Multi-Factor Authentication Using Hardware Information and User Profiling Techniques

A. Alnajjar and H. Janicke

Software Technology Research Laboratory, Faculty of Technology, De Montfort University, Leicester, UK
e-mail: P08041453@myemail.dmu.ac.uk, heljanic@dmu.ac.uk

Abstract

This paper presents a multi-factor authentication approach that extends traditional username-password authentication with hardware and user behaviour profiling techniques. The aim of the approach is to improve the reliability of authentication by computing trust and confidence scores against user profiles. Based on the level of trust, the access control mechanisms may then choose to (un-)lock certain functions or even classify the access as an attack and redirect the user to a honey-pot to gather additional information about the attacker that can be used for a trace-back. The novelty of the approach is that it observes the correlation between users' behaviours and their hardware usage as implicit verification procedures to discriminate the usage of the user-name and password entry.

Keywords

Authentication, Profiling, Multi Factor Authentication, Keystroke Recognition.

1 Introduction

In this paper, we present a simple password mechanism that is augmented with additional profiling techniques to create a form of multi-factor authentication. Using password keys in authentication alone is not reliable due to the user’s inability to keep them confidential; in addition passwords are often prone to dictionary or rainbow-table attacks as well as the ease with which social engineering techniques can obtain passwords. To address some of these issues our approach integrates with the traditional password authentication by using Hardware Manufacture Serial Part Numbers (*HMSPNs*) to consider the user environment. This approach can be easily integrated in existing password based authentication schemes. Additional factors that are considered in the authentication process are the users' behaviour in providing the user-name and password and the user-profile in using a variety of hardware. Both factors do not require the user to memorise or otherwise keep additional secret information.

Three widely accepted authentication principles base the identification of a user on a) something the user has, b) something the user knows or c) something the user is or does. Multi-factor Authentication Mechanisms employ various techniques, often drawing on several of the above principles to establish a user's identity. For example the credit card payment system (Kumar et al. 2008) with biometric authentication proposes to employ fingerprint verification with a credit card in a multi factor authentication scheme, combining principles a) the card and c) the fingerprint. However, such an approach would require the installation of additional equipment, thus increasing the cost. The use of additional devices such as fingerprint readers typically also adds to the time taken for authentication which affects the user acceptability for the system. Given that fingerprints can be spoofed with relative ease (Ihmaidi et al. 2006) the overall gain in security is questionable. Indeed most current approaches to multi-factor authentication (Naji et al. 2011, Trevathan et al. 2009) are typically expensive and difficult to deploy and directly affect the usability of the system, as they prolong the authentication process. The approach presented in this paper avoids the impact of the additional authentication procedures on usability and does not require extra devices to be deployed to end-users. The key novelty of the presented approach is that it integrates profiling information with established user-name/password authentication and can be used to discriminate valid use of password credentials against misuse by an attacker, without complicating the authentication process or incurring large extra costs.

This paper is organized as follows. Section 2 reviews related work of authentication techniques, HMSPNs usage in access control and tracking approaches. Section 3 illustrates our authentication approach and the main system activity. Next, the paper provides a sample analysis scenario using our approach to profile hardware and user activity. After that, the paper provides our system architecture and implements a prototype to show a case study. Finally, the paper evaluates the initial results of our technique and presents the conclusion of the paper including achievements and future work.

2 Related works

Naji et al. (2011) enhance the security of an access control system using handwritten signature. Their system employs the static and dynamic features of the signature to make a decision about the identity of the signature through a combination of matching statistical models to analyse them. Handwritten signature processing and extracting their features is time consuming and requires dedicated hardware at the user-end.

Card readers are an additional level of hardware security is using one-time password (*OTP*). The chip on the client "user" card generates the *OTP*, with the caveat that the account is rendered inaccessible if the card is lost or stolen. This additional challenge-response mechanism over a separate channel removes the need for security questions to confirm transactions and helps preventing fraud. However, this mechanism requires additional accessories and increases deployment cost (Ravi et al. 2004). With the ubiquity of mobile phones, sending *SMS* text or voice messages that include one-time password (*OTP*) is in effect extending the card-reader approach. Here the mobile phone is considered a secure channel, albeit with the increasing connectivity of smart-phones this cannot be considered as independent as the original card-reader. Whilst this approach reduces the cost in deploying readers it adds additional costs on the extra communication channel and requires these channels to be accessible to the user (Zomai & Jsang 2010).

Hardware has been used to facilitate authentication for a long time. The idea is that users register devices (e.g. based on their MAC address) so that the devices are authenticated rather than their users. Examples of devices are storage media drivers such as hard disc drives HDDs. Each storage media has a unique HMSPN as an identifier product that can be used in profiling (Patowary 2009). This HMSPNs are already actively used for identification, albeit they can be modified at a firm-ware level and thus are susceptible to spoofing, e.g. Microsoft products send product and hardware identifiers during the activation process (Microsoft Corporation 2010). This hardware information provides the opportunity to profile the users' computing environment.

