Computer Security in the Age of Information
W.J. Caelli (Editor)
Elsevier Science Publishers B.V. (North-Holland)

Experience of Using a Type Signature Password System for User Authentication in a Heavily Used Computing Environment

Mike Newberry
Jennifer Seberry
Department of Computer Science
University College, University of NSW

ABSTRACT

This paper describes a user authentication system based around the user's type signature, a statistical measure of the user's typing style. It was tested on two heavily loaded computers.

Traditionally, users have been authenticated by asking them to provide some form of password. This password has been stored securely in the computer and used to check the identity of the user at various times, such as when the he or she first logs on. However such authentication only really proves that the challenged user knows the password — it doesn't identify the user. This has often been a security problem in time-shared computer installations, when unauthorised users (the proverbial hackers) have obtained the passwords of valid users and used these to penetrate the site's security.

This has lead to much work to identify users uniquely by more secure means, such as fingerprints. Such measures all try to identify a user by checking some attribute of the person. The idea of a *type-signature* is one such approach.

In 1985, Terry Jones first proposed the idea of a type-signature [Jones1985]. The type-signature was a statistical measure of the typing style of the user. It was calculated by measuring the time that elapsed between keystrokes. This type-signature, and the usual password, were then checked whenever the user logged on. It was intended that this would provide a greater level of security than just a password, as this system would reject an intruder whose typing pattern was incorrect.

1. The System

In December 1986 we commenced testing the concept of the type-signature[1]. Over the next two months several mini password systems, based around this concept, were developed and installed by Mike Newberry on various machines to test them under different conditions.

1.1 Implementation

The following major functions were identified for a user authentication system:

- For users:
  - — to successfully authenticate valid users;
  - — to enable valid users to change their passwords;
  - — to allow the type-signature to gradually change, to match the user's changing typing pattern (for instance, as a user became more familiar with a password their

1. This was funded by a grant from ATERB.

typing would improve).

- For system staff:
  - — to redefine a user's password and type-signature;
  - — to receive regular performance summaries;
  - — to alter system parameters to make it easier or harder for users to log-on (if, for instance, valid users are having trouble logging-on).

1.2 User Authentication

The following sequence of operations were performed to authenticate a user:

1. The user would supply a user-name. If this was not a valid name, the process would halt;
2. The user would type a password. The computer would encrypt and record this password and the associated type-signature;
3. The offered password (now encrypted) would be compared with the user's actual encrypted password. If this test failed, the process would halt;
4. The offered type-signature would be compared with the user's recorded (encrypted) type-signature. This involved testing whether the interval between keystrokes was in the range

$$\left[average - (\delta * sd),\ average + (\delta * sd) \right]$$

where *average* and *sd* were the recorded average and standard deviation between that pair of keystrokes, and δ was a system parameter. Even a good typist would not always be in this range, so this test could fail one time for every ε characters in the password, and the user still be allowed onto the system. Thus a longer password made it easier for a valid user to log-on, but harder for an intruder;
5. Finally, if all the above tests had been passed, the user's recorded, and encrypted, type-signature was modified to reflect the way the user had typed the the password this time, and the user was allowed onto the system.

This system was found to work acceptably in practice. More sophisticated statistical approaches, such as applying a χ^2 test to the type-signature, were too precise and made logging-on too difficult for valid users.

1.3 Joint Accounts

Often computer accounts are shared (such as super-user accounts, or joint project accounts). For these accounts special provision had to be made in the type-signature system, as each user of the account would have a different typing style. In this system, each joint account had one password, but there was a separate type-signature for each valid user. The owner of the account would log-on normally (that is, by specifying the account name, and correctly typing the password). However, other people sharing the account would log-on using a *composite name* of the form:

<account name>:<user name>

The password for the shared account could only be changed by the account owner; when this happened, each of the composite users of the account would be allowed to log-on once more using the old password. When they logged-on they would be told that the password had been changed, and to see the account holder to have a new type-signature recorded for the new password. For example, if Jill shared her account with Jack, Jill would log-on using the user name "jill", while Jack would log-on using "jill:jack". In this case, only Jill could change the password.

2. Results

The mini system was run immediately after a user logged-on. At first users were allowed to choose their own password, and these were kept secret. Initially these users could log-on easily, however after a weekend (an absence of several days) they reported that they were consistently failing. This occurred because in the preceeding week they had become familiar with their password, their typing speed had increased, and the type-signature had been adjusted accordingly; but on Monday, when they tried to log-on their typing speed had reverted to its original level and the system rejected them. Consequently the δ and ε parameters were loosened. A better solution might have been to consider the amount of time since the account was last used when authenticating the user. Alternatively,the system might loosen or tighten the parameters depending on the time of day, or the day of the week.

