WEB SECURITY ENHANCEMENT BASED ON KEYSTROKE DYNAMICS

Michał Choraś
Institute of Telecommunications, University of Technology and Life Sciences, Kaliskiego 7, Bydgoszcz, Poland

Piotr Mroczkowski
Hewlett Packard Polska, Global Delivery Poland Center, Szturmowa 2a, University Business Center, Warsaw, Poland

Keywords: Web Security, Keystroke Dynamics, Password Hardening.

Abstract: Many online access systems (e.g. e-banking) require stronger protection than the login-id password pair can provide. Other, more sophisticated techniques of identity verification are in demand: one-time passwords, smart cards or biometric technologies. Among several biometric approaches the web-based solution that incorporates keystroke dynamics is the most relevant due to the low cost of the implementation, satisfactory results as well as the degree of transparency it offers.

1 INTRODUCTION

Rapid growth of e-bank systems and their popularity within information society as well as the dangers in the web motivate efforts to create more secure online services. Other more sophisticated techniques than login-id password pair are in demand. Among several web-security approaches identification based on keystroke dynamics is the most relevant due to the low cost of the implementation, satisfactory results as well as the degree of transparency it offers. Moreover, following the success of other biometrics methods of identification, keystroke dynamics seems to be a new emerging web technique for web-security enhancement.

Keystroke dynamics biometric systems analyze the way a user types at a terminal by monitoring the keyboard events, and thus is considered as the behavioral approach. Identification is based on the rhythm of typing patterns, which is considered to be a good sign of identity (Monrose and Rubin, 2000). In other words not what is typed, but how it is typed is important. In this approach several things can be analyzed: time between key-pressed and key-released events, break between two different keystrokes, duration for digraphs and trigraphs and many more.
Keystroke verification techniques can be divided into two categories: static and continuous. Static verification approaches analyze keyboard dynamics only at specific times, for example during the logon process. Static techniques are considered as providing a higher level of security than a simple password-based verification system (Monrose and Rubin, 2000). The main drawback of such an approach is the lack of continuous monitoring, which could detect a substitution of the user after the initial verification. Nevertheless, the combination of the static approach with password authentication was proposed in several papers [e.g. (Leggett et al., 1991)] and it is considered as being able to provide a sufficient level of security for the majority of applications. Our web identification system is based on such a combination. Continuous verification, on the contrary, monitors the user's typing behavior through the whole period of interaction (Monrose and Rubin, 2000). It means that even after a successful login, the typing patterns of a person are constantly analyzed and when they do not mach user's profile access is blocked. This method is obviously more reliable but, on the other hand, the verification algorithms as well as the implementation process itself are much more complex.

Choraś M. and Mroczkowski P. (2007).
WEB SECURITY ENHANCEMENT BASED ON KEYSTROKE DYNAMICS.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Internet Technology*, pages 337-340
DOI: 10.5220/0001264903370340

2 KEYSTROKE DYNAMICS CHARACTERISTICS

In the proposed and implemented verification system three independent methods of the identity verification are performed every time a user attempts to log in. First and second method is based on the calculation of the degree of disorder of digraphs and trigraphs respectively. The last one compares typing paths stored in the database against a typing path created at the time of logon process.

1. Digraphs and trigraphs
   Digraph is defined as two keys typed one after the other. In our case the duration of a digraph is measured between the press event of the first key and release event of the second key.
   Trigraph is defined as three keys typed one after the other. The duration of trigraph is measured between pressing event of the first key and release of the third key.
2. Degree of disorder Having two sets of key latencies of the same $Login - Password$ pair, it is possible to measure their "similarity". One way to calculate that is the degree of disorder (do) technique (Bergadano et al., 2002).
   Let us define vector V of N elements and vector V', which includes the same N elements, but ordered in a different way. The degree of disorder in vector V can be defined as the sum of the distances between the position of each element in V with respect to its counterpart vector V'. If all the elements in both vectors are in the same position, the disorder equals 0.
   Maximum disorder occurs when elements in vector V are in the reverse order to the model vector V'. Maximum disorder (do_{max}) is given by $do_{\max} = \frac{|V|^2}{2}$ where $|V|$ is the length of V and it is even, or by $do_{\max} = \frac{(|V|^2-1)}{2}$ where $|V|$ is length of V and it is odd.
   In order to get the normalized degree of disorder (do_{nor}) of a vector of N elements, we divide do by the value of the maximum disorder. After normalization, the degree of disorder falls between 0 (V and V' have the same order) and 1 (V is in reverse order to V').
3. Typing paths Typing paths can be described as a set of key code/key event pairs stored in order of occurrence. If some short sequence of chars is being retyped by a user several times (which is the case with the "Login - Password" mode), the analysis of such paths is likely to show some typical characteristics of a user's behavior:
   - moments where keys overlap (second key is pressed before the release of the first one)
   - the position of the key pressed in the case of duplicate keys (digits, SHIFT's, etc.)

