Moving Target Defense for Web Applications using Bayesian Stackelberg Games

(Extended Abstract)

Satya Gautam Vadlamudi, Sailik Sengupta, Marthony Taguinod, Ziming Zhao,
Adam Doupé, Gail-Joon Ahn, Subbarao Kambhampati
Department of Computer Science, Arizona State University
{gautam, sailik.sengupta, mtaguino, zmzhao, doupe, gahn, rao}@asu.edu

ABSTRACT

Vulnerabilities in web applications allow hackers to access and/or modify restricted data. Here the hackers have the opportunity to perform reconnaissance so as to gain knowledge about the web application layout before launching an attack, whereas the defender (administrator of the web application) must secure the application even with its potential vulnerabilities. In order to mask such vulnerabilities which are primarily associated with different individual configurations, Moving Target Defense systems were proposed wherein the defender switches between various configurations thereby making it difficult to attack with success, while maintaining a seamless experience for the genuine users. In this paper, we present a way to find effective switching strategies by modeling this ecosystem as a Bayesian Stackelberg game with the administrator as the leader and the hackers as the followers, which as we show succinctly captures various aspects of the Moving Target Defense systems. Furthermore, we propose ways to find the most critical vulnerabilities and the most sensitive attacker types, which are key issues in such scenarios.

Keywords

Cyber security; Web applications; Moving target defense; Bayesian Stackelberg games

1. INTRODUCTION

Web application vulnerabilities pose risks to the security and privacy of both businesses and users. Several techniques and tools based on static analysis (white-box) and dynamic analysis (black-box) have been proposed to discover the vulnerabilities in web applications, so that the vulnerabilities can be removed before the attackers discover and exploit them. However, such efforts are not enough due to the increasing complexity of modern web applications and the limited development and deployment time, whereas the attackers can perform reconnaissance and attack. To address these challenges, a Moving Target Defense (MTD) based approach was proposed in [2] to secure web applications, which complements the aforementioned vulnerability analysis techniques through a defense-in-depth mechanism.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

The MTD based approach dynamically configures and shifts systems over time, at different layers of the web application stack, to increase the uncertainty and complexity for the attackers with their probing and attacking, while ensuring that the system is available for legitimate users.

In this paper, we focus on the design of effective policies for movement in the MTD system that maximize the resilience of the web application, given the set of components and configurations of the system which can be "moved around." We show that many of the features of this problem such as attacker reconnaissance and attacker type uncertainty can be captured effectively by modeling it as a Bayesian Stackelberg Game (BSG). Our main contributions are: 1) An effective way to model MTD as BSG, 2) Showcase how the modeling leads to finding better MTD policies, 3) A methodology to find most critical vulnerabilities, and 4) A methodology to find most sensitive attacker types. To the best of our knowledge, this is the first work that models Moving Target Defense as a Bayesian Stackelberg game. The full version of this extended abstract is available as a technical report [3].

2. MODELING MOVING TARGET DEFENSE AS BAYESIAN STACKELBERG GAME

A Stackelberg game consists of a leader who commits to a strategy first, and then a follower adopts a strategy which maximizes its own reward based on the leader's strategy. Here the leader has an advantage because she gets to make the first move (choose a strategy and commit), and therefore can restrict the moves of the follower in a sense. A Bayesian game contains a set of n agents ($n = 2$ in this case, leader and follower) where each agent i could be of any of the T_i types. Here, each agent i has a set of strategies available to it S_i and a reward/utility function $R_i : T_1 \times T_2 \times S_1 \times S_2 \rightarrow \mathbb{R}$. The objective then is to find an optimal mixed strategy for the leader to commit to (one which maximizes her rewards), given that the follower(s) may know that strategy when choosing their strategy.

The defender in MTD can be modeled as the leader in BSG, and the attacker- as the follower. Whereas the defender (web application) is of a single type, clearly the attacker could be of multiple types consistent with our formulation of Bayesian Stackelberg games above. Note that, the movements of the defender can be viewed as choosing strategies as follows: different moves in MTD would lead to different configurations at different layers (we call each set of valid configurations as a 'combination'). So, a move in

MTD from one particular combination to another combination could be modeled as a strategy in BSG. And, each of the attack options of the attacker in MTD could be viewed as a strategy for the follower in BSG. Note that, each attacker type in MTD has a set of attack options which may or may not be overlapping with other attacker types.

Whether a particular attacker type has access to a given attack option is dependent on (and hence formulated based on) the following criteria:

- The difficulty of carrying out the action/attack.
- The popularity of the target technology, as more popular implementations of services result in higher chance of exploits and wider array of available tools.

