**Exploit Decisions Using Partially Observable MDP.**

Zackary Crosley

An Attack and Defense Capture the Flag (CTF) is a security competition designed to simulate real electronic warfare. Teams are given a publicly accessible server with applications that must remain running but have critical vulnerabilities another team can use to attack them. Successfully attacking a service rewards teams with a flag, earning them points towards victory. The game is separated into ‘ticks’, a period after which the flags are reset and another attack and defense round resumes. Like real electronic warfare, at higher levels strategy quickly becomes necessary for team success. Prior research has been done into when a team should decide to disclose, exploit, patch, or store a vulnerability (T. Bao et al.), however much of this research is assuming a single attack is available for each service, which is rarely the case. Most CTF services feature several attack vectors that a team can exploit, and a team with multiple attack vectors available must decide how to deploy them.

I propose using a Partially Observable Markov Decision Process (POMDP) to decide from a set of actions which one to deploy on a given tick. A CTF game is finite in duration, and thus can be easily modeled as a finite MDP with individual ticks to evaluate time since an exploit was used. The project will model a one vs one CTF as a simplification. States will be modeled using the number of ticks that have occurred, the game score, and a belief about whether the opponent has patched this vulnerability. The set of actions will be the set of exploits, which may increase in size as more exploits are discovered. Rewards for actions will be a function of the rewards received in the past for the exploit, with more recent reward sets being more valuable towards estimating its expected reward. This will utilize a reward implementation based on the Functional Reward MDP defined in Markov Decision Processes with Functional Rewards (Spanjaard et al.). The reward cannot be accurately modeled by using the last reward received, as the rewards may vary for any number of reasons that are temporally isolated (such as a server going down, dropping requests). In additional patches may be reversed in the event a team breaks their service and must reset it to the original, vulnerable file. Each action has a probability of yielding the reward and a probability of yielding nothing, as the exploit has been patched. Since using an exploit increases the likelihood of a team being able to identify and mitigate that exploit, the probability of success for an exploit will be monotonically non-increasing over a function that will be chosen during testing (Future additions could have this function be learned). This function will consider the ticks since last use of this exploit, the number of repeated uses of this exploit, and the base chance of a user finding and repairing the vulnerability. For simplicity we will assume each vulnerability is independent of the others.

This project will produce a program that, given a set of exploits to against a vulnerable service, will be able to play a CTF using intelligent decisions about when to deploy and when not to deploy an exploit.

1. Spanjaard O., Weng P. (2013) Markov Decision Processes with Functional Rewards. In: Ramanna S., Lingras P., Sombattheera C., Krishna A. (eds) Multi-disciplinary Trends in Artificial Intelligence. MIWAI 2013. Lecture Notes in Computer Science, vol 8271. Springer, Berlin, Heidelberg
2. T. Bao, Y. Shoshitaishvili, R. Wang, C. Kruegel, G. Vigna and D. Brumley, "How Shall We Play a Game?: A Game-theoretical Model for Cyber-warfare Games," *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, Santa Barbara, California, USA, 2017, pp. 7-21.  
   doi:10.1109/CSF.2017.34