A Page Token Prototype of OpenID Single Sign-On (SSO) to Thwart Phishing Attack

Nur Haryani Zakaria, Wan Mohd Yusoff Wan Yaacob, Norliza Katuk, Hatim Mohamad Tahir & Mohd. Nizam Omar
School of Computing, College of Arts & Sciences, Universiti Utara Malaysia.
haryani@uum.edu.my

***Abstract*—Single Sign-on (SSO) authentication was introduced to overcome the problem of password memorability issue by enabling the users to login once using a set of username and password that allows an access into multiple websites. Among several SSO protocol, OpenID is said to offer flexibility and security. Unfortunately, the existing OpenID model is prone to phishing attack due to lack of countermeasures to ensure authenticity of OpenID provider. In view of the proliferation of phishing attack that exposed users to fraud website, information theft and unauthorized disclosure, this study attempts to identify and propose a suitable countermeasure in order to thwart phishing attack in OpenID environment. Therefore, this study intends to develop a prototype that implements Page Token in order to mitigate phishing attack. The findings revealed that the Page Token is possible to minimize the potential risk of phishing attack.**

***Index Terms*—OpenID; Page Token; Phishing Attack; Single Sign-On (SSO).**

I. INTRODUCTION

There is an enormous growth of digitalized applications in the websites today as people's reliance on the Internet technology continues to grow rapidly. Despite of these contributions, security issues become the key concern to the user. It resulted in every users must have their username and password to log in into various websites which consequently creates burden for the users to cope with memorizing too many passwords. This ranges from criminals activities including phishing attack, man-in-the-middle attack and eavesdropping in which the service providers and online business operators have suffered both brand reputation damage and financial losses [1]. In the same vein, the unwary users have lost thousands of dollars due to unhealthy disclosure of their credentials at this malicious site.

OpenID was introduced as a decentralized and user-centric protocol in Single Sign-On (SSO) system environment [2]. This protocol was chosen based on main characteristic of OpenID authentication that provide a single user identifier to log in into the websites that support this technology [3]. It started with user creating an identity at the OpenID provider, and then directed to link the identity to any of the relaying party that is available and use it later as the main key in order to authenticate the application. The existing OpenID protocol gives the permission to the users to log in and entering relaying party by giving authentication permission from an OpenID provider.

This enables OpenID to accelerate up the authentication process by allowing the users to sign in with a single click. It also reduces frustration associated with maintaining multiple username and password. This protocol also can gain greater control over user online identity. Despite the advantages as mentioned, there are several vulnerabilities and security limitations with regards to the implementation of OpenID such as spam, session hijacking, authentication bugs and phishing attack. Among these issues, phishing attack was highlighted because it was very dominant and has been focused by many researchers and practitioners [1].

In OpenID, phishing attack happened when users enter OpenID URL to the redirected website. The attacker will redirect the victims to the website in order to compromise Identification Provider (IdP) and acts as legitimate user. Phishing scams have been escalating in number of sophistication. It continues to rise in quantity and quality that captured for both client and server side. The cyber criminals use phishing attack to obtain private information whether by social engineering or technical subterfuge technique [4]. The existing OpenID model is said to experience this problem and hence prone to phishing attack.

The literatures have suggested a concept of using token as the possible countermeasure for phishing attack [1], [5]. Nevertheless, this proposed countermeasure did not specifically mention about the method of token in the domain of SSO. Based on the study, the possible token is intangible and should be embedded in the architecture of OpenID environment. Therefore, this study intends to propose a prototype, which introduces an intangible countermeasure to overcome these issues in order to thwart phishing attacks.

This study has been conducted in order to achieve the following objectives: (i) to identify the type of countermeasure that can minimize the issues of phishing attack, (ii) to develop the prototype by incorporating the identified countermeasure (iii) to evaluate the functionality of the prototype.

The following sections of this paper will discuss further on related work including some of the existing solutions. A section then follows it to detail out the existing OpenID model highlighting the weakness that prone to phishing attack. Next, the research methodology will be discussed followed by the prototype of OpenID model, which was produced in this study. Finally, the section ends with some insightful discussion, conclusion as well as some future work to be undertaken.

II. Related Work

The technology advances have encouraged more online transactions such as banking, shopping and many others. The main advantage of this is on the convenient aspects among users. In order to facilitate this scenario, users need to open an account and each of these accounts requires set of username and password. As the hype of online transactions becomes part of daily life, numbers of online accounts increases and thus resulted in users having multiple sets of usernames and passwords. This has created a burden on human memory to memorize a lot of username and password which is known as password fatigue [2]. In order to solve these problems, SSO authentication relieves the user from burden of having multiple credentials by improving the user experience and also the security [6].Unfortunately, the adoption of SSO is still not very encouraging due to many skeptical existed among users especially with regards to issues like spamming, session hijacking and not to mention the most popular being phishing attack.

