2014 IEEE International Conference on Systems, Man, and Cybernetics
October 5-8, 2014, San Diego, CA, USA

Mindmetrics: Identifying users without their login IDs

Juyeon Jo, Yoohwan Kim, and Sungchul Lee
Department of Computer Science
University of Nevada, Las Vegas
Las Vegas, Nevada, USA
{Juyeon.Jo, Yoohwan.Kim }@unlv.edu, lee173@unlv.nevada.edu

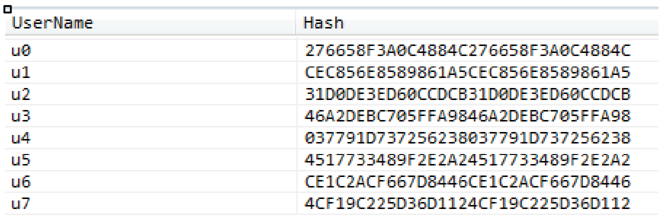
Abstract*— Authentication to a computing system is composed of two parts, identification and verification. Traditionally, login IDs have been used for identification and passwords for verification. Many schemes have been proposed to improve both parts, but they may require specialized devices or they may not be always reliable. We propose a method that can augment the current password-based system by strengthening the identification process. It utilizes personal secret data instead of a login ID to identify a user uniquely, hence mindmetrics. It then asks the user to choose a correct login ID among multiple choices of partially obscured IDs. Since it does not accept a login ID during the authentication process, a stolen or cracked password cannot be used for gaining an access to the computing system unless the attacker provides a correct identification material, i.e., mindmetrics token. This additional step raises the security of an authentication system considerably over single or double password systems. Since the stolen passwords cannot be used immediately by the attackers, account holders can have extra time to change their passwords before the attackers gain an access. This scheme does not require any specialized hardware and can be implemented easily. It may be used where biometrics schemes cannot be used cost-effectively, e.g., on public e-commerce web sites. Mindmetrics scheme separates the identification server and the verification server, thus it is scalable to a large system. We implemented a proof-of-concept system and evaluated it with test users. The survey indicates that the system is not as intrusive as other schemes, users feel better protected, and they are willing to use the scheme on public web sites.

***Keywords*— Cybersecurity, authentication, identification, password**

I. BACKGROUND AND MOTIVATION

Computer systems employ an authentication mechanism to allow access only to legitimate users. The authentication procedure is composed of two parts, *identification & verification*. The identification is for answering the question, "who am I?", and the verification is for answering, "Am I who I claim I am?". Traditionally the identification is performed with a username or login ID and the verification is done with a password. In a password-based system, the plaintext passwords are transformed into hash values with a one-way hash function, and stored in a password hash file. During the verification process, a new hash value is generated from the newly entered password, and compared with the stored hash value in the password hash file. If the hash values match, access is granted. This password verification process is the heart of the most authentication systems.

There are a number of ways to acquire other users' password for illegal access. Plaintext passwords can be captured from the network, by malware or by key logging software. When the plaintext password is not available, the attackers can try password-guessing attack where they try possible values for the victim user. In the password cracking attack, the attackers obtain a password hash file (Figure 1), and tries different inputs to find an input that produces the same hash value as the victim user's hash value.



UserName	Hash
u0	276658F3A0C4884C276658F3A0C4884C
u1	CEC856E8589861A5CEC856E8589861A5
u2	31D0DE3ED60CCDCB31D0DE3ED60CCDCB
u3	46A2DEBC705FFA9846A2DEBC705FFA98
u4	037791D737256238037791D737256238
u5	4517733489F2E2A24517733489F2E2A2
u6	CE1C2ACF667D8446CE1C2ACF667D8446
u7	4CF19C225D36D1124CF19C225D36D112

Figure 1. Sample password hash file

Password hash files are stolen quite often and cracked by hackers. Password cracking attack is a statistical attack, and some of the weak passwords can be broken through a dictionary attack or a hybrid attack. After the attackers crack some passwords, they can access the system using the known login IDs for the cracked passwords. The attack against passwords is a serious threat to current authentication systems, and additional security measures are needed to mitigate this threat. Once an account is breached, the cost from the damage is high for both victims and the companies. In 2013, the average organizational cost of data breaches in US was $5.4 million and the average per capita cost was $188 in US. [1].

