17

KEYSTROKE ANALYSIS AS A METHOD OF ADVANCED USER AUTHENTICATION AND RESPONSE

P.S.DOWLAND, S.M.FURNELL and M.PAPADAKI

nrg@plymouth.ac.uk
Network Research Group
Department of Communication and Electronic Engineering
University of Plymouth
Drake Circus
PLYMOUTH
PL4 8AA
United Kingdom
Tel: +44 1752-233521 *Fax: +44 1752-233520*

Key words: Keystroke Analysis, User Authentication, Biometrics, Intrusion Response.

Abstract: There has been significant interest in the area of keystroke analysis to support the authentication of users, and previous research has identified three discrete methods of application; static, periodic dynamic and continuous dynamic analysis. This paper summarises the approaches and metrics arising from previous work, and then proceeds to introduce a new variation, based upon application-specific keystroke analysis. The discussion also considers the use of keystroke analysis as a progressive, escalating response measure in the context of a comprehensive user authentication and supervision system, presenting an example of how this could be realised in practice.

1. INTRODUCTION

The issue of user authentication in IT systems has long been recognised as a potential vulnerability, with the majority of current systems relying upon password methods. Such methods have been repeatedly proven to be open to compromise, and can also be considered problematic in the sense

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35586-3_46

M. A. Ghonaimy et al. (eds.), *Security in the Information Society*

that they typically only serve to facilitate a one-off authentication judgement at the start of a session. A number of previous works [1, 2, 3] have consequently discussed the need for some form of monitoring to continuously (or periodically) authenticate the user in a non-intrusive manner. Although such monitoring is technically feasible, there are significant issues to be considered in selecting appropriate attributes to assess. This is particularly important, as continuous monitoring must be transparent to the end user in order to minimise any perceived inconvenience (with the exception of appropriate challenges in the event of suspected impostor activity).

A number of studies have considered the application of keystroke analysis to the problem of inadequate user authentication in modern IT system using static [4, 5, 6] and dynamic [7, 8] implementations. While these studies have evaluated the effectiveness of the proposed solutions, none have considered the implementation and necessary supporting application framework to effectively use keystroke analysis as a viable authentication and supervision mechanism.

This paper summarises the potential approaches to keystroke analysis, and presents details of a new method based on application-specific user profiling. It then proceeds to consider how keystroke analysis may be utilised as part of an intrusion response framework.

2. KEYSTROKE ANALYSIS OVERVIEW

Previous studies have identified a selection of data acquisition techniques and typing metrics upon which keystroke analysis can be based. The following section summarises the basic methods and metrics that can be used.

- **Static at login** - Static keystroke analysis authenticates a typing pattern based on a known keyword, phrase or some other predetermined text. The captured typing pattern is then compared against a profile previously recorded during system enrolment. Static keystroke analysis is generally considered to be an initial login enhancement as it can supplement the traditional username/password login prompt, by checking the digraph latencies of the username and/or password components (i.e. authenticating the user on the basis of both *what* they typed and *how* they typed it).

- **Periodic dynamic** - Dynamic keystroke analysis authenticates a user on the basis of their typing during a logged in session. The captured session data is compared to an archived user profile to determine deviations. In a periodic configuration, the authentication judgement can be intermittent; either as part of a timed supervision, or, in response to a suspicious event or trigger. This method provides distinct advantages over the static approach. Firstly, it is not dependent on the entry of specific text, and is able to perform authentication on the basis of any input. Another factor is the availability of data; in static keystroke analysis, the range of digraphs and frequency of their occurrence is likely to be significantly limited compared with a dynamic approach. Even an inexperienced typist is likely to produce sufficient digraph pairs to allow an authentication judgement to be derived. This is an important factor as it is necessary to have a statistically significant volume of keystroke data in order to generate a user profile.

- **Continuous dynamic** - Continuous keystroke analysis extends the data capturing to the entire duration of the logged in session. The continuous nature of the user monitoring offers significantly more data upon which to base the authentication judgement. With this method it is possible that an impostor may be detected earlier in the session than under a periodically monitored implementation. On the downside, however, the additional processing required will add to the computational overhead of the supervision system.

