ASSESSING DEPENDABILITY OF WEB SERVICES UNDER MOVING TARGET DEFENSE TECHNIQUES

SALVATORE DISTEFANO and MARCO SCARPA

University of Messina, Piazza Pugliatti 1, 98100 Messina, Italy.
E-mail: sdistefano,mscarpa@unime.it

XIAOLIN CHANG

Beijing Key Lab. of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, China
E-mail: xlchang@bjtu.edu.cn

ANDREA BOBBIO

DiSit, University of Piemonte Orientale, viale Teresa Michel 11, 15121, Alessandria, Italy.
E-mail: andrea.bobbio@uniupo.it

Moving Target Defense (MTD) is a quite effective solution for alleviating the impact of attacks from malicious sources or attackers taken from biological/military environments. It works as a proactive defense approach aiming at enhancing the target system security by periodically changing its configuration to reduce the exposure to vulnerabilities and opportunities for attack. Several MTD approaches have been studied in the literature, and one of the simplest and most effective is migrating the service among different nodes of a distributed computing system. To evaluate the effectiveness of migration-based MTD policies, it is crucial adopting proper cybersecurity metrics and tools for their assessment, which is highly challenging due to ICT complexity in terms of a large number of nodes, attack types, workloads, migration policies, and service delays. Current solutions in MTD evaluation, mainly based on combinatorial (attack trees), game theoretic and state space-based approaches, do not or partially address such challenges altogether. This paper aims at proposing a technique based on Petri nets to overcome limitations of existing solutions. The proposed model is highly scalable and customizable through several parameters. It also allows to stochastically characterize the underlying behaviors and phenomena through non-exponential distributions, obtaining both transient and steady-state metrics. Numerical experiments are performed to demonstrate the capability of the proposed approach in assessing the impact of MTD migration techniques on the system-service dependability, including security, availability and performance.

Keywords: Cybersecurity, Moving target defense, MTD Service migration technique, Dependability, Petri nets.

1. Introduction

Current technological landscape is dominated by Information and Communication Technologies (ICT), where distributed resources, devices (smartphones, laptops, wearables, single board computers), nodes (servers, cloudlets, communication systems) and things (smart objects, vehicles, cameras, appliances), ranging from few units to billions, exchange data through the Internet (in the order of zettabytes, 2^{70} per day altogether). On top of them, several applications transform this new commodity (data) into solutions and services for end-users, and business and revenues for providers. This is powered by huge, theoretically unlimited, computing facilities managing data to process, store, archive these services. At the heart of this system, Cloud computing plays the role of enabling technology, where services are elastically deployed and provisioned on-demand.

This widely heterogeneous and complex scenario raised up several challenges, some of which are still open and looking for proper solutions. Among them, cybersecurity is one of the big challenges concerning both data (confidentiality, privacy) and resources (integrity) management. In this context, moving target defense (MTD) (Jajodia et al. (2011), Sengupta et al. (2019)) is a quite effective solution for alleviating the impact of attacks from malicious sources or attackers taken from biological/military environments, based on the concept of changing the attack environment by moving a specific target to hide it from attackers, e.g. camouflage, weapon transfer. In ICT, the MTD cybersecurity technique has been adapted as a proactive defense approach aiming to enhance the system security by dynamics, continuously moving a service to reduce its exposure to

Proceedings of the 30th European Safety and Reliability Conference and
the 15th Probabilistic Safety Assessment and Management Conference
Edited by Piero Baraldi, Francesco Di Maio and Enrico Zio
Published by Research Publishing, Singapore.
ISBN: 978-981-14-8593-0; doi:10.3850/978-981-14-8593-0

vulnerabilities and opportunities for attack while increasing the system resiliency, as surveyed in (Lei et al. (2018)). Several MTD techniques have been proposed so far Hong and Kim (2016). Among them, those based on service migration are widely spread and successfully applied in several contexts. Proper cybersecurity metrics, methods, techniques and tools are required for assessing the effectiveness of MTD policies as also highlighted in (Zhuang et al. (2015)). Several approaches have been proposed in the literature as reported in Cai et al. (2016); Hong and Kim (2016), mostly based on attack graphs (Nguyen et al. (2017)), game theoretic approach (Zhu and Rass (2018)) and state space-based models (Connell et al. (2017)), to mainly evaluate the system survivability. However, despite their success, there is still need of further research on analytical modeling approaches for properly addressing the main challenges of current ICT scenarios, considering large number of nodes and scalability, attack types, workloads, migration policies, service delays and similar. New metrics and models should be used to also evaluate the impact of MTD migration policies in terms of dependability, i.e. security, performance and/or availability, in a realistic scenario adopting stochastic models relaxing the exponential assumption on underlying quantities and phenomena.

