Adapting with Honeypot Configurations to Detect Evolving Exploits

(Extended Abstract)

Marcus Gutierrez
University of Texas at El Paso
500 W University Ave.
El Paso Texas
mgutierrez22@miners.utep.edu

Christopher Kiekintveld
University of Texas at El Paso
500 W University Ave.
El Paso Texas
cdkiekintveld@utep.edu

ABSTRACT

Honeypots are decoy cyberdefense systems placed in a network to entice malicious entities into attacking in order to waste attacker resources and learn information about attack behavior or previously unknown exploits. We focus on the strategic selection of various honeypot configurations in order to adapt to an intelligent attacker amidst a dynamic environment. In order to infiltrate networks, attackers leverage various exploits on the system. However, these exploits and the value they provide dynamically change over time as more information is gathered about them. We introduce a model that addresses the combinatorial complexity of the honeypot selection problem and allow for these dynamic exploits. To solve this new problem, we map this model to a Multi-Armed Bandit (MAB) problem, which is a class of machine learning problems that maintain balance between exploration and exploitation. We show empirically that both stochastic and adversarial MAB solutions improve over static defense strategies.

Keywords

Honeypot, Honeynet, Exploit, Machine Learning, Adversarial Learning, Modelling and Simulation

1. INTRODUCTION

One of the central problems in cybersecurity is detecting and monitoring attacker behavior. A key aspect of attacker behavior that we want to detect is what types of exploits attackers are using to execute attacks. The types of exploits attackers are using can change dramatically over time; some forms of attacks may become more or less common, and new types of attacks may be introduced. Detecting any type of attempted attack provides useful information about attacker behavior, but an especially important case is zero-day attacks. Zero-day attacks are based on previously unknown exploits, so they are more dangerous and difficult to detect because there are no known countermeasures [3]. Our work focuses on the problem of detecting and monitoring exploits in this dynamic environment.

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.*

We focus on honeypot selection as our method for detecting attacker behavior. Honeypots are fake computer systems or services that are designed to appear like legitimate systems to an attacker [10]. Honeypots log all malicious activity with low false positive rates as malicious entities primarily interact with them. This allows for the gathering of additional information about the attacker's capabilities and intentions.

We address one of the most important limitations of honeypots for detecting exploits. A honeypot must present itself as a particular type of host or service, such as a server running a particular version of an operating system, with certain open ports, installed software packages, etc. [10]. This means that any particular honeypot presents a limited attack surface that is only vulnerable to attacks using certain types of exploits. Due to costs of deploying high-quality honeypots, we are limited to how many honeypots we can deploy, leading to a combinatorially complex problem.

The model we propose addresses this combinatorial honeypot selection problem. We use KFSensor as a motivating example of how to map this problem. KFSensor is a commercial intrusion detection system that deploys high-interaction honeypots on a network [4]. This software requires a high cost in time to develop strategic combinations of honeypots to choose from, known as scenarios, but switching between scenarios is quick and effortless. While our methodology could be used with other types of honeypot software (such as HoneyD [9]), we focus on KFSensor as an example because it is readily available and highly customizable.

Píbil et. al. consider a similar problem, but take a game-theoretic approach [8] that model the decision space as a static decision. Meanwhile, Klíma et. al. provide a similar approach as ours to the domain of Border Patrol [6]. They model an online, repeated interaction between a defender and attacker where the defender must patrol zones. Similar to our work, they leverage the Combinatorial Multi-Armed Bandit framework and provide empirical analysis on an attacker agent derived from fictitious play. Our work differs from Klíma et. al.'s due to our non-linear reward structure and dynamically changing exploits.

2. MODEL

Our model captures two important features of the exploit detection problem. First, the defender has limited resources and must selectively choose which types of honeypots to

deploy. Second, the set of exploits that can be used (and detected) changes dynamically over time, both by introducing new exploits and changing the value of existing exploits. We consider a repeated interaction between a defender and an attacker. The defender faces a choice in each round of which set of available honeypot configurations to deploy in that round. The attacker's decision in each round is which exploit to use to try to execute an attack. A high-level visualization of the interaction is show in Figure 1.

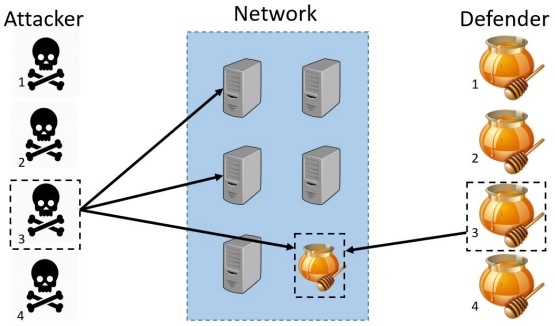

Figure 1: High-level visualization of model. Defender inserts honeypots into the network and the adversary selects an attack that targets the servers on the network (which may be honeypots).

