SAND2015-10609C

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

***Abstract*—In this paper we propose closed form mathematical equations and a Petri net that model the effectiveness of a moving target defense (MTD). The numerical results from these two models agree with one another, providing internal validation. Furthermore, the output of these models indicates the existence of parameter families that decrease the security of the protected resource and parameter families that are optimal for the attacker.**

I. Introduction

Cyber security is a critical topic: sophisticated cyber attackers are motivated by power and money, and the systems they target are growing in complexity [10].

Our attack model is the six phase attack sequence comprising: survey, tool, implant, pivot, damage/exfiltration and cleanup activities illustrated in Figure 1. 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).

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. Moving target defense, a type of intrusion prevention, can not only stop the attacker from implanting, but can also disrupt the survey phase.

Van Leeuwen et al. [12] propose the following taxonomy of moving target defense (MTD) illustrated in Figure 2: network and host based techniques comprise the first layer of classification, and host based techniques are further classified into dynamic runtime environment, dynamic code and data and dynamic platform techniques.

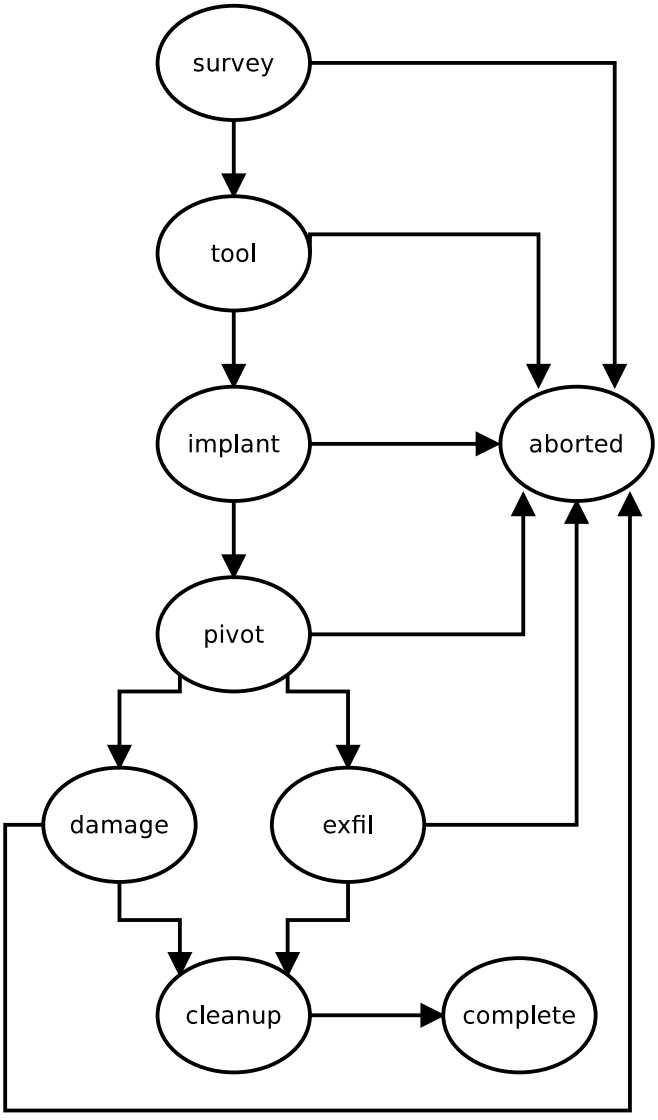

Fig. 1. Phased attack sequence.

In addition to proposing this taxonomy, Van Leeuwen et al. also discusses the possibility of MTD instrumentation doing more harm than good. That is, it is possible to provision an MTD in such a way as to decrease the security of a protected

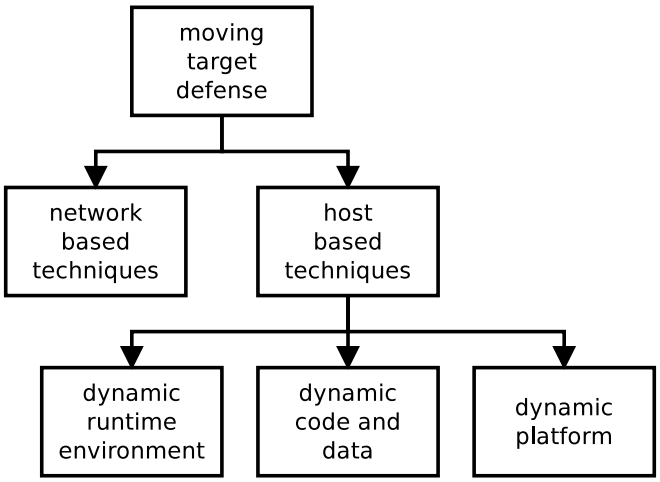

Fig. 2. Moving target defense taxonomy.

resource. To explore this concept further, we propose closed form mathematical equations and a Petri net to model the effectiveness of an MTD: specifically, a dynamic platform technique. The results of these models match one another, and each model is governed by five parameters.

