\tau)-L_{Ca_i}(t_{n-1})] \tag{12}$$

$$\pi_{Ib_i}(\tau)=\begin{cases} \pi_{I_i}(\tau) & ,i<m-y \\ e^{-[(m-y-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}\left[\sum_{j=m-y}^{m}\pi_{I_j}(t_{n-1})+\dfrac{\mu_n\sum_{j=1}^{m}\left[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})\right]}{\lambda_n+\mu_n}\left(e^{(\lambda_n+\mu_n)(\tau-t_{n-1})}-1\right)\right.+ & ,i=m-y \\ \left.\dfrac{(\beta_n-\mu_n)\sum_{j=1}^{m-y-1}\left[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})\right]}{\lambda_n+\mu_n-\beta_n}\left(e^{(\lambda_n+\mu_n-\beta_n)(\tau-t_{n-1})}-1\right)\right] & \end{cases} \tag{13}$$

$$\pi_{Cb_i}(\tau)=\begin{cases}\pi_{C_i}(\tau) & ,i<m-y \\ e^{-[(m-y-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}\left[\sum_{j=m-y}^{m}\pi_{C_j}(t_{n-1})+\frac{\lambda_n\sum_{j=1}^{m}\left[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})\right]}{\lambda_n+\mu_n}(e^{(\lambda_n+\mu_n)(\tau-t_{n-1})}-1)-\right. & ,i=m-y \\ \left.\frac{\lambda_n\sum_{j=1}^{m-y-1}\left[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})\right]}{\lambda_n+\mu_n-\beta_n}(e^{(\lambda_n+\mu_n-\beta_n)(\tau-t_{n-1})}-1)\right] & \end{cases} \tag{14}$$

$$\pi_{FIXb}(\tau)=1-\sum_{i=1}^{m-y}\pi_{Ib_i}(\tau)-\sum_{i=1}^{m-y}\pi_{Cb_i}(\tau) \tag{15}$$

$$L_{Ib_i}(\tau)=\begin{cases}L_{I_i}(\tau) & ,i<m-y \\ L_{Ib_i}(t_{n-1})-\frac{\sum_{j=m-y}^{m}\pi_{I_j}(t_{n-1})}{[(m-y-i)\beta_n+\lambda_n+\mu_n+\gamma_n]}(e^{-[(m-y-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}-1)- & ,i=m-y \\ \frac{\mu_n\sum_{j=1}^{m}\left[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})\right]}{[(m-y-i)\beta_n+\gamma_n](\lambda_n+\mu_n)}(e^{-[(m-y-i)\beta_n+\gamma_n](\tau-t_{n-1})}-1)+ & \\ \frac{\mu_n\sum_{j=1}^{m}\left[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})\right]}{[(m-y-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\lambda_n+\mu_n)}(e^{-[(m-y-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}-1)- & \\ \frac{(\beta_n-\mu_n)\sum_{j=1}^{m-y-1}\left[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})\right]}{[(m-y-i+1)\beta_n+\gamma_n](\lambda_n+\mu_n-\beta_n)}(e^{-[(m-y-i+1)\beta_n+\gamma_n](\tau-t_{n-1})}-1)+ & \\ \frac{(\beta_n-\mu_n)\sum_{j=1}^{m-y-1}\left[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})\right]}{[(m-y-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\lambda_n+\mu_n-\beta_n)}(e^{-[(m-y-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}-1) & \end{cases} \tag{16}$$

$$L_{Cb_i}(\tau)=\begin{cases}L_{C_i}(\tau) & ,i<m-y \\ L_{Cb_i}(t_{n-1})-\frac{\sum_{j=m-y}^{m}\pi_{C_j}(t_{n-1})}{[(m-y-i)\beta_n+\lambda_n+\mu_n+\gamma_n]}(e^{-[(m-y-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}-1)- & ,i=m-y \\ \frac{\lambda_n\sum_{j=1}^{m}\left[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})\right]}{[(m-y-i)\beta_n+\gamma_n](\lambda_n+\mu_n)}(e^{-[(m-y-i)\beta_n+\gamma_n](\tau-t_{n-1})}-1)+ & \\ \frac{\lambda_n\sum_{j=1}^{m}\left[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})\right]}{[(m-y-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\lambda_n+\mu_n)}(e^{-[(m-y-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}-1)+ & \\ \frac{\lambda_n\sum_{j=1}^{m-y-1}\left[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})\right]}{[(m-y-i+1)\beta_n+\gamma_n](\lambda_n+\mu_n-\beta_n)}(e^{-[(m-y-i+1)\beta_n+\gamma_n](\tau-t_{n-1})}-1)- & \\ \frac{\lambda_n\sum_{j=1}^{m-y-1}\left[\pi_{I_j}(t_{n-1})+\pi_{C_j}(t_{n-1})\right]}{[(m-y-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\lambda_n+\mu_n-\beta_n)}(e^{-[(m-y-i)\beta_n+\lambda_n+\mu_n+\gamma_n](\tau-t_{n-1})}-1) & \end{cases} \tag{17}$$

$$L_{FIXb}(\tau)=L_{FIXb}(t_{n-1})+\tau-t_{n-1}-\sum_{i=1}^{m-y}[L_{Ib_i}(\tau)-L_{Ib_i}(t_{n-1})]-\sum_{i=1}^{m-y}[L_{Cb_i}(\tau)-L_{Cb_i}(t_{n-1})] \tag{18}$$

