

Detecting Computer Intrusions Using Behavioral Biometrics

Ahmed Awad E. Ahmed, and Issa Traore

Department of Electrical and Computer Engineering, University of Victoria

P.O. Box 3055 STN CSC Victoria, B.C. V8W 3P6 CANADA

aahmed@ece.uvic.ca, itraore@ece.uvic.ca

Abstract

In this paper we introduce the idea of using behavioral biometrics in intrusion detection applications. We present a new biometrics-based technique, which can be used to detect intrusion without the need for any special hardware implementation and without forcing the user to perform any special actions. The technique is based on using “keystroke dynamics” and “mouse dynamics” biometrics.

We discuss the efficiency and applicability of such an approach.

1. Introduction

Attacks targeted by intrusion detection systems can be divided in three forms: user-level, system-level, and network-level [1]. Typical user-level attacks consist of masquerade attacks. Examples of system-level attacks include privilege escalation such as buffer overflow, program modification such as Trojan horse, and denial of service. Popular examples of network-level attacks include network denial of service and probing. Even though a typical computer intrusion involves exploiting together several of these vulnerabilities, user-level attacks remain one of the most recurring forms of intrusions because successful user-level attacks always serve as prerequisite for most forms of intrusions, denial of service being an exception. It is therefore essential to develop effective countermeasures against these kinds of attacks.

The last several years have seen significant advances in handling system and network -levels attacks. Typical approaches for detecting system-level attacks involve some form of system calls monitoring [2,3]. Network level attacks are typically dealt with through network traffic monitoring [4,5]. User-level attacks have been dealt with mostly in conjunction with system-level attacks by anomaly detectors using statistical profiling [6,7].

Statistical profile-based detection uses a set of metrics to compute some measurements of user behavior, and

compares them against a set of values that characterize normal user behavior. Any discrepancy between the computed values and the expected ones is considered an intrusion. The metrics are computed using standard statistical techniques. Existing profile-based detectors are, however, characterized by significantly high false alarm rates mainly due to the low accuracy of the profiles computed [1]. In effect, the effectiveness of the metrics used to compute these profiles can easily be questioned. For instance some anomaly detectors base users’ profiles on metrics such as the average number of files opened or emails sent daily. It is, however, easy to find several users sharing the same habits. Worse, it is easy for any user to change his habits and adopts the usage pattern of other users!

We propose in this work a new approach to user profiling based on biometrics. The profiles computed in this case are more accurate than those obtained through the traditional statistical profiling techniques, since they are based on distinctive biological characteristics of users. Biometrics technologies have been successfully used for access control and user authentication, but to our knowledge, never for intrusion detection [8]. There are several reasons to that; we discuss those reasons in Section 2 and outline possible solutions to handle them. Section 3 introduces our detection framework. Section 4 overviews the biometrics technologies used in our framework. In Section 5, we report on experiments with using behavioral biometrics for intrusion detection. In Section 6, we make concluding remarks and discuss further research.

2. Biometrics and Intrusion Detection

Different types of biometrics are currently available in the market, and are widely used in various security applications. Biometrics can be classified into two categories, “physiological biometrics” and “behavioral biometrics” [8,9]. Physiological biometrics identify the user based on physiological characteristics, such as fingerprints and eye retina/iris scanning, whereas behavioral biometrics depend on detecting the behavioral

features of the user, such as signature, voice, and keystroke dynamics.

The utilization of biometrics technology, however, has so far been limited to identity verification in authentication and access control systems. Hence important security applications such as intrusion detection systems have been left out of this technology. We have identified two primary reasons for that. First, most biometrics systems require special hardware device for biometrics data collection, which restricts their use to only networks segments that provide them, making the systems irrelevant for a significant number of remote users, who operate out of these network segments. Second, most biometrics systems require an active involvement of the user who is asked to provide some data sample that can be used to verify his identity. This excludes the possibility of passive monitoring, which is essential for intrusion detection. There are also a number of secondary obstacles to the use of biometrics for intrusion detection such as whether the technology allows dynamic monitoring, or real-time detection.

A popular biometrics system, which escapes some of these limitations, is keystroke dynamics biometrics [10,11,12]. Keystroke dynamics doesn't require special hardware device for data collection (a regular keyboard is enough), and under certain circumstances can be used for dynamic monitoring [11]. The traditional keystroke technology, however, doesn't allow passive monitoring: the user is required to type a predefined word or set of words that is used to identify him.

