Leslie Ogu
Design Document 1 - Covert Ops

Purpose

Covert Ops is a product aimed at improving the security hygiene of the system in which it is implemented. Its design of utilizing multiple biometrics to revolutionize the security of a company makes it a complex and intricate system that has a much higher rate of success of defending against, and preventing, cyber attacks. Many products in the market advertise themselves as being the best in the business and offering the best protection for customers. However, many of them only incorporate basic levels of security such as required logins to access certain information in the company, and password-protected technology such as the Wi-Fi, company network, and more. Covert Ops is a product that will bring greater assurance of protection to customers, a more effective and efficient security system, and an improved infrastructure that is less likely to be breached.

Objective

The objective of Covert Ops is to increase and improve the security measures currently used within a company, and overall, create a more secure system that can prevent malicious attacks. Individuals with malicious intent are getting more creative with each passing day, and the fear of the unforeseen future of possible intrusion attempts continues to hang over the heads of all entities who hold personal or sensitive information belonging to their customers. In order to combat and prepare for these potential attacks that a company may undergo, they must be innovative and proactive in the security measures they take, and the plans they have in place to address the problem when it arises. This design document will help give a general idea of how this system will be implemented and the overall goals it will achieve for the company.

Users

The users of this product will be companies with the intention of keeping the sensitive and personal information of its customers safe and secure - which should include all of them. There is no specific field or business type that this product is catered to as it can be used by any entity. Cyber security should be a major component of any company because it affects everyone to some degree. Given the fact that technology has become some an intricate part of our lives, it it is extremely difficult to avoid one's life being impacted by cyber security in some way.

Besides the companies that would want to use this product to improve their security, smaller entities such as schools, volunteer centers, hospitals, and pharmacies would benefit from adapting this product as well, as each of them has information they want to keep secure. Even one's personal property or assets, such as a home or small business, may need additional security measures to ensure safety and protection from outside threats. This shows how dynamic this product can be, and how versatile it is in use. Security affects everyone and should not be taken lightly, no matter the situation or context.

Use Cases

There are a significant number of uses for Covert Ops:

- Logging into a system
- Gaining access to certain information
- Entering a secured, physical location
- Regulating actions users can make in a system
- Mitigating the number of intrusions and successful attacks
- Increasing protection and assurance for customers
- Maintaining a hierarchy and chain of command within a company to ensure that employees only see what they need ("Need to Know" Basis)

The aforementioned list is not all that Covert Ops is limited to, but gives a good perspective of its capabilities and the services it provides.

If Covert Ops were used in more unconventional methods, it could also be utilized for uses such as teaching a course on cyber security, increasing awareness of cyber risk, shaping the future of how large organizations like government agencies handle their security, and coaching people on how to be mindful of how they handle technology that holds their personal information. All of these uses serve to provide a great benefit to individual people, and the world.

Major Components of Your Software

The major components of the product are the Kinect 2 sensor, its libraries, and the keystroke dynamics classifier.

Functional Requirements

Digital Signature:

- ❑ Capture digital signatures via Kinect on the screen within a time window
- ❑ Store signature image or coordinates where the signature is drawn on the screen by the user
- ❑ Compare the signature with previously saved images or coordinates and, using a coordinate comparison algorithm, provide percentage of how confident the system is that the user is authorized

Typing Rhythm:

- ❑ 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

Non-Functional Requirements

Digital Signature:

- ❑ Response to the user whether or not they are authorized within 10 seconds of analysis

Typing Rhythm:

- ❏ Determine whether or not a user is who they claim to be in 1 second or less

Classes

- Kinect - creates the object of the Kinect to begin implementing what you want it to look for
- PCDesign - used as a base class that will be extended in order to edit how the Kinect will operate the painting done by the user
- KinectDesign - extension of PCDesign used to carry out all commands of the Kinect such as initializing the Kinect, setting up the video feed, and starting to watch for Skeleton events
- CustomColorChooser - included to understand how to give the user a color to draw their signature with on the screen
- JPanel - extended by Kinect to create a unique video feed panel
- J4KSDK - library that contains all necessary methods and variables to create program based on Kinect
- Skeleton - library that contains all necessary methods to manipulate the Skeleton
- DWApp - utilized as a reference to understand how the Kinect feed is started and detects certain things on the screen
- OpenGLPanel - sets up the graphics used in the video feed to supplement in areas Java's GUI class lacks

Interfaces:

There will be three interfaces displayed by the product. The first will be an interface that prompts the user to draw their signature on the screen within a time window. From that screen, after executing a database check or coordinate matching method to confirm if the user is authorized, it can either take the user to an "Access Denied" screen or to the next security check - the typing rhythm. This interface will have the user complete a writing prompt - this will be after the authorized user's typing rhythm has already been learned by the system through training - and if the user passes this step, they will be taken to a screen displayed "Access Granted" to let them know they are authorized. Otherwise, they will be taken to the previously mentioned "Access Denied" screen.

A flowchart of how the security check will work is pictured below in **Figure 1**.

