Parameterizing Moving Target Defenses

Nicholas Anderson
Sandia National Laboratories
Albuquerque, NM 87185
Email: nbander@sandia.gov

Robert Mitchell
Sandia National Laboratories
Albuquerque, NM 87185
Email: rrmitch@sandia.gov

Ing-Ray Chen
Department of Computer Science
Virginia Tech
Falls Church, VA 22043
Email: irchen@vt.edu

***Abstract*—Moving Target Defense (MTD) is the concept of controlling change across multiple system dimensions, aiming to disrupt the adversary in the attack sequence for intrusion prevention. To date, there is a lack of progress in MTD modeling and evaluation to test the effectiveness of MTD techniques. In this paper we develop two analytical models based on closed-form solutions and Stochastic Petri Nets to analyze the effect of a dynamic platform technique based MTD on attack success rate. The numerical results from these two models agree with one another, providing cross-validation. Furthermore, the output of these models indicates the existence of parameter settings that decrease the security of the protected resource and settings that make MTD most effective in terms of minimizing the attack success probability.**

***Index Terms*—moving target defense; security; modeling**

I. Introduction

Moving Target Defense (MTD) is the concept of controlling change across multiple system dimensions in order to increase uncertainty and apparent complexity for attackers, reduce their window of opportunity and increase the costs of their probing and attack efforts [1]. In this paper, we propose two analytical models for dynamic platform technique (DPT) based MTD effectiveness evaluation for the purpose of cross-validation.

To understand how MTD can be made effective against attacks, we start with an attack model as illustrated in Figure 1. The attack model is the six phase attack sequence comprising: survey, tool, implant, pivot, damage/exfiltration and cleanup activities. During the survey phase, the attacker identifies the key locations for the attack: the vulnerable node (e.g., web server or operator workstation) through which to enter the defender system, the critical nodes (that control a critical process or store critical data) and the intermediate nodes linking the entry node and critical nodes. Survey data may include host name, subnet, network address, MAC address, operating system and security and application software. During the tool phase, the attacker configures existing attack tools or creates new tools. During the implant phase, the attacker establishes a presence on the defender system. This could be from attacking a webserver, phishing a human operator or tasking an insider. During the pivot phase, the attacker will transition from the entry node to the critical node. During a damage phase, the attacker will perform some application specific action to disrupt the defender's core mission. Alternatively, during an exfiltration phase, the attacker will transfer the defender's critical data. During the cleanup phase, the attacker will remove all artifacts from the attack (e.g., registry entries, covert file systems or tainted applications or libraries).

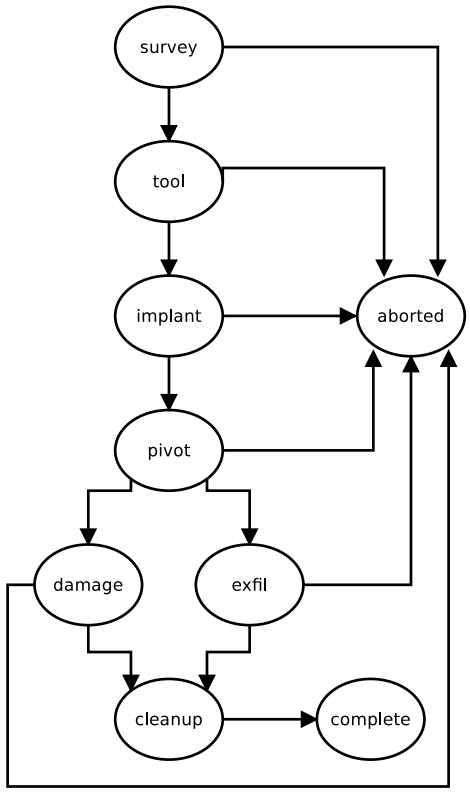

Fig. 1. Phased attack sequence.