This paper proposes a technique based on Petri nets (PN) for properly modeling migration-based MTD techniques. The use of PN in the MTD area is not new, seeing Moody et al. (2014), Cai et al. (2016) and Chen et al. (2020). But our model aims at overcoming the limitation of existing solutions in terms of scalability, realisticness and multi-objective dependability metrics. The proposed model is therefore highly scalable and customizable through several parameters. It also allows to stochastically characterize the underlying behaviors and phenomena through non-exponential distributions, obtaining both transient and steady-state performance, security and availability/dependability metrics. Numerical experiments are performed to demonstrate the suitability of the proposed approach in MTD modeling of complex ICT systems.

The main contribution of this paper can be therefore summarized by a modeling and evaluation technique overcoming some limitations of existing technologies, specifically: i) *scalability* - the proposed PN model is highly scalable, relaxing restrictions in the number of nodes of the underlying MTD infrastructure; ii) *accuracy* - it allows to represent real and actual phenomena, relaxing the memory-less/Markovian assumption that forces the time variables characterizing the system behaviours to be exponentially distributed; iii) *multi-objective* - it allows to take into account different metrics and quantities related to security and performance inside the same model.

The remainder of the paper is organized as follows: Section 2 defines the research contexts, overviews MTD techniques, with specific regard to migration-based ones, and investigates on existing solutions for their modeling. Then Section 3 describes the proposed Petri net models of migration-based MTD techniques, then evaluated in Section 4. Conclusions and future work are discussed in Section 5.

2. Problem statement

The target scenario of this research effort is a (Web) service delivering its functionalities to third parties though the network. The service deployment could be on an Internet-connected server, either physical or virtual, subject anyway to external attacks that may compromise the overall system. Redundancy policies are usually adopted to mitigate the impact of failures thus identifying a distributed deployment infrastructure for a service. This may include Cloud computing infrastructure, in which the service can be deployed in virtual machines and/or containers, including hybrid virtualization contexts where containers run on virtual machines. This way, the underlying infrastructure may range from few (mainly physical) to thousands (virtual) network connected servers.

The network, however, can be also source of problems, namely *attacks*, which can compromise data confidentiality and resource integrity. There are several types of attacks (Uma and Padmavathi (2013)): *intrusion-based*, which require the attackers to enter the system and then perform the attack (e.g malware, phishing, SQL injection, password attacks); *eavesdropping*, where attackers sniff packets from the network (e.g. man-in-the-middle); *flooding*, performed by sending data over the network to congest a node (e.g. distributed denial of service). In this paper, we mainly focus on the former category, intrusion-based attacks.

In the intrusion-based attack scenario, the attacker or intruder has to first discover target system vulnerabilities to perform the attack. The set of system vulnerabilities that can be accessed by an intruder is identified as *attack surface*. An intrusion-based attack can be modeled by a specific (intrusion or *cyber*) *kill chain* (CKC) Hutchins et al. (2010). The CKC model specifies 7 stages for characterizing an attack workflow, which can be split into two groups, the former directly involving the intruder (recoinessance, weaponization and delivery) and the latter mainly triggered by the injected weapon or malware (exploitation, installation, command and control, actions on objective). More specifically, an intruder i) first selects the target node and attempts to identify vulnerabilities in its network (*reconnaissance*), then ii) creates remote access weapon (malware such as a virus or worm) depending on the discovered vulnerabilities (*weaponization*) and iii) transmits weapon to target by USB drives, e-mail attachments, websites, etc. (*delivery*). This way, the malware starts operating by iv) acting on the target network to exploit vulnerability through program code triggers (*exploitation*), v) installing access point for the intruder (e.g., "backdoor") (*installation*) and vi) enabling the latter to have persistent access and control of the target node (*command and control*) to vii) perform attacks like data exfiltration or destruction, encryption for ransom, and so on (*actions on objective*).

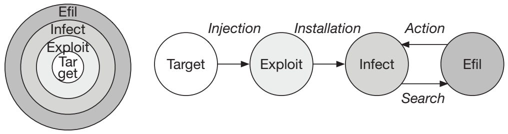

Fig. 1. Attack model.

