Improving Web Application Security by Virtual Password Authentication

Kamthorn Sarawan

Computer Science Department, Faculty of Agro-Industrial, Kalasin University
Kalasin University 62/1 Kasetsomboon Rd., Muang Kalasin, 4600, Thailand
kamthorn.sa@ksu.ac.th

Abstract— **Developing web application that handles valuable information requires an authentication using user identifications (IDs) to control the system access. An ID may contain a username and password. One of the security issues that often occur is that the ID can be stolen by a malicious user for abusive purposes which negatively causes damage to the ID owner. Today, there are many techniques and methods presented to prevent the theft. One of them is to use virtual password. The researcher has proposed a new strategy to enhance system security using the principle of virtual password by creating a virtual character set to randomly matches the real character set and then the output is sent to the server for system-access right verification. The researcher tested the proposed strategy by developing a PHP-language system and analyzed the system security. The finding revealed that the system was secure and was able to prevent itself from various threats i.e. sniffing, phishing, key-logger, and shoulder-surfing. Finally, system performance of the proposed strategy was evaluated and compared against the traditional system. The comparison indicated that the proposed strategy was suitable for use on small to medium systems.**

Keywords— **Web Security, Session, Web Authentication, Virtual Password, Password Sniffing**

I. INTRODUCTION

Evolution of information technology and the internet has allowed many organizations to incorporate web applications into their operations to improve organizational efficiency and competitiveness. These systems are designed to be used over an internet network accessible anytime and anywhere for the convenience of customers. Examples of such system include; E-Mail, E-Banking, E-Document, E-Salary, E-Payment, E-Commerce, and other systems. The operational principle of these systems contains user identification (IDs) authentication, where the ID consists of username and password to prove the system access authentication. Since these web applications handle highly secure information and property, perpetrators tend to feel attracted and attempt steal the IDs through various abusive misconducts such as packet sniff [1] using simple tools where the attack infiltrates like a man in the middle [3] of the ongoing communication which intercepts the communication between user and the server. Other misconducts include fake webpage creation which lures the victim to input ID information (web phishing) [4], keystrokes logging or key-logger [5], and one of the simplest ways which is to literally sneak peek at the victim while inputting the ID data, shoulder-surfing [6]. Losing an ID may cause damages to the owner from information, reputation, to property. However, there are currently several ways to enhance security to prevent such theft e.g. virtual password [7], dynamic password [8], and HTTPS [9]. Nevertheless, each method still contains its own problematic limitations and conditions.

This study has proposed a new strategy to enhance security to system user using the virtual password principle and it is simple. What it does is that it creates a set of randomized virtual characters stored in a form of session variable then the set is used to match with the real password character set. Users input their password through the randomized virtual character set and then the input is used to match with real password from database. The designed strategy can be realistically developed and applied without requiring infrastructural upgrade nor addition installation budget. Later, designed system was developed and security analysis against various attacks was conducted. System performance was evaluated and compared with traditional system to indicate suitability for actual implementation.

II. RELATED STUDY

A. *Virtual Password*

Virtual password [7] refers to the data set input into the system to prevent revealing the real password. Such data was created through various measures such as encryption, hashing, and randomization and, depending on the method, when the data set is sent to the server, it is decrypted and reveal the real password which the server further uses to verify with its database.

B. *Session*

Web application development essentially requires the use of a variable called session [11] to temporarily store user data who is accessing the system. Session data is stored on the host while the session ID number is stored on both the client and host and it is used as data reference. Session usage is commonly found to store temporary data on circumstances like while users stay logged in, to store account information that logs into the system, or other cases unrelated to signing in such as website traffic analytics. Session variables can be applied in various web application

development strategies and suitable for temporary data storage in PHP.

C. Relevant Study

Many studies paid attention to security or password theft protection using the principle of virtual password as well as dynamic password. Most studies proposed strategies to enhance password protection. Some studies proposed untested algorithmic designs or designs too complex for web application [13,14,15,17,18] Some studies incorporated session technique in web application implementation and design however, their complicated designs had made them impractical. [17, 19] Some other studies were flawed due to being vulnerable to other attacks such as key-logger or phishing. While the study the researcher had earlier presented contained some drawback due to being only compatible with touchscreen computer [20].