It is obvious that Eqs.(1)-(6) are a special case of Eqs.(7)-(12) when x=0 or Eqs.(13)-(18) when y=0. We now use an example to illustrate how to use Eqs.(1)-(18). Fig.7 shows the variation in system parameters $[t_0, t_4]$. According to P^{2}CA approach, there are two phases in $[t_0, t_4]$, one time slot $[t_0, t_1]$ in Phase 1 and three time slots $(t_1, t_2], (t_2, t_3]$ and $(t_3, t_4]$ in Phase 2. In each time slot, all system parameters are constant. Then Eqs.(1)-(18) are suitable for this case. In $[t_0, t_1]$, Eqs.(1)-(6) with (λ_1, γ_1) are used for security analysis. The initial state probabilities are $\pi_{I_i}(t_0)$, $\pi_{C_i}(t_0)$ and $\pi_{FIX}(t_0)$. Since $(t_1, t_2]$ is the first time slot of Phase 2, Eqs.(7)-(12) or Eqs.(13)-(18) with (λ_1, γ_1) are used to analyze the system, depending the model state number increase or decrease. Eqs.(1)-(6) with (λ_1, γ_2) and (λ_2, γ_2) are used to analyze the transient probabilities in $(t_2, t_3]$ and $(t_3, t_4]$, respectively.

Based on previous analysis and Eqs.(1)-(18), we can obtain computation formulae of m1-m4 mentioned in Section III.B, shown in Table 3. In the time slot where m is constant, the first column is applied. If m is increased, the second column is applied. Otherwise, the third column is applied.

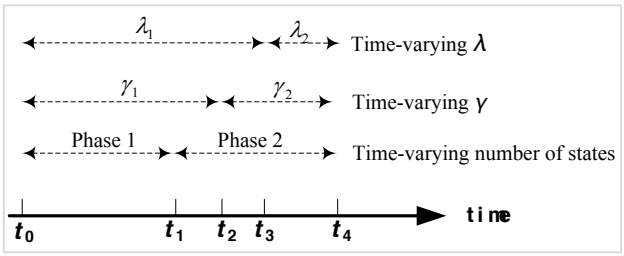

Fig.7 Time axis of dynamic model under both time-varying state number and transition rates

Table 3. Formulae for calculating metrics

m is constant	m is increased	m is decreased
m1= $\pi_{I_i}(\tau) / \pi_{C_i}(\tau)$	m1= $\pi_{Ia_i}(\tau) / \pi_{Ca_i}(\tau)$	m1= $\pi_{Ib_i}(\tau) / \pi_{Cb_i}(\tau)$
m2= $L_{I_i}(\tau) / L_{C_i}(\tau)$	m2= $L_{Ia_i}(\tau) / L_{Ca_i}(\tau)$	m2= $L_{Ib_i}(\tau) / L_{Cb_i}(\tau)$
m3= $\pi_{FIX}(\tau)$	m3= $\pi_{FIXa}(\tau)$	m3= $\pi_{FIXb}(\tau)$
m4= $L_{FIX}(\tau)$	m4= $L_{FIXa}(\tau)$	m4= $L_{FIXb}(\tau)$

IV. Numerical Analysis and Discussion

This section presents detailed evaluations of our model. All λ, β, μ, and γ are exponentially distributed but time-varying. Table 2 describes the default settings. Numerical analysis under static (i.e., not varying transition rates) settings is first presented in Section IV.A. Section IV.B and IV.C present results under varying transition rates and discussions in order to demonstrate the effectiveness of our model, compared to the model with constant transition rates. Section IV.D and IV.E present results under varying number of model states in order to demonstrate the capability of P^{2}CA in evaluating the transient security in this scenario.

A. Static Scenario

The system is assumed to be consisted of six computers and $q_i = 1/6$ for each i at the beginning. The probability and the accumulated time that there are i computers at state *INTRUDED*, at state *COMPROMISED*, or the system at state *FIX* are depicted in Fig.8-Fig.9, Fig. 10-Fig.11, and Fig.12-Fig.13, respectively. From these results, we observe that:

1) The probability that i computers are intruded increases with the increasing i, shown in Fig.8. The probability of six computers and five computers at state *INTRUDED* increases in the first 0.9 days and 0.3 days, and then decreases, respectively. However, the other probabilities decrease from the beginning. In addition, the probability of six computers at state *INTRUDED* is larger than other probabilities. The reason is that each $\pi_{I_i}(t), i \in [1,5]$, will contribute $\pi_{I_6}(t)$. Fig.9 shows that the accumulated time that i computers are intruded also increases with the increasing i. By the 10th day, it is respectively 0.053 days, 0.074 days, 0.110 days, 0.181 days, 0.355 days, 1.010 days for each i Î[1,6]. Similarly, each $\pi_{C_i}(t), i \in [1,5]$, makes contribution to $\pi_{C_6}(t)$ and then $\pi_{C_6}(t)$ is larger than other $\pi_{C_i}(t)$.

2) The probability of i computers at state *COMRPOMISED* will increase a little first and then keep decreasing no matter how many computers are compromised, shown in Fig.10. We could see that the maximum compromising probability is respectively 0.0026, 0.0034, 0.0047, 0.0071, 0.0126, 0.0312 for each i at the 0.299th day, 0.385th day, 0.525th day, 0.725th day, 1.135th day, 2.055th day. Fig.11 indicates that the accumulated time of i computers at state *COMPROMISED* also increases with the increasing i. By the 10th day, the accumulated compromising time is respectively 0.002 days, 0.004 days, 0.006 days, 0.013 days, 0.034 days, 0.143 days for each i. The reason is same as in the last section. Note that the probability and accumulated time of i computers at state *INTRUDED* are much larger than that at state *COMPROMISED*.