Currently, the most effective countermeasures for phishing attack is highly depending on user awareness. Most of the users will be using the same username and password for all the websites. In this case, if any of the websites is malicious, the attacker can easily gain personal information and obtained the user credentials to gain access to the other websites that use the same IdP. Furthermore, when there is the fake website, users easily believe the website and directly login their username and password. Therefore, this countermeasure cannot solve the problem because the lack of knowledge and awareness that leads to phishing attack [7].

The second countermeasure that commonly used to overcome phishing attack is the Secure Socket Layer (SSL). This countermeasure is embedded in the OpenID protocol. However, the main limitation of this countermeasure is that SSL requires for certificate and approval to get linked to the certificate. It has slow data transmission according to the monitoring process and takes much time to complete encryption and decryption process [8]. This makes SSL insufficient to protect OpenID protocol from being compromised due to high cost to setup trusted infrastructure from the providers.

Another countermeasure that is used to overcome phishing attack is by using One-Time Password (OTP). It is also known as dynamic password, which is an unpredictable serial of numbers generated by specified algorithm. In this countermeasure, one password only works one time to avoid account being theft. However, the vulnerability exists if the authentication server is compromised. The algorithm used to provide passwords is a possible point of attack. If the hacker collected one-time passwords for a period of months and built up a large sample, then it might be possible to guess the algorithm and seed data used [9].

These three countermeasures were prominent and classified as well known solution in order to mitigate the phishing attack. Besides that, there are several other possible countermeasures which are considered as non-prominent solutions. They are nonce, cookies and browser fingerprints. In this case, the countermeasures operate by web server, which can reset the connection if any network anomalies were detected. The system will register the web browser when the users request the services. The limitation of these countermeasures is the attacker can obtain the transmission by stealing the session ID of the user [10]. The following Table 1 summarizes all the existing countermeasures based on the literature review.

Table 1
The Summary of Existing Countermeasures to Overcome Phishing Attack

Countermeasures	Advantages	Limitations	Related Literatures
User awareness	Helping users recognize appropriately to the potential security concerns. Make the users aware of their computer; network, and mobile devices are vulnerable. Reduce the cost of control measures into system to prevent network attack.	The users are not alert about latest information related to network attacks. Time constraint to conduct security awareness training, display security awareness posters and reminder to the users. Lack of rules and regulation for each user.	[7],[11], [12],[13]
Secure Socket Layer (SSL)	Keep sensitive information sent across the Internet safely and only intended recipient can read it. Can gain the trust from the users using this protocol.	Remain high cost to setup trusted infrastructure from the providers. Downgrade the performance of server resources due to encryption process by the server.	[8],[13], [14],[15]
One-Time Password (OTP)	Only use one password to endless supply in a secure way. Dedicated into the device including mobile phones and computer.	Valid only once for authentication and may not be used twice. OTP was generated in randomized which have possibility to compromises the account.	[9],[16], [17]
Others (Nonce, Cookies and Browser Fingerprints)	Register the user system browser when user request service from web server. Web server will reset the connection if anomalies detected.	The attacker can obtain the URL, reset a user Transmission Control Protocol (TCP) connection and steal the user session ID.	[10],[18], [19]

From the above summary, the concept and characteristic that have not been covered of each solution will be improvise and implemented in the proposed countermeasure using the prototype in order to mitigate and thwart phishing attacks. The methodology undertaken for this study will be discussed in the next section.

III. METHODOLOGY

This study comprises of three main phases as summarized in Table 2 below. The first phase of begins with preliminary study where researcher critically analyzes the literature based on journals, articles, books and also some of the related research being done by previous researcher in the domain of OpenID, SSO authentication and phishing attack. The aim of this phase is to identify the problems, requirements necessary to solve the problems and also the limitation of the research. This phase has pointed out some of the principles, ideas and concepts that is required by the application of SSO authentication in more details. It has also identified Page Token as the possible countermeasure that can be used to mitigate the issue of phishing attack in OpenID.

Table 2
Software and Tools Requirements in Prototype Development

Phase	Activities Involved	Output
Phase 1: Problem Identification	Doing some literature, review of journal, article, book and related research and study.	The principles and concept that will require and applied in this study such as SSO in general, OpenID and Phishing attack.
	Identify the problem statement of this study.	The problem statement, research objective, research question, scope and significant of the research.
	Understand the requirement and limitation of the study	The requirement and limitation of the study.
Phase 2: Construct and develop prototype	Identify the tool that can be used for combining page token into OpenID.	The conceptual idea of proposed countermeasure. Software and hardware requirement.
		The implementation of conceptual idea in the proposed mechanism.
Phase 3: Testing	Testing the prototype and monitor the development process.	The evaluation of prototype implementation and verification process.