III. DESIGN

This study employed the principle of virtual password with the help of session variable. For the purpose of testing, the system was designed using PHP-language. Operational steps were tested as explained in Figure 1.

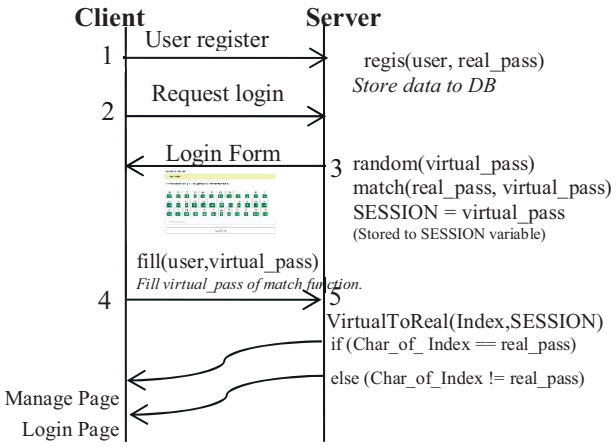


Fig. 1 Design of Virtual Password Authentication

According to Figure 1, operations could be explained as following:

A. Registration (Number 1)

User fills in the registration information as usual which is entering user account information via an HTML form. The system then records username and password in the database.

B. Sign-in (Number 2-3)

When the user access the login page, the system will generate a form prompting to input the username and password, however in the password section, the system will generate a table of character matching between real password from a-z and 0-9 and the virtual password character from a-z and 0-9. (Character variables can be increased however both sets should exactly match.) Every time the log-in form is generated, the virtual character set always randomizes its character positions while still matches those of the real password characters. The user is required to input the password through such randomized character set for instance, if the username is admin and the password is **abcd**, and assume that the randomization is generated as: | a,**f** | b,**3** | c,**y** | d,**b** | … | 9,**r** |, the user must enter **f3yb** as password.

C. Method of randomization (Number 3)

The conditions are firstly, the same character between two sets will never be matched for instance, | a,a | will never occur, and secondly, the used character will never repeat i.e. if a character is paired, it will never be used again in any other pair for example, | a,x | b,r | c,x | is not supposed to occur since x has already been randomly assigned to the first pair thus must not again be used in the third pair. Randomization method can be elaborated step-by-step as following:

1. k = random(virtual_pass)
2. if (virtual_pass[k] == real_pass[i]) new random (exclude virtual_pass[k]) k = knew
3. print real_pass[i], virtual_pass[k]
4. virtual_passrandom = virtual_pass[k]
5. unset(virtual_pass[k]) and GO 1
6. SESSION["virtual_pass"] = virtual_passrandom

The steps above can be explained below:

1. Position randomization employed mt_rand function to select the character position for the virtual character set.
2. If the random character matches the same character of the real password, execute re-randomization by excluding the previous character.
3. Match the randomized virtual character with the real character and display the result in the login page.
4. Create a variable to store position value for each randomized character.
5. Remove the already-used characters out of the virtual character set so that they are not used again.

D. Authentication (Number 4-5)

When a user fills out the username and password (the virtual password), the system sends both data sets to the server for verification. In terms of virtual password, the data is first to be decrypted into real password using character matching between the virtual and real ones. When real password is extracted, the system then checks against its database. For example, when a user enters a password, **f3yb**, the system compares each character of the obtained data back to the real password character set to find out which matches which. Assuming that the randomization at that time is generated as: | a,**f** | b,**3** | c,**y** | d,**b** | … | 9,**r** |, the decryption would result **abcd** as the real password result. Such result is then used to check against the data in the

database and if the data matches with the database the user is then redirected to the dashboard or otherwise returned to the login landing page once more.

