To increase personal computer security, keyboard biometrics is a robust option that uses common hardware. By recognizing differences in key press duration and latency between key pairs, a highly personal profile can be created, which can be used as a unique identifier for an individual. Instead of testing the user's identity once at log in as prior work has done, we focused on continuously monitoring keystrokes in the background from the user's normal typing and using that information to verify the typist's identity. After comparing the typist's current key strokes to the computer's saved profile, the program can challenge the user to verify his or her identity if appropriate. Using segments of 150 keystrokes, we obtained zero false lockouts and only 5% missed lockouts. More keystrokes decrease the percentage of missed lockouts.

Introduction

Biometrics is the science and technology of determining identity based on physiological and behavioral traits. Other physiological biometrics include retinal scans, finger and handprint recognition, and face recognition. Behavioral metrics include handwriting analysis and voice recognition. Keyboard biometrics is considered a behavioral biometric. Behavioral biometrics cannot be falsified without explicitly targeting a particular user and training extensively to match their behavior.

Using keyboard biometrics is a viable means to achieve improved computer system security. Although the data analysis techniques used here do not offer the same level of extreme scrutiny as a fingerprint or retinal scan, they also do not require end users to purchase any special hardware to use, and thus make for a worthwhile security device.

The biometric works because everyone has his or her own typing pattern. At the surface level there are clear differences in overall speed from one user to the next. Some users are heavier upon the keyboard, with long durations and small digrams, which translates to a lot of overlapping key presses. Other users are more “hunt and peck”: pressing keys down quickly and then retracting, which would translate to small durations and longer digrams. On a deeper level, each user has a unique pattern as to how quickly he or she presses each particular key, as well as how quickly he or she transitions from one particular key to another. Users vary so much in these unique patterns that they can be used for accurate biometric identification.

Our algorithms for profile creation rely on two data sets: key press duration and key digram duration. Durations are the time that a given key is pressed down, i.e. from key-down to key-up. Typical times range from 70 to 150 milliseconds. Key digram durations are measured from the key-down event of one keypress to the key-down event of the following. These can range from 50 to 200 milliseconds. We found a large amount of variance between users using these two metrics.

Summary of Prior Work

Keyboard biometrics has reached the level of having a noticeable impact on computer security. Specifically, prior work has reached close to 0.01% impostor pass rate (less than 1 successful attack out of 10,000 attempts) and a 4% false alarm rate based on a user group of 154 participants. Up to this point, there have been some hindrances in implementing the work already completed, most notably that 300 characters must be typed for proper comparison to an existing profile. Our goal is to improve on this level of accuracy.

In 1980, R. Gaines conducted experiments with seven secretaries who were asked to retype three paragraphs two times over a four-month period. Keystroke latency timings were collected and analyzed using the specific digrams that occurred during the paragraphs. This means that the two different times were averaged to create a profile for each of the secretaries. The results were very encouraging but the data offered too small of a sample to build an acceptable profile.

Later experiments conducted by John Leggett actually demonstrated identity verification, utilizing and validating Gaines’ work. Leggett’s work accepted a user if more than 60% of the comparisons were valid. However the false alarm rate was 5.5% and the impostor pass rate was 5%. The problem was that a 1,000 word sample size was required in order for proper comparison of profiles.

Rick Joyce and Goyal Gupta used keystroke information taken during the login process. The system made new users provide reference signatures by typing their username, password, first name and last name eight times, which was used to generate a profile for the new user. Later, the user was required to type his signature, which was compared to the mean and standard deviation of the profile. While this
work was very promising, it required extensive profile setup and training of the system. Although interesting research, the practical applications implied here were not realistic.

Fabian Monrose and Aviel Rubin’s later paper added onto the previous research. They examined the use of keystroke durations, developed more robust methods for examining the differences between user profiles, studied the use of unstructured text, and incorporated a larger sample set. A Euclidean distance measure was utilized for analyzing the difference between profiles, and a probabilistic measure was the actual deciding factor for accepting or rejecting the user. Utilizing these methods, they managed a 90% correct identification rate.

In 2002, research by Francesco Bergadano, Daniele Gunetti, and Claudia Picardi addressed typing errors and the intrinsic variability of typing. Their experiment used a single text of 683 characters for all participants. Their acceptance/rejection utilized the Degree of Disorder of an Array, the average differences between the units in the array. The false alarm rate of 4% and im-postor pass rate of less than 0.01% are remarkable, but this technique requires a large profile for proper authentication. The research does prove, however, that keyboard biometrics is reliable given a sufficiently large profile. Since this research was designed for authentication of the user at login, as a replacement for passwords, it is not applicable for continuous authentication: this method requires having prior data on the way the user types particular passages.

Our work is designed around building a background process that would actively collect data of the user typing naturally and would use that to build his profile. Once the profile had been built, a security tool would watch for typing patterns that did not fit that profile and simply challenge the user to prove his identity when appropriate. In this way, we eliminate the need for the user to train the system with artificial text, and do not present significant hurdles to the system-login process. We target casual users who want an extra layer of security and may, at times, forget to lock their desktop systems.