In this work we use a new behavioral biometrics based on computer mouse dynamics¹ [13]. Mouse and keystroke dynamics biometrics are two related technologies. Mouse dynamics, however, fulfills all the characteristics required for intrusion detection since it allows passive, dynamic, and real-time monitoring of users, and it simply requires a standard computer mouse for data collection. Actually, mouse and keystroke dynamics are complementary biometrics. While a mouse is very important for graphical user interface (GUI) –based applications, a keyboard is essential for command –line based applications. So our objective is to combine these two technologies in a common detector. In this respect, we have adapted the traditional keystroke technology by addressing issues such as passive and dynamic monitoring[14].

Using biometrics in intrusion detection systems opens a new dimension in the detection process. By combining traditional intrusion detection systems that focus on the actions conducted by the user, with biometrics that focus on the identity of the user, such systems are able to detect

the type of intrusion where an attacker gains access to the resources and starts performing normal non-intrusive procedures, causing information leakage or exploiting any other vulnerabilities. Differences in usage pattern cannot be detected if the attacker knows the operation sequences and his access limits; such an attack, however, can be uncovered if the detection is based on biometrics information.

Another advantage of using biometrics is the reduction of false alarms, which are considered among the major problems in current implementations of intrusion detection systems. With traditional detection systems, a false alarm can happen if the legitimate user performs an unexpected action, or changes his behavior. Eliminating these types of false alarms has a great influence on the overall accuracy of the system. With the aid of biometrics-based detection, the intrusion detection systems can ensure the identity of the user during unexpected operations, and as a consequence improve significantly the correctness of such systems.

3. Detection Framework

3.1. Basic Architecture

Figure 1 depicts the architecture of our detector. The detector is implemented as client/server software. The client module, which runs on the monitored machine (e.g. potential victim), is responsible for mouse movement and keystroke data collection. This data is sent to the server software, which runs on a separate machine. The server software is in charge of analyzing the data and computing a biometric profile. The computed profile is then submitted to a behavior comparison unit, which checks it against the stored profiles.

Like all other biometric systems, our detector operates in two modes: enrollment mode, and identification / verification mode. The operation of each mode consists of three consecutive stages. In the first stage of the enrollment mode, a data capturing process is conducted by a lightweight software module, which captures all mouse and keyboard actions, and converts them into a set of more organized and meaningful statements. These statements are directly passed to the next stage of data processing where behavioral modeling and feature extraction is conducted. This process accumulates all actions received from the previous process over a pre-defined session period and performs a number of algorithms on the data to produce the Mouse Dynamics Signature (MDS) and Keystroke Dynamics Signature (KDS) for the user being monitored [14]. Finally, in the third stage, the generated signature is stored in a database as a reference signature for the enrolled user.

¹ Patent Pending

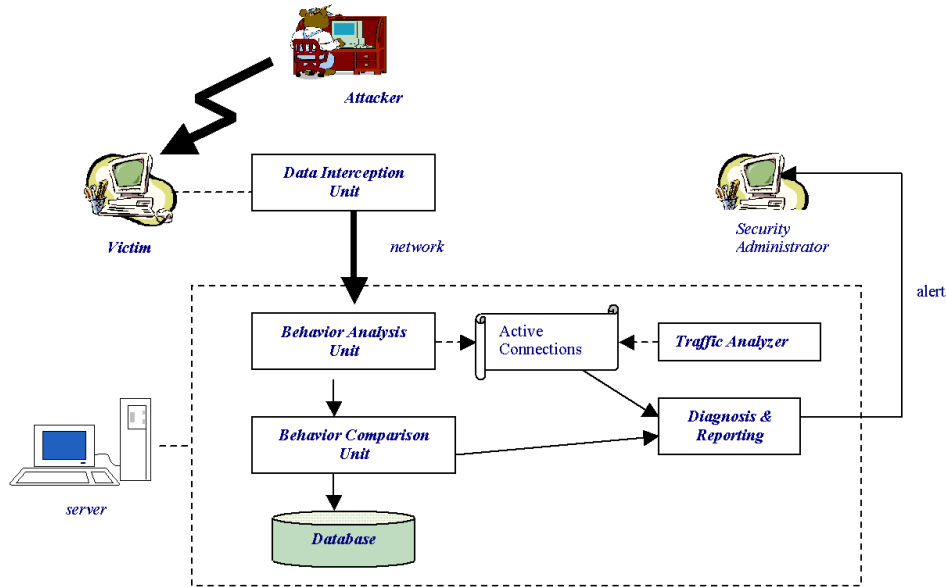


Figure 1: Detector Architecture

The detection mode shares the first two stages with the enrollment mode. The third stage in this mode is the verification process where the signature calculated during the data processing stage is compared against the reference signature of the legitimate user. Different comparison algorithms are used for each signature factor; the more the deviation from the reference signature, the less the system is confident in the identity of the user.