Based on the hypothesis that different people type in uniquely different typing measure, there are many basic methods (Shanmugapriya & Padmavathi 2009, Attila M 2007, Bergadano et al. 2002, Clarke & Furnell 2007, Yu & Cho 2004, Lee & Cho 2007) used to analyse keystroke typing.

Keystroke dynamics can be used as behavioural biometrics for users. It is an analysing technique for users typing behaviour when keyboard input is monitored (Obaidat & Sadoun 1999). However, if keystroke is not combined with particular keystrokes keys such as the password, it is insufficient to be an objective authentication factor (The et al. 2010). The keystroke approach is mostly characterised by the error rates in these following precision cases based on False Acceptance Rates (FAR), False Rejection Rates (FRR) and Equal Error Rates (EER) (Monrose & Rubin 2000).

Statistical (Bergadano et al. 2002) and *neural network* (Gunetti & Picardi 2005) techniques are the main two analysing keystroke approaches. Additionally, there are some combinations of both approaches (Monrose et al. 1999, Clarke & Furnell 2007). Statistical approaches compare a reference set of typing characteristic of specific user with test set of typing characteristic of the same user. Neural Networks use historical data that come from first usage, and then uses this data model to expect the result of new test or classify a new observation (Yu & Cho 2004, Lee & Cho 2007).

Some drawbacks have been exposed by other research (Lv & Wang 2006) that inhibits keystroke from real word applications. One research experiment provided the possibility of using modified keyboards that were based on a pressure sensor to recognize users keystroke (Lv & Wang 2006). This method requires specific keyboards that thus adding again additional cost to the user. To reduce the environment factor that may affect user behaviour in keystroke, Maxion & Killourhy (2010) explored a number pad input using a single finger. They tried to discriminate users typing style, FAR and FRR scope suggests a low level of surety that authentication using keystroke biometrics might be possible in this particular environment.

3 Our Approach

Our approach combines hardware identification with key-stroke biometrics, yielding a multi-factor authentication approach in which user biometrics can be correlated with the hard-ware that is used during the login process. The analysis of user-typing patterns on particular hardware by monitoring the keyboard inputs can visualize the significant pattern difference between the users. This correlation is reducing the FAR and FRR rates and allows the approach to be deployed throughout heterogeneous systems which are comprised of various hardware interfaces.

The key contribution of our approach is to improve the login-procedure by determining the level of trust of the user without additional cost or making the deployment of the solution overly complex. Thus, the key objective of our approach is developing a novel technique for the analysis of HMSPNs properties and patterns that are captured in the computational model. After that, an approach is developed for modelling the dynamic behaviour of the user. Then, user profiles based on analyzing and modelling users' behaviour to develop a new technique for the analysis of Internet services based on these profiles is formulated.

Hardware parts have a particular history in *HMSPNs* usage. Some computer hardware parts have not changed and have been used by the manufacturer for a long time. Therefore, every computer device has a history tracking over the time of its *Life cycle*. Thus, each computer hardware part has a particular track of usage from manufacture phase to destruction. First, if a user has been dealing with a device for every log in procedure for access control applications for a long time, this user will be more familiar with this hardware and has a particular behaviour when using it. Therefore, the user has a particular pattern scope that will be used with this hardware. Consequently, if the number of users of a particular hardware is increased, our authentication approach has to recognize the way these users behave when using this hardware, even if they use the same user-name and password. Of course, the sharing of accounts is bad practice, but still commonly encountered in both domestic and corporate environments over which the service provider has little influence. For example in Figure 1 Bob and Colin used John's hardware, however they have different behaviours in dealing with same hardware. Consequently, our approach has to find the different attribution of users’ behaviour when they use the same hardware and the same user-name and password. Ultimately, our authentication technique maps user environment hardware in order to demonstrate the user behaviour in previous pattern usage in particular hardware.