Procedure
Data Collection
We acquired raw keypress data by distributing a standalone Windows application to 42 individuals. This program recorded their keystrokes over the period of several days and reported the data to a central repository via an encrypted internet connection. The smallest profile collected was 3,746 keystrokes, and the largest was 178,392. We offered users the ability to pause data collection and delete recently collected data, in order to ensure only appropriate data was collected. Users’ keyboards each used one of two timing intervals for reporting activity: 10ms and 15ms. These data sets are not compatible for comparison. 26 users sent in data with 10ms timing information, so we exclusively used that data in our analyses.

Profile Creation
We turned raw keypress log files into profiles by collapsing data on keypress duration and digram duration. Once a user profile has been created, it can be used to continuously authenticate the person who is typing. We compare the authorized user’s profile to a second profile, which is continuously updated from the keyboard readings. To keep the profile current, only a subset of the input is used; the profile is
created from the most recent set of digrams and durations (the set size can be arbitrarily changed), and it is updated every few keystrokes.

Once the two profiles have been created, they can be compared. Our tests have shown that the top 100 digrams (or all available digrams when fewer than 100 are available) and the ten most frequent durations from the current input provide the best grounds for comparison of profiles. These parameters were determined through numerous trials with various parameters; they provided the highest levels of accuracy across various input segment sizes. However, these choices have no formal statistical grounding, and further experimentation may find better parameters.

Profile Comparison Process

The comparison process uses the chi-square goodness of fit test to see how well the individual digrams and durations fit the distribution from the user profile. For example, the timings in the input for the digram “T:H” (that is, pushing ’t’ and then ’h’) would be compared against the timings in the user profile, and the chi-square value would be the result of this test. Because the amount of data gathered within one hundred digrams is quite small, we found that the best way to combine these results into an overall score was to average all of the chi-square values together, but to keep the types of data (digrams and duration) separate. This results in an overall score for how well the input digrams fit the profile digrams, as well as how well the input durations fit the profile durations. Because this score is the average of the chi-square values, and the greater the chi-square value, the less the samples fit together, then the greater the overall score, the worse the fit is between the current user input and the profile input.

Lockout Procedure

To set the threshold at which we lock out the typist, we considered looking at the critical values for the averaged chi-square value to determine the actual probability that the profiles matched. This did not prove to be the optimal approach for several reasons. Firstly, because the majority of digrams in the current input have only one timing and the chi-square test does not give the best probability estimates for such small numbers of examples, the resultant probability would not necessarily be accurate. Also, different users have different amounts of variance for the way that they type; some users are extremely consistent, and other users have a large amount of variance. We wanted to use a method that would allow us to minimize the accepted variance without increasing the false-alarm rate.

The approach we used for finding the minimum variance to accept was more empirical than theoretical. First, we built a user profile from all of the example input we had for the authorized user. Then, we looked at every sequential subset of one hundred digrams and durations within that profile, and compared that subset against the full profile. This allowed us to test all of the typing examples we had for that user against their profile, to see how much their subsets of typing varied in comparison to their overall profile. We then took the maximum of their duration and digram scores, and set them to be the cutoff thresholds. As we noted above, while the program is running, if the current input exceeds either threshold, then the user is locked out.

Measurement of Accuracy

To measure the accuracy of these thresholds, we then ran every other user’s data file we had against the original profile. We compared the durations and the digrams separately against the original profile, to obtain the percentage of input segments from other profiles that would be accepted, if either durations or digrams were considered. This gave us distinct accuracy rates for digrams and durations. The current system does not combine the data gathered from durations and digrams in a more sophisticated way. The results of this comparison are below.

This system allowed us to accept all of the variance that occurred within a given user’s typing habits (that we have data on), without accepting any additional variance. Also, this process allowed us to empirically measure the accuracy of a given profile. We found this system to be quite accurate and reliable, and we also found that a profile of sufficient size would capture nearly all of a given user’s variations. Such a profile could be obtained within roughly 50,000 keystrokes, which should occur within some small number of days, depending on how frequently a person types on their computer.

Completeness of Profile

Currently, we do not continuously expand the profile with...
Acknowledged input, but this would be a trivial feature to add. However, we have not developed a theoretical understanding of what constitutes a complete profile. It is important to know if a profile is complete or not, as an incomplete profile is going to have a higher false-alarm rate than a complete profile. This would be useful knowledge both for lowering the false-alarm rate while in use, and also to better measure the accuracy of the profiles. Our preliminary idea is to take all of the raw data we have for a given user’s typing habits, divide it in half, and compare each half to the other. When both halves accept the other more than a given percentage of the time (say, 99%), then we can be confident that we have captured virtually all of the user’s typing habits, and the profile is complete. However, if the user’s typing habits slowly change over time, then the profile will eventually become incomplete. We have not found a method to ensure that the profile will remain complete over time, without allowing for variances that no longer occur.

