Journal of Engineering and Applied Sciences
MEDWELL PUBLICATIONS SINCE 2000

Tabnabbing: The Attack That Exploits Human Wit

Norhaiza Ya Abdullah, Muhd Sham Ashraff Bin Maskan, Wan Hazimah Wan Ismail and Herny Ramadhani Mohd Husny
Universiti Kuala Lumpur UniKLMIIT, Jalan Sultan Ismail, Kuala Lumpur, Malaysia

Key words: Tabnabbing, phishing, client-side protection, Google Chrome, information

Corresponding Author:
Norhaiza Ya Abdullah
Universiti Kuala Lumpur UniKLMIIT, Jalan Sultan Ismail, Kuala Lumpur, Malaysia

Page No.: 3101-3106
Volume: 15, Issue 16, 2020
ISSN: 1816-949x
Journal of Engineering and Applied Sciences
Copy Right: Medwell Publications

Abstract: In this era of information system where information is highly regarded, never the less, the information obtaining involves human interactions both physically and electronically. People often browse the Internetthrough browserandthey mightcomeacross the attack known as phishing. As the technology and human creativity evolved, various methods also have been deployed to assist the phishing attack. One of the miscalled Tabnabbing attack by Rask in. Users used to browse using many tabs and may fall to Tabnabbing attack. The main goal of this project is to develop a solution for Tabnabbing attack in a form of web browser extension with the capability to detect and prevent user from falling for the attack. The project will cover only Google Chrome Browser and the prototype is developed by using JavaScript programming language, HTML and CSS. The prototype, named CTabs, implements the algorithm that captures the highlighted tab, comparing its screen shots of before and after the tab is switched, highlighting the differences and notify user through pop outs. As for the testing results, out of 4 tests, CTabs managed to pass all of them, detecting all of the Tabnabbing attack attempts. In short, CTabs managed to achieve the project objectives after the testing have been done.

INTRODUCTION

Social Engineering has always been the most powerful technique used by the malicious hackers in breaching the information security defenses, either of an organization or individual. Social Engineering can be defined as the attack that manipulates the human intelligence, making people submitting the information that should not be exposed to the unauthorized person. As from the words of a researcher,“Social Engineering is a collection of techniques used to manipulate people into performing actions or divulging confidential information[1].” .One of the technology-based methods of Social Engineering is the Tabnabbing attack. Aza Raskin presented in 2010, a new type of phishing attack which he dubbed as “Tabnabbing”. In this study, a countermeasure to combat “Tabnabbing” is presented, namely CTabs.

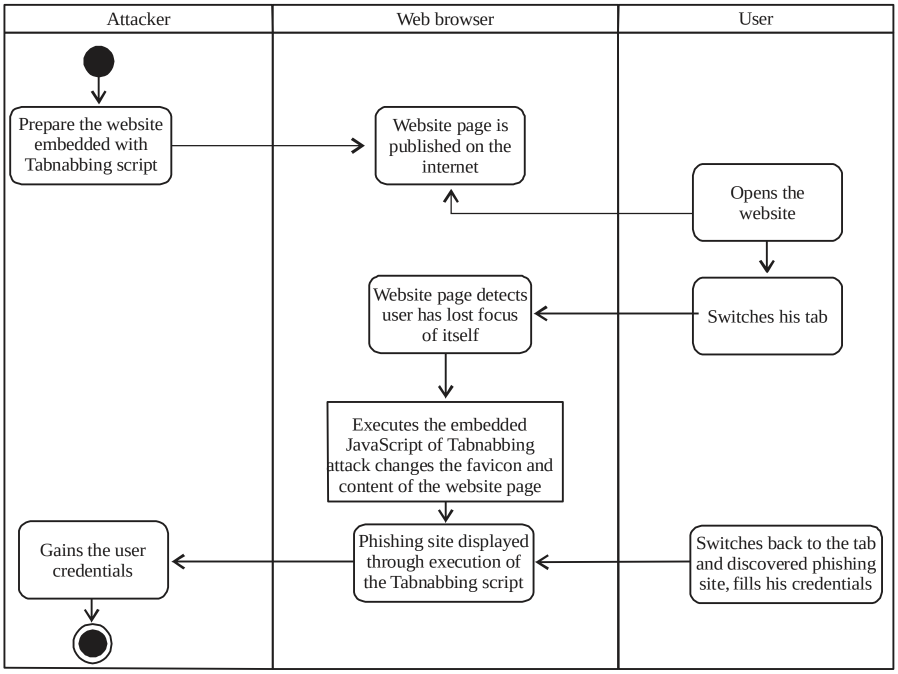

Attacker
Web browser
User
Prepare the website embedded with Tabnabbing script
Website page is published on the internet
Opens the website
Website page detects user has lost focus of itself
Switches his tab
Executes the embedded JavaScript of Tabnabbing attack changes the favicon and content of the website page
Gains the user credentials
Phishing site displayed through execution of the Tabnabbing script
Switches back to the tab and discovered phishing site, fills his credentials

