Collection and Publication of a Fixed Text Keystroke Dynamics Dataset

Luciano Bello[1], Maximiliano Bertacchini[1], Carlos Benitez[1], Juan Carlos Pizzoni[1] and Marcelo Cipriano[2]

[1] Si6 Labs - CITEFA - Inst. de Investigaciones Científicas y Técnicas para la Defensa
{lbello,cbenitez,mbertacchini,jpizzoni}@citefa.gov.ar
[2] Escuela Superior Técnica - Facultad de Ingeniería del Ejercito

Abstract. Keystroke Dynamics is a powerful technique which allows to detect and identify intruders in computer systems. In order to test keystroke data pattern matching and clustering algorithms, user data collection is a mandatory task. Si6 Labs[3] developed a web application named **k-profiler**[4] with the purpose of collecting the typing rhythm data of volunteer users. This paper describes the experiment design criteria as well as the format of the collected data which will be used for Si6 projects and will be publicly available.

1 Introduction and Previous Work

During the last 30 years many works have been published about computer user identification and/or individualisation based on Keystroke Dynamics techniques. These methods are based on measuring the latency between succesive keystrokes in a computer keyboard. The most common application of these techniques is the reinforcement of user authentication in computer systems based on the user keystroke pattern [1,2,3]. In the last years, these method was also translated to smartphones [4,5]. Later just a few papers have been published on intruder identification based on Keystroke Dynamics [6].

All of the abovemetioned papers use their own collected user data which make it difficult to compare results of different algorithms because of the lack of a common dataset. It can be pointed out that there is a related research area which is intruder or masquerader identification in UNIX systems based on user command line behavior [7,8,9,10]. Three datasets of UNIX user command line data were published and are used by most works in the area. These datasets were collected by Samuel Greenberg[11] and Mathias Schonlau[12], and a synthetic one created by Ramkumar Chinchani et al.[13].

In recent years some keystroke datasets have been published, such as [14][5] and [15][6]. These datasets are based on fixed short text, such as a particular password,

[3] http://www.citefa.gov.ar/si6/
[4] http://www.citefa.gov.ar/si6/k-profiler
[5] http://www.cs.cmu.edu/~keystroke/
[6] http://jdadesign.net/2010/04/pressure-sensitive-keystroke-dynamics-dataset/

with focus in authentication. This approach is suitable for user authentication but not for user identification. Under these conditions, the natural user typing rhythms get lost because she is not familiar with the keyboard (layout, position or size). Moreover, she types under pressure just one fixed given word, causing a deformation of her keystroke pattern along the session. In the presented dataset, each volunteer types natural sentences on her own keyboard, watching her own screen without any pressure or external disturbing factors.

The goal of this work is an attempt to provide a standard dataset to be used in future works in Keystroke Dynamics research area, particularly in user identification. This dataset will save other researchers the complex task of collecting keystroke data from different users and will allow the comparison of different algorithms.

2 The Dataset collector

The keystroke collector was designed with flexibility in mind to cover as many use cases as possible. Both depressed and released key times were recorded as the user typed 15 Spanish sentences. Since much work has been perfomed in the past in relation with profiling users in UNIX command line environments[7,8,9,10], 15 UNIX commands are showed at the last page. Volunteers were asked to type these 16 paragraphs along with some enrollment data (see Section 2.5) which were used to anonymously label them.

Nowadays, this task is performed on a regular basis, so the dataset keeps growing. At the moment, more than 66 keystroke profiles have been collected.

2.1 The Web Application

A web-based keystroke collector was developed based on PHP and JavaScript. The code was deployed at `http://www.citefa.gov.ar/si6/k-profiler/`. A web-based approach was chosen in order to increase the potential dataset size, since it is inherently multiplatform and widely accessible, besides the fact that the volunteer types on her own keyboard. **k-profiler** is capable of capturing key depress or release times using the `onkeyup` and `onkeydown` Javascript events on the client-side (i.e. the web browser). This data is sent via POST HTTP method and stored in the server.

Once the user logs in, she is asked to fill in some basic enrollment data (see Section 2.5) and later she is prompted to type the paragraphs grouped in 6 pages.

The order in which paragraphs and commands are shown to the user is generated randomly, in order to prevent biased keystroke timings in the last texts due to tiredness. As a result, the user is rewarded with a 3D chart which shows the average time for each digraph.