3) The probability that the system is fixed will keep increasing with the increase of time and finally it will be close to 1, shown in Fig.12. Fig.13 shows that by the 10th days the accumulated time of the system at state *FIXED* is 8.013 days. Namely, the accumulated time that the system is vulnerable is 1.987 days in the whole 10 days.

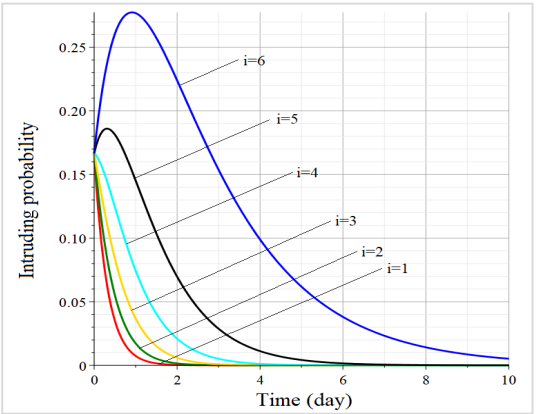

Fig.8 Intruding probability.

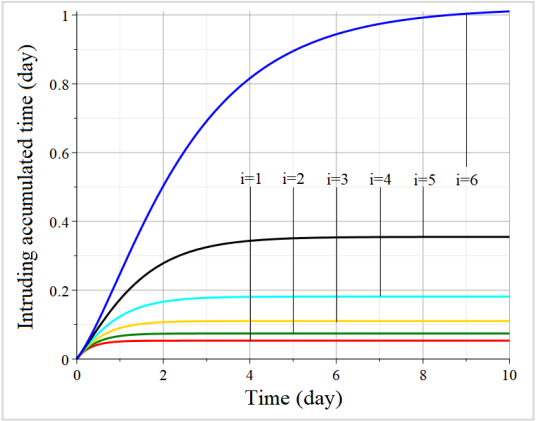

Fig.9 Intruding accumulated time.

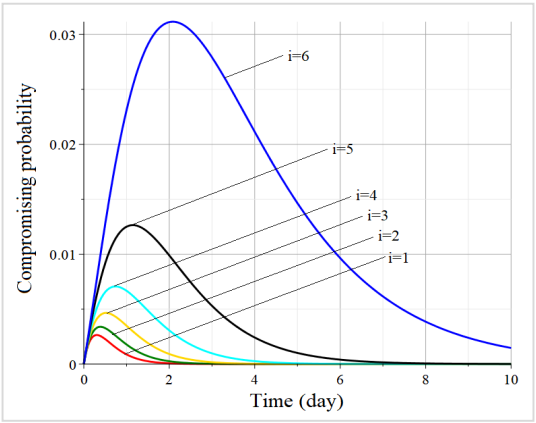

Fig.10 Compromising probability.

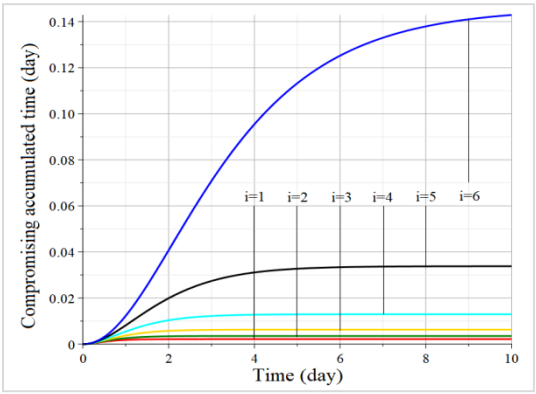

Fig.11 Compromising accumulated time.

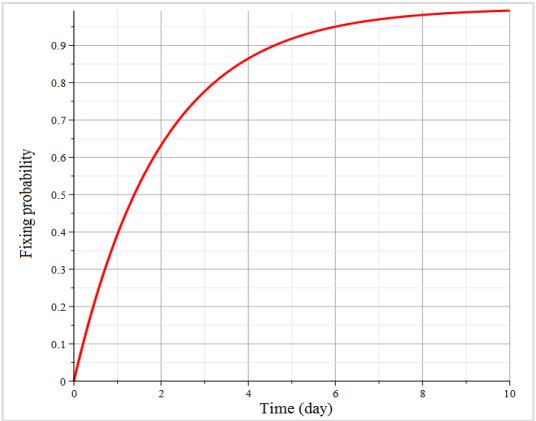

Fig.12 Fixing probability.

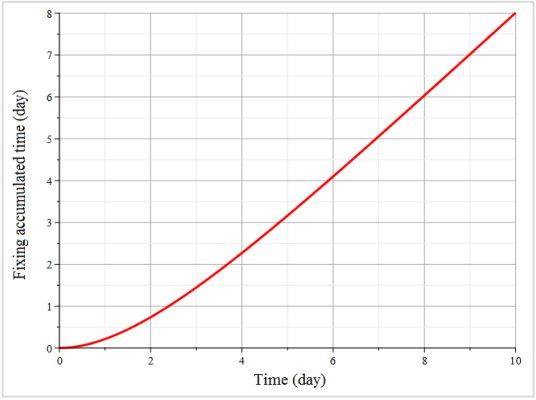

Fig.13 Fixing accumulated time .