Fig. 1: Visualization of Tabnabbing attack flow

MATERIALS AND METHODS

Anatomy of a Tabnabbing attack: In 2010, Aza Raskin dubbed anattack that will assist phishing greatly with a name of "Tabnabbing" which elaborated more[2] as Tab+Kidnapping. Firstly, Tabnabbing attack will be using tab mechanism which is widely used in all modern browsers. This attack targets in ternet users who used to open multiple tabs on their browser at once. According to Raskin, an example of successful Tabnabbing attack will start with:

- Victim opened the page of the link and discovered a normal with non-harmful looking page
- The page will detect that it has not been interacted with for a while after being left unattended for a certain moment as victim switch his/her focus to the other tabs
- The page will replace its favicon, title and its contents with the phishing site or malicious script using JavaScript
- Using the page's favicon and title visible by glancing at the tab, the victim will switch back to the page. Thereis a probability for the victim not tore-inspect the URL of the page
- If it is a phishing site, the victim will provide his/her credentials as he/she assumed that he/she has been logged out from the site. Otherwise, the malicious content of the page will be triggered and worked as the attacker intended
- After the victim has given the loginin formation and the page has sent it back to the attacker's server, the victim will be redirected the victim to the original site imitated by the phishing site (Fig. 1)

Existing solution techniques and solutions for Tabnabbing attack: In this study, some of the existing techniques and solutions for confronting Tabnabbing attack will be discussed. From the research by Hashemi and Sadat[3], there are generally two groups of technique can be used which are.

Script-blocking browser extensions: This provide protection against the script-based variant of the Tabnabbing attack. Browser extension that blocks scripts which are susceptive to perform malicious actions or violate the browser security policy. This protection is dependent on the default behavior of extensions towards preventing JavaScript code from execution on untrusted domains. These were stated by Hashemi and Sadat[3].

- Specific designed Tabnabbing detection and prevention techniques
- Recording of favicon and a screenshot of a webpage once it is visited for the first time and once after a tab switch event occurs. Page titles and favicons are recorded for each tab. Results are based on Threshold value
- Using anomaly detection techniques on heuristic based metrics for conducting the comparison with respect to syntactical similarity

Table 1: Comparison of the existing Tabnabbing detection and prevention tools and techniques[3]

Features	Script safe	Script defender	Tabs guard	CTabs
Use of whitelists	Yes	Yes	No	No
Use of blacklists	Yes	No	Yes	Yes
Browser	Chrome	Chrome and opera	Firefox	Chrome
Script-based attack prevention	Not by default	Yes	Yes (Active prevention)	Yes (passive prevention)
Script-free attack prevention	No	No	Yes (Active prevention)	Yes (passive prevention)
Technology in use to detect tabnabbing attack	Java script	Java script	HTML DOM; heuristics and data mining techniques	Screenshot comparison; threshold value

There are three existing solutions studied related to Tabnabbing attack which are Script Defender, ScriptSafe and TabsGuard.

Script defender: Script Defender is an extension for Google Chrome and Opera browser which use the whitelisting method to allow the site for user to interact with it. This extension can block the unwanted scripts, plugins and other annoying page elements. This solution includes the usage of whitelist but not blacklist. However, it does not prevent script-freeat tack but prevents script-based attack.

ScriptSafe: As defined by Williams[2], ScriptSafe is an extension for Google Chrome browser which can selectively block many types of web content and technologies and prevent multiple low-level privacy leaks.

TabsGuard: TabsGuard is an extension for FireFox browser which is proposed and developed[3]. It is a hybrid anti-tabnabbing approach which combines heuristic-based metrics and anomaly detection techniques. This solution includes usage of blacklist but not whitelist. It also prevents script-free attack and script-based attack actively (Table 1).

CTabs prototype: In this study, the idea of CTabs is discussed including the system flow and algorithm.

Core idea: A successful Tabnabbing attack will depend on the user low awareness regarding his/her browsing activity. Upon visiting the malicious site, shifting focus on different tab and returning after some time, user will discover that the malicious page has changed its looks to resemble a popular application’s login form. A Tabnabbing attack is obvious to identify, since, a phishing page will contrast from the past sub stance. Detecti is however, muddled by the tab being out of focus and the client setting some trust in previously opened and visited tabs.

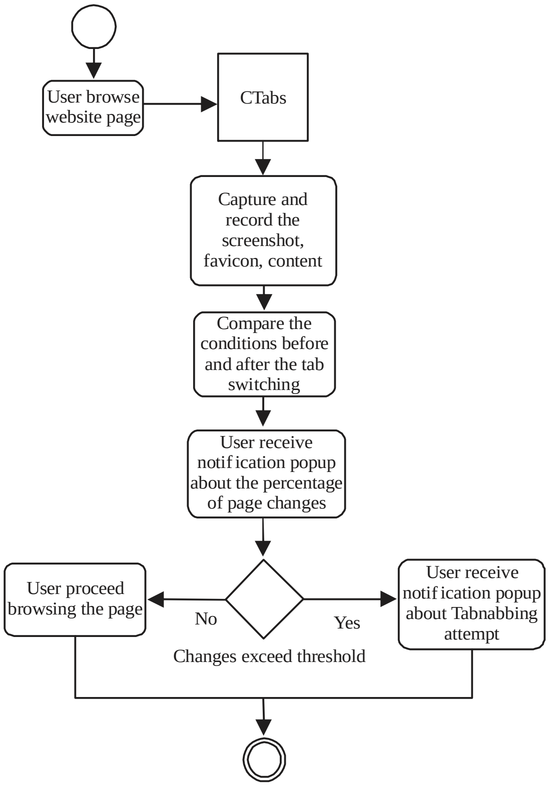

User browse website page
CTabs
Capture and record the screenshot, favicon, content
Compare the conditions before and after the tab switching
User receive notif ication popup about the percentage of page changes
User proceed browsing the page
No
Yes
User receive notif ication popup about Tabnabbing attempt
Changes exceed threshold