2.2 Limitations

The JavaScript code is executed at the client-side and the keystroke event time precision is strongly dependent on the conditions of the platform where the client browser is working (i.e. process priority, RAM and CPU state, browser load, etc.). The timing accuracy has been measured in different computers, operating systems and browsers by sending a key at a fixed time interval. The measured error was at about 10% with a slow CPU use and a maximum of 20% with high CPU load. Figure 1 shows a histogram sample of measured delays. In this example, keyboard repetition rate was set at 100 ms, CPU load was about 10% and as result, the error was of 12 ms.

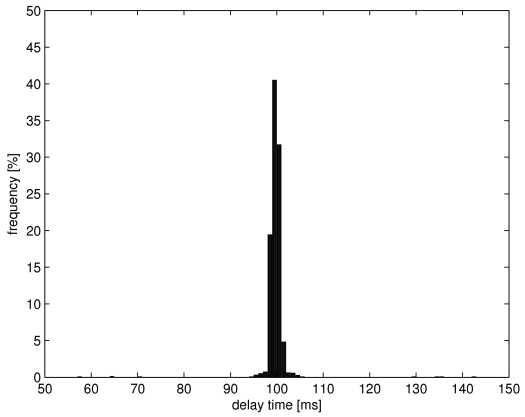

Fig. 1. Example of k-profiler time measure errors

Regarding this, the time resolution measured in JavaScript is on the order of tenths of miliseconds, while the values in experiments having special software for keystroke collection is on the order of 200 microseconds[14]. This is a system limitation, but taking into account the fact that digraph intervals usually range between 20 and 500 miliseconds, it is considered acceptable.

In browsers running on the GNU/Linux operating system, the `onkeydown` event for dead keys, used for acute accent (´) on Spanish vowels, can not be recorded. In those cases, an `onkeyup` event with keyCode `0` followed by the `onkeyup` of the vowel, without a `dn`, is detected and stored. Under these browsing conditions, when a modifier key (e.g. Shift or AltGr) is used, the event `onkeyup` is detected twice sometimes. Since the web application captures events as output of the `keyCode` and `charCode` methods, and these depend on the particular browser implementation[7], it is important to keep in mind the user's user-agent and the keyboard layout to detect particular keystrokes.

[7] http://mozilla.pettay.fi/moztests/events/browser-keyCodes.htm

2.3 Text Selection

In this dataset, a fixed but natural text approach was choosen. This text was carefully selected to hold some statistical properties (see Section 2.4) from the following pieces of literature in the public domain:

- *One Thousand and One Nights* or *Arabian Nights*
- *War and Peace*

The first one (whose translation in Spanish is *Las mil y una noches*), is a compilation of ancient arabian stories; and the second one (whose translation in Spanish is *Guerra y paz*), is the famous Russian novel by Leon Tolstoy. Plain text editions of both books were taken and their sentences were splitted using the `tokenize()` function from the NLTK Python library [16].

After unifying the capitalization and purging some special chararcters (e.g. all the simple and double quotes, dashes and hyphens), all the repeted sentences and those with less than 70 characters were discarded.

The total amount of digraphs in these sentences was counted and a ranking including the ten most popular ones was created. The same work was performed on words, resulting in two lists: top digraphs and top words.

Under the assumption that these are the most popular digraphs and words in the language, two rankings were produced, sorted by the percentage of the sentences covered by the popular digraphs or words. The intersection of the top 30 of these rankings is a set of 20 sentences, from which 15 were arbitrarily selected, excluding those that included unusual syntax forms. These sentences are listed below.