3 WEB APPLICATIONS - PASSWORD HARDENING

Password hardening using keyboard statistics can be described as login-password pair combined with the collected typing features during the logon process. A system that implements the password hardening not only performs the password verification but also checks the typing patterns. The main advantage of this approach is the significant increase of the security. On the other hand the false rejection rate (FRR) can increase, especially for users that are not very familiar with typing on a keyboard. Our application is a remote system of identity verification that is based on password hardening using typing patterns. It is implemented using J2EE platform (JSP, Servlets) and MySql database. The logon module is encapsulated in applet where both login-id password pair as well as keyboard statistics are collected and send to server for comparison.

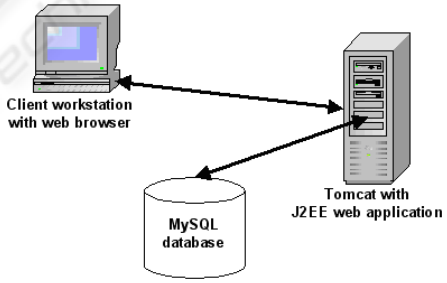

Figure 1: High level architecture of our password hardening system.

Server performs statistical analysis of typing samples, generates on the fly the typing path and the vectors of digraphs and trigraphs. Then the newly generated items are compared to their model counterparts stored in database. The model vector of digraphs and trigraphs automatically adapts to gradual changes in a user's typing patterns. High level architecture of our password hardening system designed for increasing web applications security is presented in Figure 1.
Additional use of keystroke analysis could be encouraged in many other applications and situations. Some of them are presented below:

- Identity Verification - keyboard statistics could be introduced into any verification system right after

the user's login-password pair typing stabilizes.

- Strong Authentication - root password, safety-critical systems and resources.
- Forgotten Passwords - our algorithms could be used in forgotten password recovery.

4 EXPERIMENTAL SETUP AND RESULTS

In our experiments 18 volunteers participated in testing the system. Typing skills varied slightly among them - the majority of the group type on PC keyboard every day. Every volunteer had assigned unique login-id and password. The full name of particular individual was used as her/his login-id, since it is one of the most frequently typed phrase for most of people. Collecting keyboard statistics session for single participant lasted from 20 to 40 minutes.

In our experiments we calculated False Rejection Rate (FRR) and False Acceptance Rate (FAR) for each of the users. We set the systems for different thresholds: 0.25, 0.3, 0.35 and 0.4. In the first stage every participant performed 15 attempts of log in-password authentication that were evaluated by the system in order to calculate the model vector of digraphs and trigraphs as well as to collect the typing paths. After that users performed several another logon attempts as valid users (FRR tests) and few attempts as impostors (**trying to log on somebody's else account knowing login and password** - FAR tests).

Table 1: FAR results for digraphs and trigraphs for the 0.25 threshold.

user	Digraph FAR	Trigraph FAR
user1	0.0000	15.3846
user2	0.0000	0.0000
user6	0.0000	17.5439
user8	0.0000	0.0000
user9	0.0000	12.5000
user10	0.0000	1.9231
user14	1.2346	28.3951
user15	0.0000	9.0909
user17	0.0000	0.0000
user18	0.0000	0.0000

In the experiments a participant was asked to act as impostor. She/he was trying to logon on somebody else account. In order to increase the number of logon attacks per single account, we randomly selected 10 out of 18 existing accounts to be attacked. This decision was motivated by the fact that the number of participants (and thus samples) was limited (users were not willing to spend hours trying to hack somebody's else account). Bigger number of attacks per single account will picture more clearly the FAR, so smaller number of accounts to hack was the only reasonable solution. The results showing FAR for each of the threshold for digraph and trigraph method are shown in the Tables 1-4. The results for typing path method and for all the methods combined together are shown in the Table 5.

Table 2: FAR results for digraphs and trigraphs for the 0.3 threshold.

user	Digraph FAR	Trigraph FAR
user1	1.9231	34.6154
user2	0.0000	15.3846
user6	0.0000	47.3684
user8	0.0000	1.6949
user9	0.0000	50.0000
user10	0.0000	7.6923
user14	9.8765	38.2716
user15	0.0000	45.4545
user17	0.0000	0.0000
user18	9.0909	18.1818

Table 3: FAR results for digraphs and trigraphs for the 0.35 threshold.

user	Digraph FAR	Trigraph FAR
user1	5..7962	48.0769
user2	7.6923	61.5385
user6	7.0175	66.6667
user8	5.0847	3.3898
user9	12.5000	68.7500
user10	9.6154	19.2308
user14	24.6914	59.2593
user15	0.0000	54.5455
user17	0.0000	0.0000
user18	27.2727	45.4545