While passwords are supposed to be random characters, login IDs are not random. They are used for communication or accounting purposes, and must carry a meaningful pattern. It may be part of users' first and/or last names, part of social security number, combination of names and numbers, account number, or email addresses. Thus login IDs are publicly known or can be guessed easily. In other words, obtaining the login ID is generally not a barrier for the attackers, and the success of an attack depends on the difficulty of the password. While a great emphasis was given to the verification, i.e., password system, somewhat less attention was given to the identification, *i.e.,* login ID. By fortifying the identification part, the overall authentication system can be strengthened.

The goal of this research is improving the security of the authentication system by supplementing it with a secure identification process. To make false login attempts difficult, our method does not use a publicly known login ID for identification. Instead it uses private information known only to the computer system and the user. This process makes the stolen password files unusable for the attackers.

In the following sections, we survey the previous works, describe the proposed method, explain the demonstration system, and present the survey results from the test users.

II. Previous Work

Alternative methods to replace or fortify the password-based authentication system have been developed and some are widely used at this time [2]. Generally they combine methods based on "*what you know*", "*what you have*", "*what you are*", and more recently, "*where you are*". When two methods from two different categories are combined, it is considered two-factor authentication. The methods include,

1) What you know

- Secondary password: An additional password is used. This could be in the same style as the primary password, or personalized data such as mother's maiden name. But like with the primary password, it is subject to password cracking or theft attacks.
- Graphical password: A user selects a set of images in a specific order or creates a drawing [3, 4].

2) What you have

- One-time password: A security token is generated by a special hardware device [5].
- Challenge-response: A challenge generated by a server is sent to the user. The user then creates a response and sends it back. It can be done manually by looking up a random data table or with a hardware device [6].
- Out-of-band messaging: A response request message is sent to the user's email address or to the user's mobile phone as a text message [7]. While it provides extra security, it may be too intrusive for frequent access.

3) What you are

- Biometrics: It uses the characteristics of the body of the user such as fingerprint [8], palm-print pattern [9], footprint [10], retina pattern, voice, etc. These features can be used for both identification and verification. However, digitized biometric data can be stolen, and once it is stolen, it is very hard to revoke it since it is part of the user. Also biometric identification may not be 100% accurate. To use biometric data, a special scanning device is needed, limiting the usage of it. Some people may be reluctant to supply the biometric data due to a privacy concern.
- Behavioral pattern: Some activity patterns are unique to a user and can be used to identify and/or verify a user. For example, a typing pattern [11, 12, 13], gesture [14], mouse pressing force pattern [15], etc., can be used for identification purpose. However, their accuracy is not 100% in all conditions, which may limit their adoption.

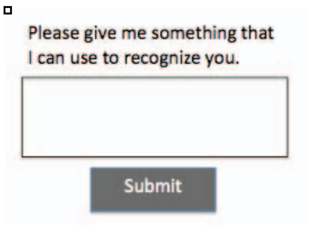
4) Where you are

- Location-based authentication: Login is limited to a particular IP address, a computer, or a specific geographic location [16].

III. Identification without Login ID

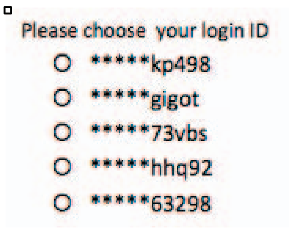
Identification is a process to recognize an unknown individual out of many (1:N relationship). All authentication systems require some form of ID for user identification, such as text-based login ID, fingerprint, or facial image. Then in the verification process, the system asks a proof of the ID since the system does not know whether the user is the legitimate ID holder (1:1 relationship).