II. Literature Search

Hong and Kim [6] 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. While it may lack some of the fidelity of Hong and Kim's approach, the closed form mathematical model we propose is more intuitive than HARM.

Collins [2] proposes a game theoretic way to assess the effectiveness of an 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 host based MTDs.

Evans et al. [4] propose a way to assess the effectiveness of an 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 can analyze dynamic platform based MTDs.

Okhravi et al. [9] propose another way to assess the effectiveness of an MTD. Their investigation focuses on evaluating dynamic platform techniques. The authors describe dynamic platform techniques 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. [14] 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, namely the valuation function, $v(\cdot)$, 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 constructed framework consists of large scale network emulation via hypervisor virtualization. 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 strong, they consider a phased attack sequence similar to what we illustrate in Figure 1. In future work, we will achieve this level of fidelity by instrumenting network emulation to further validate our results.

Crouse et al. [3] 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 probabalistic 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; good data for survey activity is hard to come by because of the large amount of noise from legitimate scanning and recreational hacking. For this reason, our predictive model focuses on the implant phase; there is no legitimate or recreational scenario for dropping malware on a protected resource.

Zhuang et al. [16] 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 [9], 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.

Xu et al. [13] survey current MTD techniques. Their MTD taxonomy comprises four categories: software based diversification, runtime based diversification, communication diversification and dynamic platform techniques. The authors propose four approaches to evaluating MTDs: attack based experiments, probability models, simulation based evaluation and hybrid approaches. Our work is a probability model for dynamic platform techniques.

Green et al. [5] survey current network based MTDs. They describe common elements of all network based MTDs: moving, access control and distinguishability. Also, the authors evaluate four contemporary network based MTDs.

Thompson et al. [11] study an implementation of the same MTD dynamic platform technique we consider in this paper: rotating OSs. They call their technique Multiple Operating System Rotation Environment (MORE). The authors' MTD taxonomy hinges on whether movement is proactive or reactive. Thompson et al. propose three metrics: probability of exploit, impact of exploit and availability; unfortunately, they do not provide numerical results.

Okhravi et al. [8] study a similar MTD dynamic platform technique to what we consider in this paper: platform diversity. They called their proposed design Trusted Dynamic Logical Heterogeneity System (TALENT); TALENT pauses an arbitrary critical infrastructure application written in C, serializes it, transfers it to another (potentially OS and hardware dissimilar) host and resumes it. This design is the foundation of [9]. The authors evaluated TALENT via executing an approximately 2000 line C application, which contained a GUI, remotely via SSH. Then they measured the time required to complete a migration of environments: The migration of an environment for MTD purposes was roughly 1 second.

Lucas et al. [7] propose a framework, called Evolutionary Algorithm (EA), for instrumenting an MTD that evolves computer configurations. A key innovation with their design is the destination for the movement (the new configuration) is informed by the current environment. The authors' framework comprises three components: one discovers new configurations, another component instantiates the new configurations, and a third component penetration tests the new configurations. Lucas et al. used fitness (unitless) and pairwise Hamming distance as metrics and compared their EA approach with randomly generated configurations.

Zhang et al. [15] propose an MTD technique where clouds are incentivized to migrate virtual machines (VMs) to different hosts. Their attack model includes side channel attacks. The authors use a game theoretic approach to compute the next host and other aspects of the move. While the paper measures their technique's impact to the defender, it does not consider the cost to the attacker.

III. Model

A. Closed Form Math Model

Equation 1 calculates the probability an information operation (IO) will succeed. Intuitively, the probability an IO 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 probability an exploit is not available for the target is the complement of the probability an exploit is available for a given configuration raised to the number of configurations. 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 IO length divided by churn time (the victim must be re-implanted after each virtual machine (VM) reset) multiplied by configuration count divided by two (assuming a uniform distribution, the attacker will need to wait out half of the configurations on average).