Based on the outcome of phase one, it is then followed by the second phase that is developing the prototype that incorporates the proposed countermeasure. This phase was intended to implement the conceptual idea of the proposed countermeasures based on the requirement that have been finalized in the phase one. The outcome of this phase is a prototype that implements the Page Token as the countermeasures to mitigate the phishing attack. The final phase of this study involved testing procedure to verify the functionality of the prototyped developed. This process will involve testing the prototype that is expected to produce a unique token every time the user executes the authentication process. The verification process will be implemented based on unique token and the type of authentication process. As a deliverable output, this phase will determine whether the proposed countermeasure is able to function well in order to mitigate the possibility of phishing attack.

The software and tools requirement that was involved in the prototype development phase were itemized in the following Table 3.

Table 3
Software and Tools Requirements in Prototype Development

Components	Software and Tools
Operating System	Windows and Macintosh
SSO Tools	Google Platform (Google Console Developer)
Programming Platform	Relying Party (RP) (Salesforce.com)
Web Browser	Internet Explorer

The Google Console Developer and Salesforce.com were chosen in this prototype according to the features that can be integrated with OpenID authentication that embedded with Page Token in order to thwart phishing attack. These tools were integrated with core architectural elements such as multitenancy, security, performance, availability, meta-data customization and also integration via web services [20]. The next section will elaborate further on the architecture of the prototype.

IV. ARCHITECTURE OF THE PROTOTYPE

The previous study discovered a vulnerability that occurred between user and IdP whereby the intruders can interrupt the transaction using phishing attacks. The highly dependency on this single set of username and password makes it vulnerable and prone to attacks particularly phishing attack [21]. In order to mitigate the phishing attack and ensure a secure communication between user and IdP, the prototype has implemented a countermeasure known as Page Token that acts as second credential to the existing set of username and password.

This study introduces the concept of Page Token as a countermeasure that allows the authorized access to the system and services. This countermeasure was differing from others in term of method and architecture. It is applying the concept of intangible token in the form of page that carries the additional embedded code. Although the concept of token considered is not new and has been implemented previously, there is less of implementation and specific verification that related to this countermeasure in OpenID environment. In our study, the concept of Page Token will be implemented in the domain of OpenID and considered as improvise countermeasure in the sense of the token architecture and prototype in order to mitigate phishing attack.

This countermeasure provides an extra level of assurance based on an approach known as two-factor authentication. It may look similar like Transaction Authorization Code (TAC) that sends via mobile phone in order to complete the bank transaction. But in this process, we are using email transaction in OpenID environment due to the architecture and SSO characteristic that pertaining to the authorization and

authentication process.

The user has a common password, which authorized them to login the system. Using Page Token, the security code will be sent to the user in order to complete the login purposes. The security code is used to prove the user's identity and change frequently due to request from the user. It has 6 character of alphanumeric and can be combining with string format that automatically generated. Each token was unique to the specific user, cannot be re-use and case sensitive.

Figure 1 illustrates the architecture of the prototype. The normal transaction in openID environment will begin with requested resources by the user, followed by response given by IdP with OpenID authentication protocol. The user will send back the valid credential that is validated by IdP. Next, the coding will post by the user and finally the Relying Party (RP) will grant the access and permission to the user in order to access the system.

In this study, the concept of Page Token was implemented in step 4 onwards (refer Figure 1 below) whereby the phishing attack was detected due to lack of mutual authentication between user and IdP. The email that contained of security code was sent to the user in order to authenticate the system and execute all the process between IdP and RP. The process was performed separately with transaction of OpenID authentication protocol to the user in order to increase the security and reducing the chance of been hacked by unauthorized user. Upon receiving the security code, the users have to combine with common password in order to login to the system.