5 DISCUSSION OF THE RESULTS AND CONCLUSION

In the article we presented our keystroke feature (characteristics) extraction methods. We also proved that biometrics based on keystroke dynamics is capable of increasing security in web applications such as

Table 4: FAR results for digraphs and trigraphs for the 0.4 threshold.

user	Digraph FAR	Trigraph FAR
user1	19.2308	50.0000
user2	46.1538	69.2308
user6	12.2807	71.9298
user8	15.2542	11.8644
user9	18.7500	81.2500
user10	26.9231	36.5385
user14	33.3333	67.9012
user15	0.0000	63.6364
user17	0.0000	10.0000
user18	54.5455	63.6364

Table 5: FAR results for typing paths and combined methods.

user	Typing Path FAR	Combined FAR
user1	0.0000	**1.9230**
user2	7.6923	**0.0000**
user6	0.0000	**0.0000**
user8	3.3898	**0.0000**
user9	0.0000	**0.0000**
user10	1.9231	**0.0000**
user14	0.0000	**8.1649**
user15	9.0909	**0.0000**
user17	0.0000	**0.0000**
user18	0.0000	**0.0000**

password hardening in e-banking (the combined values of FAR **were equal to %0 for all but 2 users** (Table 5).This means that the presented methods are effective and could be implemented to increase web security in applications where logging-in is the necessity for the clients.

It is hard to determine which of the developed and implemented method gives the best performance for all users. The best solution is to make the logon algorithm adaptive. The algorithm should check which method gives the best performance for given user in order to give it the biggest weight while taking the access/no access decision. In case of non-adaptive implementation the best results were observed for thresholds: 0,25 for trigraphs and 0,3 for digraphs. The threshold for digraphs and trigraphs should not be equal. It should be higher for digraphs and lower for trigraphs. It is also noticeable that longer char sets (trigraphs) have more stable statistics for a legitimate user (the standard deviation of particular trigraph's durations is small, and thus the distance calculated from the degree of disorder is smaller), but on the other hand they are easier to forge.

Keystroke dynamics are sensitive to the emotional and physical state of the person who is verified. Very poor typing skills are another factor which can affect the process of authentication. The good thing is that this method is very likely to achieve a high level of acceptance among ordinary users. Moreover, unlike other biometric systems which usually require additional hardware and thus are expensive to implement, biometrics based on keystroke dynamics is almost for free - the only hardware required is the keyboard (Monrose and Rubin, 2000).

REFERENCES

F. Monrose, A. Rubin, "Keystroke Dynamics as a Biometric for Authentication", Future Generation Computer Systems, vol. 16 , no. 4, 351 - 359, 2000.

G. Leggett, J. Williams, M. Usnick, "Dynamic Identity Verification via Keystroke Characteristics", International Journal of Man-Machine Studies, vol. 35 , no. 6, 859 - 870, 1991.

R. Gaines, W. Lisowski, S. Press, N. Shapiro, "Authentication by Keystroke Timing: some preliminary results", Rand Report R-256-NSF. Rand Corporation, 1980.

F. Monrose, A. Rubin, "Authentication via Keystroke Dynamics", Conference on Computer and Communications Security , 48-56, 1997.

R. Joyce, G. Gupta, "User authorization based on keystroke latencies", Communications of ACM, vol. 33, no. 2, 168-176, 1990.

S. Bleha, C. Slivinsky, B. Hussein, "Computer-access security systems using keystroke dynamics", IEEE Trans. on Patt. Anal. Mach. Int, vol. 12, no. 12, 1217–1222, 1990.

M. Brown, S. J. Rogers, "User identification via keystroke characteristics of typed names using neural networks", International Journal of Man-Machine Studies, no. 39, 999-1014, 1993.

F. Bergadano, D. Gunetti, C. Picardi, "User Authentication through Keystroke Dynamics", ACM Transactions on Information and System Security, vol.5, no. 4, 367 - 397, 2002.

M. Brown, S. J. Rogers, "Method and apparatus for verification of a computer user's identification, based on keystroke characteristics", Patent Number 5,557,686, U.S. Patent and Trademark Office, Washington, D.C., 1996.

E. Yu, S. Cho, "Biometrics-based Password Identity Verification: Some Practical Issues and Solutions," XVth Triennial Congress of the International Ergonomics Association (IEA), Aug 24-29, Seoul, Korea, 2003.

P. Mroczkowski, M. Chorań, "Keystroke Dynamics in Biometrics Client-Server Password Hardening System", Proc. of Advanced Computer Systems (ACS), vol. II, 75-82, Miedzyzdroje, Poland, October 2006.