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

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

a indicates the probability of IO success; this is the output of our closed form mathematical model. Applying Equation 1 to the subsequent parameters yields a. 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

TABLE I
CLOSED FORM MATHEMATICAL MODEL PARAMETERS.

Parameter Name	Description
a	probability of IO 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	IO length (s)
h	MTD churn time (s)

IO. Applying Equation 2 to the subsequent parameters yields i. c indicates the IO length in seconds; the attacker chooses this value. The practitioner should choose a high value here (e.g., months) to model nation state attackers and low values here to model recreational hackers (e.g., hours). h indicates the MTD churn time in seconds; the defender chooses this value. Table I summarizes these parameters.

B. Stochastic Petri Net

There are three state components of the Markov chain underlying our Petri (also known as place/transition) net illustrated in Figure 3: the detection status of the IO, how many implants have been successful and the success of the IO. One place models the detection status component (PID); a token here indicates the defender has detected the IO. This is an absorbing state because a sophisticated attacker wants to avoid attribution at all costs and will abort the IO after being detected. Another place models the IO success component (PIOS); a token here indicates the IO has completed successfully. This 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 [1] which restricts each place to 200 tokens, two places model the successful implant count component (PSI1 and PSI200). A token in these places indicates one and 200 successful malware installations, respectively. The number of tokens in each place is called the marking of the Petri net; one Petri net marking equates to one node of the underlying Markov chain. Given two states for detection status, two states for success status and the need to accommodate at least 200 implants, the associated Markov chain would have at least $2 \cdot 2 \cdot 200 = 800$ nodes.

One timed transition, TCHURN, adds tokens to a vanishing state. An immediate transition, TDETECTION, moves a token from the vanishing state to the intrusion detected state with some probability. Another immediate transition, TSUCCESS, moves a token from the vanishing state to the implants successful x1 state with some probability (the complement of the TDETECTION probability). A third immediate transition removes 200 tokens from the implants successful x1 state and adds one token to the implants successful x200 state. A fourth immediate transition, TSUFFICIENT, adds a token to the IO successful state when there are sufficient tokens in the implants successful places. Table II describes the functions governing these transitions.

TABLE II
STOCHASTIC PETRI NET PARAMETERS.

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

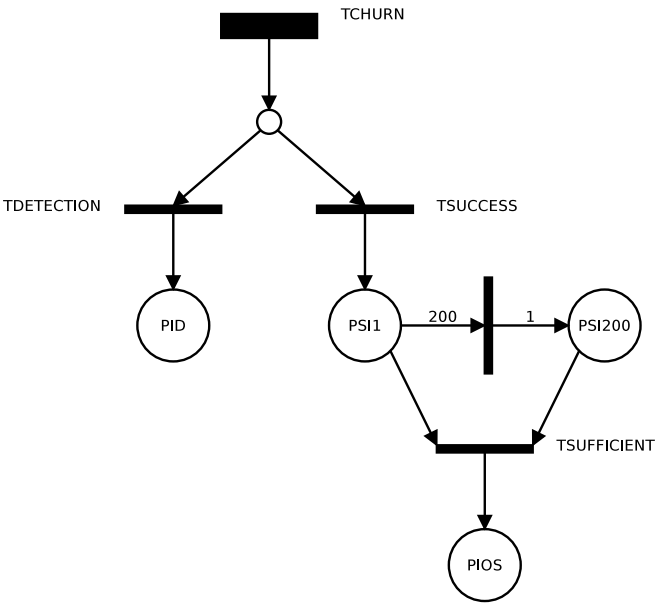

Fig. 3. Stochastic Petri net.

IV. RESULTS

A. Closed Form Math Model

The basic trends for Figures 4 through 7 are as expected. First, Figure 4 shows shorter IOs are more likely to succeed. Next, Figure 5 shows IOs are more likely to succeed if exploits are more readily available. Third, Figure 6 shows lower MTD strength (equivalent to higher churn time) will increase the likelihood of IO success. Finally, Figure 7 shows IOs are more likely to succeed if probability of implant detection is lower.