IV. TESTING

Design test was conducted through a PHP web application environment where user data was stored in MySQL database. The test was designed by running the web application on Apache Web Server the following server specifications: CPU Intel Xeon E5606 2.13GHz 4 Core, 4Gb memory, with CentOS 7.0 64bit operating system installed. The designed strategy was tested for operational accuracy through user ID input, comprising of username and virtual password, during the login. In addition, the researcher has conducted a performance test of the system compared to a web application using traditional password input method using HTTP request simulation via Apache JMeter [12]. Users were simulated and the response time was measured. The test was conducted 9 rounds and each round was designed to generate 10, 20, 50, 100, 150, 200, 300, 500, and 1000 requests/second respectively. Each round was tested 10 times to find average response time. In terms of data sniffing/interception prevention test, the researcher employed Wireshark [2] with test diagram as displayed in Figure 2.

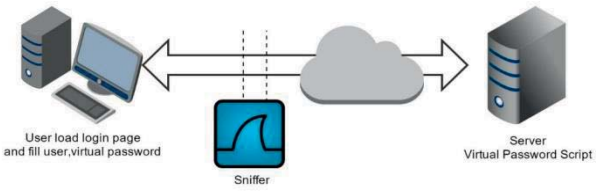


Fig. 2 Test Diagram

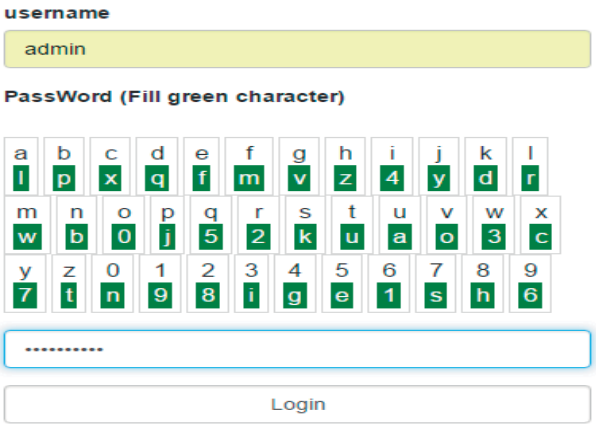
The figure indicates the system test design on data sniffing/interception which requests the login page handled by the host and then sniff the ID data using Wireshark program.

V. PERFORMANCE

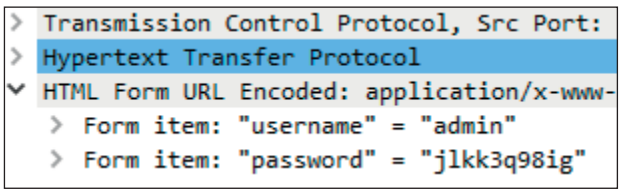
Design test was conducted through a PHP web application environment where user data was stored in MySQL database. The test was designed by running the web application on Apache Web Server the following server specifications: CPU Intel Xeon E5606 Performance results are divided into two parts i.e. firstly, system performance and security analysis against various attacks; sniffing, phishing, key-logger, and shoulder-surfing and secondly, the efficiency test of system performance when compared to the normal/traditional system.

A. Security Analysis

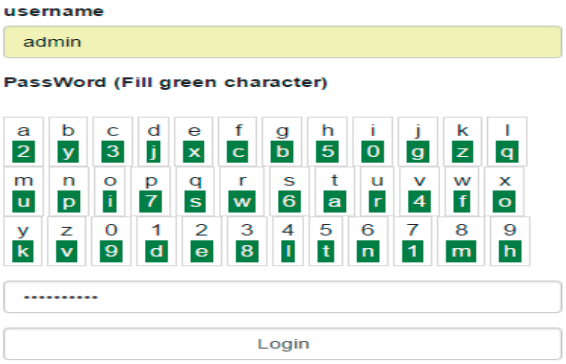
After a performance test of the developed system from the designed strategy, the test result indicated that the developed system could function properly i.e. the system was able to conduct authentication and the system could generate virtual passwords. During the test, the researcher created a test ID with the username: **admin** and password: **passwd1234** and experimented 10 times. The finding revealed that there was no duplication in virtual password generations. As exhibited in Figure 3.1-A and 3.2-A which were the first and second test sequence respectively, there was no duplication in virtual password generations. When the user input the username: admin and the password using the virtual password (green characters) by inputting the characters that matched real password (black characters) as previously registered. The results indicated that the user was required to input **jlkk3q98ig** during the first test and **7266fjde8l** during the second test. Once the user pressed login button, the system sent virtual passwords to verify user identity with the host which the host responded appropriately with access permissions. Since the system always compared the character set of real password with matched character position of virtual password and the data was stored in session variables, every time data interception test was conducted from the same ID data, it was found that the intercept data came in different values due to the fact that the intercepted data were the values of virtual password. The test results of password interceptions are exhibited in Figure 3.1-B and 3.2-B.