- **Keyword-specific** - Keyword-specific keystroke analysis extends the continuous or periodic monitoring to consider the metrics related to specific keywords. This could be an extra measure incorporated into a monitoring system to detect potential misuse of sensitive commands. For example, under a DOS/Windows environment it may be appropriate to monitor the keystroke metrics of a user attempting to execute the FORMAT or DELETE commands. This could represent a significant enhancement, as a command with a high misuse consequence (e.g. DEL *.*) is unlikely to cause sufficient profile deviation when observed from a system-wide context, due to the limited selection of digraphs. By contrast, static analysis could be applied to specific keywords to obtain a higher confidence judgement.

- **Application-specific** - Application-specific keystroke analysis further extends the continuous or periodic monitoring. Using this technique, it may be possible to develop separate keystroke profiles for distinct applications. For example, a user may be profiled separately for their word processing application and their email client. The potential of this new technique is discussed in more detail in section 3.

In addition to a range of implementation scenarios, there are also a variety of possible keystroke metrics that can be profiled as the basis for subsequent comparison:

- **Digraph latency** - Digraph latency is the metric that has traditionally been used for previous studies, and typically measures the delay between the key-up and the subsequent key-down events, which are produced during normal typing (e.g. T-H). In most cases, some form of low and high pass filter is applied to remove extraneous data from the session data.

- **Trigraph latency** - Trigraph latency extends the previous metric to consider the timing for three successive keystrokes (e.g. T-H-E).

- **Keyword latency** - Keyword latencies consider the overall latency for a complete word or may consider the unique combinations of digraph/trigraphs in a word-specific context.

- **Mean error rate** - The mean error rate can be used to provide an indication of the competence of the user during normal typing. Whilst this may not be user specific, it may be possible to classify users into a generic category, according to their typing ability, which can then be used as an additional authentication method.

- **Mean typing rate** - A final metric is that of the mean typing rate. As with the mean error rate, individuals can be classified according to their typing ability and hence evaluated based on their average typing speed.

While the final two metrics indicated above are unlikely to provide a suitably fine-grained classification of users for direct authentication judgements, they may be used to provide a more generic set of user categories that can contribute to a combined measure.

It should be noted that all of the above techniques and metrics can be implemented on a standard PC platform, without the need for special hardware.

3. EXPERIMENTAL DYNAMIC KEYSTROKE ANALYSIS

The idea of using keyboard characteristics for authentication is not unique, and there have been a number of previous published studies in the area. To date, however, virtually all published studies have focussed upon static or context-independent dynamic analysis, using the inter-keystroke latency timing method. From the earliest studies in 1980 [9], the focus has been on the analysis of digraph latencies. Later studies [6, 8] further enhanced the work, identifying additional statistical analysis methods that provided more reliable results.

In [7], the concept of dynamic keystroke analysis was first proposed, with the introduction of a reference profile that could be used to monitor a live user session. Brown and Rogers [5] also explored the idea of dynamic analysis, presenting preliminary results.

A summary of some of the main results from studies to date is presented in *Table 1* below, which illustrates the effectiveness observed (in terms of false acceptance and false rejection errors), as well as the type of keystroke analysis technique employed (digraph/trigraph etc.) and the analysis approach taken (statistical/neural network etc.).

Table 1: Previous keystroke analysis studies

Authors	Method	%FAR	% FRR
Umphress & Williams (1985) [10]	Digraph Statistical	6%	12%
Legget & Williams (1988) [11]	Digraph Statistical	5%	5.5%
Joyce & Gupta (1990) [6]	Digraph Statistical	0.25%	16.67%
Bleha et al. (1990) [12]	Digraph Statistical	2.8%	8.1%
Legget et al. (1991) [7] ^{1}Static, ^{2}Dynamic	Digraph Statistical	5% ^{1} 12.8% ^{2}	5.5% ^{1} 11.1% ^{2}
Brown & Rogers (1993) [5] ^{1}Group 1, ^{2}Group 2	Digraph Combined Neural Network & Statistical	0%	4.2% ^{1} 11.5% ^{2}
Napier et al. (1995) [13]	Digraph Statistical	29.5% / 3.8%	
Mahar et al. (1995) [8]	Digraph Statistical	35% / 17.6%	
Furnell et al. (1996) [14] ^{1}Static, ^{2}Dynamic	Digraph Neural Network ^{1,} Statistical ^{2}	8% ^{1} 15% ^{2}	7% ^{1} 0% ^{2}

