Proceedings of Student-Faculty Research Day, CSIS, Pace University, May 6th, 2011

# Investigation of Freeware Biometric Products

Michael Isola, John Granger, Arthur Gadayev, and Wojciech Hojdysz
*Seidenberg School of CSIS, Pace University, White Plains, NY 10606 USA*
{mi04392w, jg06242n, ag75101w, wh07613p}@pace.edu

## Abstract

*This study investigates and tests biometric products focusing on Face, Voice and Keylogger biometrics. Biometrics depends on the uniqueness of human characteristics and is used as a form of identity management and access control. There are two main classes of biometric characteristics – physiological characteristics related to the shape of body parts and behavioral characteristics related to learned behavior. Face recognition software uses key features such as algorithmic measurements of nose, eye and mouth of an individuals face to compare against a database of photos or prerecorded face images and their respective algorithmic measurements looking for similarities to gain access or as a way for identification. Voice biometrics uses the pitch, tone and rhythm of an individual's speech to create a numerical model and then compare it to a database of prerecorded voice biometrics. Keystroke biometric use the time for which keystrokes are made and the length of time keys are held down so as to take advantage of the fact that most computer users type in a consistent manner.*

## 1. Introduction

Biometrics provides a unique capability to confirm personal identity without reliance on less accurate methods such as names, numbers or demographic information identity [3]. However, it is no cure-all and biometrics can be fooled by simple methods such as using a high quality color laser printer of the iris of an eye and punching a hole in the center of the image and the iris security system was tricked into false positive recognition [1]. All biometrics have authentication and identification applications. In authentication (verification) applications a user is either accepted or rejected (binary response, *yes* you are the person you claim to be or *no* you are not). A pattern recognition system must be trained to become useable, so the data are usually separated into two parts, one for training the system to create decision boundaries and one for testing the system to determine its performance [8].

## 2. Research Methodology - Voice

Initial testing was performed on trial versions of several biometric products. Various technologies are used to process and store voice prints. The Programs perform a speaker identification process and create a list of best matches and executes a verification process to confirm the match. A procedure was developed for a security demonstration that compares voices of several test subjects and then tests the procedure on the voice biometric of the test subjects to obtain a similarity matrix, so it is easy to determine which test subjects sound most similar. The test subjects needed to record just the phrase 'My name is' thus allowing for accurate comparisons to take place for similarity. Results were obtained using 'My name is' and further testing was performed with a longer phrase 'I am a Pace University student'. The two types of results that we are attempting to measure and are relevant in this study are False Acceptance Rate (FAR) also known as false positive and False Rejection Rate (FRR) also known as false negative. FAR occurs when a person who is not authorized is actually approved for access by the biometric product. The FAR is the frequency that a non authorized person is accepted as authorized. Because a false acceptance can often lead to damages, FAR is generally a security relevant measure [2]. FRR occurs when an entity is not approved for access or is not positively identified and is rejected even though the entity is the same entity that is in the system and scanned. The FRR is the frequency that an authorized person is rejected access.

## 2.1 Voicecipher

Voicecipher voice biometric allows the recording of many voices, however, after verifying with the manufacturer they are not kept as wav files so manipulation is not possible, only a voice numeric print is stored which is not a wav file nor does it contain any security data. An 80 character key is created from the voiceprint that is spoken. Each voice can then be used to secure a file(s) by simply the name of the voice biometric that has been algorithmically recorded.

### 2.1.1 System Environments & Install

VoiceCipher system requirements are very simple. The requirements are a computer running Windows XP/2000/Vista/7 and a microphone attachment to that computer. A good quality microphone is also suggested for use. Voicecipher is simple to install. The software can be downloaded from Voicelatch [9]. On the Voicelatch website click download trial and save the executable file, then double click on this exe file to install VoiceCipher on your machine and you will see VoiceCipher under your Program Files directory.

