Survivability Model for Security and Dependability Analysis of a Vulnerable Critical System

Xiaolin Chang[a], Shaohua Lv[a], Ricardo J. Rodríguez[b], Kishor Trivedi[c]
[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]Department of Electrical and Computer Engineering, Duke University, USA
Email: {xlchang, 16120401}@bjtu.edu.cn, rjrodriguez@unizar.es, ktrivedi@duke.edu

***Abstract*— This paper aims to analyze transient security and dependability of a vulnerable critical system, under vulnerability-related attack and two reactive defense strategies, from a severe vulnerability announcement until the vulnerability is fully removed from the system. By severe, we mean that the vulnerability-based malware could cause significant damage to the infected system in terms of security and dependability while infecting more and more new vulnerable computer systems. We propose a Markov chain-based survivability model for capturing the vulnerable critical system behaviors during the vulnerability elimination process. A high-level formalism based on Stochastic Reward Nets is applied to automatically generate and solve the survivability model. Survivability metrics are defined to quantify system attributes. The proposed model and metrics not only enable us to quantitatively assess the system survivability in terms of security risk and dependability, but also provide insights on the system investment decision. Numerical experiments are constructed to study the impact of key parameters on system security, dependability and profit.**

Keywords*— *Reactive defense strategy*; *Quantitative analysis*; *Stochastic Reward Nets*; *Survivability*; *Security

I. INTRODUCTION

A software vulnerability is a defect in software which can be exploited by attackers/malwares (malicious software) to compromise a system for their benefits. A typical malware goes through the following four phases: (1) gaining access to a targeted system by means of the declaimed vulnerabilities, (2) trying various methods to make itself persistent into the system, (3) looking for data of interest to be stolen or modified, and (4) damaging security by modifying unauthorized data, exfiltrating sensitive data, or infecting new vulnerable computer systems (denoted as host in the following). In addition, the various activities undertaken during the attack may crash the system and then degrade system dependability. The security and/or dependability damage may lead to loss of customer confidence and lead to the other possible long-term consequences due to loss and theft of information.

In this paper, we assess the survivability of a vulnerable critical system from a severe vulnerability announcement until the vulnerability is fully removed from the system. We define survivability as a transient measure of a system's capability in withstanding vulnerability-related malicious attacks and executing pre-specified mission even when parts of the system are damaged. By severe, we mean the vulnerability-based malware could cause significant damage to the infected system in terms of security and dependability while infecting new vulnerable computer systems.

We develop a survivability model to capture the system behaviors under reactive defense strategies and the actions performed by an attacker to cause such an attack by exploiting the vulnerability. All relevant event times are assumed to be exponentially distributed and thus the model is a homogeneous continuous time Markov chain (CTMC). In this paper, the generation and solution of the proposed Markov model are automated using a variant of stochastic Petri Nets called Stochastic Reward Nets (SRNs), which could easily represent common characteristics of computer systems such as concurrency, synchronization, conditional branches, looping, and sequencing.

This paper is close to the work in [1]. The key differences from [1] are detailed in Section II. We summarize the major contributions of this paper as follows:

(1) We investigate the scenario where two reactive defense strategies are deployed to reduce or prevent the security damage caused by malware. Moreover, we not only investigate the attacking activity of affecting the local system security, but also investigate the attacking activity of infecting new hosts.

(2) We develop a survivability model by using Stochastic Reward Nets so as to capture the system behaviors. To the best of our knowledge, we are the first to apply state-space analytic model to analyze the survivability of such a vulnerable critical system. We also define survivability metrics and propose the corresponding calculating methods.

(3) Numerical experiments are constructed to study the impact of key parameters on system security, dependability and profit.

