PassPage: Graphical Password Authentication Scheme Based on Web Browsing Records

Xian Chu[1], Huiping Sun[2], Zhong Chen[2]

[1] School of Software & Microelectronics, Peking University, China
xavierchu@pku.edu.cn
[2] School of Software & Microelectronics, Peking University, China
{sunhp,chen}@ss.pku.edu.cn

Abstract. This paper proposes a two-factor graphical password authentication scheme, PassPage, which is suitable for website authentication with enhanced security. It leverages the implicit memory based on the user's web browsing records. Whenever the user tries to log in, the server returns 9 small pages as a challenge, and asks the user to select all the pages the user has browsed besides inputting a text password. We performed user experiments on 12 volunteers. The experiment results showed that the average login success rate on a news website is steadily over 80% when the users are familiar with the login process, and the login success rate does not decrease sharply in 6 days.

Keywords: PassPage, Authentication, Two-factor authentication, Graphical password, Implicit memory.

1 Introduction

Currently, most websites still use username and text password as a basic user authentication. Using only a text password is not safe, since the length and strength of text passwords that users can remember are limited, and they could be obtained by hackers using guessing attacks, shoulder surfing attacks, dictionary cracking attacks, or other attacks. Some websites use dynamic codes via phone messages or emails as a supplementary authentication. This kind of two-factor authentication is most successful, but it relies much on other devices or accounts, and it takes much time for users to receive the codes and input them. Graphical password authentication can be another alternative authentication factor which is more user-friendly. However, most existing graphical password authentication schemes cannot get rid of setting and remembering secrets. They put much memory burden on users if used on multiple websites.

We aim to propose a new authentication scheme which does not put more operation burden or memory burden on users. Our idea is to utilize the user's knowledge related to the website. It should be remembered naturally by the user when the user browses web pages. It should have a high entropy too. On many websites, including news sites,

Xian Chu, Huiping Sun, Zhong Chen, PassPage: Graphical Password Authentication Scheme Based on Web Browsing Records, Proceedings of AsiaUSEC'20, Financial Cryptography and Data Security 2019 (FC). February 14, 2020 Kota Kinabalu, Sabah, Malaysia Springer, 2020.

social networking sites, video sites, forums and blogs, the pages the user browses meet our needs.

In this scheme, the website leverages page scripts to record the user's browsing history automatically and the website server saves it for a period of time. Whenever the user tries to log in, the server returns 9 small pages as a challenge, and the user needs to select all the pages he or she has visited. Besides, the user needs to input a text password, which is served as another factor. This kind of two-factor authentication scheme is more secure than traditional text password authentication scheme while not increasing much memory burden. We call this scheme PassPage because the graphical passwords are taken from web pages and displayed as web pages.

In order to test the usage performance of PassPage, we developed an experimental authentication system and conducted user experiments on 12 volunteers. We developed the Chrome browser extension to record users' browsing history automatically, and wrote simulated sign-up page, login page and reset password page, so that all experimental data is sent to our experimental server, without bringing any change to the source codes of actual websites. The results showed that the average login success rate on a news website is steadily over 80% when the users are familiar with the authentication process, and the login success rate does not decrease sharply in 6 days.

2 Related Work

Our work mainly deals with two-factor authentication, combining text password authentication with graphical password authentication. There have been numerous researches working on graphical password authentication [1-13], including recall-based mode and recognition-based mode. Some researchers proposed novel graphical authentication schemes [2, 3, 6, 8, 10]. Some researchers worked on enhancing security of graphical authentication [7, 8, 9, 12, 13], especially resisting shoulder surfing attacks.

In order to enhance the security of graphical password authentication, users have to remember high-entropy secrets. But we do not intend to put more burdens on users' memorability. Thus, the proposed graphical authentication scheme is based on implicit memory. There have been some researches talking about authentication based on implicit memories [14-18].