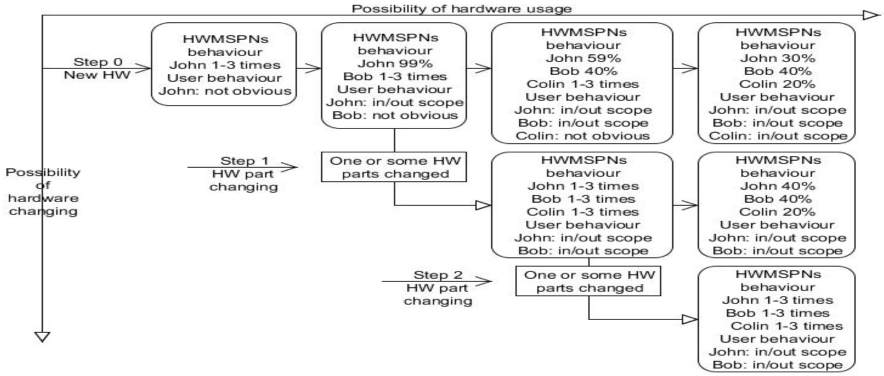


Figure 1: Hardware and users behaviour *Life Cycle*

The hardware *life cycle* in Figure 1 explains conceptually the hardware usage that supports the learning of user behaviour depending on a particular hardware configuration. However, the hardware parts may change over the time, resulting at configuration that is distinct to previous login attempts by their users. One example is the use of a tablet. E.g. the login may be typed on the touch screen or (after attaching the tablet to a docking station) through a physical keyboard. These changes in hardware configurations affect user profiling. \Step 1" and \Step 2" in Figure 1 reflect changing the hardware parts which change user environment. Therefore, the system has to recognise hardware changing and compare user's hardware at every login.

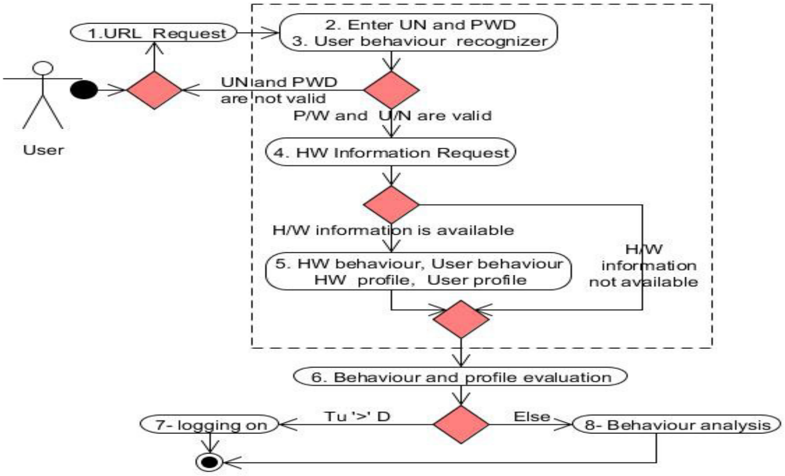

1.URL Request
2. Enter UN and PWD
3. User behaviour recognizer
User
UN and PWD are not valid
P/W and U/N are valid
4. HW Information Request
H/W information is available
5. HW behaviour, User behaviour
HW profile, User profile
H/W information not available
6. Behaviour and profile evaluation
Tu '>' D
Else
7- logging on
8- Behaviour analysis

Figure 2: System Overview

3.1 System Overview

Our authentication system uses two components in the login procedure. Whilst the user u is typing his/her user-name and password, our first component captures the current user behaviour (b_u) by calculating the keystroke (both key-press and release) speed when username and password are typed. The second component collects the HMSPNs, which make up the user's current hardware configuration (c_u). As the user or other security software installed on the client machine can prevent the gathering of hardware information, we consider this to be optional information.

However, if this information is not provided it has a detrimental effect on the accuracy of our mechanism, as the hardware profiling information is coupled with the selection of the user-profile for keystroke recognition. If the user provides access to the hardware profile, the system begins to analyse and compare the current hardware configuration (c_u) with the established profile of that user (_c_u) to determine their similarity. If the user has used the current hardware before, the system computes the similarity between the current keystroke behaviour of the user (b_u) and the behaviour that has been recorded against this hard-ware configuration previously (_$b_{u;cu}$). If the current hardware configuration is not known, the component will try to match bu against all known keystroke behaviours for that user bu; indiscriminate of the hardware configuration, which obviously reduces the effectiveness of this mechanism.

Given that the username and password checks are successfully passed, the system will compute out of the similarity between the hardware configuration and their profiles, and the associated keystroke behaviour similarity to their profiles two levels of trust. If only keystroke information is available, only one level of trust is being used in the following.

Given that usernames and passwords are not very secure, the hardware similarity test reflects the idea that hardware that has been previously used by the same user increases the likelihood of the user being genuine, as this rules out attacks in which passwords have been observed by shoulder surfing or rainbow table attacks. In addition, uncharacteristic use of hardware, e.g. the use of a company PC that has regularly been used during office-hours for 6 month and from which now an access is taking place at 2am in the night, is flagged up by a low trust-level in the hardware.