While intrusion detection, tolerance and response are important and effective defensive measures, intrusion prevention stops attackers earlier in the phased attack sequence illustrated in Figure 1. MTD is a type of intrusion prevention that aims to disrupt the attacker across many stages: First, it distorts the adversary's picture of the protected resource in the survey phase. During the tooling

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. Approved for unlimited release: SAND2015-10609 C.

phase, MTD prompts the attacker to spend time and money developing exploits for multiple platforms. Third, it stops the adversary from placing and persisting malware in the implant phase. During the pivoting phase, MTD obscures the identity of the true target. Finally, it complicates the damage and exfiltration phases.

Okhravi et al. [2] propose a taxonomy of MTD where network and host based techniques comprise the first layer of classification, and host based techniques are further classified into dynamic runtime environment, dynamic application code and data, and DPTs.

As will be discussed in Section II, there is considerable interest in developing MTD techniques such as artificial diversity, bio-inspired defenses, software-defined network configuration, system/code diversification, and code mobility. However, there is a lack of progress in MTD modeling and evaluation to test the effectiveness of these MTD techniques. In this paper we develop two analytical models based on closed-form solutions and Stochastic Petri Nets (SPNs) [3] to analyze the effect of MTD on the probability of attack success. Computational efficiency inspired us to pursue a closed-form solution, and we investigated a SPN-based solution because of its inherent consideration of time.

The rest of the paper is organized as follows: In Section II, we survey state-of-the-art MTD techniques and existing MTD evaluation methods. In Section III, we develop two analytical models for the purpose of cross-validation. In Section IV, we apply these two analytical models to various DPT-based MTD scenarios with physical result interpretations given. Finally, Section V summarizes the paper and outlines future research directions.

II. Literature Search

Hong and Kim [4] propose a Hierarchical Attack Representation Model (HARM) to assess the effectiveness of an MTD. HARM addresses the inability of flat approaches to scale due to changes in network architecture. They contrast HARM with the existing Attack Graph (AG) assessment technique. Furthermore, the authors propose Importance Measures (IM) to guide the parameterization of an MTD; Hong and Kim contrast IM with exhaustive search (ES). They categorized MTD techniques as shuffling, diversity or redundancy, and the authors incorporated each into a HARM model to measure effectiveness. Typically, redundancy is not regarded as a moving target defense. Hong and Kim found shuffling techniques had scalability issues, randomly deployed diversity strategies can be inefficient and redundancy techniques linearly increased system security risk. They use risk (unitless), probability of attack success and reliability (probability of attack success at some arbitrary time) as their metrics. The authors' model includes insider attacks. Relative to HARM [4], the closed-form mathematical model we propose is more intuitive.

Collins [5] proposes a game theoretic way to assess the effectiveness of MTD. His MTD taxonomy comprises permutation, ephemeralization and replication techniques, which are network based, and checkpointing, which is host based. The author bases his assessment on tags and assets. Collins' model includes pivoting and Denial of Service (DoS) attacks. While this works well for network based MTDs, the model we propose can analyze both network based and host based MTDs.

Evans et al. [6] propose a way to assess the effectiveness of MTD. This study discusses the utilization of a model for assessing the effectiveness for MTD utilized against various attack classes. They predict that for most cases (circumvention, deputy, brute force and probing) in their attack model, their brands of MTD provide a marginal benefit. However, their brands of MTD, given a sufficiently high rediversification rate, provide significant benefit for the incremental attack case. While their investigation focuses on evaluating dynamic runtime environment and dynamic application code and data based MTDs, our proposal in addition can analyze DPT-based MTDs.

Okhravi et al. [7] propose another way to assess the effectiveness of an MTD. Their investigation focuses on evaluating DPTs. The authors describe DPTs using four features: diversity, multi-instance, limited duration and cleanup. Their attack model is similar to our own in one sense: the attacker has exploits for some platforms, but not others. However, the authors assume a computer network attack (CNA) type attacker who seeks to disrupt system operation rather than a computer network exploitation (CNE) type attacker who seeks to keep the protected resource operating as normal while exfiltrating sensitive data. Where Okhravi et al. parameterize the attacker based on how long they seek to disrupt the protected resource, we parameterize the attacker based on how long they seek to persist on the protected resource and how well financed/skilled the attacker is.

