Moving Target Defense: A Symbiotic Framework for AI & Security

(Doctoral Consortium)

Sailik Sengupta
Advisor: Subbarao Kambhampati
PhD Student, Computer Science
Arizona State University, Tempe
sailiks@asu.edu

ABSTRACT

Modern day technology has found its way into every aspect of our lives– be it the server storing our social information, the hand-held smartphones, the home security systems or a remotely monitored pacemaker. Unfortunately, this also increases the opportunity for agents with malicious intent to violate the privacy, availability or integrity of these applications. In fact, with the advancement of Artificial Intelligence (AI) and faster hardware, the process of finding and exploiting vulnerabilities is no longer as time-consuming as before. Moving Target Defense (MTD) is emerging as an effective technique in addressing these security concerns. This technique, as used by the cyber security community, however, does not incorporate the dynamics of a multi-agent system between an attacker and defender, resulting in sub-optimal behavior. My study of such systems in a multi-agent context helps to enhance the security of MTD systems and proposes a list of challenges for the AI community. Furthermore, borrowing the example of MTD systems from the cyber security community, we can address some security concerns of the present day AI algorithms. In this abstract, I describe my research work that uses AI for enhancing security of a multi-agent MTD system and highlight research avenues in using MTD for enhancing security of present AI algorithms.

1. INTRODUCTION

Achieving software security with the present complexity in designing web applications is a difficult goal. Attackers can explore a deployed service on the web and attack it at their own leisure. Moving Target Defense (MTD) in web applications has been proposed as an effective mechanism to nullify this advantage of their reconnaissance [1]. However, MTD demands a good switching strategy when switching between multiple configurations for the web application stack. In section 2, we talk of our work [2, 3] that uses game theoretic techniques to help in addressing this issue.

In section 3, we discuss the issues involved in making AI systems robust to cyber attacks. We look at vulnerabilities in deployed machine learning systems, possible attacks on them, and whether MTD can help in securing these systems.

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

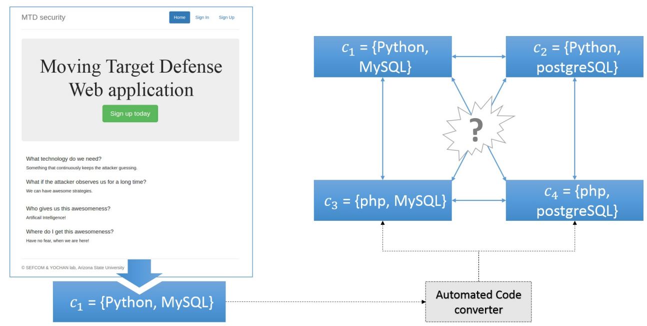

Figure 1: A Moving Target Defense web application

2. AI FOR SECURITY

For designing a good switching strategy, one first needs to formalize the notion of a switching strategy and its measure of 'goodness'. To this end, we consider the MTD system for web applications in a multi-agent framework with two players (Defenders and Attackers) and formulate a repeated Bayesian game [2]. We then show that this formulation is similar to existing frameworks developed for Physical Security Games [4]. This helps in leveraging existing solvers for finding an effective switching strategy for a web application.

After interacting with our team of cybersecurity experts to use this for a real-world application, we realized that the cost of switching from one configuration to another was non-trivial. Unfortunately, existing work had not considered this key limitation, rendering the existing solvers inefficient for our application. Moreover, populating the game matrix with reward values for the attacker and defender was a challenging requirement. In [3], we address both of these issues. For the former issue, we formulate an optimization problem that maximizes security while minimizing the non-trivial costs of switching between configurations. To address the latter limitation, which was even more relevant in the context of a real-world MTD system, we mine attack data from the Common Vulnerabilities and Exploits (CVE) Database. To generate meaningful reward values we leverage the knowledge of the security community present in the Common Vulnerability Scoring System (CVSS). With these we generate efficient switching strategies for a Moving Target Defense web-application (see Figure 1). We show that this is better than the state-of-the-art switching strategies used in current MTD web applications.

Cyber security systems provide, by default, a multi-agent context. Thus, one needs to consider both the aspects of cyber security and multi-agent environments to design system behaviours that provide formal bounds on security of the application. Rather than being a straightforward application of AI techniques, the cybersecurity domain also provides fresh research challenges to the AI community, as we saw in [3]. In case of MTD systems, incorporating evolution of defender configurations, attacker attacks and rewards values in the Game Theoretic framework raises the question of what can we say about optimal strategies in Repeated Games with evolutionary game metrics.