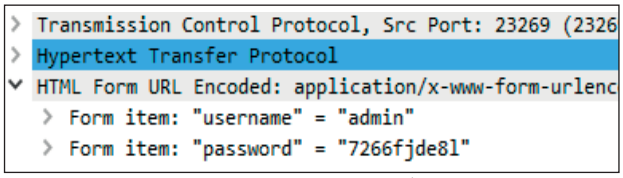
3.1-A) Login page 1st time.



3.1-B) Password Sniffing 1st time.



3.2-A) Login page 2nd time.



3.2-B) Password Sniffing 2nd time.

Fig. 3 Password Sniffing

Security analysis of test for vulnerabilities covers the following areas:

1) Sniffing: The developed and designed system could prevent theft from sniffing. Though sniffing could intercept virtual password data, the intercepted data was of no use since it did not help with further authentication due to the fact that each login generated new randomized character positions and matches between virtual and real passwords therefore there was literally no point intercepting such data for it being unusable nor decodable as the host always offered newly unpredictable randomization of matching patterns.

2) Phishing: The developed and designed system could prevent theft from phishing since the stolen password through this method is a virtual one and thus inapplicable in practicality and it is no use in decoding either.

3) Key-logger: The developed and designed system could prevent theft from key-logger since each login required the user to input different virtual passwords with uncertain randomized characters depending on generation pattern of each request.

4) Shoulder-surfing: Stealing passwords by sneak peeking through the shoulder-surfing works well with graphic authentication because the perpetrator can easily visualize the graphic however, when it comes to typing in keyboard characters especially with randomized required input where user is not to type different sets of character on every login makes it extremely hard for the perpetrator to memorize.

B. System Performance Compared to Traditional System

To compare the designed system with a regular one to find out whether the designed strategy is suitably practical, the researcher conducted an experiment simulating HTTP requests examine the response time. The result is shown in Figure 4.

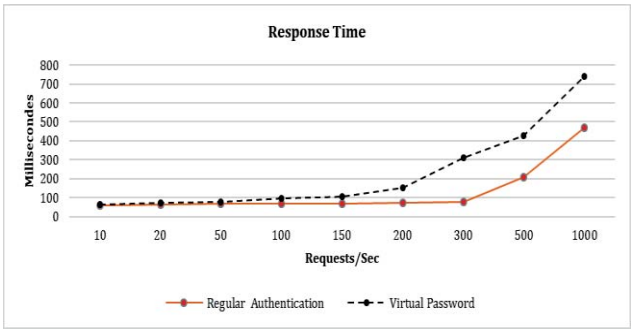


Fig. 4 Response Time

Figure 4 exhibits the comparative results of both system performances, the proposed system versus normal system. The finding indicated that response time of both systems were very similar when handling less than 100 requests/sec however when more requests/sec were simulated, the proposed system had longer response time than the normal system.

VI. CONCLUSION

This study proposed a solution according to the principle of virtual password to enhance security against password stealing through web application. The virtual password design employed matching method between real password and virtual password. Users were required to input virtual password instead of the real one and virtual password character locations were always randomized whenever the login page was requested. The system used session variables to store the randomized values each time which would later be used to track back and verify user identity on the server end. The researcher developed a system to test the proposed strategy using PHP language environment and the performance result indicated that the system was able to accurately authenticate. Further security analysis on various vulnerabilities issues was conducted i.e. sniffing, phishing, key-logger, and shoulder-surfing attacks and the finding revealed that the proposed system was able to prevent password theft from all the mentioned methods because each login, users were required to enter different virtual passwords, which made the system invulnerable from such attacks. Eventually, system performance test was conducted and the results were used in a comparison analysis between the performance of the proposed system and other regular system. The comparison indicated that the proposed system encountered longer response time when more requests/sec were generated since the proposed system was required to process the character location randomization and matching between virtual and real password characters as well as session back tracing. This can be concluded that the proposed system is more suitable to medium-sized web applications. Though, the proposed system may create difficulties to users during first uses for the requirement to always input virtual passwords, it is friendly to further development and actual implementation. In addition, it is secure against attacks due to password theft.