Among all biometrics, the mouse and keystroke dynamics biometrics are considered the most practical from an implementation point of view. Since real life usage of most of the GUI based operating systems involves a combination of mouse and keyboard actions, utilizing both mouse dynamics and keystroke dynamics biometrics increases the accuracy and speed of the detection process without much influence on the cost or performance of the system.

3.2. Enforcing Biometrics Data Collection

In the architecture presented in the previous section the client software is installed on the monitored machine. An alternative may consist of enforcing the presence of agent software on any remote accessing machine. Each of these solutions has advantages and disadvantages. The first solution allows collecting data transparently but may suffer from delays in the collection process, which may impact the collected data. In contrast, with the second solution, there is no delay in the data collection whereas transparency is not guaranteed.

An important question still remains about feasibility of the second option: how can we enforce data collection on the attacker's machine? First, for local users the security administrator can simply make sure that the data collection software is installed on local machines. For remote users our approach consists of either providing our own remote login software or extending secure remote login software such as SSH. Then the administrator can require that users use this particular remote login implementation for remote access. It is common practice in most organizations that remote access be regulated by a defined and strict policy. For example, for each machine connected to the protected domain the administrator can enforce the following policy:

- There is NO rexec or telnet access to this machine.
- There is NO rlogin or rsh access to this machine from outside of DOMAIN_P.
- FTP is NOT secure and may be removed from this machine in the near future.
- To access this machine remotely, use Secure Shell protocol 2 (SSH2), Secure FTP (SFTP), and / or Secure Copy Protocol (SCP)
- Bio Client Version 1.0 should be running on the remote side in order to access the machine remotely.
- Software available on this machine is listed at: http://Web_Domain/computing/software.shtml
- Use of this facility must adhere to: 'Policy 6030: Organization Computing and Telecommunications User Responsibilities', http://Web_Domain/policies/pol6000/6030CTUR.html AND 'Organization Standards for Professional Behavior', http://Web_Domain/policy/professional-behaviour.html
- Note that this machine will usually be rebooted at the end of every month. Please schedule your jobs accordingly.

In order to ensure that only users abiding by this policy access the monitored network, our biometric detector is extended with a network traffic analyzer, which monitors both attempted and established connections to the target machine. The connections list established by the traffic analyzer is compared against the active users list maintained by the core biometrics detector, and possible discrepancies are then reported as intrusions to the security administrator. This applies even when the data collection module is installed on the target machine.

If the network analyzer detects resource usage on the target machine while there is no biometric data collected during a session, this will raise the possibility that corresponding network traffic is due to a malicious process, which is not being executed by a legitimate user. On the other hand, if the biometric detector is able to monitor activities on the target machine while the network analyzer failed to detect the network traffic resulting from such activities, this will raise the possibility that the attacker managed to modify the behavior of the running application.

The proposed architecture raises two important issues. First the client/server communication scheme used is subject to protocol attacks. This is common to all distributed intrusion detection systems. How to ensure for instance that an intruder cannot intercept and modify the collected data. Obvious solution for that consists of using secure communication protocols for client and server interactions. Second the biometrics system is subject to forgery; this is common to all biometrics technologies. Forgery can happen by observing the biometrics generation process or by stealing biometrics samples. In the particular case of mouse and keystroke dynamics forgery by observation is extremely difficult to achieve. In contrast with physiological biometrics, the impact of biometrics sample theft can be alleviated for behavioral biometrics by using secure communication protocols. The main reason being that behavioral biometrics data such as mouse and keystroke dynamics vary with the application. Samples collected from a word processing session will be different from samples collected from a web browsing session.

4. Biometrics Technologies

Our detection framework is based on mouse and keystroke dynamics biometrics, which represent two separate but related biometrics. Data collection is performed using a common detection module, but processing for each biometrics technology is based on different algorithms. In the sequel, we give an overview of the biometrics technologies involved in our detector.

4.1. Keystroke Dynamics Biometrics

Keystroke dynamics is considered a strong behavioral biometric [10,11,12]. The functionality of this biometric is to measure the dwell time (the length of time a key is held down) and flight time (the time to move from one key to another) for keyboard actions. After these measurements have been collected, then the collected actions are translated into a number of digraphs or tri-graphs to be analyzed in order to produce a pattern that identifies the user who generated these keyboard actions.

Table 1 shows a combination of tri-graphs generated from three sessions for two different users, and the corresponding time used to perform the tri-graphs in milliseconds. The tri-graphs shown are centered by the character ‘a’ (ASCII code 65). From the table we can notice the similarity between the response time for the first user’s sessions, we can also notice obvious difference in behavior between the two users which can easily be detected for some of the tri-graphs (marked in bold).

