**Secure Authorization Web API**

**Tyler Pokos**

**Description:**

Secure Authorization API is an ASP.Net Web API that can be used to inject an extra layer of security into login pages. With this API, organizations can ensure that their login in procedures have an extra layer of security, primarily from brute force login attacks. This API aims to inject latency into login attempts, provide a means of login IP tracking, and provide an effective lockout model for invalid login attempts.

**Developer Notes:**

This Web API is still in the very early stages of development, as I found that I had very little time to work on it between balancing school and lining up a career position after I graduate this semester. This would be a good application to pick up for someone who is experienced with Visual Studio, C#, and ASP.net development, or for someone who has a strong desire to learn these skills. As a result of the little work I was able to invest in this project, many aspects are still incomplete and up for much development.

**Features:**

**Latency Injection for login pages**: his feature will create a delay on each login attempts and aims to increment the delay with each failed login attempt until total account lockout. (Work In Progress)

**Login IP Address tracking:** This feature will fetch the IP address of the user attempting a login, ideally this feature should be capable of fetching the user’s real IP address, even if they are behind a proxy. These IP address will be encrypted and stored within the server the Web API is being hosted on.(Incomplete)

**Account Lockout Model:** This feature will hold encrypted username and password information along with account login attempts, and if the account has been locked out and needs to be reset. (Incomplete)

**Set-Up Instructions:**

To set up this WebAPI, you will need to take the Models, Views, Controllers, and App\_Start data from my Github repository, and place it into a fresh ASP.net WebAPI solution. From there you should be able to perform testing of the WebAPI from the ASP.net built in login page provided within the solution. Much of the WebAPI functionality can be modified from the Values Controller, and testing should be done using the Account Controller.