### 2.1.2 Training & Recognition

Ten test subjects were initially chosen – five female and five male. All individuals used the same phrase ‘My name is’ and I made sure they pronounced the phrase clearly and in a quiet environment. Each individual had their voice recorded 3 times with each time separated by a 10-15 minute break as suggested by the product for what is known as training the software for the individual voice. During these attempted training recordings the product was unable to compile 2 of the 5 female voices. I am not sure why but it may be because the voices are not as deep sounding as a male voice. Obviously this was not scientific, but just my observance of the individuals who were rejected as compared to the female voices that was approved. I then decided to increase the number of male voice test subjects to 6 and I was able to use the 3 female test subjects for a total of 9 test subjects. Directly after each individual has recorded their voice 3 times for training, then the individual needs to speak into the microphone one more time and then compare it to the 3 recorded previously and if successful the biometric is complete for the individual and that person can start securing data files, etc. using their voice biometric algorithm that was saved to encode the file. This same procedure was used to record the longer phrase ‘I am a Pace University Student’ for all 9 test subjects.

### 2.1.3 Testing results

The reliability of the voice biometric product was tested by using multiple known voices and attempt to access secured files using other known voices. Table 1 shows the success / failure rate of the 9 test subject voices using the phrase ‘My name is’. A ‘1’ indicates accepted access. The top row of the matrix shows what test subject encoded/secured the file and the left most column shows what test subject attempted to decode/access the file. Where they intersect is the results for that particular pair on a particular file.

Table 1 Voicecipher Confusion matrix ‘My name is’

| secured by--><br>Attempt by \/ | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| A = Mike | 1 | | 1 | | | | | | 1 |
| B = Andy | 1 | 1 | | | | 1 | | | |
| C = Anthony | 1 | | | 1 | | | | | |
| D = Sam | | | | 1 | 1 | | | | |
| E =Joe | | | 1 | 1 | | | | 1 | |
| F = Len | | | | | | 1 | | | |
| G = Christina | | | | 1 | | 1 | | 1 | |
| H = Louise | | | 1 | | | 1 | | | 1 |
| I = Theresa | | | | 1 | | | 1 | | 1 |

High failure rates of test subjects trying to decode their own encoded file(s) may be due to length of time between encoding a file and attempted access of the encoded file because of change of voice pitch, inflection or other similar change in voice. All testing was done with a low security setting therefore the voices would be more likely to give a false positive. Also, I had the test subjects attempt to access their own secured file using their own voice and as you can see their was still a high failure rate (False Rejection Rate). Each person attempted access and some attempts were accepted accurately as the correct person (diagonal entries) while others were accepted as other people or not at all. The overall correct acceptance rate was 5/23 or 22%. The False Acceptance Rate is 16/72 (16 times access granted to incorrect person / 72 attempts) or 22%. The False Rejection Rate is 4/9 (4 times access falsely denied / 9 attempts) or 44%. This result may be misleading since there was a high False Rejection Rate thus either a higher quality microphone needs to be used, a longer phrase needs to be used or the voice biometric product is of poor quality. I would tend to think it would be less likely the microphone was the problem since it was the same microphone used at all times so there was no inconsistency. After these results were obtained I performed further analysis using the longer phrase ‘I am a Pace University student’ with much better results. Table 2 shows the success / failure rate of the 9 test subject voices using this phrase. A ‘1’ indicates accepted access. The overall correct acceptance rate was 7/23 or 30%. The False Acceptance Rate is 10/72 or 14%. The False Rejection Rate is 2/9 or 22%. As you can see from the results, the longer phrase has substantially improved the performance; however, the percentages are still too high.

Table 2 Voicecipher 'I am a Pace University student'

| secured by--><br>Attempt by \/ | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| A = Mike | 1 | | 1 | | | | | | |
| B = Andy | | 1 | | | | | | 1 | |
| C = Anthony | 1 | | | 1 | | | | | |
| D = Sam | | | | 1 | 1 | | | | |
| E =Joe | | | 1 | | | | | | |
| F = Len | | | | | | 1 | | | |
| G = Christina | | | | | | | 1 | | |
| H = Louise | | | 1 | | | | | 1 | 1 |
| I = Theresa | | | | 1 | | | 1 | | 1 |

### 2.1.4 Conclusion & Follow-up

As was demonstrated, a longer phrase helps to create additional voice vectors and a more accurate voice biometric security system. However, care must be taken not to create too long of a phrase as users will not be able to remember the phrase. Further testing on VoiceCipher should be performed using telephone or tape recorder instead of live voices to attempt to gain access to encoded files and record the results. Also, further testing should be performed using a higher quality microphone.

