ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/335608673

Secure Authentication Mechanism for Resistance to Password Attacks

Conference Paper · September 2019

DOI: 10.1109/ICTer48817.2019.9023773

CITATIONS
13

READS
536

1 author:

Vijayanathan Senthooran
University of Jaffna
12 PUBLICATIONS **64** CITATIONS

SEE PROFILE

All content following this page was uploaded by Vijayanathan Senthooran on 04 November 2019.

The user has requested enhancement of the downloaded file.

Secure Authentication Mechanism for Resistance to Password Attacks

S.Subangan, V.Senthooran

Department of Physical Science, Faculty of Applied Science, Vavuniya Campus of the University of Jaffna, Sri Lanka

subangan01@gmail.com
senthooran@mail.vau.jfn.ac.lk

***Abstract*- Authentication is a process that provides access control of any type of computing applications by inspecting the user's identification with the database of authorized users. Passwords play the vital role in authentication mechanism to ensure the privacy of the information and avert from the illicit access. Password based authentication mechanism suffers from many password attacks such as shoulder surfing, brute forcing and dictionary attacks that crack the password of authentication schema by the adversary. Key Stroke technique, Click Pattern technique, Graphichical Password technique and Authentication panel are the several authentication techniques used to resist the password attacks in the literature. This research study critically reviews the types of password attacks and proposes a matrix based secure authentication mechanism which includes three phases namely, User generation phase, Matrix generation phase and Authentication phase to resist the existing password attacks. The performance measure of the proposed method investigates the results in terms existing password attacks and shows the good resistance to password attacks in any type of computing applications.**

***Keywords*- Shoulder surfing, Brute forcing, Matrix based authentication.**

I. INTRODUCTION

Passwords are the most general form of user authentication technique used in various computing applications like banking ATM, websites, operating systems login and mobile phones. However, user's passwords are cracked and bargained under different vulnerabilities [1]. This paper summarizes the different types of password attacks and describes the authentication techniques by justifying the resistance to existing password attacks. Finally, it presents a password based secure authentication mechanism to resist some password attacks in different computing applications. Shoulder surfing is one of the password attacks in which the adversary detectives the user's movements to steal their passwords. The adversary observes how the users enters the password. Eventually, the attackers can observe and use all the options related to the password length [2]. The brute force attacks pattern all possible combination of password until the correct one is received to break the authentication method [3]. It is a time consuming as searching all combinations and mostly used to crack the encrypted passwords. It is effective for small length passwords. The dictionary attack is a form of brute force attack but faster the brute force attack and it attempts to match the password with mostly used words in our daily life. In this technique, the attacker creates the dictionary of most commonly used words and they can use these words to break the authentication mechanism [4]. Another password attack is Key Loggers which are software programs installed in user computer and monitors the user activities by copying the key pressing activities of user [5]. The attackers crack the authentication technique using the log file which stores all the history of key pressing activities and the log file will be forwarded to the attacker's e-mail. A web-based password attack is called phishing in which the attackers redirect the user to the fake website whose interface is like real website to attract the user to crack the password by retrieving the login information from the fake website [6, 7]. Another password attack is replay attacks also called as reflection attacks which targets response user authentication method. In this attack, the attacker initially enters his/her password first time login phase. The receiving device sends the trial to the sender to authenticate the method. The attacker utilizes s the process and responses to receiving device. The receiving side accepts the challenge and responds the query of attacker [8]. The standard process of password scheme allows the users to log into the system by his/her username and password then the system validates the user by matching the user database and grants the access to the users [9]. Although, the benefit of this scheme is to provide the security of data by handling only the authenticated users, this schema is vulnerable to password attacks such as shoulder surfing, key loggers, phishing and brute force [10]. To resist the password attacks in different applications, many types of authentication techniques have been developed. Those are challenges to password attacks. The key stroke dynamics is a one type of authentication techniques which records key press and its timings [11]. It deals with hands movements of users during the typing and stores the time patterns of users. This authentication mechanism prevents the password attacks like shoulder surfing, phishing and key logging. The click pattern [12, 13] is another type of authentication mechanism that uses mouse to enter password into system. The user is facilitated with a click pad which contains different color grids or combination of different symbols on the monitor when he/she enters the password. The attacker is deceived by with the click patterns as password by users. It also resists shoulder surfing and key logger attacks. Graphical password is an authentication method in which the user first enters the user id to login and then some graphical objects are displayed that are to be selected by user [14]. The authentication is processed based on the elected graphical objects which are pre-processed by hierarchical matching technique. It is more secure authentication for shoulder surfing attack. Another type of authentication is biometrics which is based on the recognition upon image processing. This authenticates by matching selected features of user's image and database image and comprises real and unique