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 an MTD if the parameterization is unfavorable. The left most point in each curve (configuration count equal to 1) represents a protected resource without MTD instrumented. 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

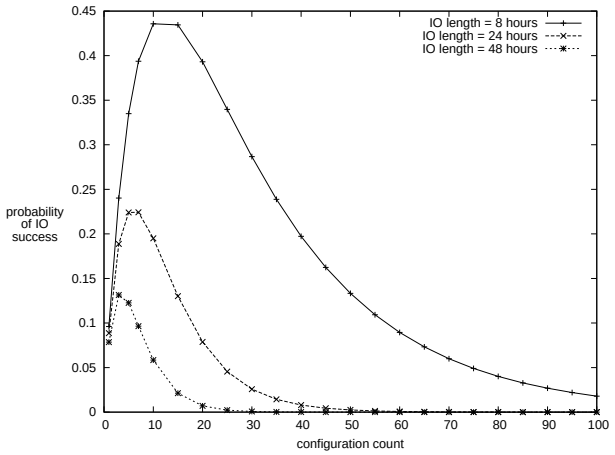

Fig. 4. Probability of IO success versus configuration count and IO length (closed form model).

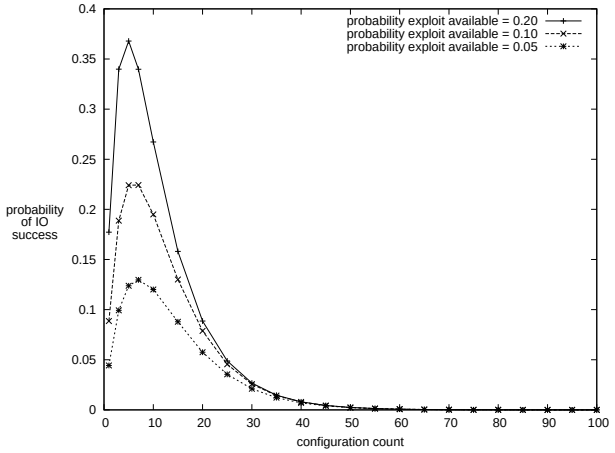

Fig. 5. Probability of IO success versus configuration count and probability of exploit availability (closed form model).

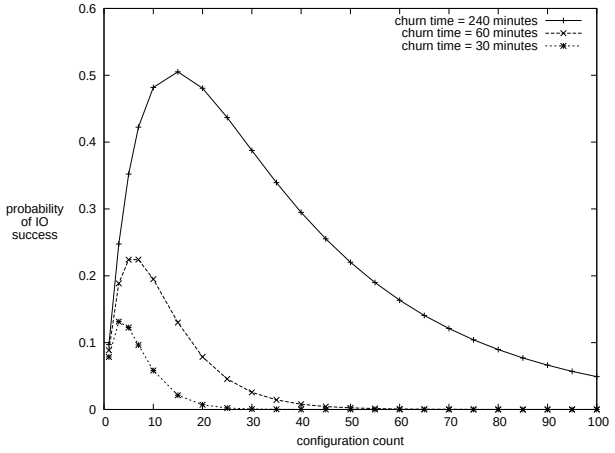

Fig. 6. Probability of IO success versus configuration count and churn time (closed form model).

detection. Also, there is an optimal configuration count for the attacker. This optimal configuration count is lower for longer IOs, higher exploit availabilities, lower churn times and higher probabilities of detection. These relationships make sense intuitively.

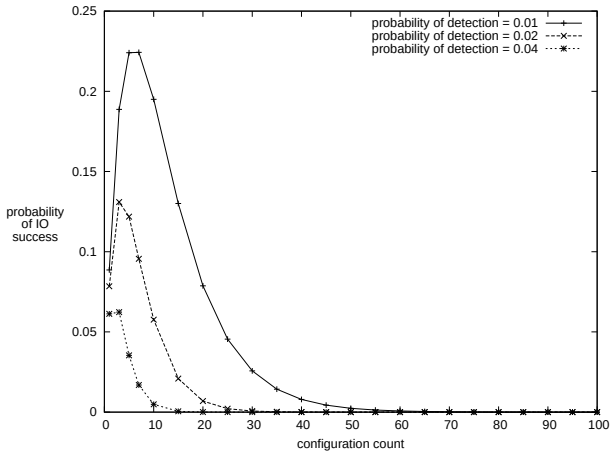

Fig. 7. Probability of IO success versus configuration count and probability of implant detection (closed form model).