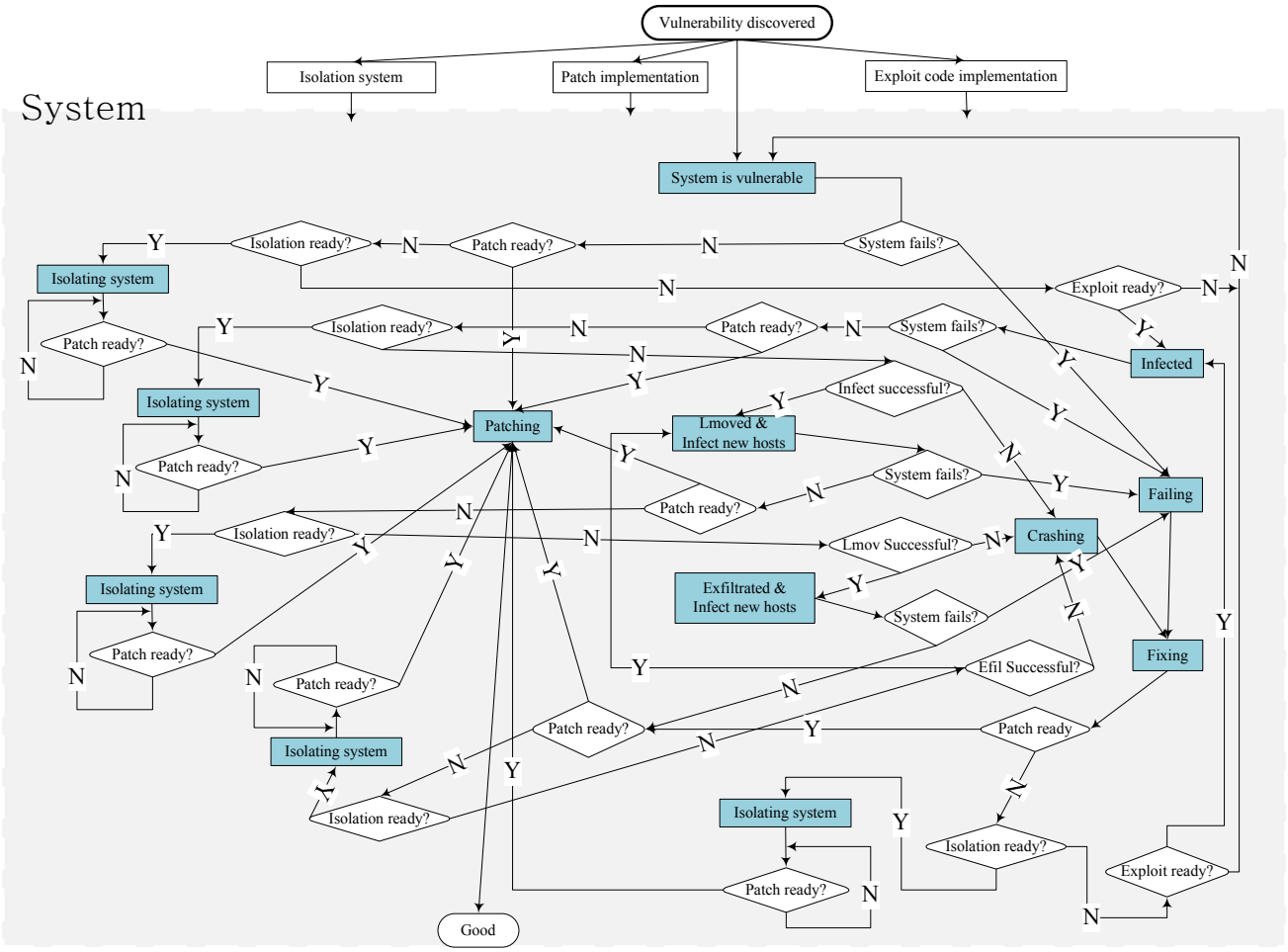

Fig. 1 Flowchart depicting events in a system under an attack, after a vulnerability is announced and during the implementation and deployment of the reactive defense strategies

The paper is organized as follows. Section II presents related work. Section III presents the system model and survivability measures. In Section IV, we present evaluation results. The conclusion is drawn in Section V.

II. Related Work

Kinds of efforts have been made to advance the improvement in the dependability and security of various infrastructure systems, including communication network, transportation, power and water distribution and so on. However, undesired events still occur to those systems. For example, natural disasters, security attacks and hardware/software failures. Timely and quick recovery from the unexpected events is critical to infrastructure systems. It is known that an undesirable event occurrence may only degrade the system performance instead of crashing the system.