The CKC model is considered as the reference model for cyber attacks, and it has been also widely used in security assessment Chen et al. (2020), mainly restricting the scope to malware activities rather than intruder ones, since the latter are quite hard to be precisely quantified. They are mainly related to human operation (human factor) and therefore widely variable, with high uncertainty due to the application domain, specific security requirements, intruders habits, needs, wills, political beliefs, and so on. In this paper we thus adopt the model of Chen et al. (2020), composed of 4 main stages extrapolated from the CKC:

(1) *Injection-Exploitation*: intruder gains access to a target system by means of a disclosed vulnerability, after injecting (recoinessance, weaponization and delivery) the malware and then activating (exploitation) it (steps i)-iv) of CKC).
(2) *Installation-Infection*: the malware tries various methods to conquer the system and giving access to the intruder (step v) of CKC).
(3) *Access and control-Search*: the malware looks for critical data in the infected platform ensuring persistent access on and control of the node (CKC step vi).
(4) *Actions on objective-Efil*: performing the attack acting on data (exfiltration, corruption, etc. - CKC step vii).

Moving Target Defense (MTD) is a proactive defense strategy that alters the attack surface for intrusion, e.g. by periodically migrating the service, thus reducing the probability of success of an attack. This is however quite effective also in mitigating the impact of a successful attack, which will be interrupted by a *"movement"* (e.g. migration). From literature (Hong and Kim (2016)), different MTD techniques, mainly grouped into 3 categories depending on the adopted strategy, have been specified. *Shuffle* techniques mainly operate on the system configuration, at different layers, e.g. changing server (migration), IP addresses, topologies, proxies, data and related functionalities. Techniques mainly implementing different equivalent solutions for the same service (different languages, OS, libraries, stacks, ...) belong to the *diversity* category, while those replicating components, functions and basically the service, fall into the *redundancy* group. It is also possible to combine such techniques to define adhoc strategies customized to the problem at hand. In this paper we mostly focus on the most widely adopted MTD technique, based on migration and categorized as shuffle.

To evaluate the effectiveness of an MTD solution to the service dependability we need to identify some specific metrics. Referring to the attack model of Fig. 1, several meaningful dependability metrics to statistically characterize the service and related MTD solutions adopted can be defined on the states identified in the model of Fig. 1 following the rule

$$Pr_* = \{\text{Probability the service is in the } * \text{ state}\}$$

This way, we can define Pr_{target}, $Pr_{exploit}$, Pr_{infect}, and Pr_{efil}. All the above quantities can even be defined over time t, thus identifying the probability to be in the state * at time t. Of course, $Pr_{target}(t)+Pr_{exploit}(t)+Pr_{infect}(t)+Pr_{efil}(t) = 1$. Furthermore, from these basic metrics we can define derived ones such as: $P_{safe} = 1 - P_{efil}$, i.e. the probability that the service information are not disclosed; $P_{notarget} = 1 - P_{target}$, i.e. the probability that the service is under attack; and $P_{notinfect} = 1 + P_{target} + P_{exploit}$, i.e. the probability that the service is not infected. Furthermore, $F_\Theta(t) = Pr\{\Theta \leq t\}$ is the cumulative distribution function of the time to complete random variable Θ representing the service completion.

3. The PN Model

In this work, a Web service running on a distributed infrastructure composed of N servers or nodes, with N ranging till thousands (e.g. in Cloud setups), can randomly migrate between nodes according to a migration-based MTD technique. The migration is triggered by a timer with a probabilistic or deterministic period. All the nodes have the same characteristics.

To model such a system focusing on the dependability quantities above discussed, Non-Markovian Stochastic Petri Nets (NMSPN) (Bobbio (1990); Bobbio et al. (1998); Longo et al. (2016)) have been adopted, extending the NMSPN of (Chen et al. (2020)) for scalability and generalization (general distributions, new metrics) pur-

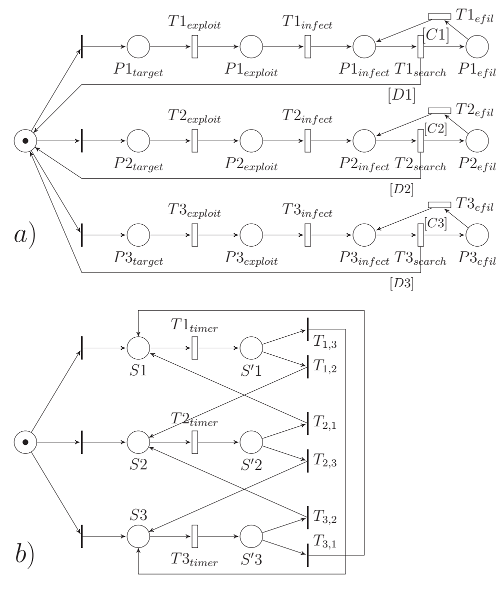

Fig. 2. NMSPN 3-node service execution model under MTD

Table 1. Enabling Functions for Fig. 2

Name	Function
$[C_i], i = 1, 2, 3$	if $\#S_i = 1$, return 1, else 0
$[D_i], i = 1, 2, 3$	if $\#S_i = 1$, return 0, else 1