Similarly the key-stroke behaviour is evaluated, linked against the used hardware configuration (c_u) if available. The system will authenticate normally if the username and password are correct and a threshold in both levels of trust is passed. If the user-name and password do not match, the authentication is considered failed. If the username and password are correct, and only a low level of trust is established based on the hardware or keystroke behaviour the system can be configured to adapt to the level of trust. E.g. the authentication can be failed; the user can be authenticated with reduced privileges such as only being able to view his account details; the system can increase the threshold for an intrusion detection system that identifies fraudulent activity based on the transactions that are undertaken or even redirect the user to a honey pot trapping system to explore if the user is a hacker using a spoofed username and password. In an e-banking context, this could e.g. mean to delay the transactions and attempt to contact the user via a different channel such as email or phone. Figure 2 shows the basic steps in the system operation.

3.2 System Activities

Our technique depends on the matching of the current hardware configuration cu against the users' previous hardware behaviour $\_c_u$ and the associated user behaviour b_u against the previous user behaviour $\_\mathrm{b}_\mathrm{u}$ as part of the login procedure.

On the client side, the login prompt performs three data-collection functions. Firstly the username and password is collected in the traditional way. Secondly the keystroke behaviour of the user is gathered during the typing of the username and password. Functions like auto completion and provision for copy & paste are turn off, as they would effectively disable the recognition of the keystroke behaviour. Thirdly the login prompt will attempt to collect the hardware configuration from the user's operating system. This may require the user to white list the login software or the server address from which the login prompt is loaded.

On the server side the authentication module will first check the username and password hash against the stored credentials. If this is successful, the additional two components hardware recogniser and keystroke recogniser are invoked to further qualify the login request, thus providing additional scrutiny.

3.2.1 Hardware Recogniser

The hardware trust is computed by the hardware recogniser, which matches the current configuration against previously used hardware configurations for the same user based on the parts' serial numbers. This process takes into account the previous usage patterns of the user over time and also considers other aspects such as concurrent usage of the same hardware configuration or hardware parts in different login processes, which e.g. could indicate a spoofing attack. Essentially there are three key results that can are generated by this component:

1. Trust level based on usage of hardware configuration
2. Known configuration for use in behaviour recognition (or matching configuration)
3. Cross login analysis for attack detection.

The trust level is computed against the history of previous login-attempts and their associated hardware configurations _cu which is essentially drawn from the sequence of previous successful login attempts by this user.

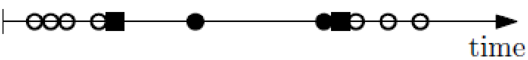


Figure 3: Hardware history

Figure 3 shows a simplified example. Every node on the timeline represents a successful login by the user in question. The used hardware configuration is depicted by the shape of the node, e.g. the empty circle could be the user's office machine, the square a mobile device, the filled circle a user's home computer. The first step is that the hardware is checked whether it has been used before, ie. it is known to the system, which is important for the keystroke recogniser in subsequent checks. This establishes a baseline trust for the access in case the hardware is known.

Secondly the access is viewed in the context of the other accesses (left neighbours), the time and the day of the access. We chose metrics based on time of day and day in week as these constitute the majority of repetitions we have encountered. We currently do not support more complex analysis of these events in our prototype, but envision the use of neural networks or support vector machines to establish a behaviour baseline against which the check can be performed. Based on the “fit" of the hardware configuration used in the login the trust level is adjusted.

Thirdly, the hardware recogniser maintains a cache of recent and current login activities over the entire user-base. If there is a current login from the same hardware configuration or configurations that share particular hardware components there is a chance that one of the logins is fraudulent and based on spoofed hardware information. It is known that some hardware manufacturers fail to provide unique serial numbers for their components. For the known cases we have a black-list of manufacturer ids which are excluded from this analysis step. A collision here reduces the trust level established by the hardware recogniser.

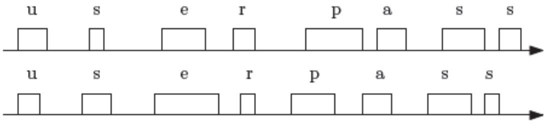
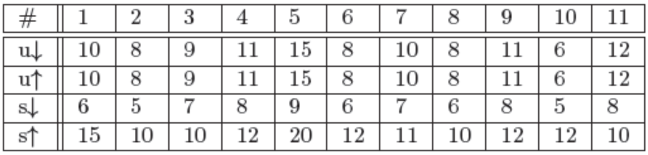


Figure 4: Keystroke patterns



#	1	2	3	4	5	6	7	8	9	10	11
u↓	10	8	9	11	15	8	10	8	11	6	12
u↑	10	8	9	11	15	8	10	8	11	6	12
s↓	6	5	7	8	9	6	7	6	8	5	8
s↑	15	10	10	12	20	12	11	10	12	12	10

Table 1: Keystroke profile _$b_{u;cu}$ against hardware configuration c_u

3.2.2 Keystroke Recogniser

The keystroke recogniser takes the current keystroke pattern entered by the user (b_u) and matches it against the previous recorded keystroke behaviour of that user using that hardware (_$b_{u;cu}$).