In the scenario where there are multiple actions to be taken for recovering the system, the recovery process could be modeled as a single-phase recovery model or a multi-phase recovery model. In the latter type, each recovery action or a set of parallel actions are modelled as a phase. Phase input determines the sequence of the phases. A multi-phase recovery model could capture the fine-grained characteristics of the restoration process [2][5], compared to a single-phase recovery model.

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 [2]. Its quantitative analysis could help improve the systems' capability in critical service provision when damage occurs to part of the system or the whole system get damaged.

The tremendous increase in the number of vulnerabilities discovered and disclosed and the severity of their damage have prompted various research to the survivability modeling and analysis in various fields and from different perspectives [1][6][7] and the references therein. Recently, the authors in [1] carried out a quantitative assessment of the system secure survivability. There are three major differences between [1] and this paper:

(1) Only one mitigation strategy, namely the patch implementation, is considered in [1]. This paper, besides patch strategy, also considers the isolation strategy to

separate vulnerable part in the infected system, which avoids vulnerability-related damage but may degrade system dependability and performance.

(2) The security loss is quantified in terms of sojourn time in [1]. This paper proposes a new calculation method by in terms of the times of successfully stealing/modifying sensitive information.

(3) The model proposed in this paper could capture the activities of infecting new vulnerable hosts.

III. System Description and Model

This section first overviews the system of interest in this paper. Then a Stochastic Reward Net model for survivability analysis of this system is presented.

A. System Description

We now describe the system considered in this paper, shown in Fig.1. It could be regarded as an extended system of [1]. There are nine system states: Vulnerable, Isolating, Patching, Fixing, Failing, Crashing, Infected, Lmoved, and Exfiltrated. Isolating and Patching denote two reactive defense strategies considered in this paper. All the assumptions made in [1] are applied in this paper, in order to highlight the differences of this paper from [1]. More assumptions are given in the following.

When a vulnerability is fully disclosed, the system is in the Vulnerable state. Meanwhile, the attacker starts the exploit implementation. In addition, the defender designs and deploys the two reactive defense strategies. Thus, there are three rectangles in the second row of Fig.1, denoted by *Isolation system*, *Patch implementation* and *Exploit code implementation*, respectively.

The shaded part denoted by *System* describes the system state changes under the attack actions and the two reactive defense strategies. After the isolating strategy is deployed, the attack could not degrade the system security but the system performance is degraded. When the patch is ready, it must be deployed into the system immediately and the system is recovered to a secure state. When the attacked system is in Lmoved or Exfiltrated state, the malware also could infect new vulnerable hosts which have not been infected before. System may fail or crash due to attacker behaviors or software bugs, such as Mandelbugs [8]. If the system crashes or fails, it must be fixed immediately even the isolation or patching strategy is ready to be deployed. In the fixing process, both the defender and the attacker can do nothing to the system.

The metrics used to quantify survivability vary according to the system and system attributes of interest. We assume that as long as system service is provisioned, there is revenue. But revenue decreases after the isolation strategy is deployed or the malware enters into the vulnerable system. When system service cannot be provisioned, there may be economic loss to the service provider due to the pre-defined SLA (Service Level Agreement) with customers. In addition, both each successful infecting of a new vulnerable host and each successful stealing/modifying sensitive information could result in some loss to the service provider. We define *profit* equals *total revenue* minus *total cost*.

The metrics considered in this paper include:

- **Metric** m_1. Mean security loss of the local system at time t.
- **Metric** m_2. Mean number of new infected hosts at time t.
- **Metric** m_3. Mean accumulated security loss of the local system in the interval [0, t].
- **Metric** m_4. Mean accumulated number of the new infected hosts in the interval [0, t].
- **Metric** m_5. Mean accumulated cost in the interval [0, t].
- **Metric** m_6. Mean accumulated revenue in the interval [0, t].
- **Metric** m_7. Mean accumulated profit in the interval [0, t].

Note that although the definitions of some metrics are same as in [3], the computation formulas are different. The first three metrics are *transient* metrics that capture the state of the system at time *t* after the occurrence of an undesired event. The left metrics are c*umulative* metrics which are expected accumulated rewards in the interval (0, *t*]. Note that survivability metrics are computed after the announcement of a vulnerability. In the remainder of this paper, time *t* refers to the time immediately after a severe vulnerability announcement and is measured in days.