Generally the login ID is not considered a secret. For example, a social security number or an email address may serve as a login ID. It is impractical to keep the login ID secret because it is used for many other purposes such as accounting or email. So an alternative secret is needed for identification to recognize a user uniquely. This secret is referred as *Mindmetrics token*. We coined the term *Mindmetrics* due to the similarity to biometrics. In case of biometrics, only the legitimate holder of the physical trait (e.g., fingerprint) can pass the identification stage. Similarly, with some kind of personal secret knowledge (mindmetrics token), a user can pass the identification stage, thus the effect of biometrics is simulated. Without this secret knowledge, the attackers cannot pass the identification stage before they can even try the password The process of identification in Mindmetrics uses something in user's mind instead of something on their body.


Please give me something that I can use to recognize you.

Submit

(a) Presenting an access token


Please choose your login ID

- ○ *****kp498
- ○ *****gigot
- ○ *****73vbs
- ○ *****hhq92
- ○ *****63298

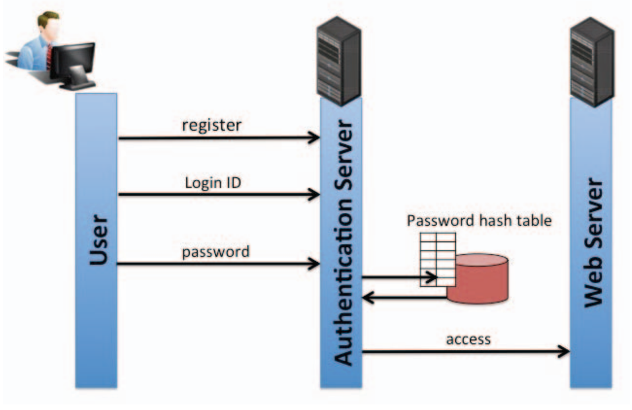
(b) Choosing a login ID

Figure 2. Deriving a login ID from an access token

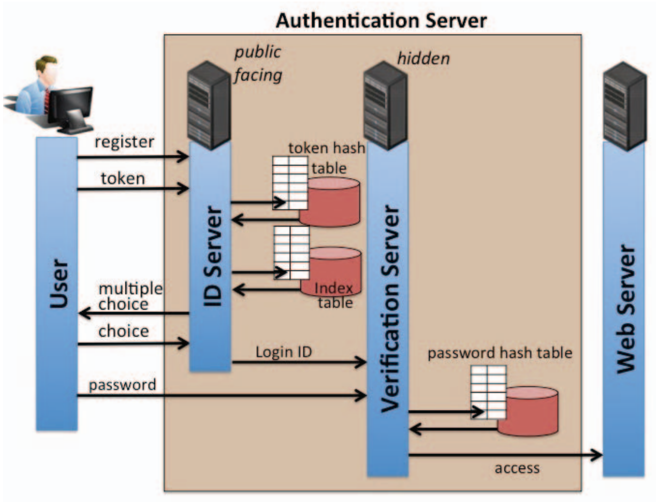
There are two parts in the mindmetrics-based authentication process. First, mindmetrics token is requested in the login page (Figure 2 (a)). A user specifies the token with which a computing system can identify a user account. Then the identification server looks up the registered access tokens to find a matching token and login ID. Second, the server presents multiple login IDs to the user, with one of the login IDs being the correct login ID for the user account and some more real or fake IDs. To prevent the attackers from recognizing the login IDs, the login IDs are partially obscured (Figure 2 (b)). Among these partial login IDs, a legitimate user can still recognize the correct login ID and choose it.

This completes the identification stage, and the rest is same as password verification system. If the login ID and password match the credentials stored for the user account, the user is authenticated to access information associated with the user account. For a failed login attempt, no information is given back to the user, so the attacker will not know whether he entered a valid access token, chose a wrong login ID, or entered a wrong password.

Figure 3 compares the conventional password-based system and the proposed mindmetrics-based system. While the password-based system allows anyone to try publicly known login IDs without any restriction, mindmetrics-based system allows only the legitimate users to pass the identification stage. Now the password verification server is hidden, and users cannot access it unless they pass the identification server. Compared with the conventional password-based system, the only difference is the addition of the identification server, so an existing password-based system can be easily upgraded by just adding the identification server.