REFERENCES

[1] "Packet analyzer," *Wikipedia*, [Online]. Available: https://en.wikipedia.org/wiki/Packet_analyzer. [Accessed: 15-Sep-2017].
[2] "Wireshark Go Deep," [Online]. Available: https://www.wireshark.org/. [Accessed: 15-Sep-2017].
[3] "Man-in-the-middle attack – OWASP," [Online]. Available: https://www.owasp.org/index.php/Man-in-the-middle_attack. [Accessed: 20-Sep-2017].
[4] "Phishing – OWASP," [Online]. Available: https://www.owasp.org/index.php/Phishing. [Accessed: 25-Sep-2017].
[5] O. Zaitsev, "Skeleton keys: the purpose and applications of key loggers," *Network Security*, vol. 2010, no. 10, pp. 12–17, Oct. 2010.
[6] "Shoulder surfing (computer security)," *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Shoulder_surfing_(computer_security). [Accessed: 15-Jun-2017].
[7] S. Harris, *CISSP All-in-One Exam Guide*, 6th ed. New York: McGraw-Hill Education, 2012.
[8] X. Gao and P. Hu, "Dynamic Password Authentication System and Method thereof," US20070186115A1, 09-Aug-2007.
[9] E. Rescorla, "HTTP Over TLS," [Online]. Available: https://tools.ietf.org/html/rfc2818. [Accessed: 15-Jun-2017].
[10] "Session Management Cheat Sheet – OWASP," [Online]. Available: https://www.owasp.org/index.php/Session_Management_Cheat_Sheet. [Accessed: 17-Sep-2017].
[11] Lei M, Xaio Y, Vrbsky S V, and Li C C, "Virtual Password Using Random Linear Function for Online Service," *Computer Communications*, vol. 31, no. 18, pp. 4367–4375, Dec. 2008.
[12] "Apache JMeter - Apache JMeter™," [Online]. Available: http://jmeter.apache.org/. [Accessed: 14-Oct-2017].
[13] S. Kanagaraj, S. M. Javith Ibram, K. D. Madhan, and D. Rajkumar, "Differentiated virtual passwords for protecting users from password theft," *International Journal of Engineering Research and Science & Technology*, vol. 2, no. 2, pp. 94–100, May 2013.
[14] R. Balaji and V. Roopak, "DPASS - Dynamic password authentication and security system using grid analysis," in *3rd International Conference on Electronics Computer Technology*, 2011, vol. 2, pp. 250–253.
[15] D. Pansa, T. Chomsiri, "Dynamic Password Authentication: Designing step and security analysis," in *7th International Conference on Computing and Convergence Technology*, 2012, pp. 518–523.
[16] S. Prabhu and V. Shah, "Authentication Using Session Based Passwords," *Procedia Computer Science*, vol. 45, pp. 460–464, Jan. 2015.
[17] N. R. Rekha, Y. V. S. Rao, K. V. S. S. R. Sarma, "Enhanced Key Life in Online Authentication Systems Using Virtual Password", *Eighth International Conference on Information Technology: New Generations (ITNG)*, 2011, pp. 366–369.
[18] M. Lei, Y. Xiao, S. V. Vrbsky, C. C. Li, and L. Liu, "A Virtual Password Scheme to Protect Passwords," in *2008 IEEE International Conference on Communications*, 2008, pp. 1536–1540.
[19] Kamthorn Sarawan and Sarayut Kornwirat, "Dynamic Password using Session Technique for Web Application," *in 7th National Conference on Information Technology*, 2015, pp.259–264.