poses. The original NMSPN describing a service running on a 3-node infrastructure is reported in Fig. 2. The NMSPN of Fig. 2a represents the attack on nodes i $(i = 1, 2, 3)$ while Fig. 2b represents the migration of the service between the nodes. When a transition Ti_{timer} fires, the service migrates to a different node. Even if the system is usually governed by a single timer, in the model of Fig. 2b each node has its own clock Ti_{timer}. The connection between parts $a)$ and $b)$ of Fig. 2 is implemented by the enabling functions $[C_i]$ and $[D_i]$ reported in Table 3 that specify if, during the searching phase, the attacker is on the node where the service is running. If so, the place S_i is marked ($\#S_i = 1$) and the token moves from Pi_{infect} to Pi_{efil} due to the firing of Ti_{search}, otherwise a new attack is performed on a randomly selected node. The probability that the service is not under attack or safe is thus

$$Pr_{safe}(t) = \sum_{i=1}^{3} Pr\{\#Pi_{efil}(t) = 0\}.$$

Notice that in the model of Fig. 2, the service can migrate to a different node but the attacker may remain on the same node.

In (Chen et al. (2020)) all the transition times are exponentially distributed with rates given in Table 2. Thence, the quantitative analysis can be obtained by solving the Continuous Time Markov Chain (CTMC) isomorphic to the NMSPN reachability graph. Increasing the number of nodes in the system requires to increase the number of lines in the two sub-NMSPNs of Fig. 2. This can be done automatically by writing a proper script for the NMSPN generation, but does not avoid the explosion of the state space of the underlying CTMC.

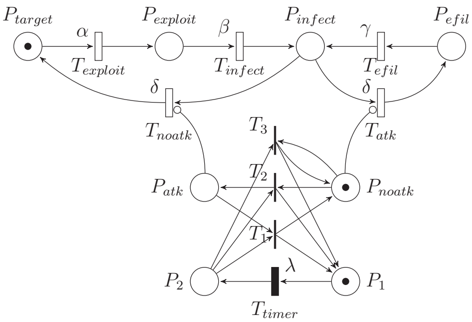

Fig. 3. Lumped PN Model.

To address the state space explosion problem, this paper proposes a more compact representation by means of the lumped NMSPN of Fig. 3, where we assume a generic number of nodes equal to N. The top part of Fig. 3 represents the phases of the attack, as in Fig. 2, but on a generic node. What changes is the control part on the bottom of the figure. Place P_{atk} models the case in which the attacker is on the node where the service is running, while the place P_{noatk} models the attack on one of the remaining $(N-1)$ nodes without service. The two places are in mutual exclusion and only one of them may contain a token. Hence the system switches from the condition in which the attacker is in a wrong node (P_{noatk} is marked as in Fig. 3) to the service attack (P_{atk} is marked) and consequently transitions T_{atk} and T_{noatk} cannot be enabled at the same time. When P_{infect} contains a token, if P_{atk} is marked, the transition T_{noatk} is inhibited and the attack proceeds from P_{infect} to P_{efil}. Viceversa, when P_{noatk} is marked, the transition T_{atk} is inhibited and the search phase returns to P_{target} (compare the behavior with the NMSPN of Fig. 2). Places P_1 P_2 represent the migration: when the transition T_{timer} fires, the token is moved from P_1 to P_2 triggering a migration to one of the $(N-1)$ different nodes. The migration is governed by the immediate transitions T_1, T_2 and T_3 as follows:

P_2 and then P_{atk} are marked - After migration timer firing, the service, previously running on

the node under attack, lands on a safe node. The only enabled transition is T_1 and with probability $Pr_{T_1} = 1$ a token is moved from place P_{atk} to place P_{noatk} and the timer restarts.