In access control applications the extracted group of digraphs and tri-graphs are pre-defined since the user is asked to enter a paragraph containing them. In intrusion detection applications, however, this scenario is not applicable. Detecting the behavior from an unexpected set of digraphs requires large amount of data to be collected in the enrollment mode so as to cover a higher percentage of the captured data in the verification mode.

Our detection algorithm generates a Keystroke Dynamics Signature or KDS, which is used as a reference user profile and matched against active user profiles to dynamically detect masqueraders.

| Tri-graph ASCII Code | User 1 Session 1 | User 1 Session 2 | User 2 |
|-------------------------|---------------------|---------------------|------------|
| 87-65-68 | 86 | 85 | 73 |
| 83-65-89 | 83 | 82 | 69 |
| 77-65-78 | 76 | 70 | 60 |
| 70-65-69 | 134 | 112 | 62 |
| 82-65-72 | 122 | 92 | 80 |
| 77-65-78 | 74 | 76 | 68 |
| 87-65-68 | 80 | 81 | 71 |
| 83-65-89 | 71 | 75 | 111 |
| 83-65-76 | 62 | 62 | 59 |
| 83-65-76 | 67 | 64 | 63 |
| 76-65-77 | 143 | 205 | 56 |

Table 1: Time used to perform different tri-graphs for two different users

To construct the KDS, we propose a key oriented neural network based approach, where a neural network is trained for each keyboard key to best simulate its usage dynamics with reference to other keys. We also propose a technique which can be used to approximate a digraph/tri-graph value based on other detected graphs and

the locations of the keys with reference to each other, aiming to speed up the user enrollment process.

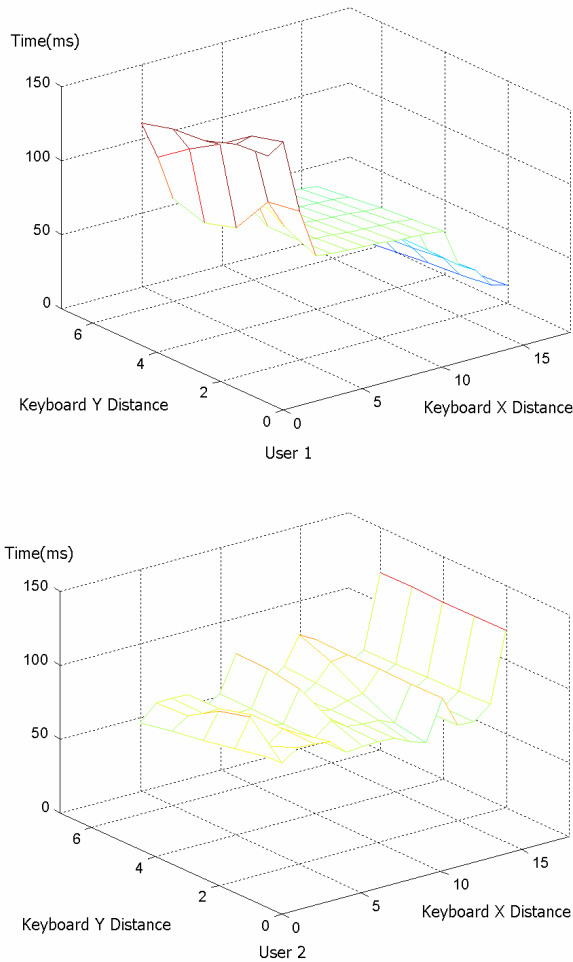


Figure 2: Di-graph approximation matrices for two different users

Figure 2 shows two approximation matrices for two different users performing a set of di-graphs originating from the same keyboard key. Values of the x and y axis represent the location of the destination key on the keyboard in reference to the originating key. From the figure we can notice a remarkable difference between both users' behaviors. Enrolled user's KDS consists of a set of approximation matrices for all keyboard keys. Data captured in the detection mode are formatted and compared against its corresponding matrix.

The higher the deviation from the matrix the less confidence that the actions belong to the same user.

4.2. Mouse Dynamics Biometrics

Mouse dynamics is a new behavioral biometric recently introduced [13]. The idea behind this biometric

is to monitor all mouse actions generated as a result of user interaction with a graphical user interface, and then process the data obtained from these actions in order to analyze the behavior of the user. Mouse actions include general mouse movement, drag and drop, point and click, and silence (i.e. no movement).