978-1-7281-5156-4/19/$31.00 ©2019 IEEE

signs that are not be stolen. Authentication panel is an authentication mechanism in which users are enabled to select the position of password words in the grid or panel instead of pressing button [15]. It is a fast authentication mechanism and resists brute force attack, dictionary attack and shoulder surfing attack. The reformation-based authentication mechanism shifts the password into new form before storing the original password. It provides an extra layer for the original password. The reformation is applied to authenticate user dynamically. Therefore, the attacker is no knowledge of original password even if the stored password is stolen [16]. The primary advantage of this scheme is, it is strongly against the attacks such as dictionary attack, shoulder surfing and brute force attack. Time signature is another method of keystroke, click patterns for practical. It is a hybrid password scheme with the combination of conventional password, keystroke dynamics and click patterns [17]. The goal of this hybrid password technique is to provide greater security of data for end users. Time signatures provide more security than the regular password systems. Time signatures can be the good prevention mechanism to the attacks such as shoulder surfing, dictionary attack, key loggers and replay attacks. Even if the attacker got the password also he cannot enter into the system. Because the attacker can't enter the password with the prescribed time sequence by the actual user.

If security breach happened in the authentication system, the valuable and private data confidentiality will become vulnerable. In some cases, it may cause to unauthorized access to the data and the hardware resources, loss of wealth, privacy issues. For example, our g-mail, google drive accounts, google contacts, hangout chats are handled by a single google account. More than that our android smartphones also integrated with the google account. If any attacker got access into our google account, it will be a catastrophic damage for our data and the privacy. In this paper we are proposing a secure password authentication mechanism like click pattern technique and authentication panel technique and it resists shoulder surfing, brute force and key logging attacks in any computing applications.

.

II. RELATED WORKS

In this section, we review and analyze the recent studies of panel-based authentication techniques proposed by researchers in the literature. Divyans Mahansaria et.al [18] proposed an authentication method that works with 8*6 matrix or grid which consists entire 26 English alphabets in capital, 10 numerals and 12 chosen symbols. The characters are organized in the matrix in random manner. In the authentication process the user will enter the position (row number followed by column number) of each character in the password and for last three characters he will enter the same characters which is in the password. In the Figure 1, sample matrix if the password is '1DEI*2DTA#3' then the user needs to enter as '6463131582226316A#3'.

In this mechanism, some complexities are added but it is not sufficient to resist the shoulder surfing and other password steeling attacks such as key loggers and spy cams and any combination of them. If anyone got the entered sequence and the matrix (by help of key loggers, spy cams and screen recorders), they can interpret the password by some attempts. Because the entering sequence only have numbers which are row and column numbers, except the last three characters. So, it's easy to break the method. Furthermore, the user needs to enter more characters than the actual password. If password contains N characters, the user needs to enter 2N-3 characters. It's overhead work for the user. Another authentication method proposed by Mohammed et.al [19] is with 6*6 matrix which contains English Alphabets (Capital and Small), numerals and set of characters in random manner without repetition. Under the matrix there are 6 buttons with arrow for each column. When the user authenticates to the system, he will click the button, which is under the column where the password character is available. Then the system will perform the transpose operation for the same matrix. After that the user again need to click the button, which is under the column where the password character is available. Through these operations the system will identify the first character of the password. There steps need to be repeat for each of the password characters.