(a) Conventioanl password-based system



(b) Mindmetrics-based system

Figure 3. Conventional vs. Mindmetrics system

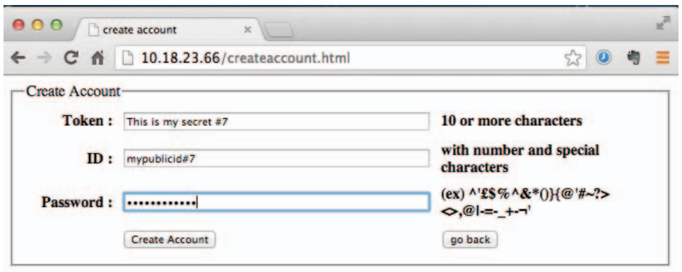
Mindmetrics system has several benefits over biometrics. The characteristics in mindmetrics can be changed if needed. It does not requires any specialized device, so it can be used for accessing local or remote computing systems from any conventional private or public computers. It is a deterministic process, and thus there is no uncertainty. Thus mindmetrics is more practical, cheaper, and can be used by any public web sites such as e-commerce web sites.

IV. Detailed Description of Mindmetrics System

We describe the operation of mindmetrics system in detail with sample screenshots and algorithms from the proof-of-concept system.

A. Step 1: Mindmetrics token registration

Figure 4 shows a webpage for creating an account with a mindmetrics token, account login ID, and password.


create account

10.18.23.66/createaccount.html

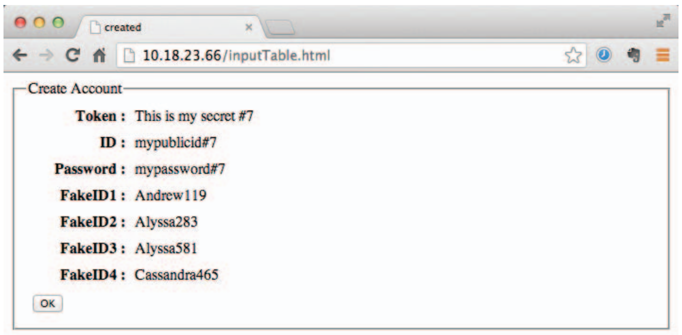
Create Account

Token :	This is my secret #7	**10 or more characters**
ID :	mypublicid#7	**with number and special characters**
Password :	•••••••••••••	**(ex) ^'£$%^&*(){@'#~?> <>,@|-=-_+-¬'**
	Create Account	go back

Figure 4. Creating an account with mindmetrics token

In this illustration, a user types the access token "This is my secret #7", and specifies a desired login ID and password. Alternatively, the login ID may be given by the server. After the user has entered them, the user selects the "Create Account". Then the authentication server validates the entered information, for example, to ensure that the login ID is unique among all login IDs and that the password satisfies the password requirements (e.g., a length of the password). If there is an error, it may prompt the user to enter another login ID or password. To increase security, it may use special keys (e.g., CTRL and ALT) in combination with other alphanumeric keys or insert spelling errors intentionally in random locations.

During the account creation process, we can examine the existing tokens and reject similar tokens. In particular, we use *similar_text()* function [17], a recursive divide-and-conquer alogrithm with *$O(n^3)$* complexity where *n* is the longest string. Given two strings, the function returns a similarity value in percentage. Figure 5 shows the result of the account creation. The webpage displays the token, the username, and the password (which may be displayed masked or unmasked). A number of fake IDs are created together and will be displayed during normal login process. In some implementations, the user may be able to specify the fake usernames.


created

10.18.23.66/inputTable.html

Create Account

Token :	This is my secret #7
ID :	mypublicid#7
Password :	mypassword#7
FakeID1 :	Andrew119
FakeID2 :	Alyssa283
FakeID3 :	Alyssa581
FakeID4 :	Cassandra465

OK

Figure 5. Confirmation of account creation

The account information is stored in separae places. The plaintext tokens are used only for similarity test, and not used for normal login process. The Mindmetrics tokens are stored in a hashed fom in the token hash file as {token hash value, index} tuple and the matching login IDs are stored in the index file as {index, fake login ID, fake login ID, true login ID, fake login ID…} tuple. The password hash file contains <login ID, password hash value> same as before.