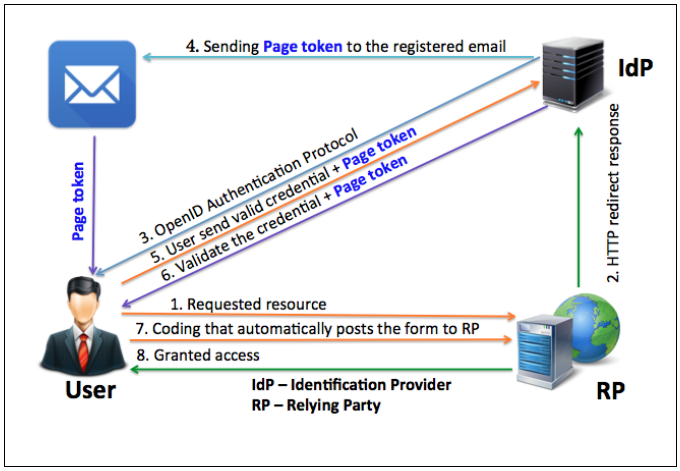

Figure 1: Architecture of the Prototype

In the context of implementation, this countermeasure attempt to filter the fake website by requesting the security code as a second credential to the user. If the website have been counterfeiting by third party in order to perform phishing attack, the system will fail to send the security code to the user's email. At the end of this process, the authentication will be aborted and terminated due to prevent the system from being access by the unauthorized user.

V. Prototype Development

Prototype development section will be elaborate accordingly to the architecture of the prototype in the previous section. In order to implement the architecture of prototype that proves the Page Token mechanism, we have chosen the Google Console Developer and Salesforce.com as the tools and programming platform to complete the task.

Figure 2 and Figure 3 shows the screenshot of WebTest development in Google Console Developer and the sample of login page in Salesforce.com that have been customized in order to develop the Page Token prototype in OpenID environment. In this case, the user has two options to complete the authentication process, whether to use the conventional method using username and password or using OpenID approach that connected to Gmail account that implement the concept of SSO. The development process is simultaneously operated between user, RP and IdP starting with the user request for authentication in Salesforce.com. Next, RP will response with point redirects into the web and launch the OpenID authentication protocol.

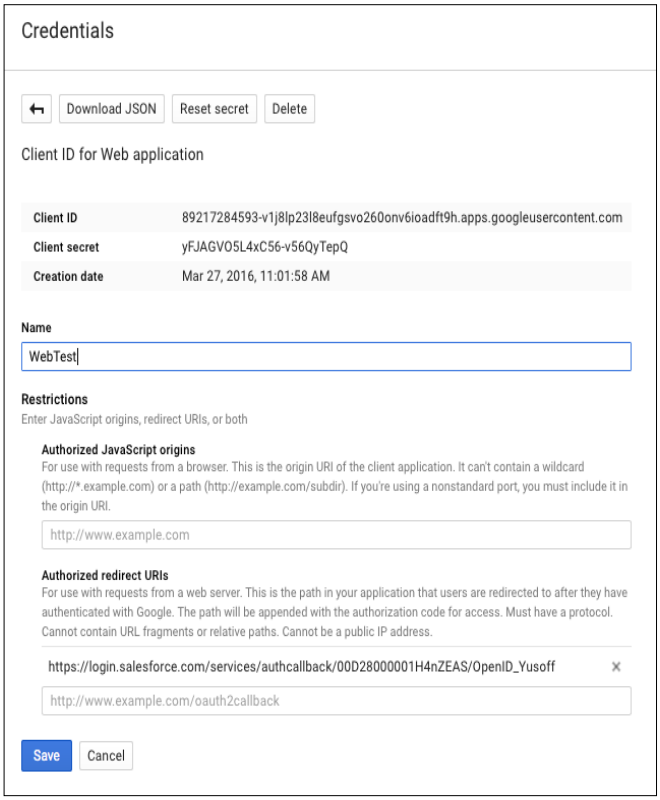

Figure 2: Web Test Development in Google Console Developer

In order to complete this process, The domain setting in Salesforce.com need to be configured and filled as shown in Figure 4. This section is required in order to create the relationship between Google Console Developer and Salesforce.com platform in OpenID environment using Gmail account for the middle communication.