ks_00 en aquel momento estaba tan seguro de ello como si se encontrase a su lado al pie del altar.
ks_01 en todas partes se hablaba de la guerra y de que el enemigo estaba a las puertas de la ciudad.
ks_02 pedro se daba cuenta de que era el centro de la atención general y se sentía contento y cohibido.
ks_03 pero le era penoso que el estado de espíritu de las personas que tenía delante estuviera tan alejado del que nacía en ella.
ks_04 porque las paredes de la casa y las de la cuadra se han derrumbado encima de todo lo que había en la casa, sin excluir a los carneros, los gansos y las gallinas.
ks_05 se los veía en los patios y en las ventanas de las casas; otros se agrupaban en la calle.
ks_06 aprendí también la ciencia de los astros y las palabras de los poetas.
ks_07 y sentáronse los tres ante las bandejas de oro debidas a los cuidados del genio de la lámpara; y aladino estaba sentado en medio, con su esposa a la derecha y su madre a la izquierda.
ks_08 a causa del juego de luces entre las copas de los tilos, no podía darse cuenta del cambio de las caras.
ks_09 al darse cuenta de la presencia del príncipe, se detuvo perpleja en el umbral de la puerta.
ks_10 la condesa se dirigió a la sala de los iconos y sonia la halló arrodillada delante de las pocas cruces que todavía pendían de las paredes.
ks_11 no es que dijera aquello que pudiese complacerla, sino que juzgaba desde el punto de vista de ella todo lo que decía.
ks_12 una de ellas estaba junto a la cabeza del califa y la otra a sus pies.
ks_13 después de despedirme del rey y de todos los amigos que me hice durante mi estancia en aquella isla tan encantadora, me embarqué en la nave, que enseguida se dio a la vela.
ks_14 y efectivamente, me dio la tentación de deshacerme de aquel collar de oro y de perlas.

During the data collection, sentences are displayed to the volunteer in a random order and can be identified in the dataset by the prefix "`ks_`". In the last page, 15 UNIX commands are added, also randomly sorted. These commands are the 15 most frequent commands from a combination of 3 datasets ([11], [8] and a private one obtained from Si6 Labs honeypots. They can be identified inside the dataset with the "`cm_`" prefix. These commands are listed below.

cm_00 ls -a
cm_01 bash
cm_02 rm -rf /var/log/lastlog
cm_03 unset HISTSAVE
cm_04 cat /etc/passwd
cm_05 wget
cm_06 fg %2
cm_07 more Makefile
cm_08 lpq -Pip
cm_10 tar xvfz *.tgz
cm_11 mv a.out /var/tmp
cm_12 touch wtmp
cm_13 ps aux
cm_14 kill -9 0

2.4 Statistical Features of the Fixed Text

The ten most frequent digraphs in the dataset, sorted by its repetition rate in the selected text, are show in Table 1. Most digraphs include a white space. The top ten digraphs without white spaces and their frequency are: 'de' (57), 'la' (52), 'en' (38), 'as' (33), 'ue' (25), 'es' (24), 'el' (24), 'os' (23), 'ta' (22) and 'nt' (22).

The ten most popular words are listed in Table 2. With 333 words in the 15 sentences, the repetition rate is calculated as $100 * sum/333$, where column *sum* is the sum of ocurrences of each digraph in each sentence. These listed words comprise nearly 44% of the total words in the text.

2.5 Enrollment Data

The initial form of the web application asks the volunteer for an e-mail (or name or nickname), occupation, age, handedness and keyboard layout. Additionaly, the IP address and browser User-Agent are recorded. These two pieces of data together with the e-mail are used to detect user reentrance or whether the same person performed the experience twice. The IP address is discarded during the anonymization process (see Section 2.6) and the e-mails/names/nicknames from the same user are joined.

2.6 Anonimization and Publication

The anonymized dataset is published at `http://www.citefa.gov.ar/si6/k-profiler/dataset/`. The anonimization process removes sessions with less than 12 completed and accepted sentences. The finished ones are tagged as `finished`, and as `unfinished` otherwise. The occupation and field of each session are normalized and translated. The e-mail/name/nickname is replaced by a generic string with the prefix "`user_`". The date, age and user-agent is kept unchanged. Each session is saved in a file the following filename: `<UNIX-timestamp>_<user>.[un]finished`. Finally these files are archived and compressed as `kprofiler-<date>-<UTC time>.tar.gz`.

digraph	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	sum
'e '	4	5	5	7	6	3	2	5	5	4	6	8	1	10	6	77
'a '	2	6	5	4	8	2	2	9	4	5	9	4	9	7	1	77
' d'	2	3	3	4	4	1	2	5	6	5	5	5	2	6	6	59
'de'	2	3	2	4	5	1	2	6	5	4	6	5	2	5	5	57
'la'	1	4	1	3	6	3	3	5	2	2	8	3	3	5	3	52
' l'	1	3	1	2	9	5	4	7	4	2	5	1	2	3	1	50
's '	**0**	4	**0**	2	7	7	5	7	5	**0**	6	1	2	4	**0**	50
' e'	3	3	2	7	3	3	**0**	3	1	2	0	3	2	6	1	39
'en'	3	2	6	3	2	4	2	4	2	3	1	**0**	**0**	4	2	38
'o '	5	1	3	4	3	**0**	**0**	4	3	1	**0**	6	1	1	2	34