Results

For the final comparison of the results, we ran several round-robin tests utilizing all of our collected data. For the first test, we loaded the 27 user profiles, and then split all of the collected data into 100 digram segments. We then compared every one of these 3,500 segments against each profile. The segments that came from the profile currently being tested were used to set the thresholds for acceptance, and thus a user’s own input segments were never rejected. After comparing the input segments from other users against each user’s profile, we found the percentage of 100 character input segments from other users that were accepted by each profile. We then repeated this test for 50, 150, 200, 250, 300, 350, 400, 450, and 500 character input segments.

The following graphs show individual profile false positive rates and the overall false positive rate for all profiles. The false positive rate indicates not what percent of users are wrongly accepted, but what percent of input samples from other users are wrongly accepted. In other words, even if a user is wrongly accepted at first, he is increasingly likely to be rejected with each keystroke.

By 150 digrams the lockout rate has reached nearly 95%, and by 500 digrams the lockout rate exceeds 99%.

The point of diminishing returns is around 150 digrams, as further data has a reduced impact on the accuracy of the profiles. The very high false positive rates for 50 and 100 digram segments is a reflection of the very limited amount of data that is available within small segments.

The above are graphs of the false positive rate when using duration data versus digram data, in 100 durations segments.

The much lower accuracy from keypress durations shows how a user’s typing pattern for keypress durations is much less characteristic of himself than digram times. However, taking duration data into account in combination with the digram data allows for accuracy to be improved, if only marginally in many cases. For a few users, however, their duration timings are unusually unique, and this data alone results in very high levels of accuracy.

Discussion

The goal of this research was to utilize keyboard biometrics in a real-time non-invasive security system. The system’s ability to continuously attempt to authenticate the user based on normal computer usage is vital. The most successful prior work, which utilized logins authentication with a pre-determined passage of 683 characters, had a false lockout rate of 4% and an imposter passing rate of .01%. Our continuous user authentication system, when using an input size of 683 digrams from normal computer use, yields a theoretical false lockout rate of 0% and an imposter passing rate of 0.8%. However, the system is subject to the law of diminishing returns, with an increasing number of key strokes needed to lower imposter passing rates. The best performance/data size ratio occurred at around 150 digrams, which resulted in a 95% correct lockout rate. We found that duration data was generally not highly characteristic of individual users, but for a few users it was very unique, and resulted in a false positive rate below 5%.

Usage Concerns

The most often posed question regarding our research has been: won’t you falsely lock the user out if she is eating and typing with only one hand? The answer is yes, we will, but that doing so is the correct behavior. We only know about the user’s typing patterns based on the data we get. If the user frequently types with a sandwich in one hand, that will become part of her profile and will be accepted. If she types abnormally, however (in the sense that she is typing in some way we have not seen before), the proper response is to challenge her to prove her identity, which is what we do.
Future Work

There are several ways to improve upon this research in the future. Methods of keeping the user profile up to date need to be looked into. Currently, our system can either not accept new typing patterns that the user slowly develops over time, or it will accept them, and also accept old, unused patterns as well. The different outcomes result from whether or not the system will add new verified input to the user's profile. Research into continually pruning and updating the profile may allow for a more optimal solution. However, this would be difficult without keeping around a large key log of the user, which is undesirable from a security standpoint.

The addition of mouse biometrics would increase the uniqueness of each user profile, as well as providing a way to completely secure input-device interactions with the computer; a user can accomplish a lot with just a mouse. Unfortunately, mouse biometrics involves a completely different type of data, as mouse movements are difficult to analyze into discrete, comparable features. Also, our initial investigation into mouse behavior found that a lot of mouse usage behavior is application specific. Future research will have to either look for general mouse behavior features, or a way to catalog mouse behavior for each different application.

Another goal is to reduce the number of keystrokes needed to determine who the user is. The problem is that below one hundred keystrokes, nearly all of the digrams have only one example data point, and it is very difficult to get accurate estimates of whether a single timing came from a particular user; several timings are needed for a good estimate. However, we believe that it may be possible to group several digrams together, and thus group several single points into a distribution, to compare against the profile’s digram groups. The question is whether there are digrams or durations that are sufficiently similar in order for them to be grouped together without losing accuracy. We have found that many digrams and durations share the same average and standard deviation of timing, but we have not looked at the complete distributions of timings for different digrams or durations from the same user. If the distributions match as closely as the average and standard deviations do, then it would be possible to group the digrams or durations, and thus have more data points within each grouping to do calculations with. Because each user has a unique pattern for their durations and digrams, the groups would need to be specific to each user. The result of this would be that we would be able to reduce the number of keystrokes needed to accurately determine a user’s identity.


In the spring of 2004, the Keyboard Biometrics team emerged out of CSC200: Undergraduate Research Seminar. Consisting of two sophomores (Lu and Ganzhorn) and three seniors (Ordal, Norwood, and Fong), the group was advised by Professor Michael L. Scott. Ordal is currently working for UR Admissions. Norwood is employed by Gestalt LLC. Fong is training to be a Radiation Treatment Dosimetrist at Memorial Sloan Kettering Cancer Center. Both Ganzhorn and Lu are currently working on their honor senior thesis, in Simulation of Societal Evolution, and the Automatic Transcription of Music using Genetic Algorithms and MIDI synthesis, respectively.