Tamara Denning et al. first proposed the idea of implicit memory for authentication [14]. The author believes that a good implicit memory authentication scheme needs to meet the following conditions: the secret can remain in the brain for a long time; the registration process and authentication process do not require the user to remember something purposely; the secret is random and has a high entropy; the process of forming the memory and the process of authentication are different. The authors proposed an implementation, but the experiment did not perform very well: the correct rate of users who have seen the complete graphics is only 7% higher than those of users who have not seen the complete graphics. Also, this scheme consumes much time for users to study it.

Sauvik Das et al. conducted a detailed investigation and analysis of people's daily memory [15]. Among the 2,167 daily problems coming from mobile phone data, only

1381 problems (about 64%) were answered correctly by the experimental users, most of which are recognition problems, and the correct rate is not affected by time. According to the survey, users' answers to social-related issues (such as calls and text messages) are much more accurate than those related to mobile phones usage (such as usage time and duration). The downside is that the authentication takes too long, on average more than one minute, so it only works when password authentication is not working properly.

The PassApp proposed by Huiping Sun et al. built a mobile phone unlocking system based on users' familiarity with the applications installed on their mobile phones [16]. The system randomly selects 4 apps installed on the phone and 12 apps not installed, and puts the icons of these 16 apps together in a random order. Users need to select the icons of all 4 installed apps to unlock the phone. This system works well for ease of use and security. On average, testers can remember 89% of their applications on their phones, with a success rate of 95%. However, this scheme can only be used on mobile phone unlocking.

The PassFrame proposed by Ngu Nguyen et al. uses a wearable miniature camera to record anything the users see in their daily lives [17]. When user authentication is required, some images are taken from the video for the user to select or sort. It is a flexible authentication scheme using implicit memory, which confirms the possibility that what users have seen can be used as passwords. But it is not practical as it is extremely privacy invasive.

The LEPs proposed by Simon Woo et al. is a textual password for asking questions about life experiences [18]. Users need to pre-set some facts about life experience, including birth, party, graduation, wedding, travelling and other aspects, and enter the answer to the question in text when logging in. The system authenticates the user through fuzzy matching. LEPs are 30-47 bits stronger than traditional 8-character passwords. Force attacks and dictionary attacks are almost ineffective, and the possibility of being guessed by friends are only 0.7%.

All these schemes utilizing implicit memory do not utilize the user's knowledge related to the website. Unlike these schemes, PassPage is suitable to be used for website authentication, and the secret pool keeps growing when the user is browsing.

3 Design of PassPage

PassPage mainly consists of four system modules: the sign-up module, the browsing history recording module, the decoy web pages maintenance module, and the login module. The order relation of them is shown in Fig. 1.

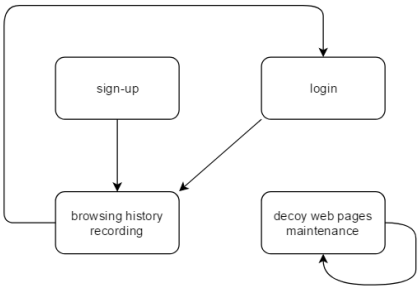

Fig. 1. Order relation of system modules

3.1 Sign-up Module

The user submits the email address, username and password to sign up an account. The email address is necessary in case the user forgets the secrets. The information is sent to the server. If the information is legitimate, the server saves it into the database and returns a session to the client. Then the client automatically logs in to the website.

3.2 Browsing History Recording Module

Once the user logs in to the website, the browsing history recording module starts to work. There is a client script automatically running when the page is open. The script identifies the user by checking the session, and then records the user's browsing history. Some pages with high click rates and low recognition rates (such as the homepage and website information pages) are not recorded. If the page needs to be recorded, the script uploads its HTML content as well as the username to the server. The server saves the HTML content in a file with a random file name, and then stores the username and the file name in the user page database. After that, the file name is returned to the client.

Besides, if the page needs to be recorded, the script keeps recording the time of the user staying between scrollings in the page. When the user closes the page, the script sends the file name and the time array to the server. The server saves the total browsing time, the total scrolling count and the time array in the user page database.