A further variation in the data analysis can be introduced through the consideration of application specific keystroke profiles. If we accept from previous work that individual users have a distinct typing pattern, it can be hypothesised that an individual's typing pattern may also vary depending upon the application in use. For example, a user participating in a chat session may type in a fairly relaxed style, while the same user may type in an significantly different way when producing a document. It should also be noted that certain categories of user might use the numeric keypad for large quantities of data entry. Under these circumstances the volume and diversity of the keystroke digraphs will vary tremendously when compared to the more usual alphanumeric typing encountered with most user profiles. Previous research has been carried out in this area [15], which has shown that analysis of numeric keystrokes can provide a viable authentication measure. This is an area receiving on-going attention through a separate research project at the authors' institution.

In [16] the authors described a trial in which keystroke data, obtained within Microsoft Windows NT, was evaluated across all applications. While the results from this trial were encouraging, the quantity of data collected was insufficient to make a true, statistically valid, conclusion. Instead it was determined that further trials were necessary. Following the first trial, the authors conducted a second round of monitoring in which eight test subjects were profiled. Over a period of 3 months, a total of 760,000 digraph samples were captured and stored for analysis. In this case, however, the analysis was conducted with a view to determining viability of application-specific keystroke profiling. To this end, it was necessary to identify a series of applications for profiling, with the selection criteria being those for which sufficient keystroke data had been logged during the sampling period. A review of the keystroke data revealed that the applications satisfying this requirement were Microsoft MSN Messenger, Internet Explorer, Word and PowerPoint. While the authors considered that a numerically intensive application such as Excel would have provided an interesting candidate, insufficient keystrokes were captured to enable the creation of a profile. Additionally, of the eight users sampled during the trial, only five produced sufficient data to analyse from all of the aforementioned applications. Although the resulting sample group was very small, it was sufficient to yield interesting results in relation to an initial assessment of application-specific profiling.

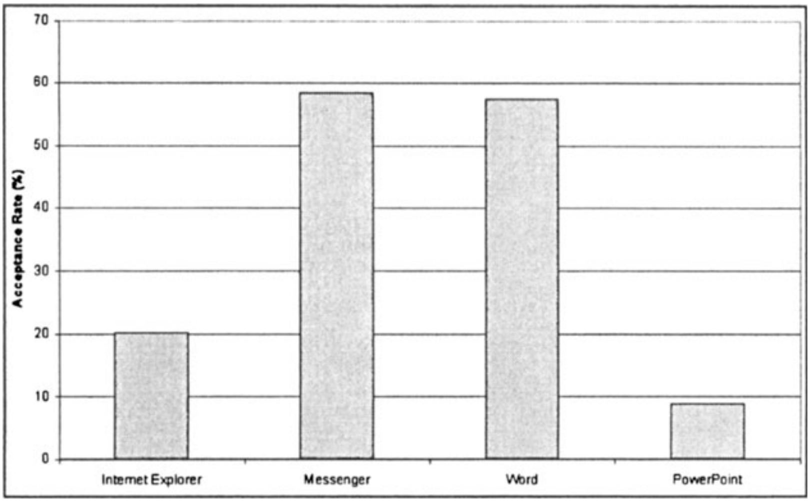

Figure 1: Acceptance Rate for application specific keystroke data compared against a system-wide context user profile

In *Figure 1* above, a single user's application-specific keystroke data is compared against the reference profile from the same user. The reference profile was based on all keystroke data acquired from all applications. Although the figure does not show distinct differences in all cases, there is a clear distinction between all applications apart from Messenger and Word. This can be explained when the nature of these applications is considered. Messenger and Word are both significantly textual in their usage, and users will typically type within Messenger and/or Word for considerable periods of time. In contrast, while Internet Explorer and PowerPoint sessions may both involve significant elements of keyboard activity, the typing is more likely to occur in sporadic bursts. As such, any dynamic that emerges is likely to be markedly different to that which would emerge in applications where more sustained typing is the norm. Considering the information portrayed above, the creation of application specific profiles would be likely to increase the acceptance rates observed.