Table 1. Digraph repetition by sentence number

word	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	sum	repetition rate
'de'	1	3	2	2	3	1	2	2	3	2	3	2	1	2	4	33	9.91
'la'	0	2	1	0	3	1	1	3	0	2	3	0	2	2	1	21	6.31
'y'	0	1	2	0	2	1	1	3	0	0	1	0	1	1	2	15	4.50
'que'	0	1	1	3	1	0	0	0	0	0	1	4	0	2	0	13	3.90
'las'	0	1	1	3	1	0	0	0	0	0	1	4	0	2	0	13	3.90
'los'	0	1	1	3	1	0	0	0	0	0	1	4	0	2	0	11	3.30
'en'	1	1	0	1	1	3	0	1	0	1	0	0	0	2	0	11	3.30
'a'	1	1	1	2	2	1	0	3	2	0	2	1	2	1	0	11	3.30
'se'	1	1	2	0	1	2	0	0	0	1	1	0	0	1	0	10	3.00
'del'	1	0	0	1	0	0	0	1	2	1	0	0	1	1	0	8	2.40

Table 2. Word repetition by sentence number

3 The Published Dataset (kprofiler-20100716-1442)

The published file was relesed with the data collected up to July 16th, 2010. It includes 66 sessions (58 finished and 8 unfinished) from 63 unique volunteers, from whom 54 of them finished the whole process.

3.1 Data Format

A session file (see Figure 2 for an example) is named after the UNIX timestamp in which the user started and the unique volunteer id. The files are in UNIX format, ASCII encoded. The first line contains the following fields separated by semicolons: *Action*, *local UNIX timestamp*, *date and hour*, *user id*, *occupation*, *field*, *age*, *gender*, *handedness*, *keyboard layout*, *obfuscated IP* (normaly "na") and *user-agent*.

The *Action* can be `SESSION` or `RESUME`. In the first case, the user started a new session. If the user is resuming an existent session, `RESUME` will appear and the date will indicate when this happened. If the user paused the data collection in order to continue later, a banner saying `-----PAUSED----` and the date and hour are shown.

```
;SESSION;1269898779;2010-03-29 18:39:39;user_146;Other;Art;22;Female;right-handed;\
latam;na;Mozilla/5.0 (Windows; U; Windows NT 5.1; es-AR; rv:1.9.2.2) \
Gecko/20100316 Firefox/3.6.2 (.NET CLR 3.5.30729);text/html,application/xhtml+xml,\
application/xml;q=0.9,*/*;q=0.8;es-ar,es;q=0.8,en-us;q=0.5,en;q=0.3;gzip,deflate;\
ISO-8859-1,utf-8;q=0.7,*;q=0.7

ks_14 1269898815055 dn 85
ks_14 1269898815135 up 85
ks_14 1269898815462 dn 8
(...)
ks_13 1269898881950 up 65
ks_13 1269898881982 dn 190
ks_13 1269898882054 up 190

---------PAUSED---------
2010-03-29 18:41:35
---------PAUSED---------
;RESUME;1269904718;2010-03-29 20:18:38;user_146;Other;Art;22;Female;right-handed;(...)

ks_06 1269904803245 dn 65
ks_06 1269904803332 up 65
ks_06 1269904803452 dn 80
(...)
cm_13 1269905257644 up 85
cm_13 1269905257876 dn 88
cm_13 1269905257940 up 88

---------END---------
2010-03-29 20:27:37
---------END---------
```

Fig. 2. Session example

Keystroke data follows line by line in the format `[invalid-]<model id> <UNIX timestamp in miliseconds> <direction> <char code>`. The `invalid` tag appears when the detected amount of keystrokes is out of bounds (when they are more than 1.3 or less than 0.95 times the expected amount of keys). In those cases, the sentence or the command is showed again to the volunteer and she is asked to retype it. The invalid keystrokes are surrounded between the tokens `invalid-data-start` and `invalid-data-end`.

Instead of comparing the typed data against the original data, keystroke counts were used as aprove/reject parameter. So, if the user typed less than 90% or more than 130% of the total sentence characters, the sentence will be rejected and she will be asked to type it again. Mistakes were allowed so that data on typing errors can also be used for user identification [17].