P_2 and then P_{noatk} are marked - Two conflicting actions are possible, modeled by the two conflicting transitions T_2 and T_3. Either the service migration lands on a node under attack (firing of T_2 and a token moved to P_{atk}) with probability $Pr_{T_2} = 1/(N-1)$ or on one of the $(N-2)$ "safe" nodes (firing of T_3) with probability $Pr_{T_3} = (N-2)/(N-1)$. In both cases after migration the timer starts over again. In the NMSPN of Fig. 3, the number of nodes N affects only the value of the switching probabilities between the immediate transitions T_2 and T_3 and does not influence the structure of the NMSPN and therefore the dimension of the underlying model state space. This way, the lumped NMSPN is a scalable model of the Web service execution under migration-based MTD policy.

4. Validation and further experiments

The validation of the proposed model is investigated in Section 4.1 by comparing the results obtained by the original PN (Fig. 2) against those of the lumped one (Fig. 3). Then, to demonstrate the flexibility and suitability of the proposed (lumped) model in dealing with more generic (stochastic) behaviors and other dependability metrics, further evaluations are discussed in Section 4.2. All numerical analysis is performed with the tool WebSPN (Longo et al. (2016)).

Table 2. Parameter values of the MTD PN Model.

Par.	Definition	Value
$1/\alpha$	Mean time to exploit a node	25 h
$1/\beta$	Mean time to infect a node	0.25 h
$1/\delta$	Mean time to search in a node	0.25 h
$1/\gamma$	Mean time for exfiltrating from the node	0.5 h
$1/\lambda$	Mean period of the timer	1 h
$1/\mu$	Mean migration time	10 m

4.1. *Validation*

In the validation, all the timed transitions are exponentially distributed with rates, taken from Chen et al. (2020), shown in Table 2. The original model of Fig. 2 and the lumped one of Fig. 3 have been compared considering the probability that the service is secure at time t, i.e. $Pr_{safe}(t) = 1 - Pr_{efil}(t)$, expressed in the PN models as the probability places Pi_{efil} (P_{efil} in the lumped NMSPN) are not marked at time t, for $N = 3, 4, 6, 8$ nodes, as specified in Section 3. Fig. 4 shows the results thus obtained, where original and lumped are related to the models of Fig. 2 and 3, respectively. Such results are very close, with an error lower than 10^{-4}, thus validating the lumped model.

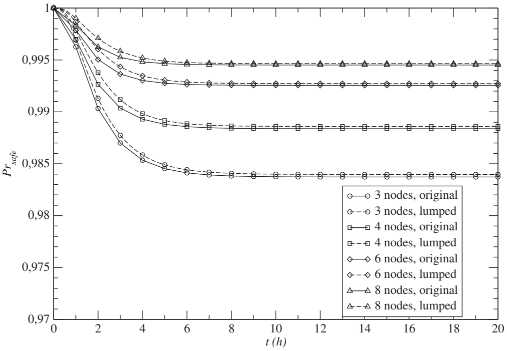

Fig. 4. Transient Pr_{safe} probability comparison between the original and the lumped NMSPN results.

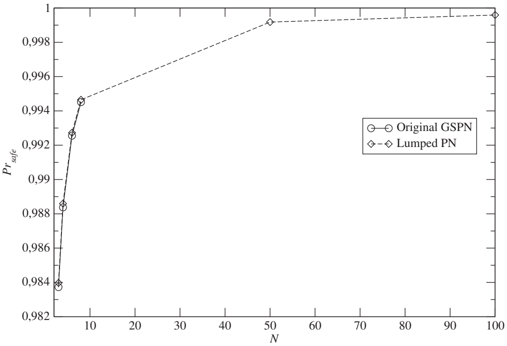

Fig. 5. Steady state Pr_{safe} probability obtained by the original and lumped NMSPN.

The scalability of the lumped NMSPN is then demonstrated by Fig. 5 showing the steady state value of the Pr_{safe} probability on the number of nodes from $N = 3$ to $N = 100$. The figure also reports Pr_{safe} of the original model till $N \leq 8$, to demonstrate that, while for the lumped model there is no upper bound to the value of N, in the original NMSPN, an increment on N brings to the state space explosion, thus restricting the computation to few nodes (8).

4.2. *Further experiments*

This section aims to evaluate the suitability of the proposed lumped model in the evaluation of large systems, even including non-exponentially distributed timer periods (Section 4.2.1), non-instantaneous migration times (Section 4.2.2), or also performance metrics such as the time to complete a service (Section 4.2.3).