B. Stochastic Reward Net Model

There are two major challenges for modeling the system:

(1) How to model two attack activities which occur simultaneously. Namely, damaging the infected system security and infecting new vulnerable hosts.

(2) How to model the priority of the patch-based defense strategy over that of the isolation-based defense strategy when both strategies are ready to be deployed.

Fig.4 describes an SRN model for the survivability analysis. The shaded part is the extension to the model proposed in [1]. As in [1], survivability focuses on capturing the evolution of the system after an unexpected event occurs. Thus, the model in Fig.2 does not include the vulnerability detection process. TABLE I and II show the variable definitions and guard definitions, respectively. The following focuses on the explanation on the shaded part. The left part explanation is referred to [1].

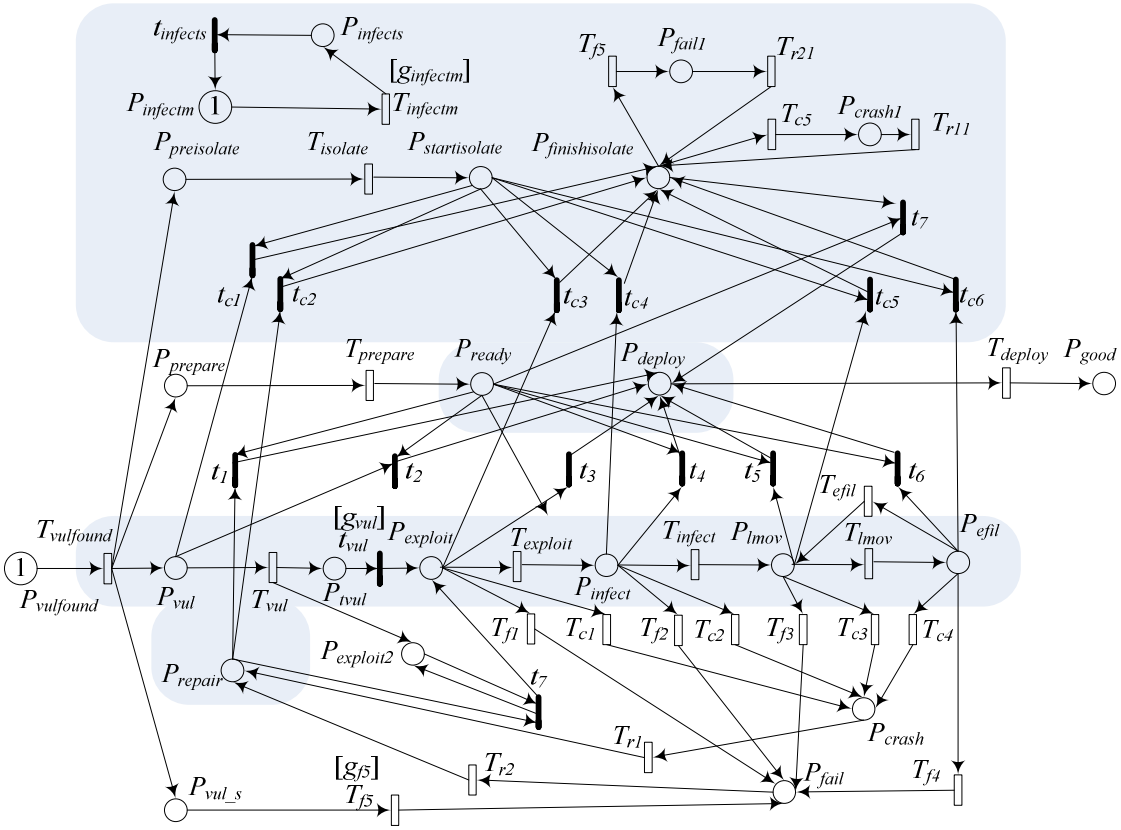

Fig. 2 Stochastic Reward Net model

When a software vulnerability is identified, one token is removed from $P_{vulfound}$ with rate δ and put in P_{vul_s}, P_{vul}, $P_{prepare}$, and $P_{preisolate}$ each. This means that system failure, exploitation code implementation, patch implementation, and the vulnerability-related service isolation implementation occur parallelly. A token in place $P_{preisolate}$ denotes that the isolation strategy is under implementation. When $T_{isolate}$ fires, one token is taken from $P_{preisolate}$ and one token is put in $P_{startisolate}$, representing that the isolation strategy is ready for deployment. When there is a token respectively in $P_{startisolate}$ and P_{vul} (P_{repair}, $P_{exploit}$, P_{infect}, P_{lmov}, or P_{efil}), the immediate transition t_{c1} (t_{c2}, t_{c3}, t_{c4}, t_{c5}, or t_{c6}) fires. Then, a token is taken from $P_{startisolate}$ and P_{vul} (P_{repair}, $P_{exploit}$, P_{infect}, P_{lmov}, or P_{efil}), and deposited in place $P_{finishisolate}$. $P_{finishisolate}$ represents that the system is isolated from the malicious software. In this situation, the system may fail or crash. As long as there is a token in P_{ready}, the system enters into the state of deploying the patch.

The priority of t_1, t_{c2} and t_7 is set as $t_1 > t_{c2} > t_7$ with the aim to achieve the following goals: whenever the patch strategy is available, the patch must be deployed immediately; then are the service isolation strategy and exploit code. Similarly, we set the priority: $t_2 > t_{c1}$, $t_3 > t_{c3}$, $t_4 > t_{c4}$, $t_5 > t_{c5}$, and $t_6 > t_{c6}$.