	1	2	3	4	5	6
1	A	O	E	N	I	T
2	9	2	H	J	K	Q
3	6	0	Y	S	3	L
4	U	V	7	B	W	P
5	C	M	8	4	R	F
6	X	Z	D	1	G	5
7	!	@	#	$	%	:
8	&	*	(	)	?	"

Fig. 1 Sample Matrix [18]

In the above case , if the first character of the password is 'Q' then the user need to select the 3rd column in matrix (M) and in transpose matrix(M^T) he need to select the 4th column. For each of the characters in the password he needs to do these same operations. In this method also, there are some complexity added. But it's still vulnerable for the persons who know about this authentication mechanism. They also can interoperate the actual password by noticing the click stream and the matrix. This method also too much overhead to the user, because for each character of the password he needs to click two times. If the password length is N, then the user needs to click 2N times to complete the authentication process.

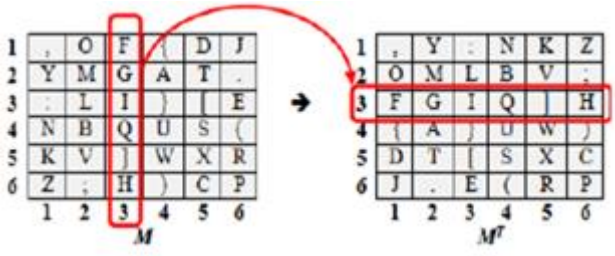

Fig. 2 Matrix Transpose [19]

A pair based scheme was proposed by Kasar Santosh et.al [20] with 6*6 matrix which is filled with 26 capital English alphabets and 10 numerals. In the authentication process the user take the characters of the passwords as pairs. The row will be selected from the first letter of the pair and the column will be selected by the second letter of the pair. The intersection point of the row and column will be taken as the part of the session password. After this the distribution of the characters in the matrix will be change for the next pair.

1	A	J	R	H	7
0	K	9	I	Q	G
3	B	O	C	P	6
Z	L	4	S	T	2
M	Y	W	D	5	F
8	X	N	V	E	U

Fig. 3 Pair Based Scheme [20]

In the above case if the first pair of password characters are 'SA' then the first character of the session password is 'L'. In this schema the user also facing complexity to input the password due to go through the row wise and column wise and then clicking the intersection point. This scheme is limited with 26 capital English alphabets and 10 numerals for selecting the password character.

III. PROPOSED METHODS

A. Matrix Based Authentication Mechanism

1) *User registration phase:* In the registration phase the user will be asked to enter the user name and password. We are recommending the password must have at least 8 characters and it should be mixture of capital, small letters, numerals and symbols. More than the above credentials the user will be asked to enter the selection method (row wise or column wise) for each characters of the password, in the same order. For example let's take "butter$fly" as the password and "CCRRCCCRRR" as the selection method order (here "C" indicating "column" and "R" indicating "row").

Password character	b	u	t	t	e	r	$	f	l	y
Selection method	C	C	R	R	C	C	C	R	R	R

Fig. 4 Mapping between password characters and the selection methods

2) *Matrix generation criteria:* The 10*10 matrix will be filled by the 26 English alphabet (capital and small), 10 numerals, symbols and some special characters in random manner. Even though the English capital letters will be in first three rows, English small letters will be in 4th to 6th rows, numbers will be in 7th row and symbols will be in last three rows. It will be reducing the seeking time of the characters to the user. The remaining empty cells will be filled with some special characters such as α, β, μ, π, €, £, ¥ and Ω.