4.2.1. *Non-exponential timer period*

In Fig. 3, the transition T_{timer} is drawn as a black box to indicate that the associated transition time can be generally distributed. Fig. 6 shows the impact on the security (Pr_{safe}) of two different timer distribution functions (exponential and deterministic) with the same expected value $E[timer] = 1\,h$, ranging from $N = 3$ to $N = 1000$. The deterministic timer results in an oscillating behavior that, in any case, provides a higher security particularly for lower values of N.

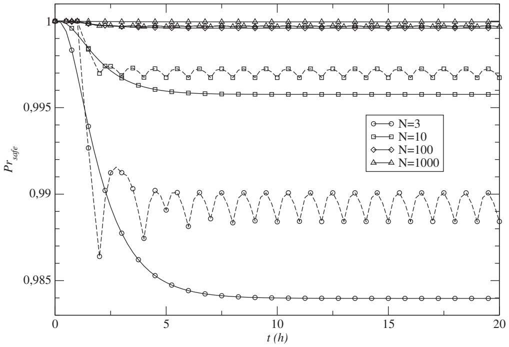

Fig. 6. Transient Pr_{safe} probability changing the timer distribution (exponential - solid lines, deterministic - dashed lines) with expected time $E[timer] = 1h$.

4.2.2. *Non-instantaneous migration time*

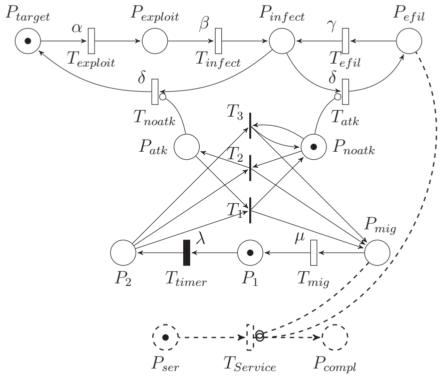

Fig. 7. Lumped PN Model with migration delay and service execution time.

In the previous sections, the migration is assumed to be instantaneous, i.e. the migration time is equal to zero. However, moving a service from one node to another through the net takes a finite time that delays the service execution and influences the performance of the service. The NMSPN including the migration time is shown in Fig. 7 (the dashed part refers to the service processing time and will be discussed in Section 4.2.3). Place P_1 represents the normal operating condition of a service. When T_{timer} fires, a token is moved to P_2 and the new node is immediately selected by firing one of the immediate transitions T_1, T_2 or T_3 according to the same logic of Fig. 3. Then the token instantaneously moves in place P_{mig} where the transition T_{mig}, modeling the migration latency, is thus enabled. Once the migration is done, the NMSPN moves a token back to P_1 representing the service online in a new node.

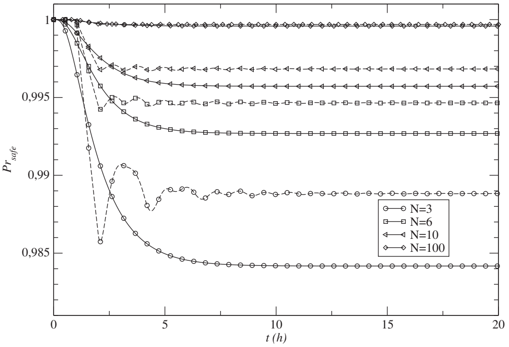

Fig. 8. $Pr_{safe}(t)$ with exponentially distributed (solid lines) vs deterministic (dashed lines) timers and non-instantaneous migration time.

In all the following numerical experiments the migration time is assumed to be exponentially distributed with mean value $1/\mu = 10\,min$ (see Table 2). Fig. 8 shows the transient $Pr_{safe}(t) = \{\#P_{efil}(t) = 0\}$ probability comparing exponential and deterministic timer with the same expected value $E[timer] = 1\,h$ for $N = 3, 6, 10, 100$. Again, a deterministic timer provides, on the average, a better security, in particular for low values of N. To further investigate the impact of the timer distribution by increasing the number of nodes, Fig. 9 plots the steady state value of Pr_{safe} on N (in logarithmic scale) for exponential and deterministic timer with $E[timer] = 1\,h$. The deterministic timer performs always better but the improvement tends to disappear for large N, when the shuffle effect prevails on the shape of the distribution.