The choice of reward values for the defender and the attacker are guided by the following considerations:

- The effect of the action/attack on the target website. Note that, the "effect" here is dependent on the particular website, since a very complex and clever attack may sometimes only be able to uncover data which is less critical to the defender (website). Also, it is dependent on how much an attacker can leverage out of the data acquired. Therefore, the rewards here are based on both the defender (the criticality of various data/functionality) and the attacker type (ability to leverage the success of various attacks).
- Whether the attack leads to detection of the attacker, which may in-turn depend on aspects such as the number of exploits they target in each attack.

The uncertainty in the attacker types is captured by associating probabilities with each of them that serve as relative estimates of each of such attackers attacking the web application under consideration. Once the attacker types, the associated probabilities, their attack options, and the reward values for the defender and the attacker types are decided for each attack, an optimal mixed strategy for the defender could be obtained by using any BSG solver such as [1].

In our experiments comparing the BSG based strategy with the uniform mixed strategy (UMS) on a representative example, we observed that, on an average (and also in most cases), the UMS results in a negative reward value of -0.50 for the defender, whereas the BSG based mixed strategy results in a positive reward value of 1.14, demonstrating its clear advantage (winning as opposed to losing). For more details on the methodology and results, please see [3].

3. FINDING CRITICAL VULNERABILITIES

We formulated and solved the problem of finding most critical vulnerabilities in web applications, which is a much needed feature for defenders, since they usually have to put their limited time and energy to harden the weakest link in their system once the system is deployed (or even before the system is deployed). In this scenario, it is useful to focus on those vulnerabilities or attacks first which are not effectively masked by the optimal mixed strategy with which the system is deployed. These are the most critical vulnerabilities for the current deployment of the system.

We consider the generic problem of finding k-most critical vulnerabilities where k is an input that can be chosen based on the resources available to work on them simultaneously. Our approach works by performing posterior analysis of addressing various vulnerabilities thereby knowing which subset of them would have led to larger gains. For more details on the methodology and results, please see [3].

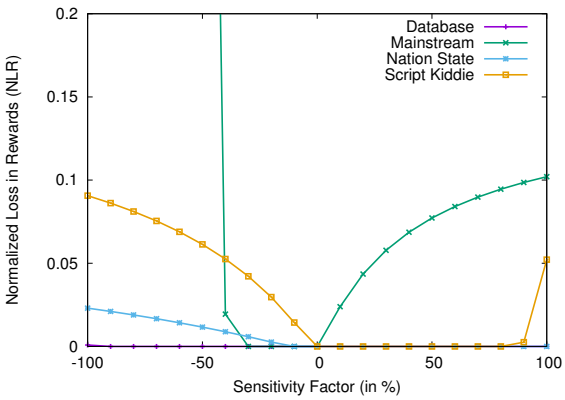

Figure 1: NLR values for various attacker types when their probabilities are modified ranging from -100% to 100%.

4. FINDING SENSITIVE ATTACKER TYPES

It is often the case that the web application administrator (defender) does not know accurately the probability with which a particular type of attacker might attack the system. Therefore, we have formulated and solved the problem of finding the most sensitive attacker types. This is important since a change in their probability will influence the optimal reward for defenders the most, which in-turn helps in knowing which probabilities are to be estimated with as high precision as possible for an effective MTD deployment.

We propose a methodology to find a real-valued quantity $\in [0, 1]$ called the *Normalized Loss in Rewards (NLR)* to measure the sensitivity of different attacker types. The attacker types for which the loss (NLR value) is higher than the rest are the most sensitive attacker types. Figure 1 shows the results obtained on an example with four attacker types: database, mainstream, nation state and script kiddie, in which the mainstream hacker and the script kiddie turn out to be the most sensitive ones.

5. SUMMARY

We have shown how MTD can be effectively modeled as BSG. For more details on the modeling, solving BSGs, working examples, criticality and sensitivity analysis, and experimental results, please see [3].

Acknowledgments

This work was partially supported by the grants from ONR N00014-13-1-0176, N00014-13-1-0519 and N00014-15-1- 2027, the ARO grant W911NF-13- 1-0023, the NSF grant NSF-SFS-1129561 and the Center for Cybersecurity and Digital Forensics at ASU.

REFERENCES

[1] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus. Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. In *Proceedings of the 7th AAMAS-Volume 2*, pages 895–902, 2008.

[2] M. Taguinod, A. Doupé, Z. Zhao, and G.-J. Ahn. Toward a Moving Target Defense for Web Applications. In *Proceedings of 16th IEEE International Conference on Information Reuse and Integration (IRI)*, 2015.

[3] S. G. Vadlamudi, S. Sengupta, S. Kambhampati, M. Taguinod, Z. Zhao, A. Doupé, and G.-J. Ahn. Moving Target Defense for Web Applications using Bayesian Stackelberg Games. *CoRR*, abs/1602.07024, 2016.