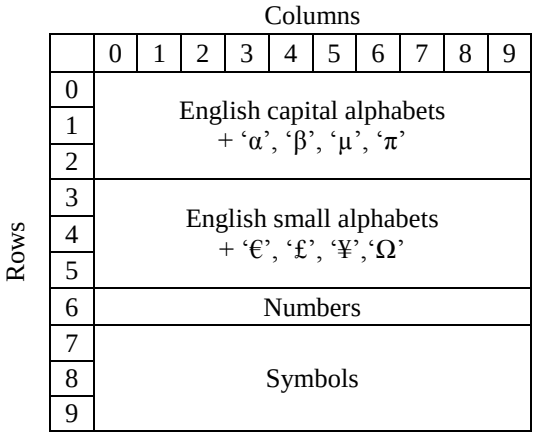

Fig. 5 Characters organization in the matrix

3) *Authentication phase:* Initially the user will be asked to enter his user name. If the user is registered person, he will be encountered with our authentication scheme. The user need to enter the session password character for each of the actual password characters according to the selection method. If the selection method is "C", then user need to find the actual password character in the matrix and enter its column index as the session password character. If the selection method is "R", then user need to find the actual password character in the matrix and enter its row index as the session password character.

Algorithm:

Step 1: Fetch the password and selection method from the database.

Step 2: Get the next character (initially first character) of the password and its selection method.

Step 3: Insert that character into the empty matrix randomly with the consideration of matrix generation criteria, and get that character's row/column index value according to the selection method, and append that index value with system generated session password.

Step 4: Fill the rest of the matrix according to the matrix generation criteria.

Step 5: Display the matrix to the user and get the input from the users.

Step 6: Append user input with the user's session password.

Step 7: Repeat the steps 2 to 6 for all the password characters.

Step 8: If the system generated session password is matched with user's session password, it's a successful authentication.

In the figures 6, 7 & 8 "butter$fly" is the password and "CCRRCCCRRR" as the selection method order for example.

First character of the password is "b" and its selection method is "C". So the user need to search the character "b" in the above matrix and enter its column index. That is "4". It will be added in the session password as the first character. Then the matrix will be rearranged and characters will be distributed randomly according to the matrix generation criteria.

	0	1	2	3	4	5	6	7	8	9
0	Q	α	E	T	O	W	A	U	µ	K
1	F	X	P	L	Z	D	β	H	R	Y
2	M	I	B	V	G	S	N	J	π	C
3	c	r	j	e	u	q	h	z	t	w
4	o	d	¥	n	x	k	£	a	Ω	g
5	€	i	p	f	b	v	l	s	m	y
6	5	3	7	1	8	4	6	0	9	2
7	$	{	+	!	.	*	]	)	`	-
8	=	^	(	:	%	~	,	\	@	/
9	;	|	#	_	[	&	<	}	>	?

Fig. 6 Initial Matrix

Second character of the password is “u” and its selection method is “C”. So the user need to search the character “u” in the matrix and enter its column index. That is “5”. It will be added as the second character of the session password. Then the matrix will be rearranged and characters will be distributed randomly according to the matrix generation criteria.

	0	1	2	3	4	5	6	7	8	9
0	G	X	Q	N	J	Z	E	α	A	µ
1	P	K	U	B	π	S	O	L	Y	H
2	R	C	W	I	T	F	β	D	V	M
3	s	a	r	¥	i	u	c	y	n	g
4	p	j	€	d	£	x	h	w	q	l
5	k	Ω	e	t	b	v	m	z	f	o
6	0	7	1	6	3	4	2	8	5	9
7	\	=	]	@	(	<	&	;	_	`
8	>	?	!	,	:	}	$	-	^	/
9	#	[	%	+	~	)	|	*	.	{

Fig. 7 Rearranged matrix after first session character entered

Third character of the password is “t” and its selection method is “R”. So the user need to search the character “u” in the matrix and enter its row index. That is “3”. It will be added as the third character of the session password .Then the matrix will be rearranged and characters will be distributed randomly according to the matrix generation criteria. In the above example first three letters in the session password are “453”. These operations need to be repeat until the last actual password character. At the end, session password length will be equal to the original password length, because for each of the actual password characters, there will be a session password character (index value). Now this session password will be compared with the session password which is created by the system. If both are matched, the user will be allowed to the further process.