B. Step 2:Mindmetric token submission

Figure 6 shows a webpage for supplying a token. The text that is typed may be masked so that a nearby individual is not able to discern the token. The identification server receives user input specifying a token, and determine whether the token matches a token stored for a user account. For example, it may identify that the login ID "frank1982" is assigned to a user account to which the token "ilikemathematics" is assigned.

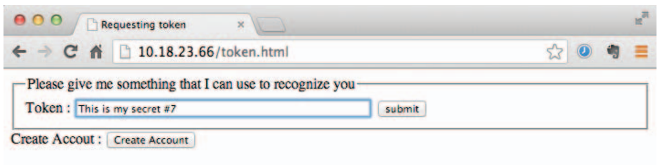

Requesting token
10.18.23.66/token.html
Please give me something that I can use to recognize you
Token : This is my secret #7
submit
Create Accout : Create Account

Figure 6. Mindmetrics token submission

We expect that the tokens will be long, diverse, and mixed with special keys and intentional misspellings. It is difficult to type such a complex token without an error. Moreover, with non-text inputs such as voice are used as a token, the accuracy may be lower, too. In this case, the *similar_text()* function can be used to allow a certain level of error. The tolerance level can be configured dynamically by the user. For example, the token hash value may be used at the first try, then we can gradually increase the toleance level upon each failure.

C. Step 3: Login ID selection

After the matching login ID is determined, the server does not display it immediately because an attacker can discover the token and the matching login ID. It presents multiple login IDs and let the user select one. The number of choices can be configured as desired by the identification server or the user. Figure 7 shows the login ID selection process.

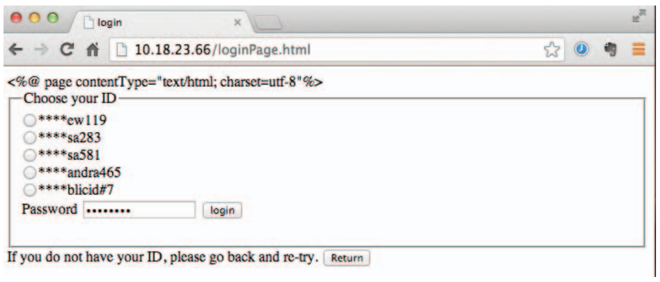

login
10.18.23.66/loginPage.html
<%@ page contentType="text/html; charset=utf-8"%>
Choose your ID
○ ****ew119
○ ****sa283
○ ****sa581
○ ****andra465
○ ****blicid#7
Password ••••••••
login
If you do not have your ID, please go back and re-try. Return

Figure 7. Selecting a login ID

Several strategies are employed to enhance the security in this step.

1) Partial obscurity

The displayed login IDs are partially-obscured by replacing some characters in the username with asterisks. For example, a user knows that his username is "frank1982," and therefore selects a radio button to indicate that the text "****k1982" matches the username. Any part of the username can be obscured, for example, f*a*k*9*2 or frank****. This is not a problem for a legitimate user, but it makes it difficult for the attackers to recover the full login ID.

2) Multiple login IDs

The authentication server identifies the associated fake login IDs from the index table (Figure 3 (b)). They may include real login IDs of other users.

3) Mutual authentication

The names and the order of the login IDs are fixed at time of account creation. Over time, a user is able to learn through user experience which partially-obscured login ID matches the true login IDs. In this way, the user is able to quickly log on to the account without much additional effort. This helps the user to recognize a fake server performing a man-in-the-middle attack (MITM). When the displayed login IDs are different from what the user expects, it is either due to a typo in the token input, or due to an MITM attack.