The keystroke pattern is characterised by the press and release times of the keys that are used in entering the username and password and is gathered on the client side. Figure 4 gives an example of such a pattern.

Our current prototype only considers the press and release times as a proof of concept and does not use other correlations between subsequent keypress events that may be further improving the accuracy. As the contribution of this paper is not a novel keystroke recognition scheme, but the integration of multiple approaches this mechanism can be replaced with more sophisticated techniques such as specific keystroke recognition (Shanmugapriya & Padmavathi 2009).

We currently build a trust-metrics based on whether the current keystroke pattern fits the users profile information, where the profile is created based on the previous user inputs. For example with respect to Figure 4 the first keyevent is the time the letter \u" is pressed. Previous logins e.g. recorded the times in Table 1 which forms the user profile, depicted in Figure 5. Currently the system looks at the variance of the data and the percentile into which the current keystroke pattern falls with respect to each of the keypress and release events and computes an accumulated trust level over all events contained in the keystroke pattern. In comparison to e.g. specific keystroke recognition (Obaidat & Sadoun 1999) this is a very simple approach which we plan to refine in the future.

3.3 System analysis

Our technique depends on the matching of the current hardware configuration cu against the user’s previous hardware behaviour *_cu* and the associated user behaviour bu against the previous user behaviour *_bu* as part of the login procedure. On the client side, the login prompt performs three data-collection functions. Firstly the user-name and password is collected in the traditional way. Secondly the keystroke behaviour of the user is gathered during the typing of the user-name and password. Functions like auto completion and provision for copy & paste are turn off, as they would effectively disable the recognition of the keystroke behaviour. Thirdly the login prompt will attempt to collect the hardware configuration from the user's operating system. This may require the user to white list the login software or the server address from which the login prompt is loaded.