	0	1	2	3	4	5	6	7	8	9
0	J	W	I	π	N	T	Z	M	G	R
1	µ	C	K	E	U	B	Y	S	β	H
2	D	X	P	V	F	α	L	O	A	Q
3	k	€	x	p	t	e	i	n	g	v
4	s	a	q	h	c	o	d	y	¥	m
5	Ω	l	£	b	z	j	r	f	u	w
6	3	4	2	9	7	5	0	1	8	6
7	>	*	!	_	+	-	`	[	\	^
8	.	(	<	@	,	{	&	%	$	|
9	/	#	?	}	)	~	:	=	;	]

Fig. 8 Rearranged matrix after second session character entered

B. Sector Based Authentication Mechanism

1) *User registration phase:* In the registration phase the user will be asked to enter the user name and password. We are recommending the password must have at least 8 characters and it should be mixture of capital, small letters, numerals and symbols.

2) Matrix generation criteria: The 10*10 matrix will be filled by the 26 English alphabet (capital and small), 10 numerals, symbols and some special characters in random manner. Even though the English capital letters will be in first three rows, English small letters will be in 4th to 6th rows, numbers will be in 7th row and symbols will be in last three rows. It will be reducing the seeking time of the characters to the user. The remaining empty cells will be filled with some special characters such as α, β, µ, π, €, £, ¥ and Ω. This matrix will be divided as five sectors, each sector has two columns and ten rows.

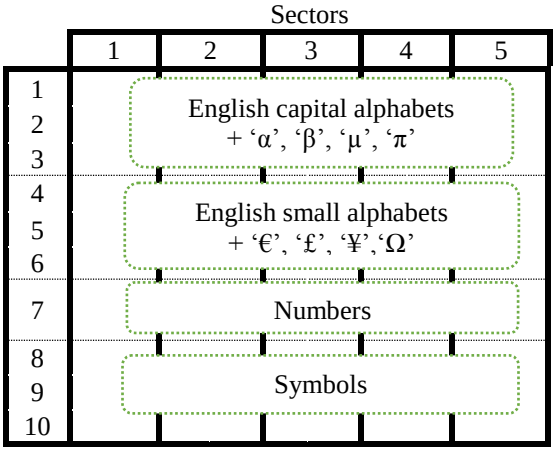

Fig. 9 Characters organization in the matrix

Authentication phase: Initially the user will be asked to enter his user name. If the user is registered person, he will be encountered with our authentication scheme. The user needs to enter the session password character for each of the actual password characters. The user needs to find the actual password character in the matrix and enter its sector number as the session password character.

Algorithm:

Step 1: Fetch the password from the database.

Step 2: Get the next character (initially first character) of the password.

Step 3: Insert that character into the empty matrix randomly with the consideration of matrix generation criteria, and get that character's sector value, and append that sector value with system generated session password.

Step 4: Fill the rest of the matrix according to the matrix generation criteria.

Step 5: Display the matrix to the user and get the input from the users.

Step 6: Append user input with the user's session password.

Step 7: Repeat the steps 2 to 6 for all the password characters.

Step 8: If the system generated session password is matched with user's session password, it's a successful authentication.

In the figures 10, 11 and 12 password is "<Atom888" for example.

1		2		3		4		5	
Q	α	E	T	O	W	A	U	μ	K
F	X	P	L	Z	D	β	H	R	Y
M	I	B	V	G	S	N	J	π	C
c	r	j	e	u	q	h	z	t	w
o	d	¥	n	x	k	£	a	Ω	g
€	i	p	f	b	v	l	s	m	y
5	3	7	1	8	4	6	0	9	2
$	{	+	!	.	*	]	)	`	-
=	^	(	:	%	~	,	\	@	/
;	|	#	_	[	&	<	}	>	?