Zaffarano et al. [8] propose a technique to assess the effectiveness of an MTD. They propose four metrics each for the attacker and defender: productivity, success, confidentiality and integrity. While the authors provide equations to calculate these eight metrics, critical pieces such as valuation functions are missing. In this paper, the authors construct a framework to quantify the impact of the various MTD systems on the traditional Confidentiality, Integrity, Availability (CIA) model of information security. Further, they expand upon these traditional aspects to measure MTD systems which might fail to prevent an attack, however still successfully monitor and log said attack to offer aid in attribution and remediation. The authors then deployed enterprise level tasks in an effort to create measurable network activity from which to gauge the effectiveness of MTDs. The attack model of Zaffarano et al. is a phased attack sequence similar to what we illustrate in Figure 1. While their approach comprises a constructed framework consisting of large scale network emulation via hypervisor virtualization, we propose a closed-form mathematical model and a stochastic model to predict system performance.

Crouse et al. [9] propose a method to assess the effectiveness of an MTD. Their MTD taxonomy classifies techniques into movement or deception categories. This

paper attempts to model the probability of success for an attacker attempting to perform reconnaissance on a network in the presence of either a honey pot defense strategy or a network address shuffling strategy. The model developed to gauge the effectiveness of the employed defenses is a probabilistic measure of the reconnaissance success given an undefended network. The model is then expanded to account for employing the above mentioned defenses, and the results show that honeypot defenses outperform network shuffling, or deception defenses outperform movement defenses, but that a combination of both defenses yields the greatest gains in disrupting attacker reconnaissance. The authors' attack model considers probing and surveillance attacks; Crouse et al. formulate these attacks into foothold, minimum to win and shuffling drop scenarios. The probing and surveillance attacks they consider fall into the survey phase of our attack model (shown in Figure 1). The drawback is that good data for survey activity is hard to come by because of the large amount of noise from legitimate scanning and recreational hacking.

Zhuang et al. [10] propose an approach to assess the effectiveness of an MTD. Their model considers five parameters: attack interval, adaptation interval, number of nodes, adaptations per adaptation interval and attack success likelihood. Like [7], this is interesting work, but does not consider a persistent attacker who wishes to remain implanted on a protected resource for a long time rather than an adversary who gets in once and claims victory. In contrast, our study focuses on the attacker who persists on the target system for a long time.

Xu et al. [11] survey current MTD techniques. Their MTD taxonomy comprises four categories: software based diversification, runtime based diversification, communication diversification and DPTs. The authors propose four approaches to evaluating MTDs: attack based experiments, probability models, simulation based evaluation and hybrid approaches. In contrast, we propose closed-form mathematical modeling and stochastic modeling.

III. Model

In this section, we develop two DPT-based MTD modeling and evaluation techniques for the purpose of cross-validation. The first technique is based on closed-form solutions. The second technique is based on SPN modeling techniques. We will apply these two modeling and evaluation techniques to example DPT-based MTD techniques in Section IV.

We highlight four particular facets of our threat model: First, we assume a persistent attack that takes a certain amount of time and can be resumed. We note that in some cases attacks may have to restart from scratch; if this is the case, the attack can never succeed if the churn rate is faster than the completion rate. Second, we assume the attacker must implant some malware and persist on a targeted host. For example, our model does not apply to a DoS attack. Third, we assume that detecting and attributing an attacker will deter further efforts. In particular, the political consequences for a nation state or legal consequences for a criminal outweigh the potential rewards of a success attack. Finally, we assume a nondeterministic implant detection process.

A. Closed-Form Mathematical Model

Equation 1 calculates the probability a cyber attack will succeed. Intuitively, the probability a cyber attack will succeed is the likelihood an exploit is available for the target (first term) multiplied by the likelihood the implemented technique is successful (second term). The probability an exploit is available for the target is one minus the probability an exploit is not available for the target. The likelihood the implemented technique is successful is the complement of the probability of implant detection raised to the number of implants required. Equation 2 calculates the number of implants required: The number of implants required is the cyber attack length divided by churn time (the victim must be re-implanted after each virtual machine (VM) reset) multiplied by configuration count divided by two. To allow for closed-form solutions, we make two simplifying assumptions: First, configurations are distributed uniformly so the attacker will need to wait out half of the configurations on average. Second, the probability an exploit is available is the same for all configurations. As we will see later in Section IV, these assumptions greatly improve solution efficiency without compromising solution accuracy.

