Joseph Haaga
Senior Design
Design Document

Overall Objective

This product aims to enhance information security without sacrificing convenience for the user. Whereas traditional multi-factor authentication schemes are inflexible, time-consuming and demand mobile connectivity to SMS and email, this product provides system administrators with the ability to decide which authentication factors the users must pass. Furthermore, additional authentication factors are introduced as needed via administrator-designated confidence intervals, providing information assurance without hassling the user or wasting their time.

User Description

The most common user will be a member of an enterprise network, and will ideally interact with the system as little as possible. As they enter their user credentials, keystroke dynamics data is collected and analyzed to predict whether or not the user is actually who they claim to be, and the confidence of these predictions is used to dynamically introduce additional authentication factors as needed.

The user interacting more deeply with the system will be the system or network administrator, who is tasked with choosing the specific authentication factors the users must successfully complete, and confidence intervals that define successful completion. The sysadmin also has the opportunity to implement their own authentication factors as they see fit.

Use Case Description

As previously mentioned, the primary user is [ideally] unaware that the system is running. Entering their user credentials to login triggers the keystroke detection classifier, and typing characteristics sufficiently similar to that user's profile will provide them with system access without any intervention. If, however, their typing characteristics are not sufficiently similar to the profile constructed by the algorithm for that user, the user will be prompted to complete an additional authentication factor. This additional factor will be chosen by the system administrator, but for our purposes will default to virtual signature evaluation. The user will sign their signature in front of the webcam, and another algorithm will determine the similarity between that signature and the known-correct signatures recorded when the user account was established.

The system administrator will first install and deploy the software to their entire network of machines. From there, the admin will be responsible for designating the specific authentication factors that the user faces, and the requisite confidence intervals (0-100%) for each factor that determine whether a user has successfully proven their identity.

Joseph Haaga
Senior Design
Design Document

Major Components & Requirements

User login & keystroke dynamics classifier

- Functional requirements
    - Collect timestamped keystroke events, including key press and key release
    - Parse, analyze and feed keystroke events into binary classification model
    - Output a user/not-user prediction, and a percentage describing confidence in that prediction
- Nonfunctional requirements
    - Determine whether or not a user is who they claim to be in 1 second or less

Virtual signature analysis

- Functional requirements
    - Capture signature via webcam (or Microsoft Kinect), and display on-screen in realtime
    - Convert signature to list of coordinate pairs
    - Pass list of coordinate pairs to distance-approximation algorithm
    - Approximate similarity between current input and known-correct signatures
- Nonfunctional requirements
    - Display webcam feed with digital signature overlay, following finger with sub-50ms delay
    - Pass coordinates to algorithm, evaluate and output similarity score within 4 seconds

Administrator Control Panel

- Functional requirements
    - Provide sysadmin with controls that alter behavior/order-of-operations of other system components
        - Alter which authentication factors appear in what order
        - Change confidence intervals that denote success in a given authentication factor
    - Allow sysadmin to enter address at which to receive emails in the event that a user fails all authentication factors
    - Allow sysadmin to introduce additional authentication factors that they've implemented
    - Provide list of machines that have the software deployed, and indicate whether changes to the authentication scheme have been installed successfully on each machine.
- Nonfunctional requirements
    - Alert admin of failed logins in real-time

- Deploy updated administrator-defined authentication schemes to all machines in a network, and have each machine report back within 5 seconds of successful update installation.

Implementation Details

Preliminary implementation of the machine learning algorithms for keystroke/virtual signature detection and classification, as described above, will be built in Python using Scikit Learn, Pandas & NumPy, all provided via Continuum Analytics' Anaconda distribution. Following an exploratory tuning process, production-level implementation will be done in Java using the Weka library, providing us with the portability to operate on any system with the JVM installed.