We assume that the defender has a static network N of real systems to protect each with a value. In each round, the defender selects from a list of honeypot configurations to deploy in the network. Each honeypot configuration contains a unique combination of features that define its vulnerabilities. Meanwhile, the attacker selects an exploit to attempt an attack on the network. Each exploit has a list of minimum required features that help determine if a system is said to be vulnerable to said exploit. This forms a many-to-many bipartite graph between exploits to honeypot configurations. The defender is aware of the features each configuration contains, but not the links between honeypot configurations and exploits.

If the defender, selects a honeypot configuration that is vulnerable to the selected exploit, the defender detects the exploit and receives a positive payoff proportional to the total value of the affected real systems and the current value of the exploit. If the defenders does not detect the selected exploit, she receives a negative payoff proportional to the total value of the affected real systems and the current value of the exploit.

We also model the severity/effectiveness of different exploits. The National Institute of Standards and Technology uses the Common Vulnerability Scoring System (CVSS) to assess the severity, difficulty of implementation, and impact of exploits [7]. We capture the severity of different exploits, but also model how this severity evolves over the typical lifecycle of an exploit [5]. Initially, an exploit may be known only to the initial discoverer, and there are no known patches or mitigations. These are the most dangerous exploits, and are known as *zero-day attacks*. At some point the exploit becomes broadly known. During this time it may be widely used to conduct attacks by many different groups, while mitigations are still being developed. Eventually, patches and other mitigations are developed to reduce or eliminate the exploit. However, they may take time to be fully adopted and distributed, so the effectiveness of the exploit is gradually reduced over time. To model this, at the end of each round, new exploits may be added to the attacker's actions.

3. EXPERIMENTS

We conducted several experiments to evaluate different online learning methods for the exploit detection problem in comparison with several baseline methods described previously. All defenders interact with our Adversarial Attacker, modeled after fictitious play.

In our initial experiments, we consider three baseline defenders and two Multi-Arm Bandit defenders. Multi-Armed Bandits (MAB) capture a central tradeoff in machine learning: balancing exploration vs exploitation. We observe the Upper Confidence Bound (UCB) stochastic bandit [1] and the Exponentially-weighted algorithm for Exploration and Exploitation (EXP3) adversarial bandit [2], since our model does not map directly to a stochastic problem, nor a classical adversarial problem. We compare these two bandit algorithms to uniform random, fixed random, and a pure strategy (only plays one configuration) defenders. We evaluate each defender by the difference in expected payoff compared to playing the optimal honeypot configuration in each round they get. This performance is known as regret, which we seek to minimize. As seen in Figure 2, EXP3 and UCB outperform the naïve defender strategies and begin learning the attacker's strategy.

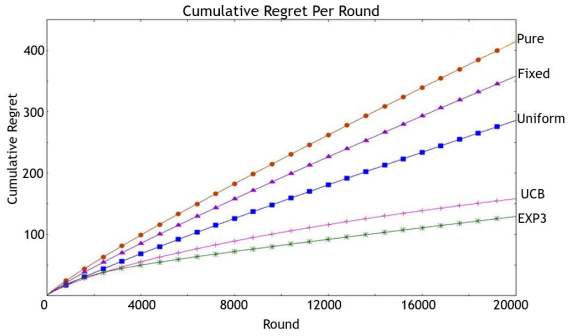

Figure 2: Cumulative Regret Over Time

UCB and EXP do not account for combinatorial actions, so we evaluate naïve combinatorial extensions to these algorithms. When each algorithm plays a honeypot configuration and successfully detects an exploit, we update the beliefs of all configurations that could have detected the deployed exploit. This naïve extension to UCB and EXP3 show improvements to both algorithms, by outperforming their noncombinatorial counterparts with combinatorial UCB performing the best among all evaluated algorithms.

4. CONCLUSION

We have introduced a new model for using dynamic honeypot selection to improve exploit detection. We have empirically demonstrated that applying learning methods dramatically improves performance over non-learning methods. We further demonstrate that we can exploit the combinatorial structure of the problem to further improve exploit detection.

REFERENCES

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[2] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 322–331. IEEE, 1995.

[3] L. Bilge and T. Dumitras. Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 833–844. ACM, 2012.

[4] K. Focus. Kfsensor overview, 2003.

[5] S. Frei, M. May, U. Fiedler, and B. Plattner. Large-scale vulnerability analysis. In *Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense*, pages 131–138. ACM, 2006.

[6] R. Klíma, V. Lisỳ, and C. Kiekintveld. Combining online learning and equilibrium computation in security games. In *International Conference on Decision and Game Theory for Security*, pages 130–149. Springer, 2015.

[7] P. Mell, K. A. Kent, and S. Romanosky. *The common vulnerability scoring system (CVSS) and its applicability to federal agency systems*. Citeseer, 2007.

[8] R. Píbil, V. Lisỳ, C. Kiekintveld, B. Bošanskỳ, and M. Pěchouček. Game theoretic model of strategic honeypot selection in computer networks. In *International Conference on Decision and Game Theory for Security*, pages 201–220. Springer, 2012.

[9] N. Provos. Honeyd-a virtual honeypot daemon. In *10th DFN-CERT Workshop, Hamburg, Germany*, volume 2, page 4, 2003.

[10] L. Spitzner. *Honeypots: tracking hackers*, volume 1. Addison-Wesley Reading, 2003.