## 2.2 Bio-ivault/My-iwallet

The *bio-iVault* and *my-iWallet* voice biometric application products are made by the same company (*myBiodentity Corporation* and *Ardeun*). The voice verification component enables users to create virtual partitions; which are encrypted and accessed only by an authorized user via successful biometric scan. However, this program is designed for online biometric authentication (e.g. web-site account access) and files are a propriety format. A speech evaluation/audio extractor program would be required to compare voices.

With a voice scan, my-iWallet will log you onto your authorized online web site account. These programs include noise-cancellation technology for better voice recognition and superior speech recording.

### 2.2.1 System Environments & Install

Both *bio-iVault* and *my-iWallet* voice biometric application products do not require any drivers to download or install. These products support the following: OS Windows XP, Vista, 2000, 2003, Minimum screen resolution 640x480, P4 processor and above, 512 Mb RAM (minimum total memory). An optional advanced digital USB noise-canceling microphone provides superior sound clarity with a plug-and-play connection. Information is encrypted via AES-256 such as Site URL and Passwords.

For the trial versions, an application activation license is required [6]. You will be provided with a trail product activation key and a Download ID. Trail applications will expire in 7 days or with 5 runs (whichever is first). Requests can be made for a new Download ID.

### 2.2.2 Training & Recognition

Once a successful download of the trial version is completed; the program offers a quick tour, video clips, demos and tutorials. A setup process is required to add a profile with an emergency (override) password. In addition, the program has a geometric plot, which requires to select four random squares as part of the initial setup (as an added authentication layer). You can select the default scan type (e.g. voice).

### 2.2.3 Testing Results

The program does not allow any editing of files and the audio files are not in a standard .WAV format. Another program is offered by Ardeun. This is a server-based, integrated biometric authentication (that supports voice) to log-on to a secure web page. However, Ardeun’s program is proprietary and is limited with customization and access to voice print files/directories.

Six voice samples were recorded for “My name is” + name from three individuals. An audio extractor program was used to edit these recordings and delete the beginning of the recorded sample (“My name”).

### 2.2.4 Conclusion & Followup

Biometrics (e.g. voice verification) is used to provide security for online identity and prevent fraudulent access. Voice biometric applications can be used for several real-world functions. They are used for identity confirmations for law enforcement (e.g. policing a parole or sex offender), for border control and to remote monitor alcohol testing for DUI felons.

A challenge for voice biometrics is accuracy. Voice recognition for access to systems is not very accurate and has high error rates. Poor quality voice samples due to changes in the speaker’s voice (e.g. due to illness and mood) can impact access via the voice biometric system. Bio-ivault/My-iwallet while being a voice biometric it is not believed to be the kind of product needed for the purpose of our effort since it is for securing websites.

## 3. Research Methodology-Face

Face detection and recognition is a challenging problem because faces vary significantly in size, shape, color, texture and location. Their overall appearance can also be influenced by lighting conditions, facial expression, occlusion or facial features, such as beards, moustaches and glasses. Another challenging problem comes from the orientation (upright, rotated) and the pose (frontal to profile) of the face. The crucial first step of face detection is to determine whether or not there are any faces in the image and, if present, their location. Thus, accurate and fast human face detection is the key to a successful operation. Face recognition has been an active research area for more than 30 years and different systems are now capable of correctly recognizing people's faces under specific environments (near frontal faces and controlled imaging conditions).

### 3.1 KeyLemon

KeyLemon is a simple solution to log on to your personal Windows account by using your face. If your computer has multiple users the software automatically logs you into the right Windows account. When you leave the computer, it will automatically lock it and then unlock it when you are back. KeyLemon works as a password manager for popular internet sites. When you connect to a website (Facebook, Twitter and / or LinkedIn), KeyLemon automatically logs you into your account by using your face.