Fig. 10 Initial Matrix

First character of the password is "<". So, the user needs to search the character "<" in the above matrix and enter its sector number. That is "4". It will be added in the session password as the first character. Then the matrix will be rearranged, and characters will be distributed randomly according to the matrix generation criteria.

1		2		3		4		5	
G	X	Q	N	J	Z	E	α	A	μ
P	K	U	B	π	S	O	L	Y	H
R	C	W	I	T	F	β	D	V	M
s	a	r	¥	i	u	c	y	n	g
p	j	€	d	£	x	h	w	q	l
k	Ω	e	t	b	v	m	z	f	o
0	7	1	6	3	4	2	8	5	9
\	=	]	@	(	<	&	;	_	`
>	?	!	,	:	}	$	-	^	/
#	[	%	+	~	)	|	*	.	{

Fig. 11 Rearranged matrix after first session password character entered

Second character of the password is "A". So the user needs to search the character "A" in the matrix and enter its sector number. That is "5". It will be added as the second character of the session password. Then the matrix will be rearranged, and characters will be distributed randomly according to the matrix generation criteria.

1		2		3		4		5	
J	W	I	π	N	T	Z	M	G	R
μ	C	K	E	U	B	Y	S	β	H
D	X	P	V	F	α	L	O	A	Q
k	€	x	p	t	e	i	n	g	v
s	a	q	h	c	o	d	y	¥	m
Ω	l	£	b	z	j	r	f	u	w
3	4	2	9	7	5	0	1	8	6
>	*	!	_	+	-	`	[	\	^
.	(	<	@	,	{	&	%	$	|
/	#	?	}	)	~	:	=	;	]

Fig 12 Rearranged matrix after second session character entered

Third character of the password is "t". So, the user needs to search the character "t" in the matrix and enter its sector number. That is "3". It will be added as the third character of the session password. Then the matrix will be rearranged, and characters will be distributed randomly according to the matrix generation criteria. In the above example first three letters in the session password are "453". These operations need to be repeat until the last actual password character. At the end, session password length will be equal to the original password length, because for each of the actual password characters, there will be a session password character (sector value). Now this session password will be compared with the session password which is created by the system. If both are matched, the user will be allowed to the further process.

IV. Performance Measure of the Proposed Scheme

The proposed method works with 10*10 matrix which elements are randomly arranged and user will enter the session password which will change for each and every authentication attempt. Capturing the session password using some attack methods such as shoulder surfing, interception, key logging and phishing will be useless. Because that session password will not be valid for the next time. In brute forcing and dictionary attack the attacker will try the possible values one by one and at some point, it will match with the actual password. The session password is different for every attempt, it's not fixed one. So brute forcing and dictionary attack will not help to match with the session password. If an attacker got the all the randomly generated matrixes and the session password for a particular successful attempt by the authorized user, then the attacker has the number of possible combinations for actual passwords is denoted by N. Number of actual password characters is X.

For Matrix based method: $N=19^x$

For Sector based method: $N=20^x$

These are comparatively very height than the existing works [18][19][20] . In related works [18][19], the attackers can identify the actual password, If he got the all the randomly generated matrixes and the session password for a

particular successful attempt. In another work [20], the attacker has the number of possible combinations for actual passwords is denoted by N. Number of actual password characters is X. $N=6^x$.