3.3 Decoy Web Pages Maintenance Module

Since the server should return a challenge consisting of correct pages and decoy pages when the user logs in, the server must maintain a decoy web page database. The server should have access to all the pages on the website. It must request web pages once every day and save them in the decoy page database. The topics of pages should be as diverse as possible, but pages with high click rate and low recognition rate should not be saved. The server saves the pages in HTML files and stores file name, title, and adding time of each web page in the database.

If the number of decoy pages increases too fast, old decoy pages can be deleted at regular time. Assuming that 500 new decoy pages are added every day, then the server

find out the pages added 3 days ago but no longer than 10 days ago from the database, and randomly delete half of them. In this way, the number of old decoy pages will gradually decrease, but will not reach zero. There is a high possibility that several pages will retain permanently. The reason why the long-lasting web pages are not removed completely is that the server should return decoy pages with adding times close to those of correct pages when the user attempts to log in, so the long-time non-login user can see decoy web pages added long time ago. The page selection algorithm is described in the next section.

3.4 Login Module

The user inputs the registered username and password, and then clicks 'next step' button. The lower layer will display 9 small pages in 3*3 format. The user can scroll the lower layer up and down. Fig. 2 is one implement of login page. The small pages might be transformed beforehand.

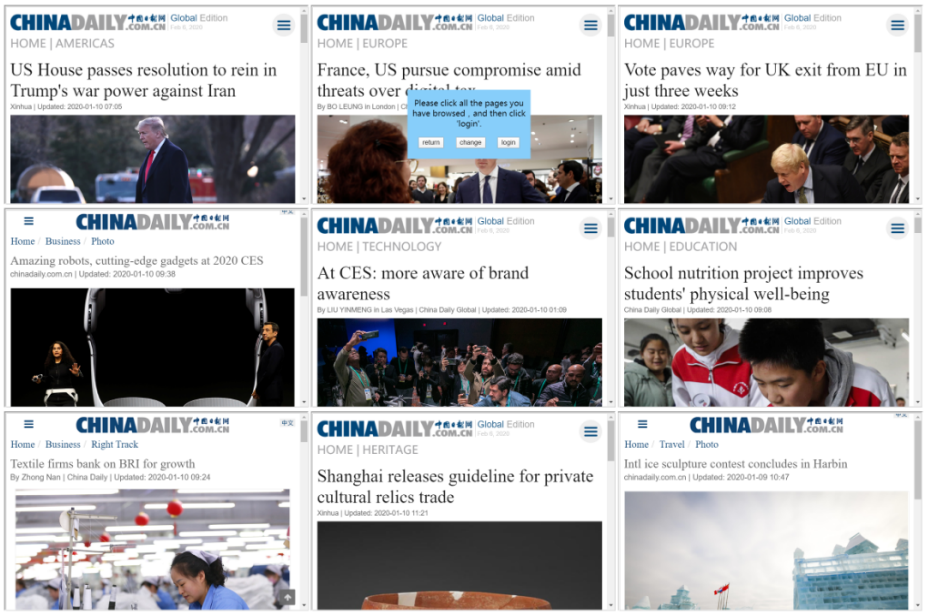

Fig. 2. One implement of login page (after clicking 'next step')

The user must select all the visited pages from the 9 pages by clicking on them. After selecting all the visited pages, the user can log in. If the graphical authentication is passed and the password is correct, the login is successful. If the login fails, the user can select the pages again or change them by clicking the 'change' button. Each user is given only three chances to log in. If the user still attempts to log in after failing three times, the server refuses it and the user has to reset the password through email authentication.

After the user clicks ‘next step’ button, the client first sends the username and the password to the server. Only if the password authentication is passed, the graphical authentication could start. The client sends the username to the server. The server returns the file names of 9 web pages (see Page Selection Algorithm), and then the client requests 9 HTML files from the server according to the file names. These 9 web pages consist of pages which have been visited by the user (called real pages) and pages which have never been visited by the user (called decoy pages), and the client will display them. When the user clicks ‘login’, the client sends the username and the file names of all selected pages to the server. If the authentication is passed, the server returns a session to the client.