### 3.1.1 System Environments & Install

KeyLemon supports the following operating systems:
Windows XP Sp2/Sp3, Vista, 7
Minimum hardware requirements:
Pentium 500 MHz (Recommended: Pentium 1GHz or greater), 100 MB RAM (Recommended: 128 MB RAM or greater), 25 MB hard drive space, USB webcam or integrated webcam (laptop). To install KeyLemon choose and download the right installer, for your system (Windows XP 32/64bits or Windows Vista 32/64bits), from the KeyLemon downloads page [4]. Run installer and follow instructions. You need administrator privileges to run installation. After reboot, the KeyLemon wizard will launch automatically.

### 3.1.2 Training & Recognition

After launching the KeyLemon wizard and Control Center you must choose your webcam. For optimal performance be sure your webcam is as in front of you as possible, you have good luminosity, take a natural position and smile. While you are in front of the webcam you must stay in that position and click on the 'Create my model' button and wait until the status bar is 100%. Now you can verify the quality of the picture and move on to the next step. To perform login on Windows, KeyLemon need your Windows account password. This information is confidential and accessible only by KeyLemon during the login process. If your password has changed, don't worry Keylemon will ask you automatically to enter the new one. Now you are ready to use your Face recognition for access.

### 3.1.3 Testing results

KeyLemon testing results on a scale of 1 thru 10 is 5. It was not able to recognize my face and log me in when I woke up in the morning and went straight to test the software from my bed. Also when trying to log in I have to move my body to and from the camera until it recognizes who it is that is trying to login. There is no way to compare faces on this software.

## 3.2 FaceSDK

Luxand FaceSDK is a cross-platform face detection and recognition library that can be easily integrated into the customer's application. FaceSDK offers the API to detect a face and facial features and to match faces. Following face detection, the SDK provides the coordinates of 40 facial feature points for further processing such as eyes, eye corners, eyebrows, mouth corners and nose tip. The library is webcam-capable and able to retrieve frames from DirectShow compatible cameras.

### 3.2.1 System Environments & Install

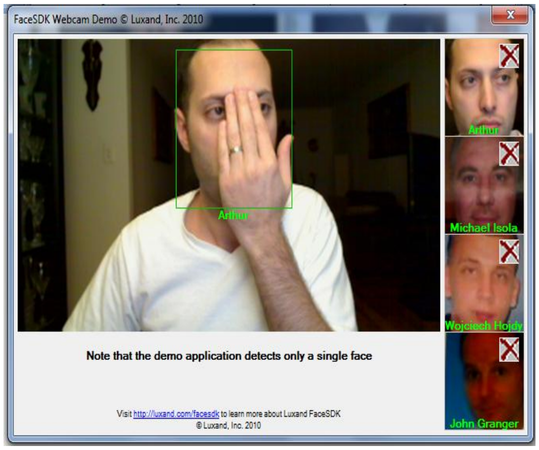
FaceSDK supports Windows 2000, XP, 2003, Vista, 7 on Linux (RHEL 5+,CentOS 5+ and others). Mac OS X 10.4+ x86_64. Minimum requirements are 1.6GHZ processor, 256 MB RAM and 150MB free disk space. To Install FaceSDK on a Windows environment Run the installation file Luxand_FaceSDK_Setup.exe from the site [5] and follow the instructions. FaceSDK is installed to the C:\Program Files\Luxand\FaceSDK director.

### 3.2.2 Training & Recognition

After launching the FaceSDK application, choose 'Camera Demo' and select the camera you are using. The demo version allows the creation of four profiles, such as one for each team member. The application takes several samples of the individual's face for training, taking about 30 seconds. As soon as the software recognizes a face that is in the application profile it displays name. Because we are using the demo version only one face is detected at a time.