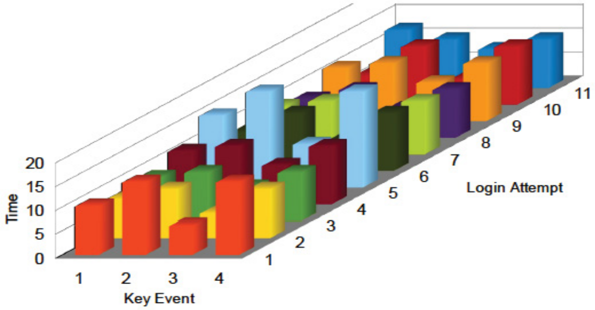


Figure 5: Keystroke Profile

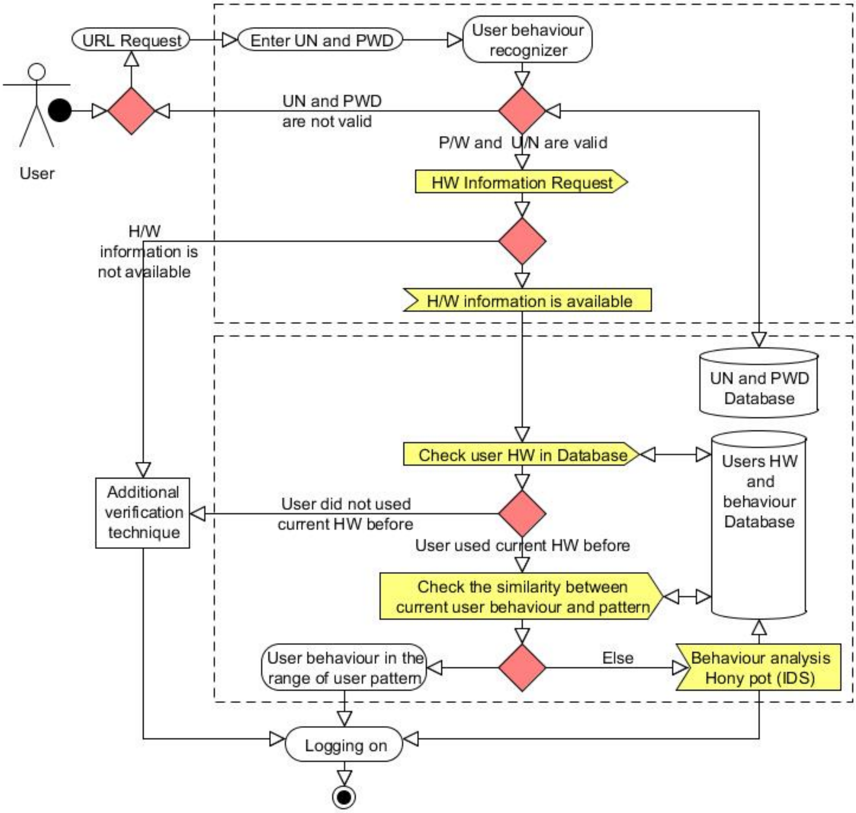

URL Request
Enter UN and PWD
User behaviour recognizer
User
UN and PWD are not valid
P/W and U/N are valid
HW Information Request
H/W information is not available
H/W information is available
UN and PWD Database
Check user HW in Database
Users HW and behaviour Database
Additional verification technique
User did not used current HW before
User used current HW before
Check the similarity between current user behaviour and pattern
User behaviour in the range of user pattern
Else
Behaviour analysis Hony pot (IDS)
Logging on

Figure 6: Flow chart

4 Case Study

We developed a simple Java application to apply our approach in the login process as an implicit login procedure. Every log in, our system captures user behaviour using a keystroke function to calculate users typing speed and response time among the keys of the user-name and password. The user-name and password contains characters and number. Then, when the user typed his/her valid user-name and password the system collects three parts of *HMSPNs*. These parts are the BIOS device number, MAC address number and the hard disk drive number. After that, the system recognizes if the user used current hardware before and if and to what extend the hardware was used by other users. Figure 7 shows the percentage of hardware usage and user pattern stamp by determining how the current user behaviour is related to previous usage patterns. In this case study, system improves the ability of observe the levels of trust to reflect the different bu when the user uses different hardware. In this scenario, the user performed 200 succeeded log in using username and password as key to log. However, the user used two devices representing two different hardware environments.

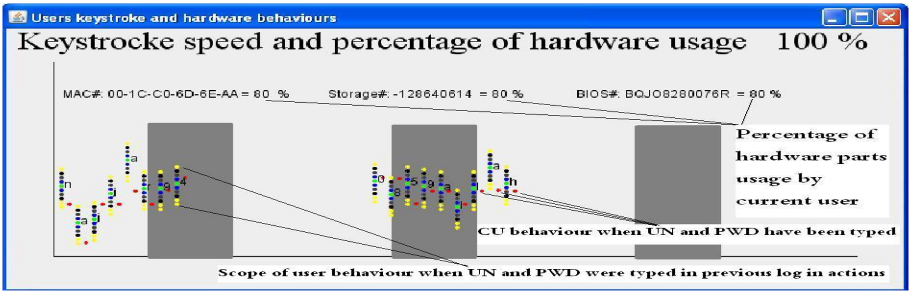

Users keystroke and hardware behaviours
Keystrocke speed and percentage of hardware usage 100 %
MAC#: 00-1C-C0-6D-6E-AA = 80 %
Storage#: -128640614 = 80 %
BIOS#: BQJO8280076R = 80 %
Percentage of hardware parts usage by current user
CU behaviour when UN and PWD have been typed
Scope of user behaviour when UN and PWD were typed in previous log in actions

Figure 7: ***HMSPNs*** **usage** c_u **and profile** $\_b_{u;cu}$ **against keystroke pattern** b_u**.**

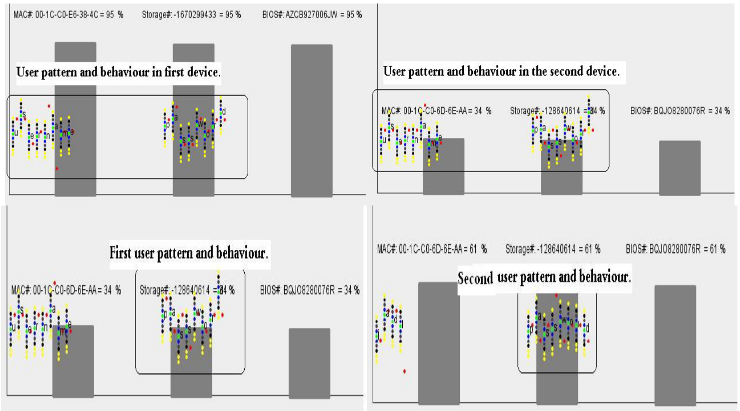

MAC#: 00-1C-C0-E6-38-4C = 95 %
Storage#: -1670299433 = 95 %
BIOS#: AZCB927006JW = 95 %
User pattern and behaviour in first device.
User pattern and behaviour in the second device.
MAC#: 00-1C-C0-6D-6E-AA = 34 %
Storage#: -128640614 = 34 %
BIOS#: BQJO8280076R = 34 %
First user pattern and behaviour.
MAC#: 00-1C-C0-6D-6E-AA = 34 %
Storage#: -128640614 = 34 %
BIOS#: BQJO8280076R = 34 %
MAC#: 00-1C-C0-6D-6E-AA = 61 %
Storage#: -128640614 = 61 %
BIOS#: BQJO8280076R = 61 %
Second user pattern and behaviour.