4) Preventing inference attack with fixed choices

The displayed list of partially-obscured login IDs is the same and appears in the same order every time. An attacker cannot simply supply the same token with a different computing system (using a different IP address) in order to identify which partially-obscured login ID does not change.

5) Handling non-existant tokens

If the token does not match any token in the token hash file, the identification system creates random login IDs and show it the user. This prevents an attacker from knowing whether the entered token matches any user accounts. To prevent an inference attack, the choice of the displayed login IDs do not change when the attacker types the non-existing token again. A hashing algoithm can be used to generate a fixed set of loginIDs consistantly for a given non-exsting token.

D. Step 4: Password verification

After selecting a radio button to choose a login ID, the user types the password that matches the login ID. The login ID and password is given to the verification server. If they match the credentials assigned to the user account, the verification server grants an access. Should the user have entered credential information that did not match all the information assigned to the user account, the server indicates that the user entered incorrect credentials and try it again from the beginning. The verification server may not indicate to the user which of the token, login ID, or password caused the failure.

V. ANALYSIS

A. Level of safety

By making the token long and sophisticated, the level of protection can be increased. By combining intentional misspelling and special keys, a high-complexity token can be contrsucted. However, mindmetrics is still an authentication system based on "what you know", not based on "what you have" or "what you are". Therefore mindmetrics system may look like a double-password system where a user provides two independent passwords. However, it has a significant advantage over the conventional password system as it makes a targeted attack on a particular user account nearly impossible.

We compare the attack complexity on mindmetrics and on a double-password system. We assume that the size and strength of all the passwords and the tokens are same. In a double-password system, the attackers need to crack two password hashes. Assuming each password is *n* letters long with *c* character set, the average search space size for each password is $(c^n)/2$. Therefore the average search space for two passwords is (c^n). Note that this attack does not depend on the size of the password hash file. The attacker only need to work on a single targeted user account.

To crack the mindmetrics system, on the other hand, the attacker must locate the correct token first to get the particular login ID. Assuming the size of the token is same as the size of the password, the average search space for each token is $(c^n)/2$. However, there are multiple accounts in the token hash file, say *t*, so the search space becomes $t * (c^n)/2$ on average. The attacker also must crack the password as usual, which takes $(c^n)/2$. Therefore the total complexity to attack a user account is $t * (c^n)/2 + (c^n)/2 =(t + 1)(c^n)/2$. As the number of accounts in the token hash file (*t*) and the length of token (*n*) grow, the search space grows quickly. So mindmetrics has a significant advantage over a double-password system.

Another adavantage of mindmetrics is the letter pattern. In a double-password system, the the complexity and the patterns in both passswords are likley to be similar for agiven user. If one of them is weak, the other one may be also weak. This increases the effectiveness of an attack. In fact, two passwords of length *n* has an average search space of (c^n), while a single password with a length of $2n$ has (c^{2n}). In other words, using a single long password is far stronger than using two short passwords. However, by using one password, the risk from a theft of a password hash file increases. In case of mindmetrics, the letter pattern in the token is different from the password's. Users can use a long sentence that they can remember. The length alone renders a cracking attack impractical. Even if a user's passwods is weak, the user may employ an entirely different letter pattern for the mindmetrics token so that it is not subject to the same type of cracking attack.

With these barriers, the attackers will face a great challenge in trying to derive the correct login ID out of millions of potential login IDs before trying a password attack. This barrier provides valuable extra time to combat the password attack in case the password hash file gets stolen. Password guessing attack is virtually eliminated because the attacker has to retrieve the victim's login ID first before even trying a password guessing attack. The correct user ID can be retrieved only when the correct access token is provided. Thus the chance of successful password guessing attack is greatly reduced.

B. Security

1) Separate storage of files

The fist line of defense comes from the storage of multiple files. The authentication system stores the token hash file, index file, and password hash file separately in order to mitigate the risk of theft. Unless the attacker obtains them altogether, a successful cracking attack is very difficult.