3. SECURITY FOR AI

In my ongoing work, I am looking at using MTD for ensuring safety of AI agents. With the use of Machine Learning algorithms in applications that affect our day to day life, we are vulnerable to attacks that seek to guide the intelligence of these approaches for malicious purposes. Consider an automated handwritten check reader at the ATM machine near you. If a malicious depositor were to put a few dots over the digit 1 so that the machine interprets it as a 9, (s)he might be able to withdraw $900 instead of the $100 you had planned to give him(/her). There are existing works that show that such manipulation of state-of-the-art machine learning algorithms is feasible if one can guess the type of network architecture used for such classification [5]. Although it is possible to design solutions for preventing such security compromises by reverse engineering specific attacks or incorporating adversarial examples into the train data, it is worth investigating if foundations for a general security measure is possible here.

An interesting approach would be to design multiple learners from the same testing data to keep an adversary guessing about the correct classification boundary which would make designing model-based attacks tougher. Although this Moving Target Defense approach seems related to the notion of Ensemble models, the goal of the system is to prevent adversarial samples from being misclassified as opposed to increasing classification robustness. For an MTD system to succeed in thwarting attacks, the different configurations need to have *differential immunity*. This means that adversarial samples generated for one model are ineffective (i.e. correctly classified) by all other models in the system. Existing literature has investigated such measures in the context of linear classifiers with binary labels [6], but lacks formal guarantees when these frameworks are investigated in an attacker-defender multi-agent context.

A simple idea would be to divide the data set into parts and use them to train different models for creating the configurations for the MTD framework. For the case of learning networks, as shown in [7], this idea does not provide differential immunity. On the other hand, using different network architectures to obtain the various models for creating the configurations of the MTD framework is something we are investigating at present (Figure 2). In such cases, maximizing security without sacrificing classification accuracy becomes a challenging requirement.

In the context of model-based scenarios like Markov Decision Processes or Automated Planning, it seems to be possible for an adversary to design reward shaping mechanisms for agents (without complete information) or use an agent's reward function to make them behave in an unforeseen manner, which may lead to dire consequences. Consider a case where a ball-catching robot is learning its reward function. An attacker learns that it has extremely high reward for catching a ball thrown at it. As the agent is deployed, the adversary can throw a ball off a cliff and tempt the robot to jump off the cliff. Notice that such instances are different from unintended consequences resulting out of bad reward function designs [8] where the robot exploits the knowledge about reward functions for itself. I plan to investigate these directions, identifying concrete problems that can lead to adversarial compromise of AI agents in these scenarios.

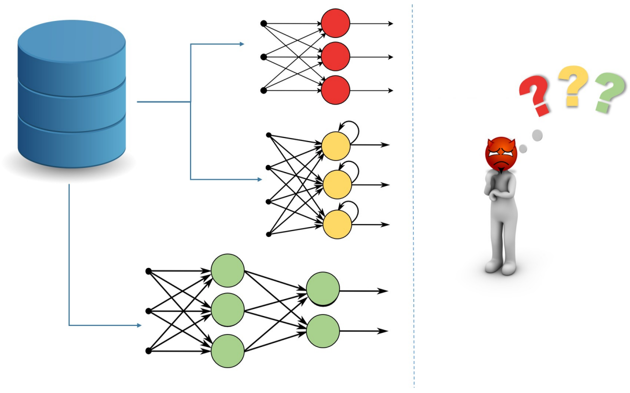

Figure 2: Learning networks with different architectures from the same data can make it difficult for an attacker to craft malicious examples for intended mis-classification by a particular network

Acknowledgments. This research is supported in part by ONR grants N00014161-2892, N00014-13-1-0176, N00014-13-1-0519, N00014-15-1-2027, & the NASA grant NNX17AD06G.

REFERENCES

[1] M Taguinod, A Doupé, Z Zhao, and G Ahn. Toward a moving target defense for web applications. In *IEEE International Conference on Information Reuse and Integration (IRI), 2015*, pages 510–517.

[2] S. Vadlamudi, S. Sengupta, M. Taguinod, Z. Zhao, A. Doupé, G. Ahn, and S. Kambhampati. Moving target defense for web applications using bayesian stackelberg games. In *AAMAS, 2016*.

[3] S. Sengupta, S. Vadlamudi, S. Kambhampati, A. Doupé, M. Taguinod, Z. Zhao, and G. Ahn. A game theoretic approach in strategy generation for moving target defense in web applications. In *AAMAS, 2017*.

[4] A. Sinha, T.H. Nguyen, D. Kar, M. Brown, M. Tambe, and A. X. Jiang. From physical security to cyber security. *Journal of Cybersecurity*, 2016.

[5] N Papernot, P McDaniel, S Jha, M Fredrikson, Z Celik, and A Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE*.

[6] B Biggio, G Fumera, and F Roli. Adversarial pattern classification using multiple classifiers and randomisation. In *Joint IAPR International Workshops on Structural, Syntactic, and Statistical Pattern Recognition*, 2008.

[7] C Szegedy, W Zaremba, I Sutskever, J Bruna, D Erhan, I Goodfellow, and R Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[8] D Amodei, C Olah, J Steinhardt, P Christiano, J Schulman, and D Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.