B. *Comparison between NHCTMC and CTMC Models under Varying Intruding Rate*

This section shows the comparison of CTMC and NHCTMC model under varying intruding rate β . The settings of the other parameters are the same as in Section IV.A. The system still consists of 6 computers and $q_i = 1/6$ for each i at the beginning. Fig.14 shows β variation over time where solid line represents time-varying β and dashed line represents average β . In the first five days, $1/\beta$ is 2.0 days. Then it changes to 1.5 days. At the 8th day, it changes to 1.3 days. The average of these values is 1.66 days, denoted by the dashed line in Fig.14. Without loss of generality, we only present the transient probabilities of four computers at state *INTRUDED* and *COMPROMISED*, namely, *i*=4 (see Fig.15 and Fig.16, respectively). Fig. 17-Fig.18 show the results about the system at state *FIX*. In the following figures, "CTMC" denotes the results under average β and "NHCTMC" denotes the results under time-varying β .

From Fig.15-Fig.16, we can observe that the probability that four computers in the system are at state *INTRUDED* or *COMPROMISED* under "NHCTMC" is a bit larger than that under "CTMC". In Fig.17 the probability of the system at state *FIXED* under "NHCTMC" is a bit lower than that under "CTMC". Furthermore, the probabilities of "NHCTMC" that four computers are intruded and the system is fixed also have some fluctuation compared with that of "CTMC". From Fig.18 we can see that the accumulated time for "CTMC" and "NHCTMC" in 10 days is 8.014 days and 7.476 days, respectively. These results indicate that CTMC model could not capture some transient behaviors occurring in the system recovery, compared to NHCTMC.

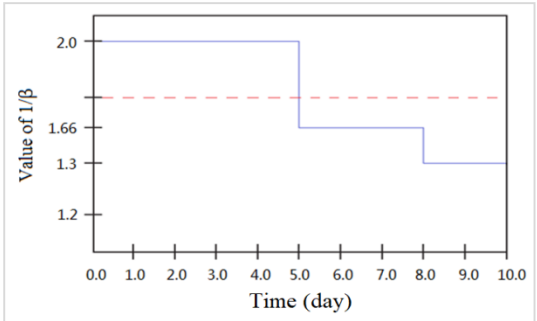

Fig.14 Intruding rate variation.

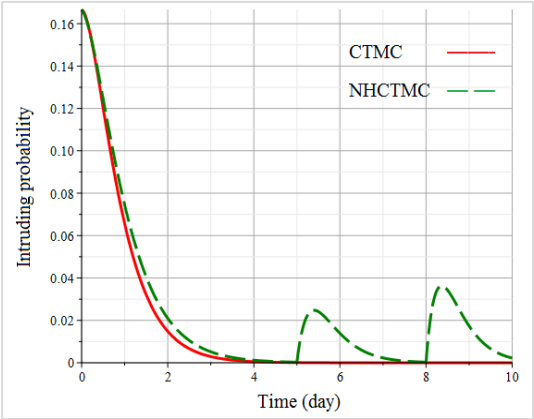

Fig.15 Intruding probability of I_4 under varying intruding rate.

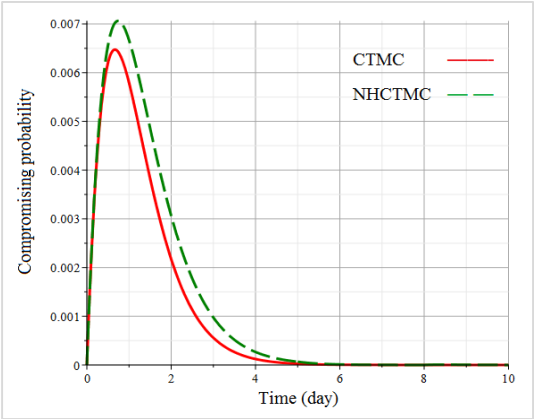

Fig.16 Compromising probability of I_4 under varying intruding rate.

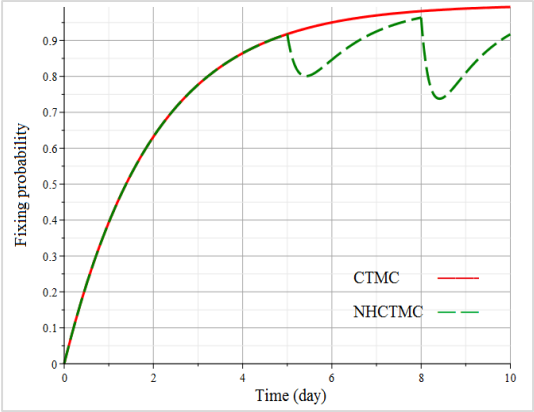

Fig.17 Fixing probability under varying intrusion rate.

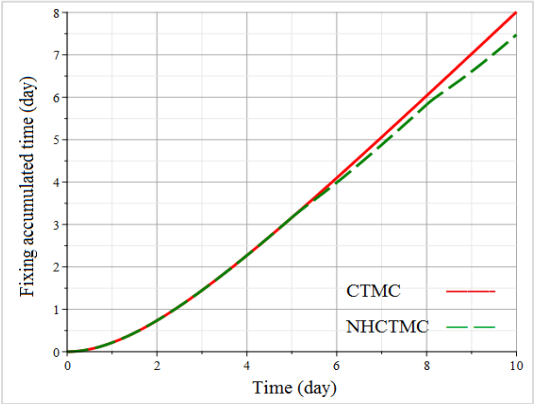

Fig.18 Fixing accumulated time under varying intrusion rate.