To bind the ability of the system to run a service with the security provided by the implemented MTD technique, we introduce a new measure called *Secure Service Availability* (SA), defined as the probability that the service is running (token in

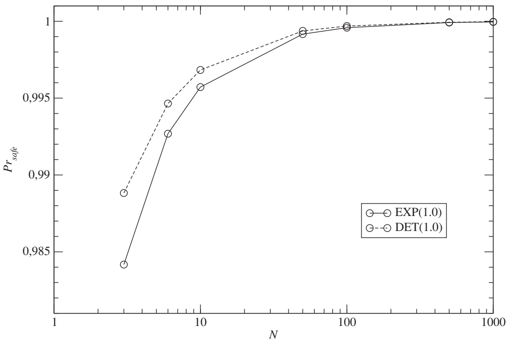

Fig. 9. Steady state Pr_{safe} probability for exponentially distributed vs deterministic timer with non-instantaneous migration time varying N.

P_1) and the node on which the service is running is not under attack (no token in P_{efil}). More formally

$$Pr_{SSA}(t) = Pr\{\#P_1(t) = 1 \wedge \#P_{efil}(t) = 0\}$$

The transient Secure Service Availability Pr_{SSA} is plotted as a function of the time t in Fig. 10 for a system with $N = 3$ nodes, exponentially distributed and deterministic timers with expected values $E[timer] = 0.5\,h$ and $E[timer] = 2\,h$. As expected, by decreasing the timer, higher values of Pr_{SSA} are obtained, also confirming the fluctuating transient trend of deterministic timers.

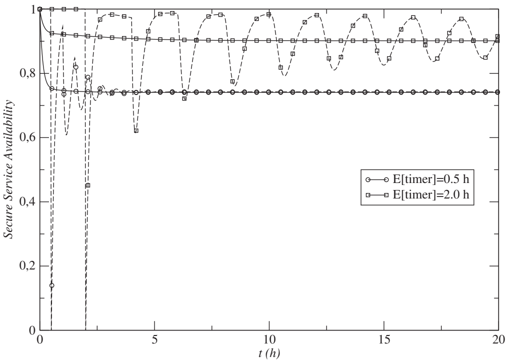

Fig. 10. Secure service availability Pr_{SSA} considering both exponential (solid) and deterministic (dashed) timers varying the expected value of the timer with $N = 3$.

4.2.3. *Secure service completion time*

Let us assume the service requires a processing time θ on a node to be completed. However, the service processing is affected by two phenomena: *i)* the migration, since the service cannot be executed while migrating, and *ii)* the security, since during a successful attack (a token in P_{efil}) the processing is not secure and can be compromised. The processing time for the service θ can be either a random variable or a constant, but in any case the time to complete the service without data disclosure Θ is a random variable due to the stochastic fluctuations of the MTD system running the service. As discussed in Section 2, the cumulative distribution function of the secure service completion time is given by $F_\Theta(t) = Pr\{\Theta \leq t\}$. In the bottom part of Fig. 7, transition $T_{service}$ is associated with the processing time θ. Place P_{ser} represents the service processing and is initially marked by a token. The firing of $T_{service}$ moves a token to P_{compl} representing the service processing completion. Hence

$$F_\Theta(t) = Pr\{\Theta \leq t\} = Pr\{\#P_{compl}(t) = 1\}$$

However, the secure service processing is inhibited when either P_{mig} or P_{efil} are marked. In the following numerical example, θ is assumed to be exponentially distributed with mean value $E[\theta] = 4\,h$. Fig. 11 evaluates $F_\Theta(t)$ for a system with $N = 3$ nodes, and reports the results obtained by assuming an exponential timer (solid line) or a deterministic timer (dashed line), with variable timer mean values $E[timer] = 0.5, 1, 1.5, 2\,h$. Of course, $F_\Theta(t) = 0$ for $t \leq \theta$ since the service cannot be completed before θ. By increasing the expected value of the timer a sharper distribution (lower variance) is obtained since the expected number of migrations before service completion decreases accordingly.

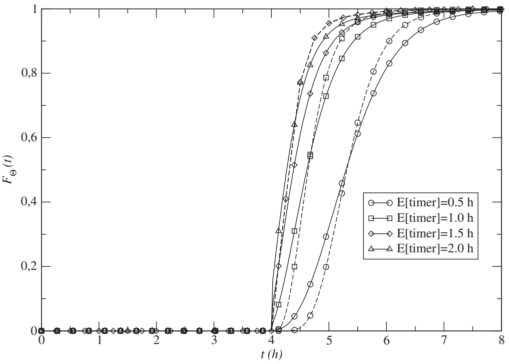

Fig. 11. Cumulative distribution function of the secure service time to completion $F_\Theta(t)$ with $N = 3$, varying timer distribution and expected value.