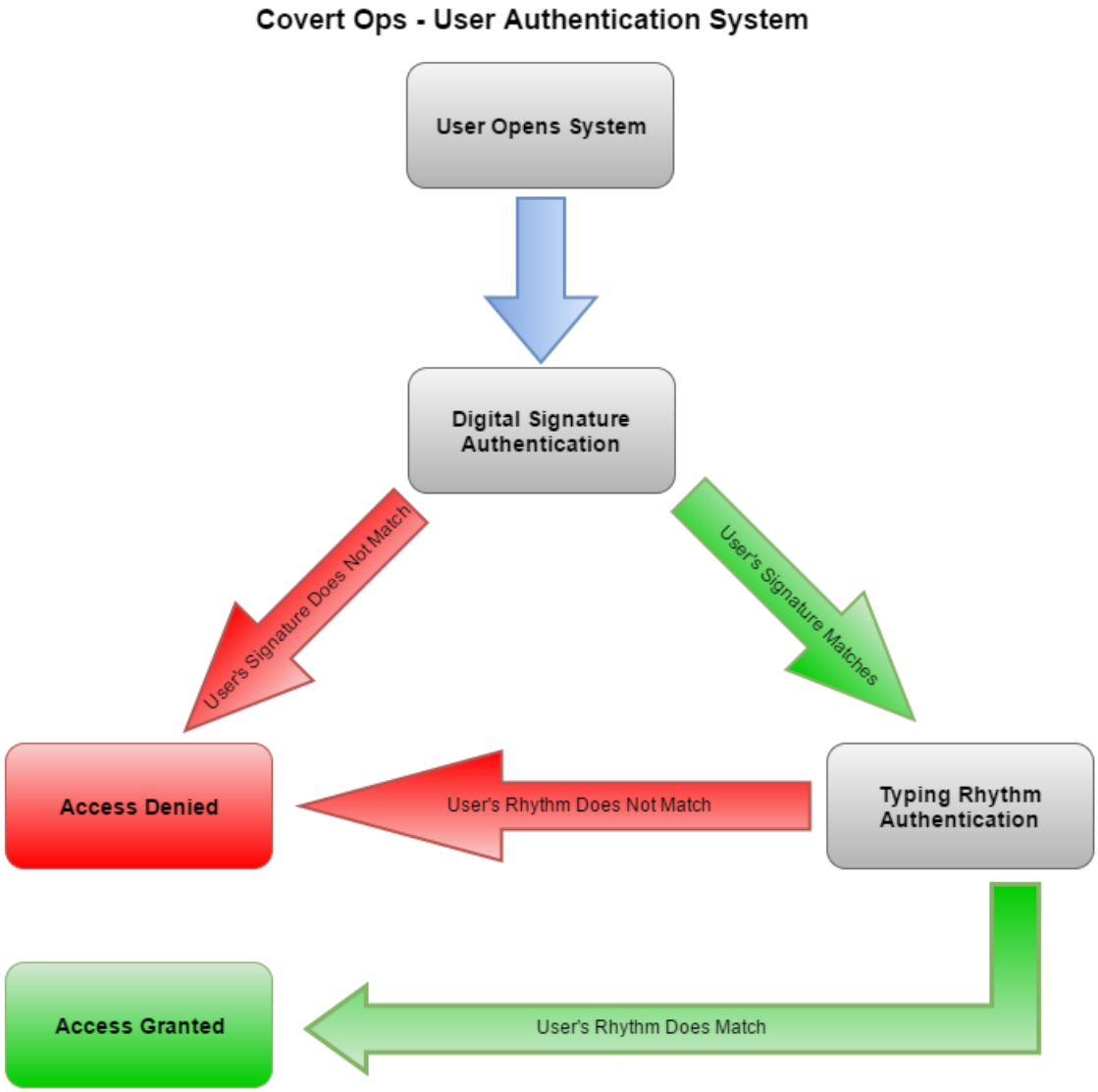

Figure 1: Displays a flow of how the Covert Ops product authenticates users.

Logical Software Module

Timeline - Creation of Covert Ops

- ❏ Month One: Research the biometrics and current implementations available
- ❏ Month Two: Attain necessary libraries, software, and components to begin coding the digital signature and typing rhythm authentication
- ❏ Month Three: Begin coding
  - ❏ Digital Signature: Create a boundary for the user to draw their signature that is based on the skeleton coordinates of the hand
  - ❏ Typing Rhythm: Write out a prompt and save time stamps of each user's typing speed.
- ❏ Month Four: Continue coding

        - ❏ Digital Signature: Code a Kinect library to allow the user to draw on the screen with their hand
        - ❏ Typing Rhythm: Get additional users to write prompts and begin training system to accurately differentiate between them.
    - ❏ Month Five: Continue coding
        - ❏ Digital Signature: Develop a method of comparing digital signatures of users
        - ❏ Typing Rhythm: Have a system created to have users complete multiple prompts to refine how accurate it is in determining the user
    - ❏ Month Six: Testing
        - ❏ Digital Signature & Typing Rhythm: Have multiple users draw their signatures and type prompts to increase accuracy of system in determining which users are authorized
    - ❏ Month Seven: Continue testing and reach 85-90% accuracy

Timeline - After Covert Ops has completed

- ❏ Month One: Covert Ops is installed onto the system.
- ❏ Month Two: Covert Ops is run on the system with diagnostic reports created on a daily basis. The information will be compiled at the end of the month.
- ❏ Month Three
    - ❏ First Two Weeks: The reports are analyzed.
    - ❏ Second Two Weeks: Necessary changes are made for improved performance.
- ❏ Month Four and Five: Covert Ops is run on the system again with the same diagnostic reports being generated each day and compiled at the end of two months. In the second month, security professionals and programmers will attempt to gain unauthorized access to the system.
- ❏ Month Six
    - ❏ First Two Weeks: Analysis of diagnostic reports and creating an understanding of the impact Covert Ops has on the system. Also, check how successful the product is in preventing attacks.
    - ❏ Second Two Weeks: Make additional changes to the system.
- ❏ Month Seven, Eight, and Nine: The final test run of Covert Ops will be carried out with the same methodologies as the second test, except the attempted intrusions will occur throughout all three months.
- ❏ Month Ten: Final analysis of Covert Ops progress and creation of next steps to be taken.
- ❏ Month 11: After final revisions, releasing Covert Ops to the wider market.