C. *Comparison between NHCTMC and CTMC Models under Varying Fixing Rate*

This section compares CTMC and NHCTMC model under varying fixing rate γ. The settings of the other parameters are as same as before. Fig.19 shows γ variation over time where solid line represents time-varying γ and dashed line represents average γ. Similarly, in the first five days, $1/\gamma$ is 2.0 days. Then it changes to 1.5 days. At the 8th day, it changes to 1.3 days. The average of these values is 1.66 days. As in Section IV.B, we only present the transient probabilities of four computers at state *INTRUDED* and *COMPROMISED* in Fig.20 and Fig.21, respectively. Fig.22-Fig.23 show the results about the system at state *FIX*. Here, "CTMC" and "NHCTMC" denote the results under average γ and time-varying γ, respectively.

These figures indicate that there are still some differences between "CTMC" and "NHCTMC". The probability of four computers at state *INTRUDED* or *COMPROMISED* under of "NHCTMC" is a bit larger than that of "CTMC" with increasing fixing rate γ, and the probability of "NHCTMC" that the system is fixed is a bit lower than that of "CTMC". Fig.23 shows that the accumulated time for "CTMC" and "NHCTMC" in 10 days is 8.348 days and 8.049 days, respectively. Compared with Section IV.B, we can observe that different time-varying values cause different influences to the system. Although the variation of β and γ are the same, as depicted in Fig.14 and Fig.19, their impacts on the system performance are quite different: increasing β will cause more influence than increasing γ. Furthermore, the results of Fig.20-Fig.23 confirm that some transient behaviors in the system recovery cannot be captured by the CTMC model and thus, an CTMC model is unsuitable for this purpose.

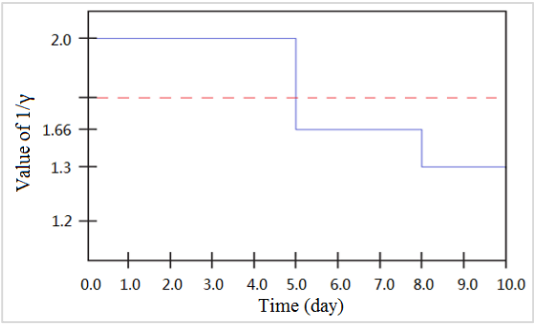

Fig.19 Fixing rate variation.

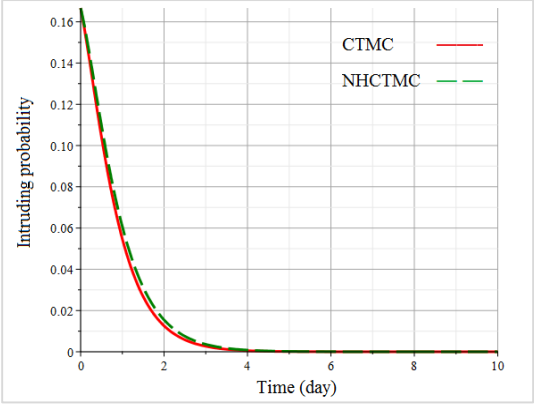

Fig.20 Intruding probability of I_4 under varying fixing rate.

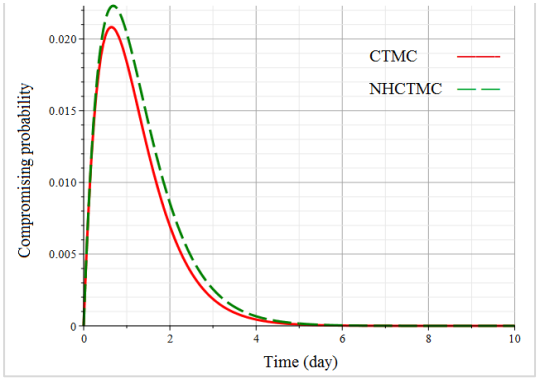

Fig.21 Compromising probability of I_4 under varying fixing rate.

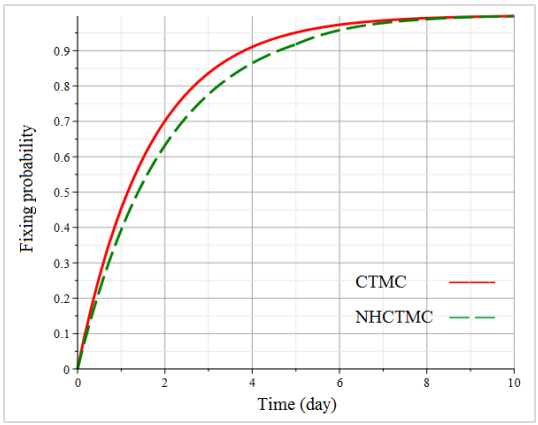

Fig.22 Fixing probability under varying fixing rate.

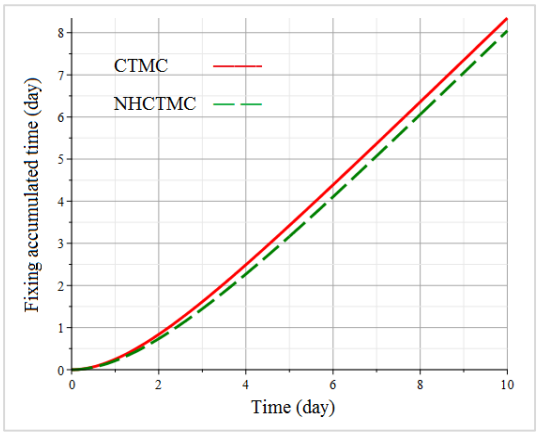

Fig.23 Fixing accumulated time under varying fixing rate.