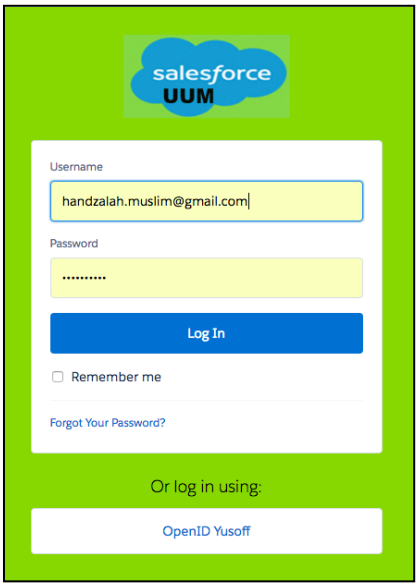

Figure 3: Login Page using OpenID

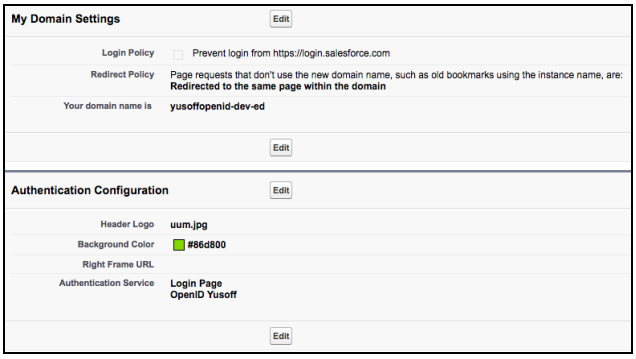

Figure 4: Domain Setting in Salesforce.com

The authentication provider also has been setup as shown in Figure 5. The setting was proposed in order to prepare the Page Token for mitigating the phishing attack. Basically, this section provides two functions, one for creating the authentication provider that connecting to Google Console Developer and Salesforce.com and the other are related to the process of determining the token and user endpoint based on provided consumer key. The Gmail account was chosen as Email OpenID Profile due to Google requirement and specification. In the end of this process, the default Salesforce Configuration URL will be display for initializing process that consist of test, SSO and callback pages which will insert into WebTest development in Google Console Developer setting.

In order to give the privilege to the legitimate user, we have managed the user and give the authorization based on profile and type of user. Figure 6 illustrates the process of user management according to registered email.

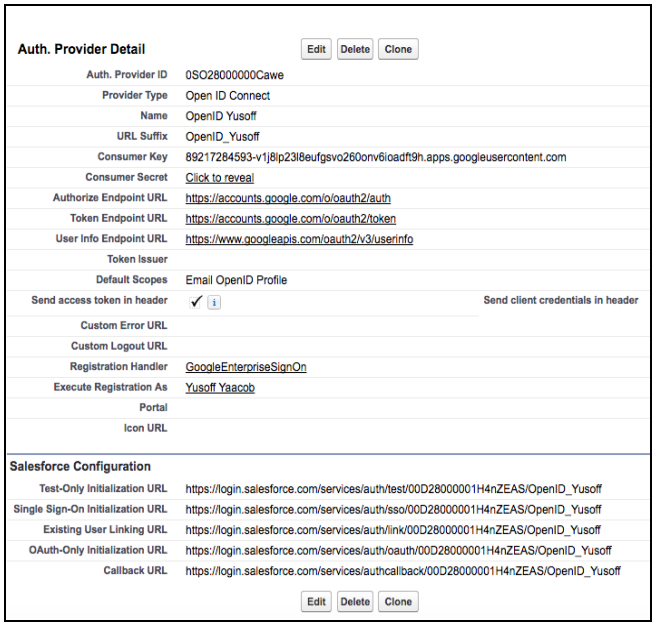

Figure 5: Authentication Provider Setting in Salesforce.com

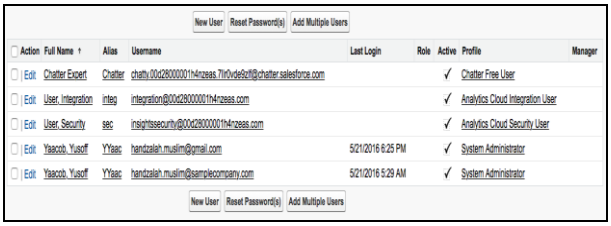

Figure 6: Managing Users in Salesforce.com

This process used to identify and control the state of users logged into the system and reduce the risk of phishing attack. This filtering method was implemented and unauthorized users are not allowed to authenticate the system. The following