The behavioral analysis utilizes neural networks and statistical approaches to generate a number of factors from the captured set of actions; these factors are used to construct what is called a Mouse Dynamics Signature or MDS, a unique set of values characterizing the user's behavior over the monitoring period. Some of the factors consist of calculating the average speed against the traveled distance, or calculating the average speed against the movement direction. In [13] up to seven factors that exhibit strong stability and uniqueness capability are reported.

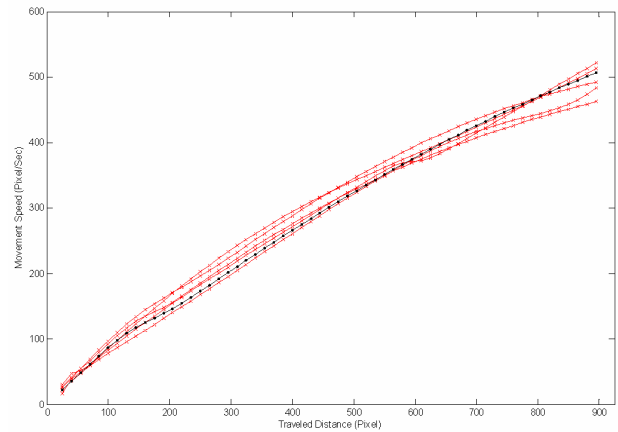


Figure 3: Active profiles compared to the reference profile for the same user

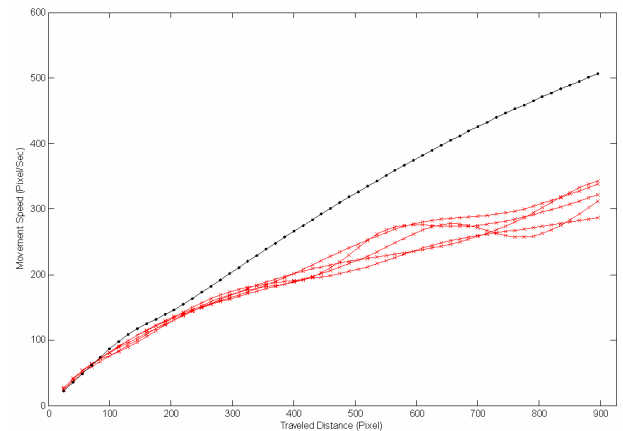


Figure 4: Active profiles of a given user compared to the reference profile of a different user

For instance, Figures 3 and 4 show examples of biometrics profiles based on the average speed against

the traveled distance factor denoted MSD. The x-axis represents the traveled distance and the y-axis represents the movement speed. Each point on these figures represents an intercepted mouse action. Figure 3 shows data from five different sessions for the same user compared to a reference signature of the same user. As we can see from the figure the curves are very close to each other and to the reference signature as well. To give an idea about the detectable deviation, Figure 4 shows a comparison between five signatures for the same user compared against a reference signature of a different user.

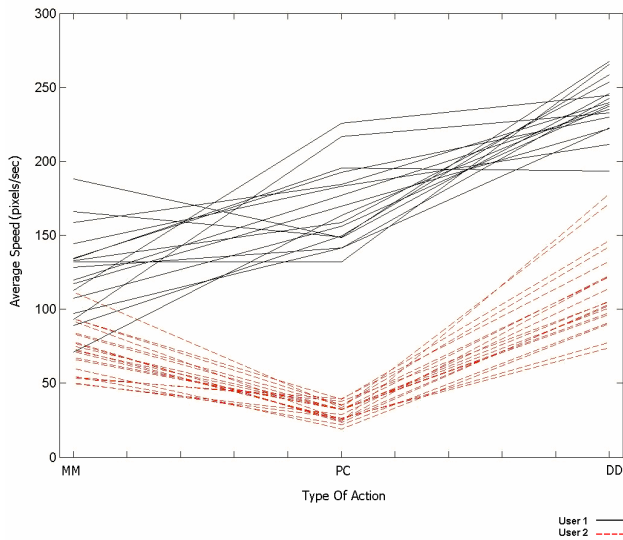


Figure 5: Average Speed for Different Types of Actions, comparing large number of sessions for two different users.