### 3.2.3 Testing Results

FaceSDK testing results on a scale 1 thru 10 is a 9. I was able to create team 3 profiles by pointing the camera on to my monitor, at the pictures on our students' website. Then I printed those pictures and the software recognized who it is on a picture at any time. I have created my profile live, then showed a headshot of my picture blocked by my hand and was not recognized, see Figure 1. It recognized there is a face in front of the camera but did not give the profile name, meaning that the face is not in the library.



FaceSDK Webcam Demo © Luxand, Inc. 2010

Note that the demo application detects only a single face

Visit http://luxand.com/facesdk to learn more about Luxand FaceSDK
© Luxand, Inc. 2010

Figure 1.

## 3.3 Verilook 4.0

VeriLook facial identification technology is intended for biometric systems developers and integrators. The technology assures system performance and reliability with live face detection, simultaneous multiple face recognition and fast face matching in 1-to-1 and 1-to-many modes.

### 3.3.1 System Environments & Install

A PC with x86 (32bit) or x86-64 (64bit) compatible processors or Mac with x86 or PowerPC compatible processors and 2GHz or better processor is recommended. At least 128 MB of free RAM should be available for the application. Additional RAM is required for applications that perform 1-to-many identification, as all biometric templates need to be stored in RAM for matching. For example, 10,000 templates (each containing 1 face record) require about 24 MB of additional RAM. Free space on hard disk drive (HDD): at least 1 GB required for the development. 10,000 face records require approximately 30MB of free HDD space. T he database engine itself requires HDD space for running. Please refer to HDD space requirements from the database engine providers. Any camera that is accessible by the following methods can be used with Verilook:DirectShow interface for Microsoft Windows platform, Video4Linux interface for Linux platform, QuickTime interface for Mac platform. Verilook can be used with various database engines and with Windows 2000/XP/2003/2008/Vista/7 and various other environments. To install you must download VerilookStandardSDKSetup.exe.

### 3.3.2 Training & Recognition

VeriLook 4.0 performs fast and accurate detection of multiple faces in live video streams and still images. All faces on the current frame are detected in 0.01 - 0.14 seconds and then each face is processed in 0.03 - 0.11 seconds depending on defined template size. A conventional face identification system can be easily cheated by placing a photo of another person in front of a camera. VeriLook is able to prevent this kind of security breach by determining whether a face in a video stream belongs to a real human or is a photo. A quality threshold can be used during face enrollment to ensure that only the best quality face template will be stored into database. VeriLook allows 360 degrees head roll. Head pitch and yaw can be up to 15 degrees in each direction. Biometric template record can contain multiple face samples belonging to the same person. These samples can be enrolled with different face postures and expressions, from different sources and in different time thus allowing to improve matching quality. For example a person could be enrolled with and without eyeglasses or with different eyeglasses, with and without beard or moustache, with different face expressions like smiling and non-smiling etc. VeriLook functions can be used in 1-to-1 matching (verification), as well as 1-tomany mode (identification). The VeriLook 4.0 face template matching algorithm can compare up to 800,000 faces per second.. A face features template can be only 2.3 Kilobytes, thus VeriLook-based applications can handle large face databases. Larger templates can be used to increase matching reliability. This mode generates the collection of the generalized face features from several images of the same subject. Then, each face image is processed, features are extracted, and the collections of features are analyzed and combined into a single generalized features collection, which is written to the database. This way, the enrolled feature template is more reliable and the face recognition quality increases considerably.

## 3.4 Conclusion & Follow-up

Humans often use faces to recognize individuals and advancements in computing capability over the past few decades now enable similar recognitions automatically. Early face recognition algorithms used simple geometric models, but the recognition process has now matured into a science of sophisticated mathematical representations and matching processes. Major advancements and initiatives in the past ten to fifteen years have propelled face recognition technology into the spotlight. Face recognition can be used for both verification and identification (open-set and closed-set). The computer-based face recognition industry has made much useful advancement in the past decade; however, the need for higher accuracy systems remains. Through the determination and commitment of industry, government evaluations, and organized standards bodies, growth and progress will continue, raising the bar for face recognition technology [7].