$$a = (1 - (1 - e)^o)(1 - p)^i \tag{1}$$

$$i = (c/h)(o/2) \tag{2}$$

In Equation 1, a indicates the probability of cyber attack success; this is the output of our closed-form mathematical model. e indicates the probability an exploit is available for a given configuration. Because all software has vulnerabilities, e is a function of the budget and/or skill level of the attacker. o indicates the number of MTD configurations; the defender chooses this value. p indicates the probability of implant detection; this is a function of the skill level of the attacker and the skill level of the defender. i indicates the number of implants required for the cyber attack. c indicates the cyber attack length in seconds. The attacker chooses this value because a mature attacker, even a recreational one, will have an objective in mind before going after the subject. For example, they may want to deface a web page, steal a database or compromise a program. The practitioner should use a high value for c (e.g., months) to model nation state attackers and a low value to model recreational hackers (e.g., hours). The lower success rate associated with higher cyber attack length does not mean that nation state attackers are less dangerous than recreational hackers because not all cyber attacks are equal: A recreational attack will likely have less severe consequences than a nation state attack. h indicates the MTD churn time in seconds; the defender chooses this value. Table I summarizes these parameters.

B. Stochastic Petri Net

Figure 2 shows our SPN model describing the ecosystem of a cyber engagement. The underlying model of the SPN

TABLE I
CLOSED FORM MATHEMATICAL MODEL PARAMETERS.

Parameter Name	Description
a	probability of cyber attack success
e	probability an exploit is available for a given configuration
o	number of MTD configurations
p	probability of implant detection
i	number of implants required
c	cyber attack length (s)
h	MTD churn time (s)

model is a continuous-time semi-Markov process with a state representation (PID, PSI1, PSI200, PCAS) where PID is a binary variable with 1 indicating the defender has detected the attacker and 0 indicating the attacker is undetected, PSI1 is the number of successful malware installations modulo 200, PSI200 is the number of successful malware installations integer divided by 200, and PCAS is a binary variable with 1 indicating the cyber attack has completed successfully and 0 indicating otherwise. PID is an absorbing state because a sophisticated attacker wants to avoid attribution at all costs and will abort the cyber attack after being detected. PCAS is an absorbing state because a sophisticated attacker will end the mission after accomplishing the objective to reduce the risk of detection and attribution. Due to a limitation of the analysis software [3] which restricts each place to 200 tokens, two places model the successful implant count component (PSI1 and PSI200). The number of tokens in each place is called the marking of the Petri net. The SPN model is constructed as follows:

- We use places to hold tokens. Initially, the attacker is undetected, no implants have occurred and the cyber attack has not succeeded. The initial state thus is (0, 0, 0, 0) in the underlying semi-Markov model.
- We use transitions to model events. Specifically, TCHURN models an MTD reconfiguration; TDETECTION models the defender detecting the attacker; TSUCCESS models the attacker implanting the defender; TSUFFICIENT models the attacker persisting on the defensive target long enough to complete the attack. Table II describes the functions governing these transitions.
- One timed transition, TCHURN, adds tokens to a vanishing state.
- After each churn, the attacker will need to reimplant the defender. During this process, the defender may detect the attacker; this is modeled by associating immediate transition TDETECTION with a probability. In this case, a token moves from the vanishing state to PID. On the other hand, the attacker may elude the defender; this is modeled by associating immediate transition TSUCCESS with the complement of TDETECTION's probability. In this case, a token moves from the vanishing state to PSI1.
- When 200 tokens accumulate in PSI1, they all move to become a single token in PSI200 via an immediate transition; this is a technical constraint of the SPN analysis software.
- If the attacker persists for long enough on the defender's system, they will reach their objective (i.e., exfiltration or disruption). In this case, a token is placed in PCAS.

TABLE II
STOCHASTIC PETRI NET PARAMETERS.