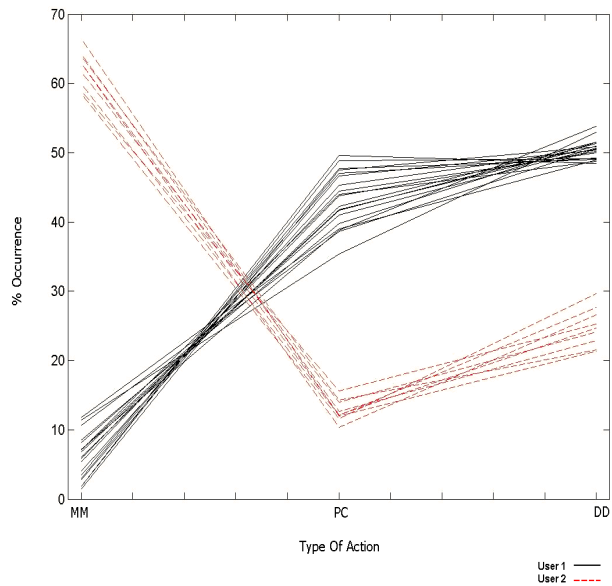


Figure 6: Types of Actions Histogram, comparing large number of sessions for two different users.

Here also we notice the closeness of the profiles of the same user, and their distinctiveness with respect to the reference signature of a different user.

The comparison technique used for this factor consists of computing the sum of the absolute difference between the curves; this represents how far the curves are from each other; if it is higher than a threshold then those curves belong to two different users. The threshold can be determined for each user during the enrollment phase when the reference mouse signature is generated.

Figure 5 shows the relation between the movement speed and the type of performed action for the three recognized types of actions (MM: Mouse Movement, PC: Point and Click, DD: Drag and Drop). A large number of sessions for two different users are shown. Each session is represented by a line connecting the three readings involved in this factor. From the figure we can notice the deviation between signatures of the two users as well as the reproducibility of this factor.

Figure 6 shows the histogram of the types of actions for a number of sessions for two different users. Behavior differences can be easily detected for the two users and values and ratios between entries can easily be identified. From the figure we can notice that the first user performs very low number of regular mouse movements and depends mostly on point click and drag drop types while the second performs very high number of regular mouse movements, and very low number of point and click actions.

Including more factors in the user signature has a great influence on the accuracy of the detection process. The detection algorithm developed in [13] utilizes the seven detectable factors. It calculates the significance of each factor with respect to the other factors in the same signature, and with respect to its corresponding values in other users signatures. A neural network is trained for each enrolled user resulting different detection scheme to be used for each of them. The developed approach proved its effectiveness in responding to the simulated attacks in the conducted experiments; more details are given in the next section.

5. Experiments

We have conducted some experiments involving 22 participants, and collected experimental data over 9 weeks. Participants installed the client software and used their machine for their routine activities. Mouse and keystroke data was collected transparently and sent to a central server. At the end of the data collection phase, we used the collected data to conduct an offline evaluation of our detection system.