## 4. Basic Key logger

The evaluation of a keystroke dynamic key logger application needs to occur in an environment clean from other conflicting applications. Basic Key Logger is a standalone key logger which has the ability to capture keyboard and mouse inputs from any application running in parallel. Basic Key Logger also has the ability to record the detailed timing information that describes exactly when each key was pressed and when it was released as a person is typing at a computer keyboard. This aspect of timing will be the main focus of Basic Key Logger.

### 4.1 System Environment & Install

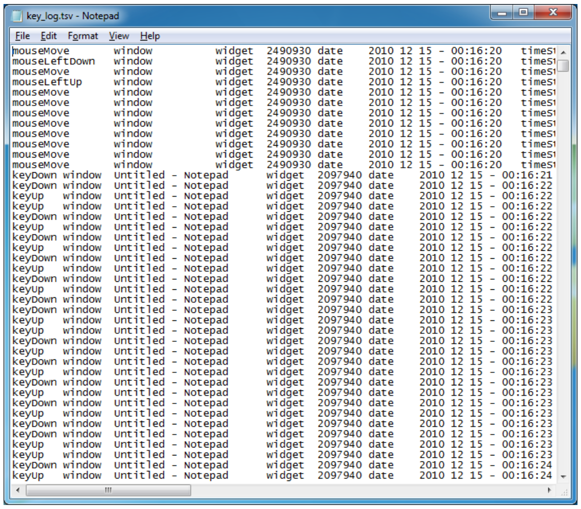
The only installation requirement for Basic Key Logger is that the Operating System be Windows XP or later. Windows XP, Windows Vista, Windows 7 are all acceptable platforms including their respective 64-bit variations. No additional requirements are specified. Basic Key Logger is written in Python which the installation includes a standalone version of Python.

### 4.2 Training & Recognition

No training is needed for the use of Basic Key Logger since it will be acting as a data driver for the Pace University Keystroke System (PKS). Basic Key Logger has the ability to monitor all keystrokes as soon as it is told to start monitoring. Keys and timings are logged no matter what application is used. Basic Key Logger is not application bound.

### 4.3 Test Results

Basic Key Logger generates two types of logs once monitoring is stopped. Key_log.tsv and KPC_log.tsv are both generated conjointly. Both of these logs contain event which correspond to one line of the log file. The first word on each line describes the type of event. Date and timestamps are both generated at the point each event has occurred. Key_log.tsv log files contain key press/release timings, mouse movements and mouse button press/release logs as can be seen in Figure 2. Since only keystroke data is relevant to this study, this log is not as important as the other KPC_log.tsv which only logs keystroke data.

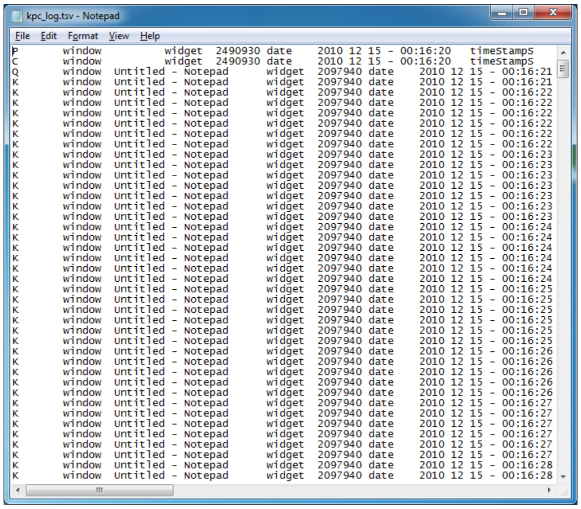


```
mouseMove       window          widget  2490930 date    2010 12 15 - 00:16:20   timeS
mouseLeftDown   window          widget  2490930 date    2010 12 15 - 00:16:20   timeS
mouseMove       window          widget  2490930 date    2010 12 15 - 00:16:20   timeS
mouseLeftUp     window          widget  2490930 date    2010 12 15 - 00:16:20   timeS
mouseMove       window          widget  2490930 date    2010 12 15 - 00:16:20   timeS
mouseMove       window          widget  2490930 date    2010 12 15 - 00:16:20   timeS
mouseMove       window          widget  2490930 date    2010 12 15 - 00:16:20   timeS
mouseMove       window          widget  2490930 date    2010 12 15 - 00:16:20   timeS
mouseMove       window          widget  2490930 date    2010 12 15 - 00:16:20   timeS
mouseMove       window          widget  2490930 date    2010 12 15 - 00:16:20   timeS
mouseMove       window          widget  2490930 date    2010 12 15 - 00:16:20   timeS
mouseMove       window          widget  2490930 date    2010 12 15 - 00:16:20   timeS
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:21
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
keyDown window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:24
keyUp   window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:24
```