Page Selection Algorithm. First, the server selects all the pages the user visited and puts them into the *allRealPages* set, including the file names, the titles, browsing logs and adding time, and gets the size of the set. If *size* = 0, the graphical authentication is invalid and the email authentication has to be used. If 1 <= *size* <= 5, the server randomly selects 1~2 page(s) from *allRealPages* as the real pages. If *size* >= 6, it randomly selects 3 pages from *allRealPages* as the real pages. The server should select real pages which are more likely to be remembered and recognized by the user. After that, the server takes out decoy pages from the decoy page database, ensuring that the titles of selected decoy pages are different from those in *allRealPages*, and the adding times of selected decoy pages are close to those of real pages. The server takes out 9 pages in total. All the pages are loaded from HTML files, which means the size of each page is only about 1 KB and all 9 pages can be loaded in a moment.

In order to prevent the adversary from constantly changing pages to find out the recurring real web pages, the server requires each user to log in before changing pages again. Based on this, the number of consecutive login failures of the user is limited. If the user fails for 3 times consecutively, the account will be locked and can only be restored by resetting the graphical password.

4 User Experiments

4.1 Experiment Procedure

In order to test the usage performance of PassPage, we developed an experimental authentication system. We developed the server using Java and wrote pages and Chrome browser extension using HTML, CSS and JS. The pages are simulated sign-up page, login page and reset password page. The Chrome browser extension is used to record users’ browsing history. We chose the news website www.sohu.com for test. All the experimental data is sent to our experimental server, without bringing any change to the source code of the actual website.

We sent a recruitment invitation in online chatting groups. Everyone could join our experiments regardless of age, sex, major or computer skill as long as he or she would have 10 minutes of free time in successive 6 days. Anyone interested is invited to sign on an experiment agreement. Finally, we received agreements from 23 volunteers.

Their ages are between 19 and 29. About half of them major in computer science or software engineering.

We wrote a detailed experiment guide on a web page, so that volunteers could follow it and do the experiment by themselves. On the first day, they downloaded and installed our extension in their Chrome browsers. Then they opened the simulated sign-up page, and registered test accounts. After that, they browsed the pages in www.sohu.com for several minutes. From the second day to the sixth day, they opened the simulated login pages and tried to log in with their test accounts. Everyone was asked to log in for 5 times every day. After that, they browsed the pages in www.sohu.com for several minutes. All the login processes were recorded by the system.

After completing the whole experiment, every volunteer filled in a feedback table, which contains seven statements about user experience. Every statement is followed by five levels of consent. Every volunteer chose one level of consent as a score for every statement.

Finally, nine volunteers completed their experiments. Each of them was paid 50 Chinese yuan as a reward. Three volunteers only completed part of their experiments. Each of them was paid 10 Chinese yuan.

4.2 Experiment Results

The volunteers performed 277 login trials in total. We first calculated the accuracy rates of text passwords and the success rates of logins for volunteers with more than 5 logins. The success of the two-factor authentication, requiring both the accuracy of text password and the success of graphical authentication, is regarded as the success of login.

Table 1. Password accuracy rates and login success rates

User ID	Total login count	Password accuracy rate	Login success rate
1	27	100.0%	63.0%
2	30	100.0%	93.3%
3	33	100.0%	93.9%
4	26	100.0%	96.2%
5	27	100.0%	77.8%
6	10	100.0%	50.0%
9	20	100.0%	85.0%
10	33	100.0%	93.9%
11	24	41.7%	29.2%
12	35	82.9%	68.6%

As is shown in Table 1, the login success rates varied a lot on different volunteers. The login success rates of 4 volunteers are over 90% while those of 2 volunteers are not more than 50%. Through our investigation, User 6 did not notice that the number of real pages was always 3 and he chose 1 or 2 page(s) for several times, so his login success rate is only 50%. User 11 was so careless that he forgot his text password and inputted wrong password for many times, so his login success rate is only 29.2%.

We also calculated the average used time of inputting usernames and passwords as well as the average total used time of logins.