Transition Name	Function
TCHURN	$1/\text{churn time}$
TDETECTION	probability of detection
TSUCCESS	$1 -$ probability of detection
TSUFFICIENT	$\frac{\text{cyber attack length} \cdot \text{configuration count}}{\text{churn time} \cdot 2}$

Given churn time, probability of detection, cyber attack length and configuration count as input the underlying semi-Markov model of our SPN model can be solved using techniques such as SOR, Gauss Seidel or Uniformization [12] to yield the probability the cyber attack will be successful, as well as the expected values of PID, PSI1, PSI200 and PCAS at time t.

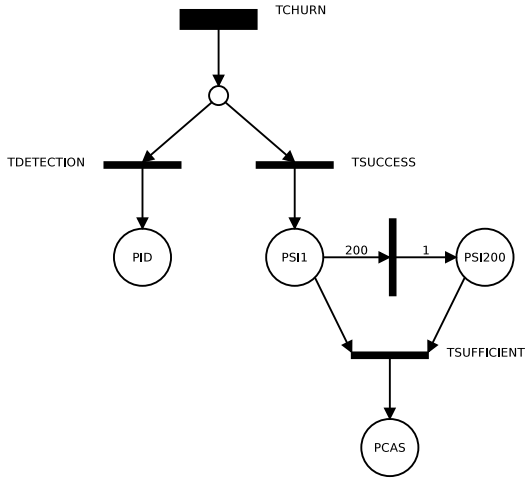

Fig. 2. Stochastic Petri net.

IV. RESULTS

In this section, we apply the two analytical models developed in Section III to DPT-based MTD scenarios. Figures 3 through 10 reflect the probability of attack success at a given moment in time; they do not consider the attacker goals which may have more (nation state) or less (recreational) consequences.

A. Closed-Form Mathematical Model

Figure 3 plots the probability of cyber attack success as a function of configuration count; it shows curves for cyber

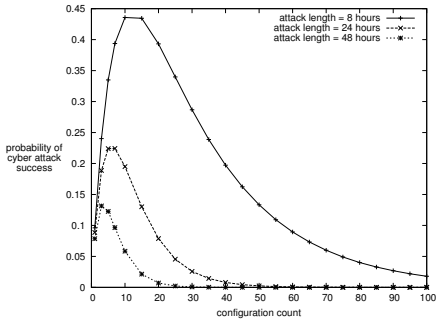

Fig. 3. Probability of cyber attack success versus configuration count and cyber attack length (closed-form model).

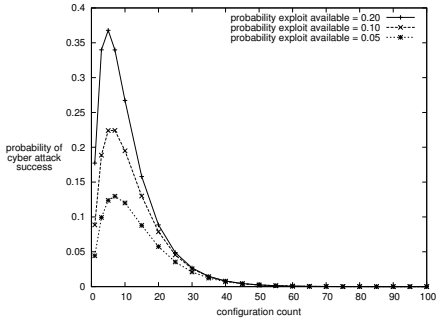

Fig. 4. Probability of cyber attack success versus configuration count and probability of exploit availability (closed-form model).

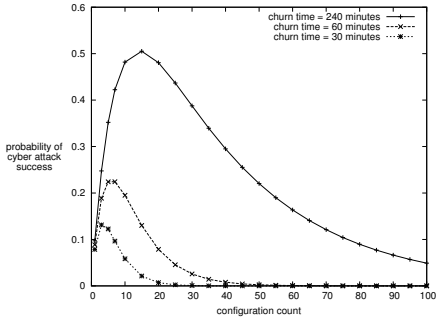

Fig. 5. Probability of cyber attack success versus configuration count and churn time (closed-form model).

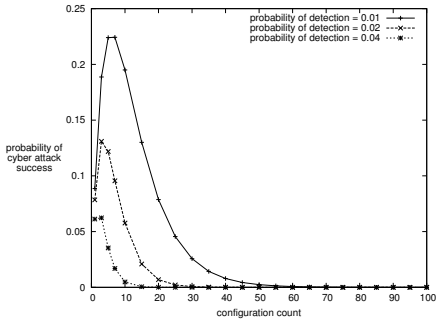

Fig. 6. Probability of cyber attack success versus configuration count and probability of implant detection (closed-form model).