Figure 2.

KPC_log.tsv log files contain user operations which mostly correspond to keystroke data such as key type (ASCII and non-ASCII), duration of key press, duration of overlapping keystroke events, key release timings and the ability to log keys which are automatically generated (auto-repeat). All of the keystroke events recorded include the name of the key (uppercase letter or identifier), ASCII code, scan code of the physical keys that have been touched, numerical identifier of the key, and the extended characters (0 if the key pressed was on the main keyboard, positive integer otherwise) as can be seen in Figure 3. The events logged in KPC_log.tsv correspond mostly to key functions. KPC_log.tsv uniquely stores entries when a key is held down, instead of putting an entry for each auto-repeating key and also records quiet periods where the keyboard is idle for more than 100ms.

.



```
P       window          widget  2490930 date    2010 12 15 - 00:16:20   timeStamps
C       window          widget  2490930 date    2010 12 15 - 00:16:20   timeStamps
Q       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:21
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:21
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:22
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:23
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:24
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:24
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:24
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:24
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:24
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:24
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:25
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:25
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:25
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:25
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:25
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:25
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:26
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:26
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:26
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:26
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:26
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:27
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:27
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:27
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:27
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:27
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:27
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:28
K       window  Untitled - Notepad      widget  2097940 date    2010 12 15 - 00:16:28
```

Figure 3.

### 4.4 Conclusion & Follow-up

Log samples have been sent to Ned Bakelman for analysis. These log samples are needed to develop middleware which will format the logs into an acceptable data stream for the Pace University Keystroke System (PKS), which would accept these log files as data input. Five individuals have entered short paragraphs of about 100 words in a simulated e-mail application while Basic Key Logger has generated logs in the background. Each of the five users has entered ten free-text paragraphs which five will be used to train the Pace University Keystroke System and five will be used for biometric testing. Once the appropriate middleware is developed, training and testing the system will begin.

## 5. References

[1] A. Brandt, "Biomeric Security Barely Skin-Deep," , 2008, http://www.pcworld.com/article/103535/biometric_security_barely_skindeep.html, accessed October 2010.

[2] Bioidentification FAQ, http://www.bromba.com/faq/biofaqe.htm#Messgroesen, accessed October 2010.

[3] D. Campbell, "The Importance of Biometrics in the U.S. government's Response to 9/11," Biometric Consortium 2005 Conference, 2005, http://www.biometrics.org/bc2005/Presentations/Conference/2%20Tuesday%20September%2020/Tue_Ballroom%20B/CampbellBiometricsConsortium2005Conference.pdf, accessed October 2010.

[4] KeyLemon Face Recognition Company Website, http://www.keylemon.com/product/, accessed October 2010.

[5] Luxand Face Recognition Company Website, http://www.luxand.com/facesdk/download/, accessed October 2010.

[6] MyBioidentity Company Website, http://www.mybiodentity.com/generalsite/applications.asp, accessed October 2010.

[7] National Science and Technology Council (NSTC). http://www.biometrics.gov/Documents/FaceRec.pdf, accessed October 2010.

[8] C. Tappert, "Biometrics Background," http://www.csis.pace.edu/~ctappert/it691-10fall/projects/biometrics-background.htm, accessed October 2010.

[9] VoiceLatch Company Website, http://www.voicelatch.com/voicecipher/, accessed October 2010.