2) Stolen password hash file

If the password hash file is stolen, and crcaked successfully, the attacker has a pair of login ID and a cracked password. But mindmetrics system does not accept login ID during authentication, so there is no immeidta threat.

3) Stolen token hash file

First, the lengthe of a token is much greater than passwords, so a cracking attack is very difficult. Even if a cracking attack is successful or a plaintext token file is stolen, the attacker only finds the login IDs. At that point, the security of the system comes down to the same level of a conventional password system.

4) Mutual authentication

A user can verify whether she is accessing a legitimate authentication server or not by checking the set of login IDs. If a correct list of login IDs are not presented, the user can recognize a Man-In-The-Middle (MITM) attack.

C. Scalability

The identifcation server can be separated from the verification server. Multiple identifcation servers may be employed for a large system. It is conceivable that a 3rd party company provides the identification service to customer companies, while the password verification is done in-house. Since the 3rd party company does not know the password, the identification service is safe to outsourcing. This structure also improves the security because the chance of both the token hash file and the password hash file stored in two independent companies to be stolen is very low.

D. Usability

Unlike biometrics, mindmetrics does not use any specialized hardware. As long as the user types the token correctly, it identifies a user accurately. If the token is revealed, the user can re-register a new token. Mindmetrics authentcation system can be used either for a local or a remote machine. Existing password systems can be easily upgraded to mindmetrics system by adding the identification server without any change in the existing password system.

VI. Usability Survey Results

We implemented a proof-of-concept system and asked a group of students to evaluated it. A total of 13 students volunteered. As the sample group size is rather small, the results below should be taken with a grain of salt. In addition, since the prototype system is not highly user-friendly, some testers had a difficulty in following the steps properly. So the results may be improved with a better prototype in the future.

1) *Was it annoying to have this extra step?*

• **Very annoying:**	**5**
• Somewhat annoying:	3
• Not annoying:	2
• Somewhat amusing:	3

2) *How intuitive was it?*
- Very intuitive: 1
- **Somewhat intuitive: 4**
- **Neither intuitive nor counter-intuitive: 5**

- Somewhat counter-intuitive: 2
- Steps not clear: 1

3) Compared with other technologies, is this more convenient?

- **Somewhat convenient: 9**
- Somewhat inconvenient: 2
- Very inconvenient: 2

4) Will you use it if it is adopted by popular web sites?

- Definitely: 2
- **Maybe: 6**
- Only if I am forced: 3
- Not at all: 2

5) Was the partial blocking of the ID acceptable to you?

- Very acceptable: 2
- **Acceptable: 9**
- Neither acceptable nor unacceptable: 1
- Unacceptable: 1

6) Was it hard to find your ID from the choices because some characters are blocked?

- **Very easy: 6**
- Easy: 3
- Neither easy not hard: 4

7) Did you feel more protected by having this extra step?

- Well protected: 2
- **Somewhat protected: 8**
- I am not sure: 1
- Less protected: 2

Overall we observe that mindmetrics system is acceptable to most users although the test population size was small. Users feel they are protected and are willing to use it if it is adopted widely. However, it is still an extra step in addition to password, and it does cause some annoyance. This should be improved with a better user interface.

VII. Conclusions

User authentication is done in two steps, identification and verification. The traditional password-based verification system has been challenged by sophisticated attacks, but new schemes are being made to cover the weaknesses of the password-based systems. However, the identification part is still done based on a public login ID. We proposed a new scheme called mindmetrics to strengthen the identification process with a personal secret information. In mindmetrics systems, a login ID is not asked. Instead a user must provide the correct token to pass the identification stage. In case the password file is stolen, the login attemps by attackers is blocked by the identification server. Thus it may stop or slow down attackers, and account holders can change their account credentials before attackers can gain access. Mindmetrics can simulate biometrics with its high security level where true two-factor authentication with biometrics is not feasible. It is very simple, and it does not require any specialized devices nor any methematical algorithm. So, it is more practical than other methods and can be used by any public web sites. We implemented a proof-of-concept system and evaluated it with test users. The survey indicates that mindmetrics system is intuitive and easy to use, and users are willing to use it to have extra protection.