Table 2. Comparison between average used time of two authentication schemes

Average used time of inputting usernames and passwords	Average total used time of logins
7.519 s	27.120 s

As is shown in Table 2, the average total used time of logins is about four times than that of inputting usernames and passwords. The average time consumed for graphical authentication is 20 seconds. Compared with other graphical authentication schemes and the dynamic code authentication scheme, it is a reasonable value, though.

Then we calculated the success rates of graphical authentication as the interval between sign-up and login increasing. We grouped the login records by the interval between sign-up and login (called login interval), setting 24 hours as one day, and got the results in Table 3.

Table 3. Graphical authentication success rates by login interval

Login interval (day)	Password accuracy count	Login success count	Graphical authentication success rate
1	39	25	64.10%
2	47	37	78.72%
3	45	38	84.44%
4	44	40	90.91%
5	51	46	90.20%
6	31	27	87.10%

As we can see in Table 3, the success rate of the first day was abnormally low. It might be caused by the volunteers' unacquaintance with this scheme. From the third day to the sixth day, the graphical authentication success rate was steadily above 80%. We can infer that users can remember most of the web pages they visited in 6 days. We also calculated the success rates of graphical authentication as the number of real pages increasing. The result is that the success rate is still 88.89% when the number of real pages grows to 35~40.

In addition, we calculated the average staying time and the average scrolling count of recalled real pages and missed real pages. The staying time of a real page is the total time during which the page keeps open on the foreground. The scrolling count of a real page is the total count of the user scrolling the page. The results are shown in Table 4. We can conclude that it's easier for users to recall pages on which they stayed for longer time and scrolled for more times.

Table 4. Average staying time and scrolling count of recalled real pages and missed real pages

Page type	Staying time	Scrolling count
Recalled pages	23.178 s	5.56
Missed pages	14.842 s	4.97

At last, we collected the answers of feedback tables. The average score of every statement is shown in Table 5.

Table 5. Average score of every statement

Index	Statement	Average score
1	I'd like to use this scheme.	3.4
2	I think this scheme takes up lots of memory.	3.4
3	I think this scheme is very annoying.	2.6
4	I think this scheme is hard to use.	2.3
5	I think the login success rate of this scheme is acceptable.	4.1
6	I think there are big problems with this scheme.	3.0
7	I think this scheme can be used widely.	3.5

We find that our volunteers hold positive views in general. Most volunteers think the login success rate of this scheme is acceptable, and it is not hard to use it.

5 Widespread Use of PassPage

We developed the experiment system only for the news site www.sohu.com on PC browser. There remains much to do for widespread use on multiple websites and multiple platforms.

5.1 PassPage on Multiple Websites

PassPage can be used on multiple websites, but it should be modified for specific websites. The most important matter is to find out high-entropy knowledge for every user on every website. For instance, on a shopping website, it may be the goods the user bought or watched in detail. On a social networking website, it may be the posted photos or videos the user watched. On a job website, it may be the jobs the user applied for. On a bank website, it may be the financing products the user bought. These kinds of information are implicit memories that can be transformed to passwords in pages.

In fact, PassPage cannot be used on all websites, and will be less effective or successful in certain websites. But it is still an optional authentication factor on most websites if the difficulty of authentication is reduced.

5.2 PassPass on Multiple Platforms

PassPage can be used on multiple platforms too. Since the browsing records are bound with username and stored in the server, the user can log in on any platform such as PC browser, tablet software or phone APP. The website developers need to enable their software or APP to record users' browsing history on these platforms. They also need to develop suitable login interfaces for different platforms.

6 Security and Privacy

Our scheme enhances the security of password authentication. Let's just focus on the security of the graphical password authentication. Assume that an adversary already

knows a username and the corresponding password, and he tries to log in to the user's account through graphical authentication. The adversary only knows the number of real pages is among 1~3. There is no other clue for the adversary to pass the graphical authentication easily. So he randomly selects 1~3 pages with C(9, 1) + C(9, 2) + C(9, 3) = 129 choices, but only one choice is correct, so the possibility he makes a correct choice is only $1/129 \approx 0.775\%$.

For the sake of privacy, the website must come to an agreement with users in advance so that the browsing records can be collected legally. It should be ensured that only the website server has access to these records. Furthermore, users' browsing records should be encrypted or partly encrypted in the server database.