Figure 8: LEFT top: ***HMSPNs*** **usage cu and profile** $\_b_{u;cu}$ **against one user uses two hardware. RIGHT bottom: HMSPNs usage cu and profile** $\_b_{u;cu}$ **against two users use same login information (password) in one hardware**

In the second scenario, two users used same hardware and a shared password for 100 successful logs in attempts. The system recognised the effect of the hardware in user keystroke behaviour. In addition, the system compared between the users depending on their familiarity with the hardware. This recognition comes from the hardware trust.

4.1 Trust

For all login attempts that provided the correct username and hardware we computed the hardware trust based on the hardware configuration that was used in the login attempt against the previously encountered hardware. We computed the trust-level based on precedences, ie if the hardware was encountered previously we assigned a baseline trust of 40% for previously encountered hardware. Based on whether there was a precedent of that hardware being used on that day in the week, within that hour of the day and after the use of the previously used hardware configuration, we added additional 20% as these occurrences increased our confidence. If the hardware configuration (or part thereof) was used concurrently in another login process we subtracted 60% from the trust-level.

For all three hardware configuration that were used in the case-study, we recorded 100 keystroke patterns to build up the profile. The trust was computed by calculating the deviation from the mean for each key-event (key- press and release) of the profile against the standard deviation as a percentage value. The overall keystroke trust was then computed as the mean of the individual percentage values.

We overall set relatively low thresholds for both trust levels, and proceeded with the authentication when both trust levels exceeded 70%. If only one of the trust-levels exceeded the threshold, an additional verification question was asked from the user. If this was answered correctly the authentication was considered successful. If both trust levels fell below the threshold, the login attempt was considered unsuccessful and the user was returned to the login prompt. We considered a maximum of three unsuccessful login attempts before the account was blocked.

The recorded profile information was only updated after a successful login attempt. This means that even if behaviour or hardware usage changed over time the system was able to adapt, in most cases via the provision of an additional security question. We did not yet integrate actual honey-pots into our system or linked it to the access control system.

5 Conclusion & Future Work

The availability of hardware information can enhance authentication mechanisms. The work presented in this paper shows that by capturing a wide range of statistics it is possible to perform an analysis of hardware and user behaviour. In this paper we considered keystroke as a biometrics. By combining password based authentication with hardware profiling and keystroke recognition we provided a multi-factor authentication scheme that does not require additional devices to be deployed and adds little cost to the deployment of the authentication system.

The paper reviewed related work on authentication approaches and their limitation as a motivation for this approach. We then presented our approach and showed how the additional data can be collected on the client side and what data needs to be collected. We then described in detail the server-side and the functioning of the hardware-recogniser and the keystroke recogniser and how their interaction improves the accuracy of keystroke recognition as a more specific profile can be maintained depending on the hardware that is used.

We implemented our prototype system using basic profiling techniques for the analysis and presented a trust- model that takes into account the hardware usage and the user behaviour when entering his/her username and password. The prototype is of course a proof of concept that shows that the techniques can be combined and that their combination yields a positive influence on the accuracy of the detection. In the future we will refine the individual techniques and adopt e.g. keystroke recognition approaches that have been presented in (Obaidat & Sadoun 1999). We provided a java-based prototype implementation of our authentication system and presented a small case-study as a proof of concept for our work.

In the future we will refine the profiling techniques used in our authentication framework and are looking at implementing techniques based on neural networks or support vector machines. We also investigate the use of the profile information in attack attribution, as the hardware profiles can provide indication about (fraudulent) users.

In addition, we will look at geo-spatial information and its integration in the hardware recogniser. The idea is that successive logins from different geographical areas are not plausible and can indicate fraudulent activity. In this line of investigation we will also actively deploy honey-pots to further identify behavioural traits of the user. This information can then be used two folds: a) to provide additional attribution information about the attacker; b) to retrospectively authorise the actions performed if the user is deemed to be genuine.

