SNARL: Simulating kNown Actors with Reinforcement Learning

Yasyf Mohamedali, Bryan Cai

6.858 Project Proposal

Abstract

SNARL is a set of tools for generating a model which can trick password-based authentication systems that use biometric factors, given a user and her password. SNARL will use a value function-based reinforcement learning algorithm (such as Deep-Q Learning) to learn a per-user model. An additional contribution will be a replica biometric authentication system that exposes confidence levels, for rapid training.

1. Introduction

The past few years have seen several attempts to increase security in password-based authentication systems by adding secondary factors derived from biometric patterns in user behavior. Examples of this include Yang and Haddad's *punchTimeAuth* [1] and the commercial product being developed by UnifyID [2]. However, there has not been much focus on the barrier this presents to attackers who are capable of mimicking their targets, particularly in the age of commodity cloud hardware and effective deep learning methods.

We propose an attempt to build a set of tools to rapidly build effective models that can provide indistinguishable biometric data to a remote server, so as to deceive it into authenticating a user. This model should be able to learn to mimic the user with nothing but the user's password, but should be able to have learning accelerated by bootstrapping with extra behavioral data on the user in question. The desire to learn without any prior data, combined with the ability to simulate password attempts very easily with a

[1] http://css.csail.mit.edu/6.858/2017/projects/yangjy-ddh.pdf
[2] https://unify.id/

replica system, makes this a very natural candidate for (deep) reinforcement learning.

The scope of this project will be limited to learning how a user types their password, though we acknowledge there are many other biometric factors that could be taken into account.

2. Model

SNARL will be attempt to learn the user's biometric patterns by "playing" on a simulator which feeds in each character of the correct password, along with an associated delay. The reward function for this action will be a confidence score that the simulator, which knows the user's true typing behavior, outputs.

Following this basic model, we will build a system for capturing user input (for example, a barebones online document editor), which can be used to generate data to bootstrap the model. We could, for example, use a RNN to learn the estimated delays a user has when typing a given character stream, which can be used as an initial guess for the delays in a password. Part of this project will be to evaluate how effective this is as a technique in the real world.

3. Risks

The most obvious risk of this proposal is that the proposed model will not be effective in replicating user behavior. To mitigate this, there are several simpler, more traditional modeling techniques we can use to get a baseline to compare against.

The other major risk of this proposal is that the data-collection techniques, when used in the real world, will not be effective at providing sufficient data to bootstrap the model so that it trains in a reasonable time.

A final risk for applying this technique in the real world, though not for our project, is that collecting feedback for the RL model through password attempts can be hindered by rate limits, account lockouts, and other security measures which prevent brute-forcing. We will offer some suggestions on how to avoid these, but the risk remains.