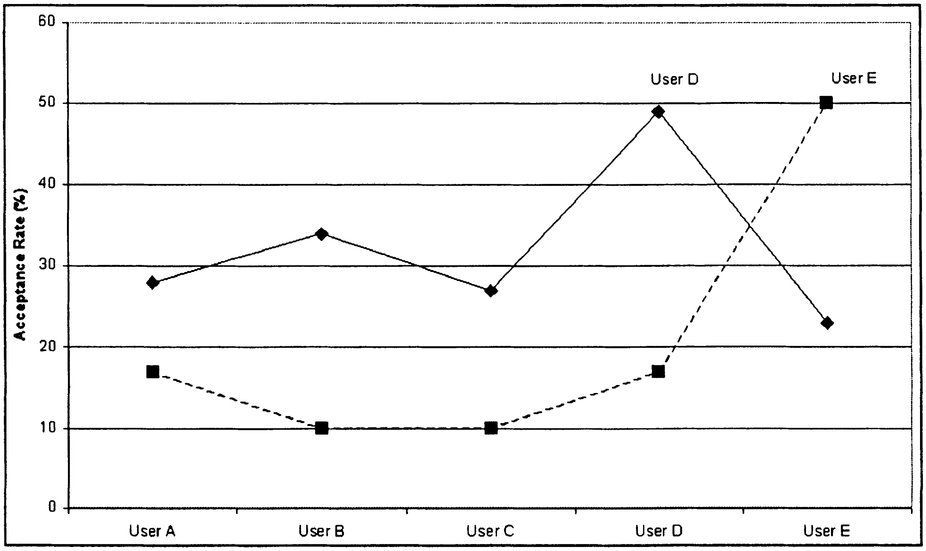

Figure 2: Acceptance Rate for two user profiles

In *Figure 2* above, a specific users' profile (users D and E when using Internet Explorer) is examined, showing there is a clear difference between other users' keystroke data (impostors) with appropriate peaks in acceptance rate for the valid users.

While the results shown do not indicate a suitably discriminative metric upon which to base a satisfactory authentication judgement, they do show a level of correlation between a user's typing pattern in an application-specific context. These preliminary results show that further work is needed to investigate the use of application-specific keystroke analysis.

4. AN ESCALATING RESPONSE FRAMEWORK USING KEYSTROKE ANALYSIS

The earlier discussion summarised the different potential implementations of keystroke analysis, and explained the operational differences between the approaches. It is possible to integrate these analysis approaches into an overall user authentication and supervision framework, with the varying techniques being invoked as responses to anomalies detected at earlier stages. A possible example of this is illustrated in *Figure 3*, which shows how the five variations discussed earlier can be incorporated within a four-level response framework. It should be noted that this is by no means the only method by which the techniques could be combined, and specific implementations could vary depending upon rule sets for a particular user, class of users, or general organisational security policy.

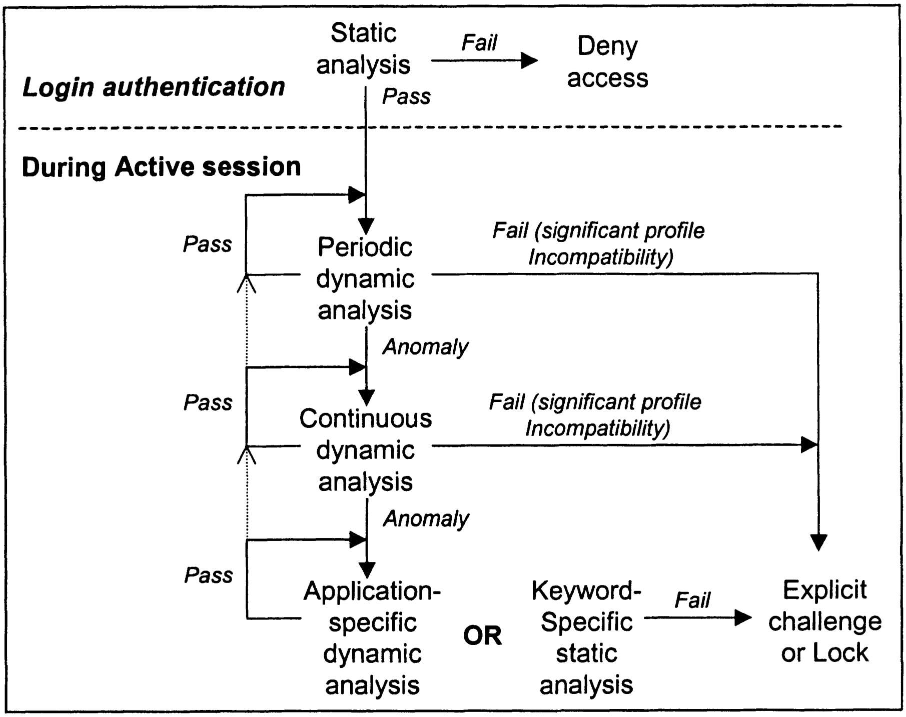
Login authentication
Static analysis
Fail
Deny access
Pass
During Active session
Pass
Periodic dynamic analysis
Fail (significant profile Incompatibility)
Anomaly
Pass
Continuous dynamic analysis
Fail (significant profile Incompatibility)
Anomaly
Pass
Application-specific dynamic analysis
OR
Keyword-Specific static analysis
Fail
Explicit challenge or Lock