The `<model id>` refers to the sentence or the command that the volunteer typed. The `Event` may be `dn` or `up`, depending whether the key was depressed or released respectively. The char code is the output of the following JavaScript command:

```
e.keyCode? e.keyCode : e.charCode;
```

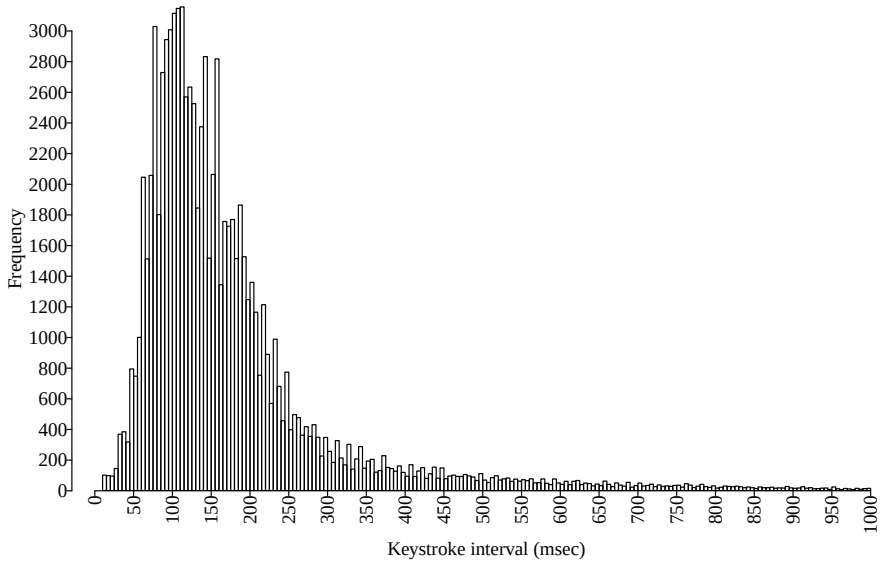

Fig. 3. Histogram of digraph times (up to 1 second)

where `e` is the keystroke event. This returns the output of `keyCode`[8] when this function is implemented for the event or `charCode` otherwise, which is the ASCII value of the resulting character. In browsers running on GNU/Linux, the `dn` of a key associated with a dead key (and the deadkey itself) is not detected and only the `up` is shown (see Seccion 2.2 for details).

A banner with the legend `-----END----` with the date and time flags is appended when the session ends.

3.2 Statistical Features

The dataset contains 282020 keystroke events (key presses and releases); 16372 of them are tagged as `invalid`. Figure 3 shows that most digraphs were typed in an interval below half a second. Most outliers can be filtered out ignoring those exceeding 500 miliseconds.

Figure 4 shows the average digraph times of 3 fixed groups of sentences for 4 particular users. Only the most frequent digraphs are included. Each user has clearly her own typing pattern, which is consistent throughout different sentences.

[8] http://lists.w3.org/Archives/Public/www-archive/2006Nov/att-0047/keyCode-ie.htm