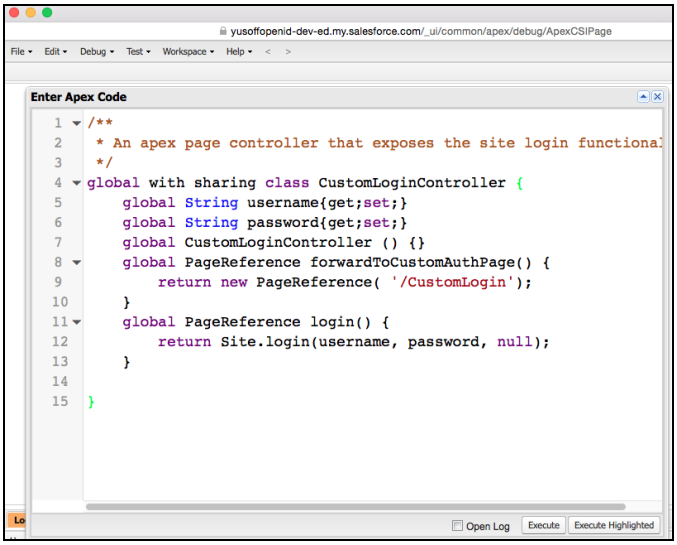

Figure 7: Sample of Apex Coding for Authentication

Figure 7 indicates the sample of code that embedded in the Apex Code to show the flow of the process in implementing the Page Token as a second credential that communicates with Gmail. The Gmail account was function as an instrument to retrieve the security code, which sending through email as a second credential. In this case, the prototype has been programmed to communicate with the email and the Page Token will be sending directly to the user. During the transaction, some information required for generating the token is hidden. If the user changes the password of the email, the Page Token also will we change.

Figure 8 shows the content of security code in Page Token that was send by the Salesforce.com to the user through Gmail. The proposed mechanism implements the unique token and has a different code for every request. Thus, the risk of missing token is limited within different request. The risk also can be reducing by giving token expiration time in order to avoid unauthorized people being access to the code. This second credential is very important to face any intercepted in the middle phishing attack in the unsecured environment. The server has capability to recognize if the request came from authorize or unauthorized user by using this mechanism.

Figure 8: The Content of Verification Code in Page Token

The verification code has shown in Figure 9 that was requested to the user in order to complete the process of authentication. This code will used as security code and has been sent to the user as a Page Token through Gmail account. The user is required to enter the code and verify the system to finish the task. Furthermore, Salesforce.com as an IdP will validate the credential and the coding will automatically post to the RP. The main page will be show to the user as granted access after successful login, as shown in Figure 10.

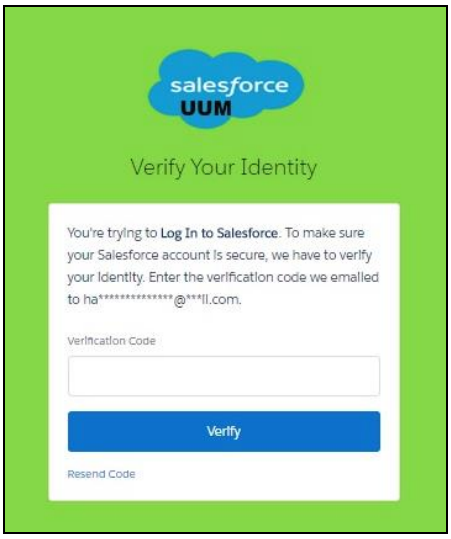

Figure 9: Verification Code through Page Token

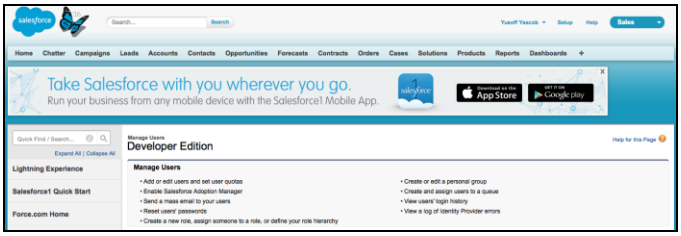

Figure 10: The Main Page after Successful Login

VI. Prototype Testing

In general, we have conducted several preliminary tests on our prototype. The testing approach was implemented based on two methods; Fake website with OpenID and without Page Token, and fake website with OpenID together with Page Token. The phishing attack was based on spam email phishing technique. With this technique, the phishers may send the same email to millions of users, requesting them to fill in and complete the personal details related to Salesforce.com. The information details will be used by the phishers for their illegal activities such as unauthorized and confidential data disclosure.

This common phishing scam messages may have an urgent note that requires the user to enter credentials to update account information, change details, and verify accounts. Sometimes, they may be asked to fill out a form to access a new service through a link that is provided in the email. The sample of spam email phishing technique was show in the following Figure 11.

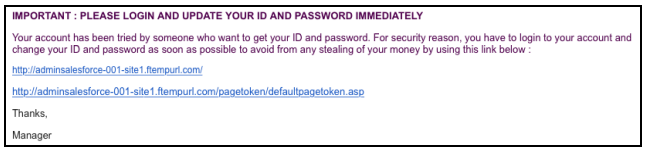

Figure 11: Sample of Spam Email in Phishing Technique

In this simple test by the administrator, we are trying to login to the system based on the link that attached in the email. We will be redirected to the fake Salesforce.com website and login using conventional techniques which requires username and password. We also try to complete the authentication process using OpenID option in the related counterfeit website. The OpenID option in the fake website need for Google login even though we have login the Gmail account in previous session. Overall, this experiment was conducted without security awareness in order to test the functionality of the Page Token.