7 Conclusion and Future Work

This paper proposes a two-factor graphical password authentication scheme, PassPage, which is suitable to be used on website accounts. It leverages users' implicit memories of browsing web pages. The experiment results showed that the average login success rate on a news website is steadily over 80% when the users are familiar with the login process, while an adversary with a random selection only has 0.775% possibility to pass the graphical authentication, not to mention that the adversary has to input an accurate password. Our experiments also showed the login success rate does not decrease sharply in 6 days. It can be concluded that this scheme integrates usability, efficiency and security.

In future work, we will study more about user behaviors of browsing web pages so that the graphical authentication challenge can be more favorable for users. For example, the server should find out pages which are more likely to be remembered and recognized by the user. Besides, the login process still needs to be optimized to increase the login success rate.

References.

1. Biddle Robert, Sonia Chiasson, and Paul C. Van Oorschot. "Graphical passwords: Learning from the first twelve years." ACM Computing Surveys (CSUR) 44.4 (2012): 19.
2. Brostoff, Sacha, and M. Angela Sasse. "Are Passfaces more usable than passwords? A field trial investigation." People and computers XIV—usability or else!. Springer, London, 2000. 405-424.
3. Bianchi, Andrea, Ian Oakley, and Hyoungshick Kim. "PassBYOP: bring your own picture for securing graphical passwords." IEEE Transactions on Human-Machine Systems 46.3 (2015): 380-389.
4. Uellenbeck, Sebastian, et al. "Quantifying the security of graphical passwords: the case of android unlock patterns." Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. ACM, 2013.
5. Stobert, Elizabeth, and Robert Biddle. "Memory retrieval and graphical passwords." Proceedings of the ninth symposium on usable privacy and security. ACM, 2013.
6. Zhu, Bin B., et al. "CAPTCHA as graphical passwords—a new security primitive based on hard AI problems." IEEE transactions on information forensics and security 9.6 (2014): 891-904.

7. Gao, Haichang, et al. "A survey on the use of graphical passwords in security." JSW 8.7 (2013): 1678-1698.
8. Rao, Kameswara, and Sushma Yalamanchili. "Novel shoulder-surfing resistant authentication schemes using text-graphical passwords." International Journal of Information and Network Security 1.3 (2012): 163.
9. Renaud, Karen, et al. "Are graphical authentication mechanisms as strong as passwords?." 2013 Federated Conference on Computer Science and Information Systems. IEEE, 2013.
10. Khan, Mudassar Ali, et al. "g-RAT| A Novel Graphical Randomized Authentication Technique for Consumer Smart Devices." IEEE Transactions on Consumer Electronics 65.2 (2019): 215-223.
11. Mackie, Ian, and Merve Yıldırım. "A novel hybrid password authentication scheme based on text and image." IFIP Annual Conference on Data and Applications Security and Privacy. Springer, Cham, 2018.
12. Mokal, P. H., and R. N. Devikar. "A survey on shoulder surfing resistant text based graphical password schemes." International Journal of Science and Research (IJSR) 3.4 (2014): 747-750.
13. Gaikwad, Anagha. "A Survey in Shoulder Surfing Resistant Graphical Authentication System." International Journal of Emerging Technology and Computer Science 2.3 (2017).
14. Denning, Tamara, et al. "Exploring implicit memory for painless password recovery." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2011.
15. Das, Sauvik, Eiji Hayashi, and Jason I. Hong. "Exploring capturable everyday memory for autobiographical authentication." Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. ACM, 2013.
16. Sun, Huiping, et al. "Passapp: My app is my password!" Proceedings of the 17th International Conference on Human-Computer Interaction with Mobilbe Devices and Services. ACM, 2015.
17. Nguyen, Ngu, and Stephan Sigg. "PassFrame: Generating image-based passwords from egocentric videos." 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEE, 2017.
18. Woo, Simon, et al. "Life-experience passwords (leps)." Proceedings of the 32nd Annual Conference on Computer Security Applications. ACM, 2016.