attacks lasting three different amounts of time. This figure shows shorter cyber attacks are more likely to succeed. This is because a shorter attack will require fewer implants which provide less opportunities for an attack-ending detection. Figure 4 also plots the probability of cyber attack success as a function of configuration count; it shows curves for three different exploit availability probabilities. This figure shows cyber attacks are more likely to succeed if an exploit is more likely to be available. This is because there will be more victim configurations that are vulnerable to a piece of malware. Figure 5 also plots the probability of cyber attack success as a function of configuration count; it shows curves for three different churn times. This figure shows cyber attacks are more likely to succeed for higher churn times. This is because less frequent churns will require fewer implants which provide less opportunities for detection. Figure 6 also plots the probability of cyber attack success as a function of configuration count; it shows curves for three different probabilities of detection. This figure shows cyber attacks are more likely to succeed if the probability of detection is lower. This is because any implant will be less likely to be detected.

In addition to the expected basic trends, in all four graphs, we see two interesting phenomena: First, it is possible to make a system less secure by instrumenting a DPT-based MTD if the parameterization is unfavorable; this is because the attacker may have exploits for one platform in the MTD but not others. The left most point in each curve (configuration count equal to 1) represents a protected resource without DPT-based MTD instrumented. Dynamic platform technique based MTD is beneficial when the configuration count is above some breakeven point. This breakeven point is higher for shorter campaigns, higher exploit availabilities, higher churn times and lower probabilities of detection. Also, there is an optimal configuration count for the attacker. This optimal configuration count is lower for longer cyber attacks, higher exploit availabilities, lower churn times and higher probabilities of detection.

B. Stochastic Petri Net

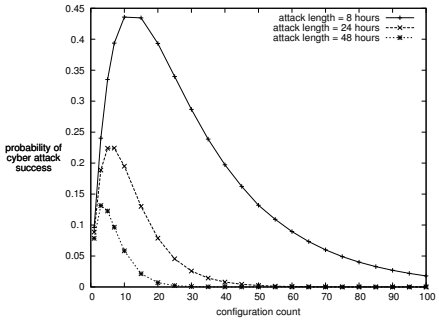

Fig. 7. Probability of cyber attack success versus configuration count and cyber attack length (stochastic model).

We apply the same MTD scenarios to the SPN model. We observe that Figures 7 - 10 generated from the SPN model match Figures 3 - 6 generated from the closed-form solutions

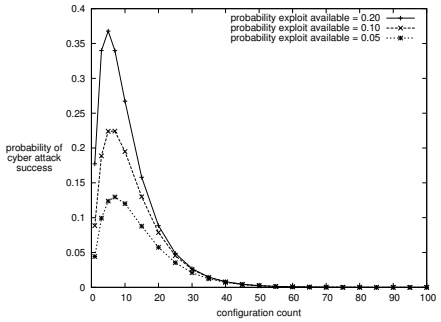

Fig. 8. Probability of cyber attack success versus configuration count and probability of exploit availability (stochastic model).

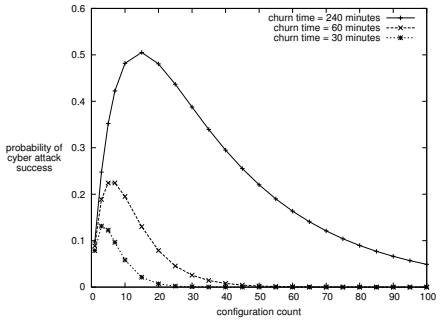

Fig. 9. Probability of cyber attack success versus configuration count and churn time (stochastic model).

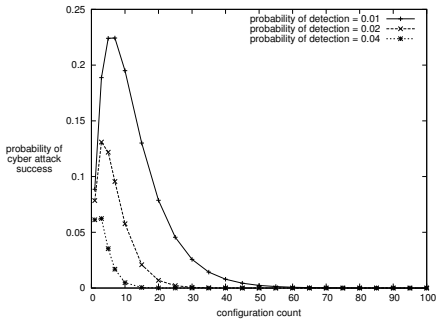

Fig. 10. Probability of cyber attack success versus configuration count and probability of implant detection (stochastic model).

very closely. This close match validates the analysis results presented in these figures. The mean square error between the closed-form and stochastic results is on the order of 10^{-9}.