From this experiment, the result shows that the Page Token can mitigate the phishing attack and blocking an authorized user to access the website. The Figure 12 shows the message for invalid username, password and security token that is refer to security code that was show in the end of this experiment.

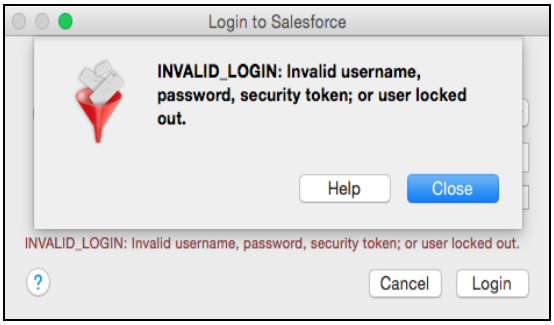

Figure 12: Message for Invalid Username, Password and Security Token

And finally, the message shows the error of the authentication as shown in Figure 13 that determined unauthorized user could not penetrate the prototype if they were supported by Page Token as second credential in authentication process.

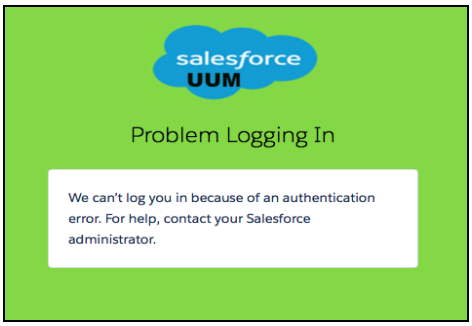

Figure 13: Error Message when Penetrated by Unauthorized User

Both experiment of phishing attack shows the same result that unauthorized user cannot complete the authentication process due to Page Token mechanism that was implemented in the prototype specifically in OpenID environment.

VII. Discussion

This study has developed a prototype to implement the proposed countermeasures for mitigating the problem of phishing attack known as Page Token. The objective of having this proposed countermeasure is to secure the communication between user and RP due to the phishing attack is proven possible to happen. In general, the prototype developed has shown that the Page Token countermeasures function well and to certain extent manage to mitigate the phishing attack problem.

From the prototype development and testing process, we have identified that using the combination of password and security code in OpenID environment are more efficient and secure rather than using single ID or password. It secures the credential and can act like additional security layer in the prototype. It also can be integrated in other platform and help the user to gain trust when reaching unsecure site and webpage.

On the hand, this countermeasure has fulfill the requirement mentioned by Lee [5] in their research regarding to new anti-phishing method in OpenID whereby the token approach was suggested in order to prevent phishing attack. Besides phishing attack, the researcher also mentions that the concept of Page Token is also possible to mitigate the problem of session hijacking. Furthermore, this suggested method easily realizable and can protect from phishing attack that do not require the third certificate from a third party.

The prototype development also has demonstrated that the Page Token is successfully implemented in the mutual authentication environment that is between user and RP. During the process of SSO in OpenID, lack of verifying process between User and RP will cause the phishing attack.

Based on the prototype developed, this study also acknowledges several limitations. First, the Page Token was tested to mitigate phishing attack that is caused by the email spams. Thus, as mentioned in the literature [22], phishing attack may exist in several other are techniques which requires further testing to confirm the suitability of the proposed countermeasures to mitigate the problem. Until then, our claim for this proposed countermeasure suit for phishing attack only. Secondly, we come to realize that the element of user awareness still plays an important role in mitigating this phishing attack. Although the proposed countermeasure can be put in place to mitigate this problem, users can still fall as prey of this vicious phishers who normally resort to social engineering technique.

VIII. Conclusion

Phishing activities have indeed reached epidemic proportions with the increasing availability to perform the attack, which simplifies the creation of fake websites. In this paper, the concept of Page Token as a countermeasure was presented and discussed using related prototype. Our architecture detected the weaknesses of OpenID transaction that is prone to phishing attack and tries to create a solution by adding the second credential technique. This is achieved through the implementation of Page Token that can strengthen the transaction and prevent interruption between user and RP in order to mitigate phishing attack.

However, this countermeasure needs to be tested with the various type of phishing attack due to the process that only can solve the common and certain scenario of phishing attack. Since the phishing attack become more sophisticated, the multiple solution that related to this countermeasure should considered and need a further evaluation in order to mitigate most of phishing attack. Future work will consider the third credential using encryption technology specifically in public and private key in order to thwart phishing attack. Moreover, we will determine the effectiveness of Page Token that can integrate with encryption technology.