| | Mouse Signature Factors | | | | | | | | | | | | | | | | | CR | | |
|-----------------|-------------------------|-------|-------|-------|-------|-------|--------|-------|-------|-------|-------|--------|--------|--------|-------|-------|-------|-------|-------|----------|
| | MDH | | | | | | | ATH | | | ATA | | | MSD | | | | | | |
| Legitimate User | 11.40 | 9.56 | 11.57 | 10.90 | 15.43 | 20.30 | 10.23 | 10.40 | 50.50 | 45.30 | 4.02 | 105.54 | 32.22 | 72.33 | 24.46 | 20.10 | 15.64 | 9.56 | 5.61 | 100 |
| | 11.57 | 11.40 | 13.75 | 11.57 | 12.91 | 17.61 | 8.72 | 12.24 | 50 | 44.46 | 5.36 | 104.94 | 37.78 | 54.12 | 23.21 | 16.18 | 14.58 | 10.02 | 5.44 | 100 |
| | 10.99 | 17.59 | 9.13 | 8.79 | 12.16 | 21.65 | 7.27 | 12.16 | 48.73 | 43.99 | 7.10 | 77.69 | 25.87 | 71.42 | 27.20 | 24.05 | 19.28 | 12.86 | 5.50 | 97.19 |
| | 12.86 | 14.40 | 11.32 | 7.71 | 13.89 | 21.26 | 7.20 | 11.14 | 50.6 | 38.59 | 10.63 | 102.64 | 24.99 | 91.80 | 32.15 | 19.35 | 13.07 | 10.21 | 6.08 | 100 |
| | 11.32 | 10.79 | 12.38 | 7.96 | 14.51 | 20.70 | 10.97 | 11.15 | 48.31 | 36.63 | 14.86 | 122.61 | 32.13 | 83.38 | 28.14 | 20.97 | 15.69 | 10.92 | 7.00 | 100 |
| Insider | 12.17 | 8.59 | 9.30 | 15.03 | 15.75 | 13.60 | 11.69 | 13.60 | 30.07 | 13.12 | 56.56 | 222.06 | 169.6 | 117.15 | 18.14 | 15.99 | 14.16 | 12.71 | 10.07 | 8.72E-07 |
| | 12.02 | 10.48 | 9.71 | 10.48 | 13.81 | 24.55 | 7.92 | 10.74 | 36.31 | 14.83 | 48.59 | 235.18 | 177.03 | 119.66 | 18.16 | 15.17 | 14.06 | 13.78 | 10 | 6.54E-09 |
| | 13.84 | 9.74 | 6.66 | 9.74 | 20 | 14.87 | 10.51 | 14.35 | 32.30 | 14.61 | 52.82 | 237.3 | 156.07 | 107.51 | 18.46 | 17.77 | 14.49 | 13.05 | 10.30 | 8.13E-10 |
| | 10.62 | 8.99 | 8.17 | 10.06 | 24.25 | 12.26 | 10.06 | 15.25 | 47.41 | 19.07 | 33.24 | 216.92 | 115.37 | 136.49 | 17.70 | 14.64 | 14.62 | 10.83 | 8.45 | 1.06E-06 |
| | 7.12 | 9.58 | 10.68 | 10.95 | 20.27 | 13.42 | 11.23 | 16.43 | 41.09 | 19.17 | 39.45 | 225.57 | 66.11 | 154.72 | 18.43 | 13.72 | 16.02 | 12.40 | 9.95 | 4.11E-07 |
| Outsider 2 | 13.45 | 7.64 | 17.73 | 13.45 | 10.39 | 16.51 | 10.70 | 9.78 | 7.33 | 2.44 | 89.90 | 272.83 | 127 | 110.82 | 19.10 | 18.95 | 17.01 | 14.56 | 11.30 | 1.43E-05 |
| | 9.02 | 10.6 | 17.76 | 9.28 | 8.74 | 14.75 | 16.66 | 12.84 | 13.66 | 3.55 | 82.51 | 150.32 | 69.846 | 107.05 | 22.88 | 20.35 | 17.88 | 15.49 | 12.83 | 1.39E-05 |
| | 12.53 | 9.11 | 7.12 | 21.08 | 13.67 | 10.25 | 12.25 | 13.67 | 18.51 | 9.11 | 72.08 | 203.2 | 86.125 | 83.02 | 21.60 | 20.26 | 18.24 | 15.21 | 10.72 | 1.43E-05 |
| | 13.05 | 5.55 | 12.22 | 15 | 13.05 | 13.88 | 13.61 | 13.3 | 13.61 | 6.66 | 79.44 | 202.69 | 105.83 | 92.629 | 20.74 | 17.83 | 16.09 | 15.01 | 10.62 | 1.43E-05 |
| | 9.39 | 8.18 | 10.90 | 16.06 | 11.51 | 14.54 | 15.15 | 13.93 | 22.72 | 11.81 | 65.15 | 174.61 | 87.282 | 105.58 | 22.69 | 22.69 | 16.75 | 16.75 | 12.5 | 1.39E-05 |
| Outsider 1 | 15.71 | 13.66 | 5.69 | 7.28 | 15.26 | 15.26 | 10.251 | 16.62 | 43.96 | 12.07 | 43.73 | 208.46 | 123.09 | 120.27 | 19.27 | 18.31 | 15.25 | 12.05 | 10.02 | 3.77E-05 |
| | 20.58 | 13.42 | 7.60 | 9.17 | 16.55 | 9.61 | 7.83 | 14.98 | 39.15 | 9.172 | 51.45 | 200.34 | 77.17 | 101.11 | 24.10 | 18.11 | 15.77 | 10.92 | 7.018 | 0.17946 |
| | 14.70 | 14.95 | 7.35 | 9.31 | 12.74 | 17.64 | 8.33 | 14.70 | 32.84 | 8.33 | 58.57 | 206.54 | 66.58 | 106.09 | 23.02 | 18.90 | 14.37 | 13.62 | 9.14 | 5.03E-07 |
| | 18.22 | 14.80 | 8.65 | 7.51 | 10.02 | 15.94 | 13.66 | 10.93 | 41.23 | 14.80 | 43.73 | 223.78 | 109.17 | 127.88 | 17.24 | 16.36 | 15 | 12.7 | 9.03 | 1.13E-06 |
| | 15.72 | 14.78 | 9.15 | 11.73 | 5.86 | 17.37 | 11.26 | 13.85 | 44.13 | 11.50 | 44.13 | 233.79 | 133.55 | 129.19 | 20.05 | 18.70 | 16.53 | 14.52 | 10.05 | 0.01187 |

MDH: Direction of Movement histogram
ATH: Type of Action Histogram
ATA: Average Movement Speed for Action Types
MSD: Movement Speed compared to Traveled Distance

Table 2: Simulated Attack: one insider and two outsiders masquerading as a legitimate user.