D. Comparison when Active Computer Number Increases

This section aims to investigate the effect of adding x computers during the system recovery on the system performance. Six computers are assumed in the system when attack-related abnormity is detected and $q_i = 1/6$ for each i. At the 3rd day, one new computer (x=1) is added in the system, thus our model will turn from current phase to the next phase where the number of model states increases. Fig.24-Fig.27 show the results in the second phase, namely after the 3rd day. "Seven active computers" denotes the results that one computer is added in the system. "Six active computers" denotes the results that no additional computers are added in the system, which is shown in order to demonstrate the capability of P^{2}CA in capturing the system behaviors. Without loss generality, we only show the results that four computers are intruded/compromised and the system is fixed.

These results of Fig.24 and Fig.25 indicate that when new computer is added in the system, the probability of "Seven active computers" is a bit lower than that of "Six active computers". The probability that the system is fixed will decrease a little in the first 3.3 days to its minimum value 0.725, shown in Fig.26. Then it will keep increasing and finally it will be close to that of "Six active computers". It suggests that although additional computer is added, the impact caused to the fixing transient probability is reduced. Fig.27 demonstrates that the accumulated time that the system is fixed by the 10th day is 8.013 days and 7.873 days under "Six active computers" and "Seven active computers", respectively. Fig.24-Fig.27 illustrate that when some computers in the system are damaged, some influences will be caused to the system.

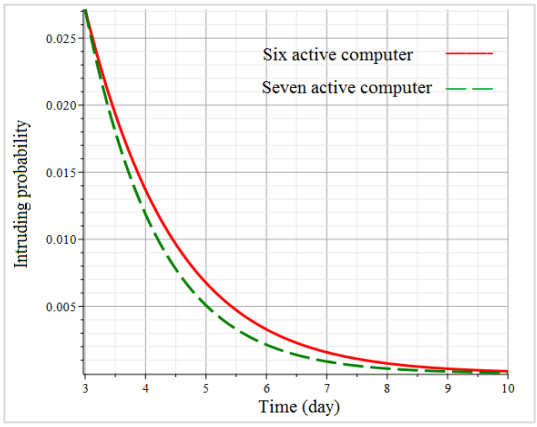

Fig.24 Intruding probability of I_4 under increasing number of active computers.

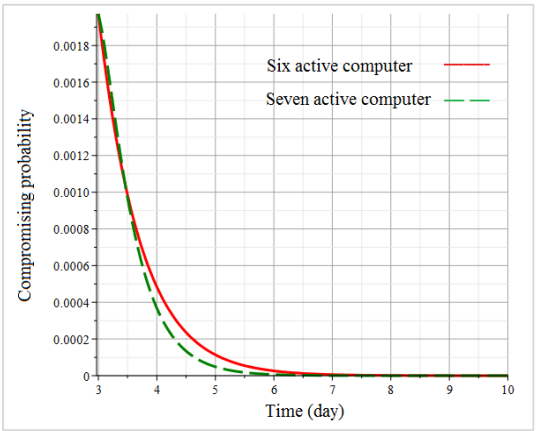

Fig.25 Compromising probability of I_4 under increasing number of active computers.

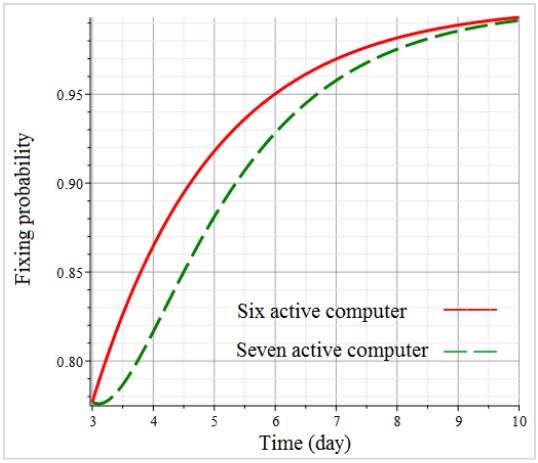

Fig.26 Fixing probability under increasing number of active computers.

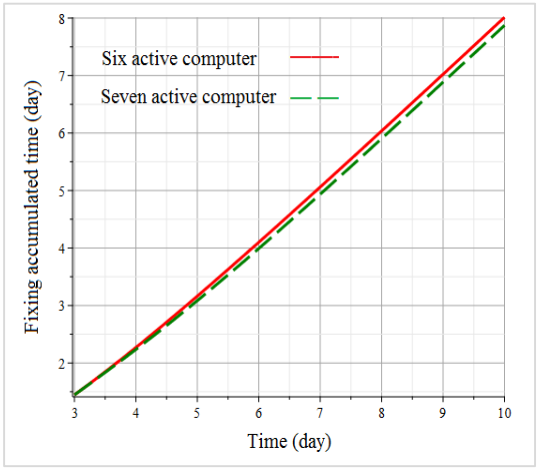

Fig.27 Fixing accumulated time under increasing number of active computers.