Acknowledgment

This research was supported by Fundamental Research Grant Scheme (FRGS - S/O Code: 13143) from the Ministry of Education (MoE) Malaysia. The content of this article is solely the responsibility of the authors and does not necessarily represent the official views of the MoE, Malaysia.

REFERENCES

[1] H. Oh and S. Jin, "The Security Limitations of SSO in OpenID," pp. 608–611, 2008.

[2] P. Sovis and F. Kohlar, "Security Analysis of OpenID," *Sicherheit 2010*, vol. 170 of LNI, no. GI, pp. 329–340, 2010.

[3] M. Uruena and C. Busquiel, "Analysis of a Privacy Vulnerability in the OpenID Authentication Protocol," *IEEE Multimedia Communication and Services*, 2010.

[4] N. Mavrogiannopoulos, A. Pashalidis, and B. Preneel, "Toward a Secure Kerberos Key Exchange With Smart Cards," *IEEE Multimedia Communication and Services*, vol. 13, no. 3, pp. 217–228, 2014.

[5] H. Lee, I. Jeun, K. Chun, and J. Song, "A New Anti-Phishing Method in OpenID," pp. 243–247, 2008.

[6] A. I. Technology, "Enhanced Kerberos Authentication For Distributed Environment," vol. 69, no. 2, pp. 368–374, 2014.

[7] K. Solic, B. Tovjanin, and V. Ilakovac, "Assessment Methodology for the Categorization of ICT System Users Security Awareness," pp. 1560–1564, 2012.

[8] C. Science and S. Engineering, "Security through SSL," vol. 2, no. 12, pp. 178–184, 2012.

[9] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes," *Proceeding - IEEE Symposium Security and Private*, pp. 553–567, 2012.

[10] R. Upathilake, Y. Li, and A. Matrawy, "A Classification of Web Browser Fingerprinting Techniques," *New Technology Mobil Security (NTMS), 2015 7th International Conference*, pp. 1–5, 2015.

[11] J. Milletary, "Technical Trends in Phishing Attacks," *Technical Trends Phishing*, pp. 1–17, 2005.

[12] Q. Feng, K. K. Tseng, J. S. Pan, P. Cheng, and C. Chen, "New Anti-Phishing Method With Two Types of Passwords in OpenID System," *Proceeding - 2011 5th International Conference Generation Evolution Computing. ICGEC 2011*, pp. 69–72, 2011.

[13] R. Wang, S. Chen, and X. Wang, "Signing Me Onto Your Accounts Through Facebook and Google: A Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services," *Proceeding - IEEE Symposium Security and Private*, pp. 365–379, 2012.

[14] A. Alsaid, C. J. Mitchell, and S. S. L. Tls, "D912F511a302775848.Pdf."

[15] C. Huang, S. Ma, and K. Chen, "Journal of Network and Computer Applications Using One-Time Passwords to Prevent Password Phishing Attacks," *Journal of Network and Computer Application*, pp. 1–10, 2011.

[16] H. Wang, C. Fan, S. Yang, J. Zou, and X. Zhang, "A New Secure OpenID Authentication Mechanism Using One-Time Password (OTP)," *7th International Conference of Wireless Communication Network and Mobile Computing, WiCOM 2011*, pp. 1–4, 2011.

[17] J. Jeong, M. Y. Chung, and H. Choo, "Integrated OTP-Based User Authentication Scheme Using Smart Cards in Home Networks," pp. 1–7, 2008.

[18] B. Li, S. J. Lv, Y. S. Zhang, and M. Tian, "The Application Research of Cookies in Network Security," *Proceeding 2013 International Conference of Network Security and Technology Private Communication System, SNS PCS 2013*, pp. 152–155, 2013.

[19] U. Fiore, A. Castiglione, A. De Santis, and F. Palmieri, "Countering Browser Fingerprinting Techniques: Constructing a Fake Profile with Google Chrome," pp. 355–360, 2014.

[20] A. Saxena, S. Sengupta, P. Duraisamy, V. Kaulgud, and A. Chakraborty, "Detecting SOQL-Injection Vulnerabilities in SalesForce Applications," *Proceeding 2013 International Conference of Advance Computing and Communication Informatics, ICACCI 2013*, pp. 489–493, 2013.

[21] J. You and M. Jun, "A Mechanism to Prevent RP Phishing in OpenID System".

[22] X.-W. Huang, C.-Y. Hsieh, and C. H. W. and Y. C. Cheng, "A Token-Based User Authentication Mechanism for Data Exchange in RESTful API," *2015 18th Int. Conference of Network-Based Information System*, pp. 601–606, 2015.