B. Stochastic Petri Net

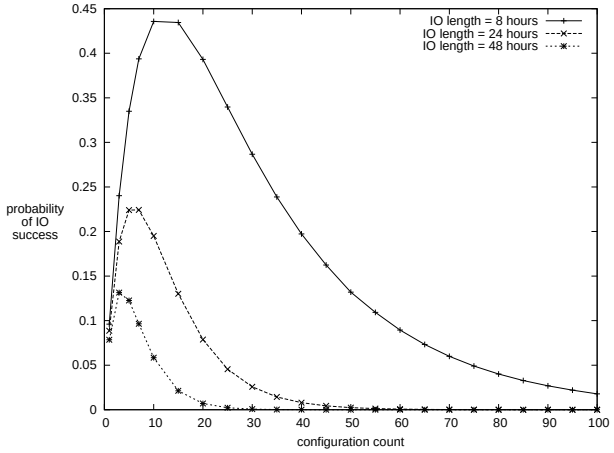

Fig. 8. Probability of IO success versus configuration count and IO length (stochastic model).

As expected, Figures 8 - 11 match Figures 4 - 7 very closely. The mean squared error between the closed form and stochastic results are on the order of 10^{-9}.

V. Conclusions

In this paper, we showed that it is possible to instrument an MTD in a way that makes the protected resource more vulnerable to attack. Furthermore, we identified parameter families for which the MTD is optimally configured from the attacker perspective. Two models, one closed form and one stochastic, independently support these results.

There are two clear next steps in this line of investigation: First, we will instrument a simulation or emulation to further

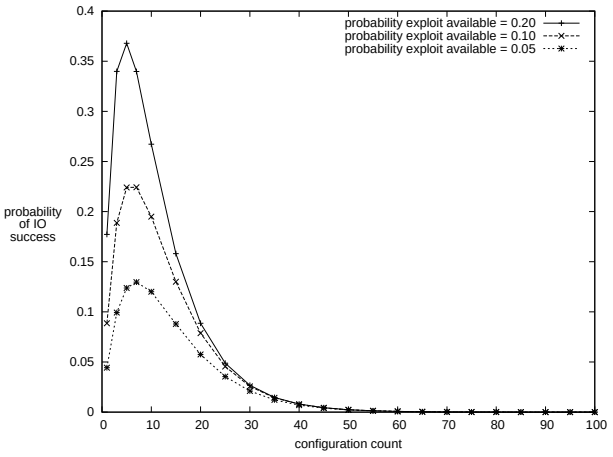

Fig. 9. Probability of IO success versus configuration count and probability of exploit availability (stochastic model).

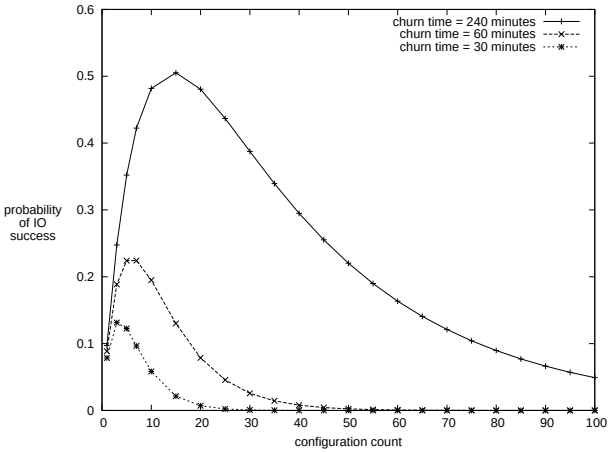

Fig. 10. Probability of IO success versus configuration count and churn time (stochastic model).

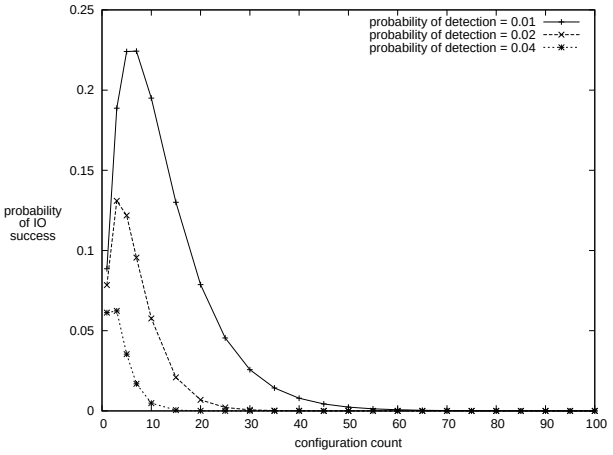

Fig. 11. Probability of IO success versus configuration count and probability of implant detection (stochastic model).

strengthen the results we already identified. Also, we will derive additional models that cover other forms of MTD, such as network based techniques, dynamic runtime environments and dynamic code and data techniques.