Figure 3: Response framework using keystroke analysis

A suitable architecture for achieving such an approach is offered by the Intrusion Monitoring System (IMS) [17]. This proposes an architecture for real-time user authentication and misuse detection, based upon a monitoring *Host* that has the responsibility for supervising a number of *Client* systems (e.g. in the form of end user PCs or workstations). Key elements of the architecture, from the perspective discussed in this paper are the *collector* (which obtains the keystroke data from the individual client systems), the *anomaly detector* (which performs the actual keystroke analysis and profile comparison, maintaining a consequent alert status metric), and the *responder* (which is responsible for initiating the different keystroke analysis approaches in response to increases in the alert status and other contextual factors). Assuming such a monitoring context, the text below describes how the response process in *Figure 3* would proceed.

Initial authentication may occur using a standard username/password pair, but supplemented by the use of static keystroke analysis to assess how the information is entered. If the user fails to authenticate at this stage (e.g. after being permitted three attempts to enter the details), then the most appropriate response is to deny access (if the correct password is provided, but the keystroke analysis aspect fails, then an alternative option could be to allow the login to proceed, but to begin the session with a higher level of subsequent monitoring – e.g. continuous rather than periodic assessment). If this login authentication is successful, the user will proceed to a logged in session, during which dynamic keystroke analysis could be applied on a periodic basis (in order to minimise the associated processing overhead in the initial instance). Assuming no anomalies, this could simply continue throughout a logged in session. If a departure from the typing profile is noted during the monitoring period, however, there would be two options for response. If the keystroke data exhibits a significant incompatibility, then a high confidence of impostor action could be assumed and the responder could proceed directly to some form of explicit action (e.g. interrupting the user session by issuing a challenge or suspending their activity pending an administrator intervention). In cases where the profile incompatibility is not conclusive, the responder could initiate an increase in the monitoring resolution – firstly to invoke continuous dynamic analysis, and then beyond this to invoke either application or keyword-specific methods. The choice in the latter case would depend upon the context of the current user's activity. For example, if they were word-processing, then application-specific dynamic analysis would potentially give a more accurate assessment of identity. If, by contrast, they were operating at a command line level, then it could be considered more appropriate to invoke keyword-specific static analysis, looking for instances of particularly sensitive commands such as 'format' or 'erase'. Profile incompatibility at this final stage would automatically result in more explicit response action.

In cases where the responder agent has initiated a more detailed level (e.g. from periodic to continuous, or from continuous to application-specific), then the monitoring would continue at this level for a period of time, in order to ensure that profile incompatibilities were no longer observed. A suitable trigger (e.g. the entry of a certain number of further keystrokes without significant profile departure) would be used to reduce the alert status of the monitoring system, and thereby allow the responder agent to re-invoke a lesser level of analysis (this is indicated by the dotted arrow lines in the figure).

The combination of mechanisms in this manner allows a system to provide a standard, and hence acceptable, user login for the initial authentication, while also providing enhanced user supervision for the duration of the users' session. Such a system should, in theory, ensure transparent operation to legitimate users. It should also be noted that, in a practical context, keystroke analysis may not be the only technique involved, and other metrics relating to user activity and behaviour might also be considered by the *anomaly detector*, and thereby used to inform the *responder* agent.