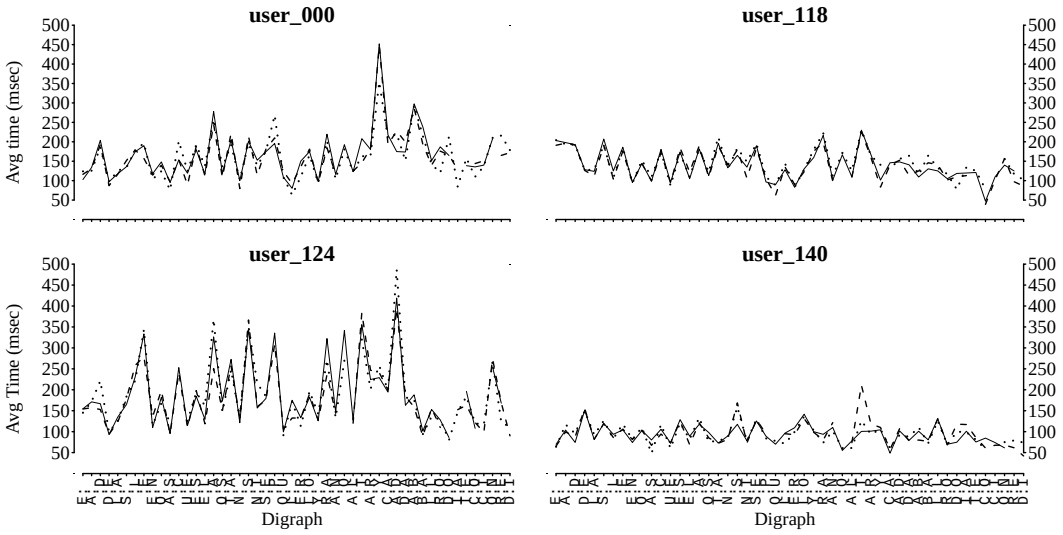

Fig. 4. Example of 4 keystroke paterns grouped in 3 sets of 5 sentences

4 Future Work

This dataset was collected to perform user identification experiments based on their keystroke pattern. In order to test the effectiveness of the identification/clustering algorithms, a fixed test dataset is the natural first dataset to use. Nevertheless, for identification purposes, free text needs to be used. Therefore, the next step of this work will be the collection of free text keystroke data from labeled users.

On the other hand, this dataset was designed with Spanish texts because most volunteers are Spanish speaking people. A next step could include the collection of the same type of keystroke data in another language.

References

1. Monrose, F., Rubin, A.: Authentication via keystroke dynamics. In: Proceedings of the Fourth ACM Conference on Computer and Communications Security, Zurich, Suiza (1997) 48–56
2. Joyce, R., Gupta, G.: Identity authentication based on keystroke latencies. Communications of the ACM **33**(2) (1990) 168–176
3. Jiang, C., Shieh, S., Liu, J.: Keystroke statistical learning model for web authentication. In: Proceedings of the 2nd ACM symposium on Information, computer and communications security, ACM New York, NY, USA (2007) 359–361
4. Clarke, N., Furnell, S., Lines, B., Reynolds, P.: Application of keystroke analysis to mobile text messaging. In: Proceedings of the 3rd Security Conference, Las Vegas, NV, 14-15 April. (2004)

5. Zahid, S., Shahzad, M., Khayam, S., Farooq, M.: Keystroke-based user identification on smart phones. In: 12th International Symposium on Recent Advances in Intrusion Detection - RAID, Sep. 2009, Saint-Malo, Francia (2009)
6. Zamonsky, G., Sznur, S.: Keystroke dynamics aplicado a la clasificación de intrusos. In: Workshop de Seguridad Informática - WSegI 2009, Ago. 2009, Mar del Plata, Argentina, SADIO (2009)
7. Maxion, R.: Masquerade detection using enriched command lines. In: International Conference on Dependable Systems and Networks. Volume 0., Los Alamitos, California, EEUU, IEEE Computer Society (2003) 5
8. Schonlau, M., DuMouchel, W., Ju, W., M. Theus, A., Vardi, Y.: Computer intrusion: Detecting masquerades. Statistical Science **16** (2001) 58–74
9. Bertacchini, M., Fierens, P.: Preliminary results on masquerader detection using compression based similarity metrics. Electronic Journal of SADIO **7**(1) (2007)
10. Benitez, C., Fierens, P.: Command dimension reduction in masquerader detection. In: V Conferencia Iberoamericana en Seguridad Informática, CIBSI 2009, Montevideo, Uruguay, Nov. 16-18,2009, Uruguay (2009)
11. Greenberg, S.: Using unix: Collected traces of 168 users. Technical Report 1988-333-45, Department of Computer Science, University of Calgary, Calgary, Alberta, Canadá (1988)
12. Schonlau, M.: Masquerading user data. http://www.schonlau.net/intrusion.html (1998)
13. Chinchani, R., Muthukrishnan, A., Chandrasekaran, M., Upadhyaya, S.: RACOON: Rapidly Generating User Command Data for Anomaly Detection from Customizable Templates. In: Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC '04), Tucson, Arizona (2004)
14. Killourhy, K., Maxion, R.: Comparing anomaly-detection algorithms for keystroke dynamics. In: IEEE/IFIP International Conference on Dependable Systems & Networks, 2009. DSN'09. (2009) 125–134
15. Allen, J.D.: An analysis of pressure-based keystroke dynamics algorithms. Master's thesis, Southern Methodist University, Dallas, TX (2010)
16. Loper, E., Bird, S.: Nltk: the natural language toolkit. In: Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2002) 63–70
17. Shepherd, S.: Continuous authentication by analysis of keyboard typingcharacteristics. In: European Convention on Security and Detection. (1995) 111–114