References

[1] Ponemon Institute Research Report, "2013 Cost of Data Breach Study: Global Analysis", May, 2013, available on https://www4.symantec.com/mktginfo/whitepaper/053013_GL_NA_WP_Ponemon-2013-Cost-of-a-Data-Breach-Report_daiNA_cta72382.pdf

[2] Joseph Bonneau, cormac Herley, Paul C van Oorschot, and Frank Stajano, "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes", 2012 IEEE Symposium on Security and Privacy, pp. 553 - 567

[3] Bin B. Zhu, Jeff Yan, Guanbo Bao, Maowei Yang, and Ning Xu,"Captcha as Graphical Passwords—A New Security Primitive Based on Hard AI Problems", IEEE Transactions on Information Forensics And Security, Vol. 9, No. 6, June 2014, pp. 891 - 904

[4] Emmanouil Georgakakis, Nikos Komninos, Christos Douligeris, "NAVI: Novel Authentication with Visual Information", IEEE Symposium on Computers and Communications, 2012 , pp. 588 - 595

[5] N. Sklavos and C. Efstathiou, "SecurID Authenticator: On the Hardware Implementation Efficiency", 14th IEEE International Conference on Electronics, Circuits and Systems, 2007, pp. 589 – 592

[6] Anna Vapen, David Byers, and Nahid Shahmehri, "2-clickAuth – Optical Challenge-Response Authentication", 2010 International Conference on Availability, Reliability and Security, pp. 79 - 86

[7] M. Alzomai, A. Jøsang, A. McCullagh, E. Foo, "Strengthening SMS-Based Authentication through Usability", International Symp on Parallel and Distributed Processing with Applications, 2008, pp. 683 - 688

[8] Dileep Kumar, Yeonseung Ryu, and Dongseop Kwon, "A Survey on Biometric Fingerprints: The Cardless Payment System" 2008 Int'l Symposium on Biometrics and Security Technologies, pp. 1-6

[9] Bob Zhang, Wei Li, Pei Qing, and David Zhang, "Palm-Print Classification by Global Features", Ieee Transactions On Systems, Man, And Cybernetics: Systems, Vol. 43, No. 2, March 2013, pp. 370 – 378

[10] Jaeseok Yun, Gregory Abowd, Woontack Woo, Jeha Ryu, "Biometric User Identification with Dynamic Footprint", 2007, 2nd Int'l Conference on Bio-Inspired Computing: Theories and Applications, pp. 225-230

[11] Alon Schclar, Lior Rokach, Adi Abramson, and Yuval Elovici, "User Authentication Based on Representative Users", IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 42, No. 6, November 2012, pp. 1669 – 1678

[12] Mariusz Rybnik, Marek Tabedzki, and Khalid Saeed, "A Key stroke dynamics based system for user identification, 7th Computer Information Systems and Industrial management Applications, 2008 pp. 225 – 230

[13] Zahid Syed, Sean Banerjee, Qi Cheng, Bojan Cukic, "Effects of user habituation in keystroke dynamics on password security policy, 2011 IEEE 13th International Symposium on High-Assurance Systems Engineering, pp. 352 – 359

[14] Jože Guna, Iztok Humar, and Matevž Pogaþnik, "Intuitive Gesture Based User Identification System", 35th International Conference on Telecommunications and Signal Processing (TSP), 2012, pp. 629 – 633

[15] Chao Shen, Zhongmin Cai, Xiaohong Guan, Youtian Du, and Roy A. Maxion, "User Authentication Through Mouse Dynamics", IEEE Trans on Information Forensics and Security, v. 8, no. 1, Jan 2013, pp. 16 - 30

[16] Feng Zhang, A. Kondoro and S. Muftic, "Location-Based Authentication and Authorization Using Smart Phones", IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2012, pp. 1285-1292

[17] Ian Oliver, "Programming Classics: Implementing the World's Best Algorithms", Prentice Hall, 1994