5. CONCLUSIONS

This paper has considered the significant variety of implementation methods and metrics that can be associated with keystroke analysis. The new concept of application-specific analysis has been introduced, along with initial experimental findings that support the feasibility of the approach. The preliminary results suggest that the technique is worthy of further investigation.

The discussion has also considered the application of keystroke analysis as a response mechanism within an intrusion detection system. The combination of analysis techniques, placed within such an authentication/supervision framework has the potential to provide a significant improvement in system-wide security against impostor attacks, as well as ensuring transparency to legitimate end users.

6. REFERENCES

[1] Morrissey J.P.; Sanders P.W. & Stockel C.T. 1996. "Increased domain security through application of local security and monitoring"; Expert Systems; vol. 13; no. 4; pp296-305.

[2] Lunt T.F. 1990. "IDES: an intelligent system for detecting intruders"; Proceedings of the Symposium on Computer Security: Threat and Counter Measures"; Rome.

[3] Mukherjee B. & Heberlein L.T. 1994. "Network intrusion detection"; IEEE Networks; vol. 8; no. 3; pp26-45.

[4] Jobusch D.L. & Oldehoeft A.E. 1989. "A survey of password mechanisms: Weaknesses and potential improvements. Part 1"; Computers & Security; vol. 8; no. 7; pp587-603.

[5] Brown M. & Rogers S.J. 1993. "User identification via keystroke characteristics of typed names using neural networks"; International Journal of Man-Machine Studies; vol. 39; pp999-1014.

[6] Joyce R. & Gupta G. 1990. "Identity authentication based on keystroke latencies"; Communications of the ACM; vol. 33; no. 2; pp168-176.

[7] Legett J.; Williams G.; Usnick M. & Longnecker M. 1991. "Dynamic identity verification via keystroke characteristics"; International Journal of Man-machine Studies; vol. 35; pp859-870.

[8] Mahar D.; Napier R.; Wagner M.; Laverty W.; Henderson R.D. & Hiron M. 1995. "Optimizing digraph-latency based biometric typist verification systems: inter and intra typist differences in digraph latency distributions"; International Journal of Human-Computer Studies; vol. 43; pp579-592.

[9] Card S.K.; Moran T.P. & Newell A. 1980. "Computer text-editing: An information-processing analysis of a routine cognitive skill"; Cognitive Psychology; vol. 12; pp32-74.

[10] Umphress D. & Williams G. 1985. "Identity verification through keyboard characteristics"; International Journal of Man-Machine Studies; vol. 23; pp263-273.

[11] Legett J. & Williams G. 1988. "Verifying user identity via keystroke characteristics";International Journal of Man-Machine Studies; vol. 28; pp67-76.

[12] Bleha S.; Slivinsky C. & Hussein B. 1990. "Computer-access security systems using keystroke dynamics"; Actions on pattern analysis and machine intelligence; vol. 12; no. 12; pp1217-1222.

[13] Napier R.; Laverty W.; Mahar D.; Henderson R.; Hiron M. & Wagner M. 1995. "Keyboard user verification: towards an accurate, efficient, and ecologically valid algorithm"; International Journal of Human-Computer Studies; vol. 43; pp213-222.

[14] Furnell S.M.; Morrissey J.P.; Sanders P.W. & Stockel C.T. 1996. "Applications of keystroke analysis for improved login security and continous user authentication"; Proceedings of the 12th International Conference on Information Security (IFIP SEC '96), Island of Samos, Greece; 22-24 May, pp283-294.

[15] Ord T. & Furnell S.M. 2000. "User authentication for keypad-based devices using keystroke analysis"; Proceedings of the Second International Network Conference (INC 2000), Plymouth, UK, 3-6 July; pp263-272.

[16] Dowland P.S.; Singh H. & Furnell S.M. 2001. "A preliminary investigation of user authentication using continuous keystroke analysis"; Proceedings of the IFIP 8th Annual Working Conference on Information Security Management & Small Systems Security, Las Vegas; 27-28 September.

[17] Furnell S.M. & Dowland P.S. 2000. "A conceptual architecture for real-time intrusion monitoring"; Information Management & Computer Security; vol. 8; no. 2; pp65-74.