To do so, we divided the participants into 2 groups: a group of 10 representing authorized users and a group of 12 representing unauthorized users.

We computed a reference signature for each member of the first group using some of their own sessions. For each legal user we used the sessions belonging to the other users (authorized and unauthorized) to conduct some masquerade attacks on their reference signature. This resulted in a false negative rate of 0.651%.

Table 2 shows signatures for one insider (from the list of the authorized users) and two outsiders (unknown users) masquerading as a legitimate user. Five signatures resulting from five different sessions are shown for each user. The table also shows the confidence ratio (CR) calculated for each session in comparison to the reference signature of the legitimate user. The algorithm was able to confirm the identity of the legitimate user as the CR values resulting from both categories of attackers are very low compared to the values resulted for the same user.

To evaluate the false positives, for each legal user we compared their own remaining sessions (not involved in the computation of the reference signature) against their reference signature. This resulted in a false positive rate of 1.312%.

6. Concluding Remarks

This paper reports on results obtained in the development of a biometrics-based intrusion detector. Our goal is to provide a lightweight and self-contained

module specialized primarily in detecting user identities misuse.

We expect our detector to play a similar role as detectors such as tripwire do in the area of system integrity checking [15].

In this regard, in order to provide a full detection solution we expect it to be used in combination of complementary intrusion detection systems such as system-calls and network traffic monitoring systems.

7. References

- [1] McHugh J. "Intrusion and Intrusion Detection", International Journal of Information Security, 1: 14-35, 2001.
- [2] C. Ko, M. Ruschitzka, and K. Levitt, "Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-Based Approach", Proceedings of the 1997 IEEE Symposium on Security and Privacy, pp. 175-187, May 1997.
- [3] Forrest, S., Hofmeyr, S., Somayaji, A., and T. Longstaff, "A Sense of Self for UNIX Processes", Proceedings of the 1996 IEEE Symposium on Security and Privacy, May 1996, pp. 120-128
- [4] L. Heberlein, G. Dias, K. Levitt, B. Mukherjee, J. Wood, and D. Wolber, "A Network Security Monitor", Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy, pp. 296-304, May 1990.
- [5] S. Snapp, J. Brentano, G. Dias, T. Goan, L. Heberlein, C. Ho, K. Levitt, B. Mukherjee, S. Smaha, T. Grance, D. Teal, and D. Mansur, "DIDS (Distributed

Intrusion Detection System): Motivation, Architecture, and an early prototype”, Proceedings of the 14th National Computer Security conference, pp. 167-176, Oct. 1991.

[6] D. Denning, “An Intrusion Detection Model”, IEEE Transactions on Software Engineering 13 (2), pp. 222-232, Feb. 1987.

[7] T. Lunt, R. Jagannathan, “A Prototype Real-Time Intrusion-Detection Expert System”, Proceedings of the 1988 IEEE Symposium on Security and Privacy, pp. 2-10, Apr. 1988.

[8] L. O’Gorman, “Comparing Passwords, Tokens, and Biometrics for User Authentication”, Proceedings of the IEEE, Vol. 91, No. 12, Dec. 2003.

[9] Matyas, V. Jr., Riha, Z., “Toward Reliable User Authentication through Biometrics”, IEEE Security & Privacy Magazine, May/June 2003, Vol. 1 No. 3, pp 45-49.

[10] Gaines, R., Lisowski, W., Press, S., Shapiro, N., 1980. Authentication by Keystroke Timing: Some Preliminary Results. Rand. Report R-256-NSF. Rand Corporation.

[11] Legget, J, Williams, G., 1988. Dynamic Identity Verification via Keystroke Characteristics. Int. J. Man-Mach. Stud. 35, 859-870.

[12] Bleha, S., Slivinsky, C., Hussein, B., 1990. Computer-access Security Systems using Keystroke Dynamics. IEEE Trans. Patt. Anal. Mach. Int. PAMI-12, 12, 1217-1222.

[13] A.A.E. Ahmed, I. Traore “A New Biometrics Technology based on Mouse Dynamics”, Technical Report ECE-03-5, University of Victoria, Department of Electrical and Computer Engineering, Victoria, Canada, September 2003.

[14] A.A.E. Ahmed, I. Traore “Security Monitoring through Human Computer Interaction Devices”, Technical Report , University of Victoria, Department of Electrical and Computer Engineering, Victoria, Canada, March 2004.

[15] G. H. Kim, E. H. Spafford, “Experiences with Tripwire: Using Integrity Checkers for Intrusion Detection”, Proc. Systems Administration, Networking and Security Conference III, Usenix, 1994.