E. Comparison when Active Computer Number Decreases

This section aims to investigate the effect of removing y active computers during the system recovery on the system performance. Six computers are assumed in the system when attack-related abnormity is detected and $q_i = 1/6$ for each i. At the 3rd day, one computer (y=1) among them are damaged, thus our model will turn from current phase to the next phase where the number of model states decreases. Fig.28-Fig.31 show the results of the second phase, namely after the 3rd day. "Five active computers" denotes the results of one damaged computer. "Six active computers" denotes the results of no damaged computer, which is shown in order to demonstrate the capability of P^{2}CA in capturing the system behaviors. Without loss generality, we only show the results that four computers are intruded/compromised and the system is fixed.

From these figures we can observe that when one computer is damaged, the probability of "Five active computers" is a bit larger than that of "Six active computers", and the probability that the system is fixed is a bit lower than that of "Six active computers". The accumulated time of "Six active computers" and "Five active computers" that the system is fixed by the 10th day is 8.013 days and 7.955 days, respectively. Compared with analyses in Section IV.D, we can observe that increasing number of active computers in the system recovery may cause more influences to the system than decreasing number of active computers.

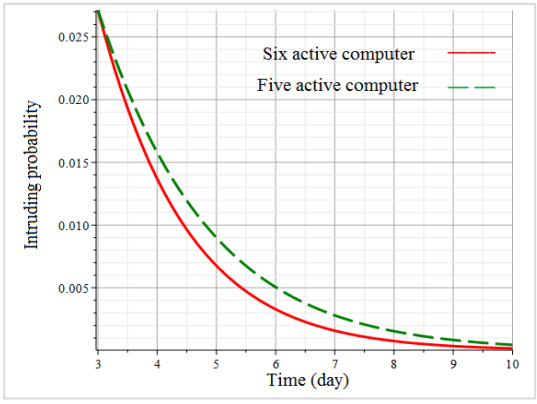

Fig.28 Intruding probability of I_4 under decreasing number of active computers.

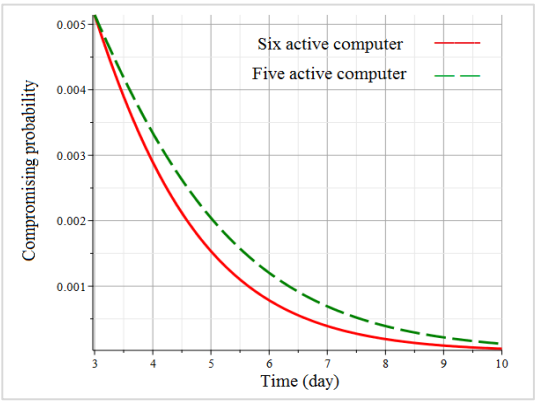

Fig.29 Compromising probability of I_4 under decreasing number of active computers.

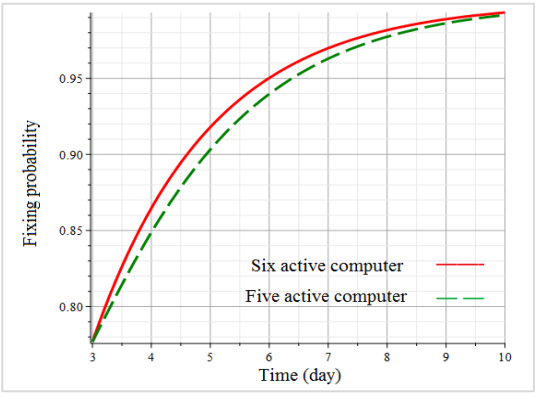

Fig.30 Fixing probability under decreasing number of active computers.

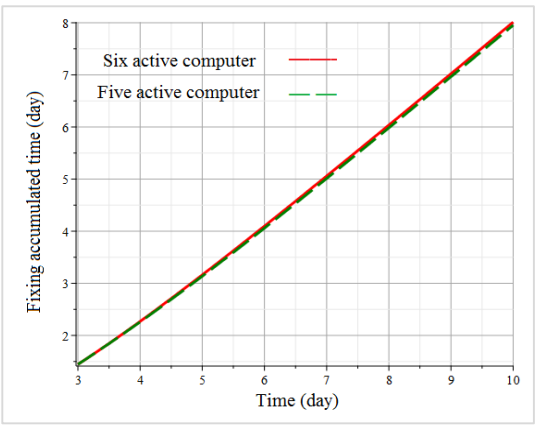

Fig.31 Fixing accumulated time under decreasing number of active computers.

V. Conclusion and Future Work

This paper applies model-based techniques to analyze the transient security of a dynamic network system under lateral movement-based attacks from the time that attack-related abnormity in the system is detected until mechanisms are designed and deployed to defend against the attacks. A survivability model is constructed and a phased piecewise constant approximation approach is proposed to derive different formulae for calculating state transient probabilities and accumulated time for scenarios of varying model state transition rates and model state numbers. Numerical analyses are also presented for evaluating the impact of various system parameters on system security dynamically.

In this paper, we assume that all intruding rates or compromising rates in the model are equal. However in real systems there exist heterogeneity in computers and therefore these rates might be different. One future work is the investigation of how to deal with such heterogeneity. In addition, it is valuable to extend our security model to a more general case so that more complex scenarios can be analyzed.

References

[1] Ross Brewer: Advanced persistent threats: minimising the damage. Elsevier Network Security, 2014.

[2] Aditya K. Sood, Richard J. Enbody: Targeted Cyberattacks: A Superset of Advanced Persistent Threats. IEEE Security & Privacy 11(1): 54-61 (2013).

[3] Karl Rauscher: Writing the rules of networkwar. IEEE Spectrum 50: 30-32 (2013).