REFERENCES

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

[2] M. Patrick Collins. A Cost-Based Mechanism for Evaluating the Effectiveness of Moving Target Defenses. In Jens Grossklags and Jean Walrand, editors, *Decision and Game Theory for Security*, volume 7638 of *Lecture Notes in Computer Science*, pages 221–233. 2012.

[3] Michael Crouse, Bryan Prosser, and Errin W. Fulp. Probabilistic Performance Analysis of Moving Target and Deception Reconnaissance Defenses. In *Second ACM Workshop on Moving Target Defense*, MTD '15, pages 21–29, Denver, CO, USA, October 2015.

[4] David Evans, Anh Nguyen-Tuong, and John Knight. Effectiveness of Moving Target Defenses. In Sushil Jajodia, Anup K. Ghosh, Vipin Swarup, Cliff Wang, and X. Sean Wang, editors, *Moving Target Defense*, volume 54 of *Advances in Information Security*, pages 29–48. 2011.

[5] Marc Green, Douglas C. MacFarland, Doran R. Smestad, and Craig A. Shue. Characterizing Network-Based Moving Target Defenses. In *Second ACM Workshop on Moving Target Defense*, MTD '15, pages 31–35, Denver, CO, USA, October 2015.

[6] J.B. Hong and D.S. Kim. Assessing the Effectiveness of Moving Target Defenses using Security Models. *IEEE Transactions on Dependable and Secure Computing*, PP(99):1–1, 2015.

[7] Brian Lucas, Errin W. Fulp, David J. John, and Daniel Cañas. An Initial Framework for Evolving Computer Configurations As a Moving Target Defense. In *9th Annual Cyber and Information Security Research Conference*, CISR '14, pages 69–72, Oak Ridge, TN, USA, April 2014.

[8] Hamed Okhravi, Adam Comella, Eric Robinson, and Joshua Haines. Creating a cyber moving target for critical infrastructure applications using platform diversity. *International Journal of Critical Infrastructure Protection*, 5(1):30–39, 2012.

[9] Hamed Okhravi, James Riordan, and Kevin Carter. Quantitative Evaluation of Dynamic Platform Techniques as a Defensive Mechanism. In Angelos Stavrou, Herbert Bos, and Georgios Portokalidis, editors, *Research in Attacks, Intrusions and Defenses*, volume 8688 of *Lecture Notes in Computer Science*, pages 405–425. 2014.

[10] Bruce Schneier, January 2013.

[11] M. Thompson, N. Evans, and V. Kisekka. Multiple OS rotational environment an implemented Moving Target Defense. In *7th International Symposium on Resilient Control Systems*, pages 1–6, Denver, CO, USA, August 2014.

[12] B. van Leeuwen, W. Stout, and V. Urias. Operational Cost of Deploying Moving Target Defense: Defensive Work Factors. In *MILCOM*, Tampa, FL, USA, October 2015.

[13] Jun Xu, Pinyao Guo, Mingyi Zhao, Robert F. Erbacher, Minghui Zhu, and Peng Liu. Comparing Different Moving Target Defense Techniques. In *First ACM Workshop on Moving Target Defense*, MTD '14, pages 97–107, Scottsdale, AZ, USA, November 2014.

[14] Kara Zaffarano, Joshua Taylor, and Samuel Hamilton. A Quantitative Framework for Moving Target Defense Effectiveness Evaluation. In *Second ACM Workshop on Moving Target Defense*, MTD '15, pages 3–10, Denver, CO, USA, October 2015.

[15] Yulong Zhang, Min Li, Kun Bai, Meng Yu, and Wanyu Zang. Incentive Compatible Moving Target Defense against VM-Colocation Attacks in Clouds. In Dimitris Gritzalis, Steven Furnell, and Marianthi Theoharidou, editors, *Information Security and Privacy Research*, volume 376 of *IFIP Advances in Information and Communication Technology*, pages 388–399. 2012.

[16] Rui Zhuang, Scott A. DeLoach, and Xinming Ou. A Model for Analyzing the Effect of Moving Target Defenses on Enterprise Networks. In *9th Annual Cyber and Information Security Research Conference*, CISR '14, pages 73–76, Oak Ridge, TN, USA, April 2014.