5. Conclusions and future work

In this paper we propose a scalable state-space-based modeling approach to quantitatively analyze Web service dependability metrics subject to migration-based MTD techniques. The proposed models relax some assumptions of existing solutions related to the maximum number of nodes (scalability), the stochastic characterization of related metrics and quantities (extending to non-Markovian behaviours), while considering new dependability metrics including security, availability and performance. Once the proposed model is validated against existing model and solution taken from literature (Chen et al. (2020)), further experiments are carried out for evaluating the suitability and effectiveness of the proposed model to more realistic setup, investigating its capability on analyzing the service dependability in a large-scale system.

The improved dependability is usually achieved with increasing costs. Studying a tradeoff analysis of cost, time, and effectiveness of a MTD system is necessary. Several factors, including definition of various costs and optimization goals, are considered in ongoing and future work, also planning to extend the modeling to other MTD techniques into a full fledged tools for the evaluation and design of MTD systems.

Acknowledgments

One of the authors acknowledges the financial support of the Universita' del Piemonte Orientale for this original work.

References

Bobbio, A. (1990). System modelling with Petri nets. In A. Colombo and A. S. de Bustamante (Eds.), *System Reliability Assessment*, pp. 103–143. Kluwer Academic P.G.

Bobbio, A., A. Puliafito, M. Telek, and K. Trivedi (1998, Feb). Recent developments in non-Markovian stochastic Petri nets. *Journal of Systems Circuits and Computers 8*(1), 119–158.

Cai, G., B. Wang, Y. Luo, and W. Hu (2016). A model for evaluating and comparing moving target defense techniques based on generalized stochastic petri net. In J. Wu and L. Li (Eds.), *Advanced Computer Architecture*, Singapore, pp. 184–197. Springer Singapore.

Cai, G.-l., B.-s. Wang, W. Hu, and T.-z. Wang (2016, Nov). Moving target defense: state of the art and characteristics. *Frontiers of Information Technology & Electronic Engineering 17*(11), 1122–1153.

Chen, Z., X. Chang, Z. Han, and Y. Yang (2020). Numerical evaluation of job finish time under mtd environment. *IEEE Access 8*, 413–416.

Connell, W., D. A. Menascé, and M. Albanese (2017). Performance modeling of moving target defenses. In *Proceedings of the 2017 Workshop on Moving Target Defense*, MTD '17, New York, NY, USA, pp. 53–63. Association for Computing Machinery.

Hong, J. B. and D. S. Kim (2016). Assessing the effectiveness of moving target defenses using security models. *IEEE Transactions on Dependable and Secure Computing 13*(2), 163–177.

Hutchins, E. M., M. J. Cloppert, and R. M. Amin (2010). Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Lockheed Martin Corporation.

Jajodia, S., A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang (2011). *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats* (1st ed.). Springer Publishing Company.

Lei, C., H.-Q. Zhang, J.-L. Tan, Y.-C. Zhang, and X.-H. Liu (2018). Moving target defense techniques: A survey. *Security and Communication Networks vol. 2018*(12), 25.

Longo, F., M. Scarpa, and A. Puliafito (2016). *WebSPN: A Flexible Tool for the Analysis of Non-Markovian Stochastic Petri Nets*, pp. 255–285. Springer.

Moody, W. C., H. Hu, and A. Apon (2014). Defensive maneuver cyber platform modeling with stochastic petri nets. In *10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pp. 531–538.

Nguyen, T. H., M. Wright, M. P. Wellman, and S. Baveja (2017). Multi-stage attack graph security games: Heuristic strategies, with empirical game-theoretic analysis. In *Proceedings of the 2017 Workshop on Moving Target Defense*, MTD '17, New York, NY, USA, pp. 87–97. Association for Computing Machinery.

Sengupta, S., A. Chowdhary, A. Sabur, D. Huang, A. Alshamrani, and S. Kambhampati (2019). A survey of moving target defenses for network security. *ArXiv abs/1905.00964*.

Uma, M. and G. Padmavathi (2013). A survey on various cyber attacks and their classification. *IJ Network Security 15*(5), 390–396.

Zhu, Q. and S. Rass (2018). Game theory meets network security: A tutorial. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, pp. 2163–2165. Association for Computing Machinery.

Zhuang, R., A. G. Bardas, S. A. DeLoach, and X. Ou (2015). A theory of cyber attacks: A step towards analyzing mtd systems. In *MTD '15*.