During this stage one user complained she could not log-on. She had developed a mild case of RSI[2] and her typing style had changed. The system detected this change, and refused to authenticate her. The system should behave similarly for any event that altered a type-signature (such as intoxication). This behaviour was anticipated in [Jones1985].

The system also refused to authenticate users when they typed with one hand, as expected. This in turn suggests that such a system could not be implemented without the users' knowledge.

For the next stage, all passwords were changed to a standard string of the form:

mynameis<user name>

to test how secure the system was when passwords became known. It was not possible to uniquely identify users from their type-signatures. Moreover, there wasn't any clear correlation between the consistency of a person's typing and their account's security. Neither did the speed of the typing appear to make much difference, although very fast typists were harder to imitate. The critical factor appeared to be the rhythm of the typing. Most people either typed the password as a continuous stream of characters (especially fast typists), or paused between words. In both these cases the rhythm could be deduced. The most secure accounts had type-signatures that paused irregularly. Some users realised this, and introduced artificial rhythms to their typing, to make an intruder's task harder. This worked well while they remembered the rhythm.

---

2. Repetition Strain Injury

Through both stages the parameter mix $\delta = 1.4$, $\varepsilon = 4$ allowed the genuine user to log-on 89% of the time, while preventing intruders (as determined by the experiments performed above) for 43% of the time. The prevention figure would be larger in a real situation where users could choose their own passwords. It should be noted that users in this experiment were experienced with computer, and so they had some typing skills — inexperienced users may have trouble logging onto this system.

Optimum parameters would vary highly from site to site — these parameters worked well in our case.

2.1 Buffering

Critical to this system was the availability of accurate statistics about the type-signature. This information was distorted by any buffering of keystrokes that occurred between the terminal and the cpu. For instance, the type-signature would be distorted if a terminal buffered all keystrokes internally, to send them in packets to the cpu. This was a major problem in this experiment. On one machine the system was tested on, machine keystrokes were buffered at up to three different places before the type-signature was measured.

This effect could be eliminated by measuring the type-signature at the terminal itself, or at the first buffering point (for instance, at the first I/O board). These kind of changes would be easier if intelligent terminals were being used, but could require changes to the operating system otherwise.

2.2 CPU load

Another factor that distorted the measurement of the type-signature was the number of times the authentication program was swapped from the cpu during this process. If this occurred frequently, the type-signature could be distorted beyond recognition. This was a problem on both machines the system was tested on - whenever the computer became heavily loaded, the ability of the system to recognise users deteriorated.

One solution to this would be to perform the user authentication at a very high priority, or to disable interrupts while type-signature calculations were being performed.

2.3 Efficiency

This type of password system was noticeably slower than a conventional system, because of the extra file I/O and calculation required. Moreover it also required more storage space especially for the type-signatures.

3. Conclusions

This system added significantly to the security of the users in the experiments described above; even when passwords were public knowledge, intruders had only a 50/50 chance of successful entry. Thus, in any implementation, password integrity would still provide the main level of defence, with type-signatures as a back-up defence.

3.1 Suitability

This system works best when the type-signature can be measured directly (without any intermediate buffering), and the authentication routines can be run without interruption.

Users would require some minimum typing skills to make the system work. This all suggests it should only be used on machines in high security areas, or for selected accounts (such as those with super-user privileges).

3.2 Should we tell the users?

One issue that arose in the trials was whether the users should know that their typing patterns were significant. Some users, because they knew they had to type in a particular manner, consciously tried to type that way, and failed. Others, who succeeded in establishing a particular rhythm in their password, later forgot this rhythm and were unable to log-on at all. Certainly the system is more secure if users aren't aware of the use of type-signatures but it seems difficult to keep this hidden.

3.3 Other Applications

This style of user authentication has applications other than in computer security. One suggested application has been in the area of academic examinations, by computer. In this application, students in remote areas studying by correspondence, could do their examinations locally by computer, rather than travelling long distances to be examined at regional centres. Such a system would only work if the examiner was sure the person doing the exam was the student, and not a friend who did the course last year — for this authentication type-signatures could be useful.

4. Summary

In any area where it is critical that users be identified correctly, the type-signature provides greater security than just a password, although it is not foolproof. However it is slower, and requires more storage space than conventional password systems. Moreover, type-signature gathering may not be practical in some circumstances because of cpu scheduling and buffering requirements.

References

Jones1985. Terry Jones, *Secrecy and Authentication*, Honours Thesis, University of Sydney (1985).

Newberry1987. Mike Newberry, *Footprints in the Snow*, Honours Thesis, University of Sydney (1987).