Dynamic Factor Blacklisting for Multi-Factor Authentication

Will Gantt

September 9 2019

1 Overview

Multi-factor authentication (MFA)—the use of two or more security credentials to authenticate to a device or a web service—has been widely adopted in recent years as cyberattacks have grown more prevalent and as more people are becoming aware of the weaknesses of traditional, password-only authentication schemes. Requiring a user to enter a one-time token or respond to a text message on her phone provides greater assurance that her personal data and accounts will remain secured even if her password is compromised. But cybersecurity has always been an arms race, and as MFA has spread, so too have efforts to outwit it. Many MFA methods are susceptible even to fairly primitive types of attack, such as phishing, SIM hijacking, and interception of text messages.

Although outright prevention of these attacks is always the ideal, it is also very difficult: Companies that use MFA for their employees and customers can do only so much to reduce users' vulnerability to phishing, for example, or to more direct types of social engineering. There is thus strong motivation to develop ways of mitigating these attacks when they are *in progress*, which is the objective of our research.

We aim to prototype a machine learning system that can decide in the moment (i.e. dynamically) to disable or "blacklist" a security credential from being used based on API- and device-level metadata about the login attempt.

2 Intellectual Merit

The notion of a security measure triggered by suspicious behavior is not new. Credit card companies, for instance, will automatically lock user accounts if unusual purchases are made or if purchasing activity seems to indicate impossibly fast travel. Moreover, machine learning models are famously useful for classification tasks and have already been employed to great effect on a variety of security problems. We do not claim to be doing something new in either respect.

The novelty and the intellectual merit of this work consists, rather, in two things: 1) Applying ML-based decision-making already in use in other domains (like finance) to a new one—namely, individual authentication credentials, and 2) Exploring what types of information can provide the best evidence of attempted fraud in this context.

3 Broader Impacts

The motivation for this project is to further enhance the level of security that MFA provides to users of all types of digital services. If MFA is a response to concerns about the ease of compromising passwords, our project extends that concern to compromise of other security factors.

A specific example may help illustrate the idea. Consider a bank, Acme Corp, that has a web application that allows its clients to do their personal banking online, and that allows those clients to authenticate with MFA. Suppose Alice is a client who has configured two factors, a one-time token sent via email and a six-digit code sent via text message—one of which must be entered, along with her password, in order to access her account. Suppose, moreover, that a bad actor was able not only to obtain Alice's password (through, say, a password-spray attack), but to intercept her text messages (e.g. by cloning her SIM card). If the bad actor then attempted to log in to Alice's account with Acme Corp using her password, our system would use information such as the IP address and porting history of the attacker's device to decide which, if any, of Alice's other configured factors to disable. In this scenario, we would likely want the system to disable the one-time SMS code factor (since there is evidence that SMS is no longer a safe channel) but still allow her to use the email factor. If Alice had had *only* the SMS code factor configured, she (and the attacker) would be unable to access the web app and she would have to go through Acme Corp's account recovery process.

The impact of our project is thus directly proportional to that of MFA itself, which is to say it is considerable. It has the potential to bolster the security systems of thousands of companies and organizations around the world, and hence the security of their clients.

4 Research Plan

4.1 Approach

4.1.1 Data Collection

To our knowledge, there is no open source dataset for classification of legitimate and illegitimate authentication attempts with MFA. Even if some MFA providers do collect a variety of authentication metadata sufficient for our purposes, we do not anticipate being able to persuade them to grant us access

to that data. A substantial portion of this work would consist in compiling a dataset of our own.

Although MFA is used in a variety of settings, we intend to limit our initial research to web apps, as this is perhaps the most common. We propose to set up a mock, publicly available web app. The data collection will then consist of three parts:

1. Configuring the web app, which will include integrating MFA and API- and device-level analytics (metadata) and setting up fake user accounts.
2. Simulating legitimate authentication attempts.
3. Simulating illegitimate authentication attempts.

Each part is discussed in greater detail below.

Configuring the web app. A simple single-page app (SPA) will be adequate for our initial inquiry and is straightforward to set up, as are fake user accounts. As is typical for any app that collects user data, all accounts will have a certain minimum set of required attributes (e.g. name, email, and home address), and certain accounts may contain additional information (e.g. phone number, security question answers, or other information required for enabled MFA factors). Each account will have exactly one factor associated with it, in addition to the password. Equipping our app with MFA would most easily be accomplished by contracting out to a company like Okta or Auth0 that provides this service for a reasonable fee. These companies support a number of different second factors, including email links, SMS codes, push notifications via mobile app, Yubikey, security questions, and one-time passwords. We would take a similar approach for obtaining device and API metadata, using a service like Twilio or TeleSign for the former and Google Analytics or MuleSoft for the latter. Collectively, these services will allow us to collect data on such things as porting status and history, subscriber status, device make and model on the device side, and IP address, location, HTTP error types and rates, number of calls, and time and date on the API side.

Simulating legitimate authentication attempts. Once the app has been configured and the fake user accounts created, it is trivial to log in as any one of those users. However, to obtain an adequately representative data set, we will have to authenticate from a variety of locations for each account and using different types of second factor. For mobile-based second factors, we may associate the same device with multiple accounts for economy's sake. Additionally, even though the companies previously mentioned support many types of second-factor, we will limit ourselves to only a subset that may be easily shared with others online. This would exclude physical keys. (The reason for this decision will be made apparent below.)

Simulating illegitimate authentication attempts. To simulate illegitimate authentications, we propose to recruit "attackers" by offering small monetary rewards ("bounties") to anyone who can successfully gain access to one of the fake accounts. To facilitate the "attacks," each attacker will be provided with the password associated with the target account, and *some* information about the second factor. In some cases, we will give the attackers enough information to compromise the second factor (e.g. by providing the email credentials for an email factor); in others, we will not and ask that they simply attempt brute force.

4.1.2 Model Selection & Evaluation

The choice of classification model for discriminating legitimate and illegitimate authentications will be in part determined by the amount and variety of data we are able to collect. However, we intend to compare the performance of at least several different types of popular model, including decision trees, SVMs, and neural networks. To reiterate, the objective of this work is not to develop cutting-edge machine learning models, but merely to explore what types of features are *relevant*.

4.2 Required Resources

The funding from this grant would be put toward the following:

1. (One-month) subscriptions to an MFA provider (Okta/Auth0) and analytics provider (Google/MuleSoft and Twilio/TeleSign).
2. Some (5 to 10) "victim" phones and tablets to be used for second-factor authentication.
3. Bounties for the "attackers."