V. Conclusions

In this paper, we developed two analytical models for evaluating the effect of DPT-based MTD on attack success rate. We showed that it is possible to mistakenly instrument an MTD in a way that makes the protected resource more vulnerable to attack. Consequently, given knowledge of the attacker strength and vulnerability in terms of attack length and exploit availability, we can identify the best defense parameter settings in terms of configuration count, implant detection probability, and churn time, under which DPT-based MTD is most effective to minimize the attack success probability. Two models, one closed-form and one stochastic, cross-validate and support these results.

There are four clear next steps in this line of investigation: First, we will instrument a simulation or emulation involving real, specific DPTs to further validate our models. Also, we will derive additional models that cover other forms of MTD, such as network based techniques, dynamic runtime environments and dynamic application code and data techniques. Third, our future threat models will consider attacks that must be restarted. Finally, we will relax four assumptions: that dynamic platform configurations are uniformly distributed, that the probability an exploit is available is the same for all configurations, that the adversary must implant malware in order to prosecute a cyber attack and that detection and attribution will deter an attacker.

References

[1] U.S. Department of Homeland Security. [Online]. Available: http://www.dhs.gov/science-and-technology/csd-mtd

[2] H. Okhravi, M. Rabe, T. Mayberry, W. Leonard, T. Hobson, D. Bigelow, and W. Streilein, "Survey of Cyber Moving Target Techniques," DTIC Document, Tech. Rep., 2013.

[3] G. Ciardo, J. Muppala, and K. Trivedi, "SPNP: stochastic Petri net package," in *Third International Workshop on Petri Nets and Performance Models*, Washington, DC, USA, December 1989, pp. 142–151.

[4] J. Hong and D. Kim, "Assessing the Effectiveness of Moving Target Defenses using Security Models," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, pp. 1–1, 2015.

[5] M. P. Collins, "A Cost-Based Mechanism for Evaluating the Effectiveness of Moving Target Defenses," in *Decision and Game Theory for Security*, ser. Lecture Notes in Computer Science, J. Grossklags and J. Walrand, Eds., 2012, vol. 7638, pp. 221–233.

[6] D. Evans, A. Nguyen-Tuong, and J. Knight, "Effectiveness of Moving Target Defenses," in *Moving Target Defense*, ser. Advances in Information Security, S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, Eds., 2011, vol. 54, pp. 29–48.

[7] H. Okhravi, J. Riordan, and K. Carter, "Quantitative Evaluation of Dynamic Platform Techniques as a Defensive Mechanism," in *Research in Attacks, Intrusions and Defenses*, ser. Lecture Notes in Computer Science, A. Stavrou, H. Bos, and G. Portokalidis, Eds., 2014, vol. 8688, pp. 405–425.

[8] K. Zaffarano, J. Taylor, and S. Hamilton, "A Quantitative Framework for Moving Target Defense Effectiveness Evaluation," in *Second ACM Workshop on Moving Target Defense*, ser. MTD '15, Denver, CO, USA, October 2015, pp. 3–10.

[9] M. Crouse, B. Prosser, and E. W. Fulp, "Probabilistic Performance Analysis of Moving Target and Deception Reconnaissance Defenses," in *Second ACM Workshop on Moving Target Defense*, ser. MTD '15, Denver, CO, USA, October 2015, pp. 21–29.

[10] R. Zhuang, S. A. DeLoach, and X. Ou, "A Model for Analyzing the Effect of Moving Target Defenses on Enterprise Networks," in *9th Annual Cyber and Information Security Research Conference*, ser. CISR '14, Oak Ridge, TN, USA, April 2014, pp. 73–76.

[11] J. Xu, P. Guo, M. Zhao, R. F. Erbacher, M. Zhu, and P. Liu, "Comparing Different Moving Target Defense Techniques," in *First ACM Workshop on Moving Target Defense*, ser. MTD '14, Scottsdale, AZ, USA, November 2014, pp. 97–107.

[12] R. A. Sahner, K. S. Trivedi, and A. Puliafito, *Performance and Reliability Analysis of Computer Systems*. Kluwer Academic Publishers, 1996.