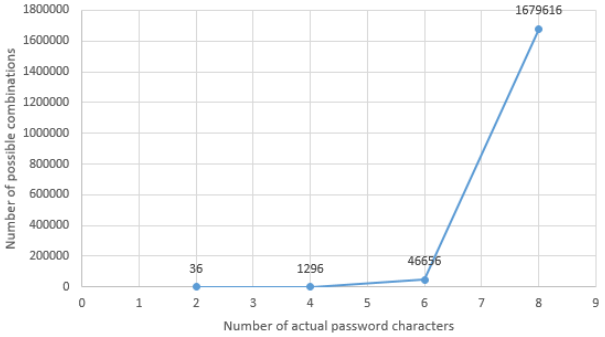

Fig. 13 Pair Based Scheme [20]

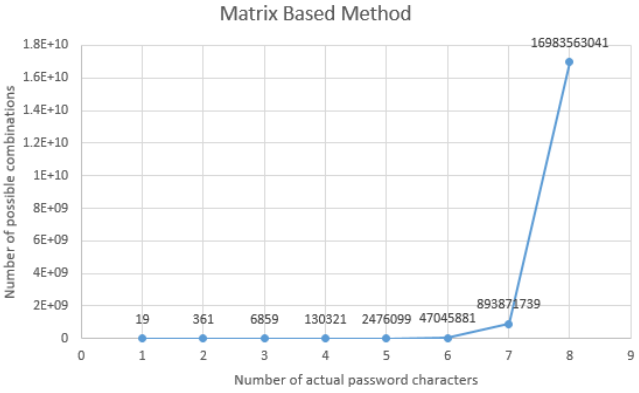

Fig. 14 Proposed Matrix based method

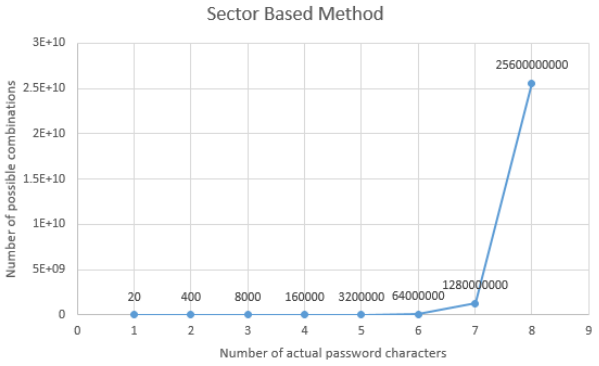

Fig. 15 Proposed Sector Based Method

The proposed sector based method is high resistance than the matrix based method for the attacks, if an attacker got the all the randomly generated matrixes and the session password for a particular successful attempt. More than that the sector based method is convenient and user-friendly than matrix based method.

V. Conclusions

The protection of password theft is an important in today's life. The standard authentication methods are subject to a variety of attacks. In this paper, we summarize the existing password attacks and authentication techniques and propose an effective and convenient secure matrix based authentication mechanism and sector based authentication mechanism to resist the type of password attacks like shoulder surfing, dictionary attack and brute force attack. It will be trustworthy method to ensure the confidentiality, privacy and deny the unauthorized access on the hardware resources. In both methods, the characters are categorized (capital letters, small letters, numbers, characters) and organized in the user convenient manner. The session password will change for each attempts, so that capturing the session password is useless to further attack. The sector based method have high performance and high user convenient than the matrix based method. Hence our sector based method is secured authentication mechanism to resist to the password attacks.

References

[1] THORAWADE M.B. and PATIL S.M, "*Authentication scheme resistant to shoulder surfing attack using image retrieval*", International Journal of Knowledge Engineering, ISSN: 0976-5816 & E-ISSN: 0976-5824, Volume 3, Issue 2, 2012, pp.-197-201.

[2] Doke Ashvini, Wagh Dhanashree, Shaikh Saddam, "*Authentication Scheme for Shoulder surfing using Graphical and Pair Based scheme*", IJARCSMS ISSN: 2321-7782 OCT 2014.

[3] Fujita, K. and Y. Hirakawa, "*A study of password authentication method against observing attacks*", 6th International Symposium on Intelligent Systems and Informatics, SISY 2008M.

[4] Wazir, Z.K., Mohammed, Y.A., Yang, X, "*A Graphical Password Based System for Small Mobile Devices*", International Journal of Computer Science, Issues, Vol. 8, Issue 5, No 2, pp.145-154, (2011).