The activity of infecting a new vulnerable host is modeled by $P_{infectm}$, $T_{infectm}$, $P_{infects}$, $t_{infects}$ and $g_{infectm}$. $g_{infectm}$ assures that only when there is a token in P_{lmov} or P_{efil}, a new vulnerable host may be infected. Before we define each metrics, some variables are defined first. We define a reward/loss to each place in Fig.2 to represent the service revenue/loss at this place per day. r_{vul} / c_{vul}^{Place} denotes the unit revenue/loss at P_{vul}. The other places have similar revenue and loss definitions. c_{lmov}^{Trans} and $c_{infectm}^{Trans}$ are defined to denote unit loss of throughput at T_{lmov} and $T_{infectm}$, respectively. Now we use the SPNP software package [9] to calculate the above metrics as follows:

- m_1 : throughput of T_{lmov} at time t.
- m_2 : throughput of $T_{infectm}$ at time t.
- m_3 : the expected accumulated rate of $T_{infectm}$ in the interval [0, t].
- m_4 : the expected accumulated rate of T_{lmov} in the interval [0, t].

- m_5 : $m_3 * c_{lmov}^{Trans} + m_4 * c_{Infectm}^{Trans}$ + the sum of mean accumulated loss of each place in the interval [0, t].
- m_6 : the sum of mean accumulated reward of each place in the interval [0, t].
- $m_7 = m_6 - m_5$.

IV. NUMERICAL ANALYSIS AND DISCUSSIONS

This section aims to evaluate the effectiveness of the proposed model. We evaluate our model solutions obtained by using SPNP software package [9] to solve the SRN model, in terms of the metrics described in Section III.B. Parameter values are set as in [1], also given in TABLE I.

We first investigate the effect of $\lambda_{prepare}$ on security loss. The other parameter values are fixed as in TABLE I. Fig.3-Fig.6 plot these results. P10, P12, P16, and P20 represent the results of $1/\lambda_{prepare}$ = 10 days, 12 days, 16 days, 20 days respectively. We observe:

- Fig.3 indicates that for each $1/\lambda_{prepare}$, the throughput of damaging the local system security increases first and then decreases. The increasing throughput is due to the increasing probability that P_{lmov} has a token. But this increase stops at some time. The decreasing throughput is due to the increasing probability that the isolation and/or patch-based defense strategies are ready for deployment. Similar explanation could be applied for the changes in the throughput of infecting new hosts, shown in Fig.5.
- With the increasing mean days ($1/\lambda_{prepare}$) for the patch implementation, the probability that the patch-based defense strategy is ready for deployment increases slowly. Therefore, more security damage is caused. Fig.3 indicates the throughputs of P20, P16, P12 and P10 at the same time instant are increasing.
- With the increasing mean days ($1/\lambda_{prepare}$) for the patch implementation, much more local security damage is caused and there are more number of new hosts to be infected, shown in Fig.4 and Fig.6, respectively.