6 References

Attila M, Zoltn B, L. C. (2007), `Strengthening passwords by keystroke dynamics', IEEE . www.knt.vein.hu

Bergadano, F., Gunetti, D. & Picardi, C. (2002), `User authentication through keystroke dynamics.' ACM Trans. Inf. Syst. Secur. 5(4), 367-397. http://dblp.unitrier.de/db/journal-s/tissec/tissec5.html/BergadanoGP02

Clarke, N. L. & Furnell, S. (2007), `Authenticating mobile phone users using keystroke analysis.', Int. J. Inf. Sec. 6(1), 1-14. http://dblp.unitrier.de/db/journals/ijisec/ijisec6.-html/ClarkeF07

Gunetti, D. & Picardi, C. (2005), `Keystroke analysis of free text.' ACM Trans. Inf. Syst. Secur. 8(3), 312-347. http://dblp.unitrier.de/db/journals/tissec/tissec8.html/GunettiP05

Ihmaidi, H.-D., Al-Jaber, A. & Hudaib, A. (2006), Securing online shopping using biometric personal authentication and steganography', in `Information and Communication Technologies, 2006. ICTTA '06. 2nd', Vol. 1, pp. 233-238.

Kumar, D., Ryu, Y. & Kwon, D. (2008), A survey on biometric fingerprints: The cordless payment system, in `Biometrics and Security Technologies, 2008. ISBAST 2008. International Symposium on, pp. 1-6.

Lee, H. & Cho, S. (2007), ‘Retraining a keystroke dynamics-based authenticator with impostor patterns’. Computers and Security 26(4), 300-310. http://dblp.uni-trier.de/db/journals/compsec/compsec26.html/LeeC07

Lv, H.-R. & Wang, W.-Y. (2006), `Biologic verification based on pressure sensor keyboards and classifier fusion techniques', Consumer Electronics, IEEE Transactions on 52(3), 1057 - 1063.

Maxion, R. & Killourhy, K. (2010), `Keystroke biometrics with number-pad input, in `Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on', pp. 201-210.

Microsoft Corporation (2010), `Microsoft office activation/registration privacy statement'. http://office.microsoft.com/en-us/help/HP010069531033.aspx

Monrose, F., Reiter, M. K. & Wetzel, S. (1999), `Password hardening based on keystroke dynamics`, in J. Motiwalla & G. Tsudik, eds, `ACM Conference on Computer and Communications Security', ACM, pp. 73-82. http://dblp.uni-trier.de/db/conf/ccs/ccs1999.html/MonroseRW99

Monrose, F. & Rubin, A. D. (2000), `Keystroke dynamics as a biometric for authentication`, Future Gener. Comput. Syst. 16, 351-359. http://dl.acm.org/citation.cfm?id=338350.338359

Naji, A. W., Housain, A. S., Zaidan, B. B., Zaidan, A. A. & Hameed, S. A. (2011), `Security improvement of credit card online purchasing system', Scientific Research and Essays 6(16), 3357-3370.

Obaidat, M. S. & Sadoun, B. (1999), `Keystroke dynamics based authentication, in In &quot; Biometrics`. Personal Identification in Networked Society&quot;. A.Jain, R.Bolle, S.Pankanti (Eds', Kluwer Academic Publishers, pp. 213-229.

Patowary, K. (2009), `How to interpret hard disk model numbers'. http://www.instantfundas.com/2009/02/howto-interpret-hard-disk-model.html

Ravi, S., Kocher, P. C., Lee, R. B., McGraw, G. & Raghunathan, A. (2004), `Security as a new dimension in embedded system design`, in S. Malik, L. Fix & A. B. Kahng, eds, `DAC', ACM, pp. 753-760. http://dblp.uni-trier.de/db/conf/dac/dac2004.html/RaviKLMR04

Shanmugapriya, D. & Padmavathi, G. (2009), `A survey of biometric keystroke dynamics: Approaches, security and challenges', CoRR abs/0910.0817. http://dblp.uni-trier.de/db/journals/corr/corr0910.htmlabs-0910-0817

Teh, P. S., Teoh, A. B. J., Tee, C. & Ong, T. S. (2010), `Keystroke dynamics in password authentication enhancement', Expert Syst. Appl. 37, 8618-8627. http://dx.doi.org/10.1016/j.eswa.2010.06.097

Trevathan, J., McCabe, A. & Read, W. (2009), `Online payments using handwritten signature verification`. In S. Lati_, ed., `ITNG', IEEE Computer Society, pp. 901-907. http://dblp.uni-trier.de/db/conf/itng/itng2009.html/TrevathanMR09

Yu, E. & Cho, S. (2004), `Keystroke dynamics identity verification - its problems and practical solutions.' Computers and Security 23(5), 428-440.http://dblp.uni trier.de/db/journals/compsec/compsec23.html/YuC04

Zomai, M. A. & Jsang, A. (2010), `The mobile phone as a multi otp device using trusted computing`, in Y. Xiang, P. Samarati, J. Hu, W. Zhou & A.-R. Sadeghi, eds, `NSS', IEEE Computer Society, pp. 75-82. http://dblp.uni-trier.de/db/conf/nss/nss2010.html/ZomaiJ10