[5] Baig, M.M. and W. Mahmood, "*A Robust Technique of Anti Key-Logging using Key-Logging mechanism*", Digital EcoSystems and Technologies Conference, 2007. DEST '07. Inaugural IEEE-IES.

[6] Nazreen Banu and Munawara Banu, "*A Comprehensive Study of Phishing Attacks*", International Journal of Computer Science and Information Technologies, Vol. 4 (6), 2013, 783-786.

[7] Antonio San Martino, Xavier Perramon, " *Phishing Secrets: History, Effects, and Countermeasures*", International Journal of Network Security, Vol.11, No.3, PP.163–171, Nov. 2010.

[8] Gagan Dua, Nitin Gautam, Dharmendar Sharma and Ankit Arora, "*Replay attack prevention in Kerberos authentication protocol using triple password*", International Journal of Computer Networks & Communications (IJCNC), Vol.5, No.2, March 2013.

[9] Savita Kamalakarrao Kulkarn, "*A Survey of Password Attacks, Countermeasures and Comparative Analysis of Secure Authentication Methods*", International Journal of Advance Research in Computer Science and Management Studies, Volume 3, Issue 11, November 2015.

[10] Mudassar Raza, Muhammad Iqbal, Muhammad Sharif and Waqas Haider, "*A Survey of Password Attacks and Comparative Analysis on Methods for Secure Authentication*", World Applied Sciences Journal, vol. 19, pp. 439-444, Jan. 2012.

[11] Dalia Abdul Hadi Abdul Ameer and Ahmed Abdulhakim Al-Absi, "*Anywhere On-Keyboard Password Technique*", IEEE Student conference on Research and development 2010 Putrajaya Malaysia.

[12] Muhammad Sharif, Tariq Faiz and Mudassar Raza, "*Time Signatures - An Implementation of Keystroke and Click Patterns for Practical and Secure Authentication*", The third International Conference on Digital Information Management (IEEE ICDIM 2008), University of east London, UK.

[13] Abdurazzag Ali Abura and Manal I. Al Fallah, "*Password generator based on mouse clicks signal and screen cursor position*", IEEE Proceedings of the International Conference on Computer and Communication Engineering.

[14] Martinez-Diaz, M. and C. Martin-Diaz, *"A comparative evaluation of finger drawn graphical password verification methods*", 12thinternational conference on frontiers in handwriting recognition 2010 Spain.

[15] Hung-Min Sun, Yao-Hsin Chen, and Yue-Hsun Lin, " *oPass: A User Authentication Protocol Resistant to Password Stealing and Password Reuse Attacks*", IEEE Transactions on Information Forensics and Security 7 (2), 2012.

[16] Safdar, S., M.F. Hassan, M.A. Qureshi, R. Akbar and R. Aamir, "*Authentication model based on reformation mapping method*", International Conference on Information and Emerging Technologies (ICIET), 2010.

[17] Emidio de Oliveira e Silva, Wallace Thierre Souza de Lima, "*Authentication and the Internet of Things*", ICSEA 2017: The Twelfth International Conference on Software Engineering Advances.

[18] Divyans Mahansaria, Samarpan Shyam, Anup Samuel, Ravi Teja, "*FAST AND SECURE SOFTWARE SOLUTION [SS7.0] THAT COUNTERS SHOULDER SURFING ATTACK*", 13th IASTED International Conference Software Engineering and Applications, Cambridge, USA, 2009.

[19] Mohammed A. Fadhil al-husainy, Diaa mohammed uliyan, "*A smooth textual password authentication scheme against shoulder surfing attack*", Journal of Theoretical and Applied Information Technology, Vol.96. No 09, 2005.

[20] Kasar Santosh R. Baig Arfan J., Gunjal Vishal S, Pawar Atul J, Dhokane Rahul M, "*Prevent shoulder surfing using graphical and duo letter authentication*", IJARIIE-ISSN (O)-2395-4396, Vol-2 Issue-2 2016.

View publication stats