We also do experiments by fixing $1/\lambda_{prepare}$ =20 days and varying $\lambda_{isolate}$. Due to space limitation, we only present results of *Times of successfully damaging local system security at time t* in Fig.7. "i8" and "i16" represent the results of $1/\lambda_{isolate}$ = 8 days and 16 days, respectively. "i0" represents that there is no isolation strategy deployment. We observe that with the increasing mean days for the isolation implementation, there is more mean sojourn time for malware to launch attack to local system. Then more security damage is generated.

TABLE I. PARAMETER DEFINITION

Symbol	Definition	Mean value
$1/\delta$	Mean time that the discovered vulnerability is known to all	30 mins
$1/\lambda_{prepare}$	Mean time for implementing a patch	20 days
$1/\lambda_{deploy}$	Mean time for deploying the patch	12days
$1/\lambda_{vuln}$	Mean time for generating the exploit code by an attacker	4 days
$1/\lambda_{fail}$	Mean time that the computer system fails	365 days
$1/\lambda_{fix}$	Mean time that the computer system completes the failure or crash fixing	2 days
$1/\lambda_{exploit}$	Mean time for injecting the exploit code into the system	7 days
$1/\lambda_{inf}$	Mean time that the exploit code is persistent	1 days
$1/\lambda_{lmov}$	Mean time that the attacker finds sensitive data of interest	7 days
$1/\lambda_{efil}$	Mean time that the attacker obtains the desired information	2 days
$1/\lambda_{isolate}$	Mean time for shutting down those services related to the detected vulnerability	8 days
$1/\lambda_{infectm}$	Mean time that the attacker injects the exploit code into another vulnerable host	7 days
ρ_1 , ρ_2	Probability that the exploit code works in the system and is persistent, respectively	0.9,0.9
ρ_3 , ρ_4	Probability that the attacker finds its target and the desired information, respectively.	0.9,0.9
ρ_5	Probability that the attacker infects a new host successfully.	0.9

TABLE II. GUARD FUNCTIONS FOR THE SRN MODEL

Guard	Values
g_{vul}	if (#(P_{vul_s})==1) then 1 else 0
g_{f5}	if (#(P_{vul})==1) then 1 else 0
$g_{infectm}$	if (#(P_{lmov})==1 ||#(P_{efil})==1) then 1 else 0

V. CONCLUSIONS

This paper presents a CTMC model for survivability analysis of a critical system under a severe vulnerability. Stochastic Reward Nets was used to facilitate the generation and solution of the Markov model. We defined survivability metrics in terms of system dependability and security. In addition, numerical results were presented to study the impact of the underlying parameters on the system survivability. These results also provided insights on investment efforts in various system recovery strategies including reactive defense strategies.

ACKNOWLEDGMENT

The research of Xiaolin Chang is supported in part by NSF 61572066 of China. The research of Ricardo J. Rodríguez is supported in part by Spanish Ministry of Economy, Industry and Competitiveness project CyCriSec (grant number TIN2014-58457-R). The research of Kishor S. Trivedi is supported in part by US NSF grant number CNS-1523994, by IBM under a faculty grant.

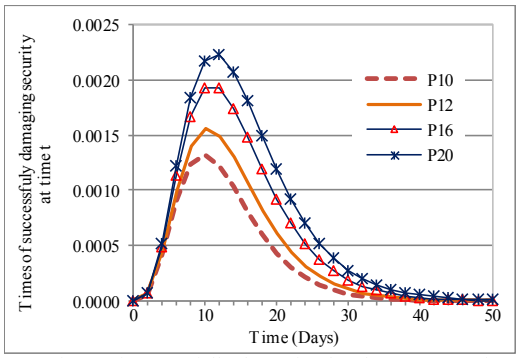

Fig. 3 Times of successfully damaging local system security at time t under different $\lambda_{prepare}$