[4] Kaspersky Lab: APT Trends report. Q1 2017. https://securelist.com/apt-trends-report-q1-2017/78169/.

[5] L. Takacs: Introduction to the Theory of Queues. Oxford University Press book (1962).

[6] Kishor Trivedi, Andreaw Bobbio, "Reliability and Availability Engineering", Cambridge University Press, August 2017.

[7] Ping Chen, Lieven Desmet, Christophe Huygens: A Study on Advanced Persistent Threats. Communications and Multimedia Security 2014: 63-72 (2014).

[8] Inkyung Jeun, Youngsook Lee, Dongho Won: A Practical Study on Advanced Persistent Threats. Computer Applications for Security, Control and System Engineering: 144-152 (2012).

[9] Martin Ussath, David Jaeger, Feng Cheng, Christoph Meinel: Advanced persistent threats: Behind the scenes. CISS 2016: 181-186.

[10] Atul Bohara, Mohammad A. Noureddine, Ahmed M. Fawaz, William H. he: An Unsupervised Multi-Detector Approach for Identifying Malicious Lateral Movement. SRDS 2017: 224-233.

[11] Ahmed M. Fawaz, Atul Bohara, Carmen Cheh, William H. Sanders: Lateral Movement Detection Using Distributed Data Fusion. SRDS 2016: 21-30.

[12] Mohammad A. Noureddine, Ahmed M. Fawaz, William H. Sanders, Tamer Basar: A Game-Theoretic Approach to Respond to Attacker Lateral Movement. GameSec 2016: 294-313.

[13] Alessandro Greco, Giovanni Pecoraro, Alberto Caponi, Giuseppe Bianchi: Advanced Widespread Behavioral Probes against Lateral Movements. International Journal for Information Security Research (2016).

[14] Alessandro Greco, Alberto Caponi, Giuseppe Bianchi: Facing lateral movements using widespread behavioral probes. ICITST 2016: 159-160.

[15] Giovanni Apruzzese, Fabio Pierazzi, Michele Colajanni, Mirco Marchetti: Detection and Threat Prioritization of Pivoting Attacks in Large Networks. IEEE Transactions on Emerging Topics in Computing. 10.1109/TETC.2017.2764885.

[16] Kamrul Hasan, Sachin Shetty, John Sokolowski, Deepak K. Tosh:Security game for cyber physical systems. Proceeding CNS '18 Proceedings of the Communications and Networking Symposium, 2018.

[17] Abdallah K. Farraj, Eman M. Hammad, Deepa Kundur: On the Impact of Cyber Attacks on Data Integrity in Storage-Based Transient Stability Control. IEEE Trans. Industrial Informatics 13(6): 3322-3333 (2017).

[18] Mohammad Ashraf Hossain Sadi1, Huaxi Zheng, Mohd. Hasan Ali: Transient Stability Enhancement of Power Grid by Neural Network Controlled BFCL Considering Cyber-Attacks. IEEE SoutheastCon (2017).

[19] Ravi Jhawar, Karim Lounis, Sjouke Mauw: A Stochastic Framework for Quantitative Analysis of Attack-Defense Trees. STM 2016: 138-153.

[20] Neal Wagner, Cem Safak Sahin, Diana Hanson, Jaime Pena, Era Vuksani, Brady Tello: Quantitative analysis of the mission impact for host-level cyber defensve mitigations. SpringSim (ANSS) 2016: 2.

[21] Neal Wagner, Cem S Sxahin, Michael Winterrose, James Riordan, Diana Hanson, Jaime Pena,William W Streilein: Quantifying the mission impact of network-level cyber defensive mitigations. Journal of Defense Modeling and Simulation: Applications, Methodology, Technology 2017: 14(3).

[22] Ricardo J. Rodríguez, Xiaolin Chang, Xiaodan Li, Kishor S. Trivedi: Survivability Analysis of a Computer System Under an Advanced Persistent Threat Attack. GraMSec@CSF 2016: 134-149.

[23] Xiaolin Chang, Shaohua Lv, Ricardo J. Rodríguez, Kishor Trivedi: Survivability Model for Security and Dependability Analysis of a Vulnerable Critical System. Pro. IEEE ICCCN workshop IoTPST 2018.

[24] Lang Xie, Poul E. Heegaard, Yuming Jiang: Network survivability under disaster propagation: Modeling and analysis. WCNC 2013: 4730-4735.

[25] Su Yao, Jianfeng Guan, Hongke Zhang: Survivable strategy set design for malicious attack propagation in NEMO scenario. EURASIP J. Wireless Comm. and Networking 2016: 234.

[26] Zhipeng Yi, Tadashi Dohi, and Hiroyuki Okamura: Connectivity-Based Survivability Analysis with Border Effects for Wireless Ad Hoc Network. Advances in Reliability and System Engineering: 53-86 (2016).

[27] L. Zhang, X. S. Liu, J. W. Pang, D. G. Xu, V. C. M. Leung: Reliability and Survivability Analysis of Artificial Cobweb Network Model Used in the Low-Voltage Power-Line Communication System. IEEE Trans. Power Delivery 31 (5): 1980-1988 (2016).

[28] Xiaolin Chang, José M. Martínez, Kishor S. Trivedi: Transient performance analysis of smart grid with dynamic power distribution. Inf. Sci. 422: 98-109 (2018).

[29] Poul E. Heegaard, Kishor S. Trivedi: Survivability Modeling with Stochastic Reward Nets. Winter Simulation Conference 2009: 807-818.