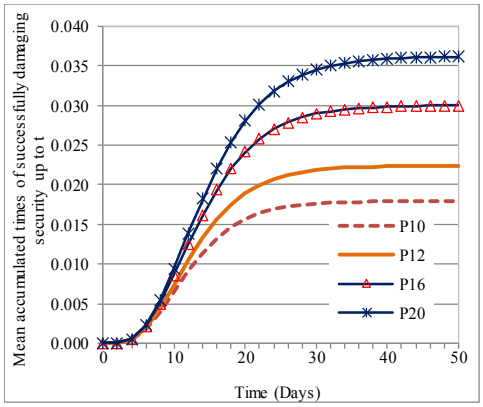

Fig. 4 Mean accumulated times of successfully damaging local system security by time t under different $\lambda_{prepare}$

REFERENCES

[1] 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.

[2] Poul E. Heegaard, Bjarne E. Helvik, Kishor S. Trivedi, Fumio Machida: Survivability as a generalization of recovery. DRCN 2015: 133-140.

[3] Kishor S. Trivedi, Ruofan Xia: Quantification of system survivability. Telecommunication Systems 60(4): 451-470 (2015).

[4] Lang Xie, Poul E. Heegaard, Yuming Jiang: Survivability analysis of a two-tier infrastructure-based wireless network. Computer Networks 128: 28-40 (2017) .

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

[6] Xiaolin Chang, Tianju Wang, Ricardo J. Rodríguez, Zhenjiang Zhang: Modeling and Analysis of High Availability Techniques in a Virtualized System. Comput. J. 61(2): 180-198 (2018).

[7] Zhi Chen, Xiaolin Chang, Zhen Han, Lin Li: Survivability Modeling and Analysis of Cloud Service in Distributed Data Centers. To be published in Comput. J. https://doi.org/10.1093/comjnl/bxx116.

[8] Michael Grottke, Kishor S. Trivedi: Fighting Bugs: Remove, Retry, Replicate, and Rejuvenate. IEEE Computer 40(2): 107-109 (2007).

[9] Gianfranco Ciardo, Jogesh K. Muppala, Kishor S. Trivedi: SPNP: Stochastic Petri Net Package. PNPM 1989: 142-151.

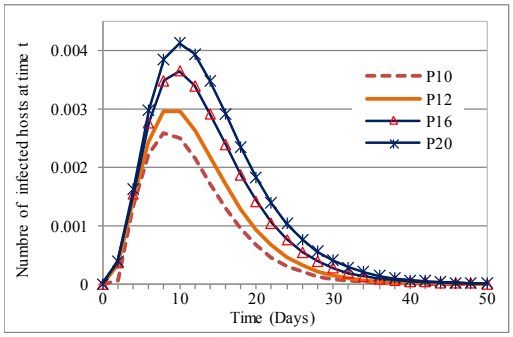

Fig. 5 Mean number of infected hosts at time t under different $\lambda_{prepare}$

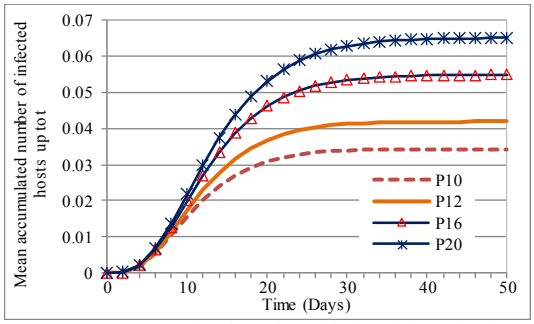

Fig. 6 Mean accumulated number of infected hosts by time t under different $\lambda_{prepare}$

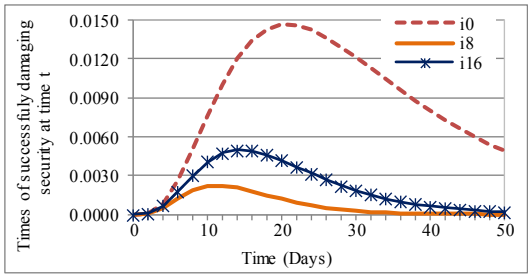

Fig. 7 Times of successfully damaging local system security at time t under different $\lambda_{isolate}$