Fig. 2: System flow

CTabs will grab the benefit of these obvious changes by recording what the tab looks like before it loses focus and comparing its condition to what it looks like when it regains focus. Any differences that happened in the background will be detected and be notified to user by the means of a red-colored overlay and few warning popup. This will allow user to decide either the changes are harmful or not.

System flow and architecture: Figure 2 and 3 depicts CTabs system flow while Fig. 4 depicts CTabs system architecture. CTabs will capture and record the condition of the tab. User then switch to another tab and then switch back to the previous tab. CTabs then will compare the current tab condition to the condition before. Percentage of the difference between the two conditions will be notified to the user through popup. If the percentage exceed the threshold, warning notification will be triggered about an attempt of Tabnabbing attack. If not exceeding the threshold, user will resume his browsing activity.

Functionality: In this study, the functionality of CTabs are discussed in detail. Currently, there are six major functions which are.

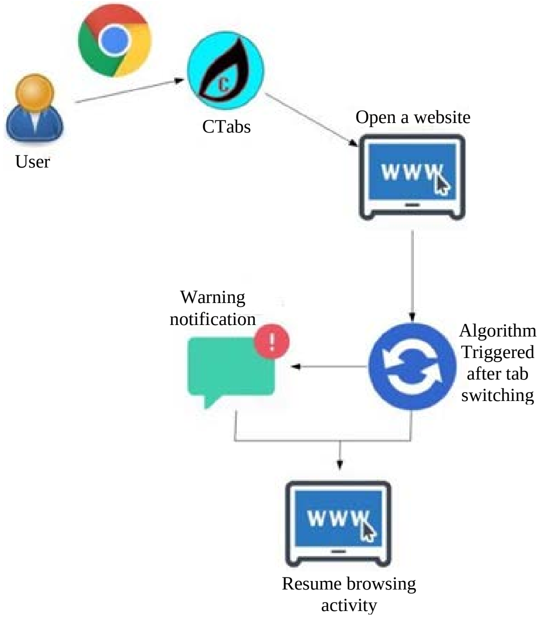
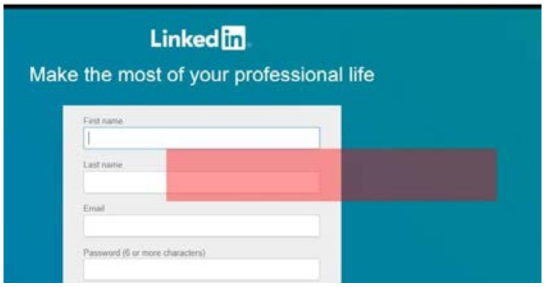


Fig. 3: System architecture



Make the most of your professional life

First name

Last name

Email

Password (6 or more characters)

Fig. 4: CTabs highlighted the area that has been changed using the red-colored overlay

Capturing tab condition: This function is consisted of methods that will capture the condition of the particular page including its favicon. Google Chrome has an API that can identify the selected tab by its ID and currently visible tab of a window. To output all of the processes that can be output, a method of Logger is developed.

Comparing tab condition: This function is consisted of methods that will compare the screen shots of the selected tab before and after it has gained focus. The screen shots will be compared using an algorithm. The algorithm will be implementing HTML5 canvas element for the screen shots, cutting and dividing the min to fixed-size. (e.g.,10×10 pixels). For the favicon, they will be compared by source.

Highlighting area changed: This function is consisted of methods that will highlight the area that has been changed after the selected tab is switched back. Implementing overlay element, CTabs will inject a red-colored overlay of the provided results in to the page, highlighting the differences in the page. Fig. 4 depicts the example of highlighting the changes on the web page.

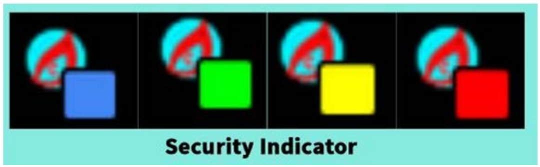
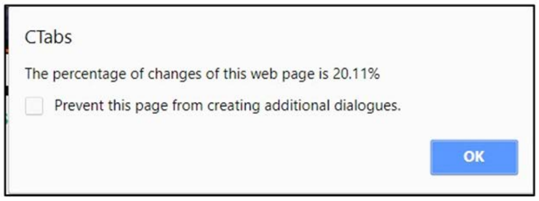


Fig. 5: CTabs icon colour

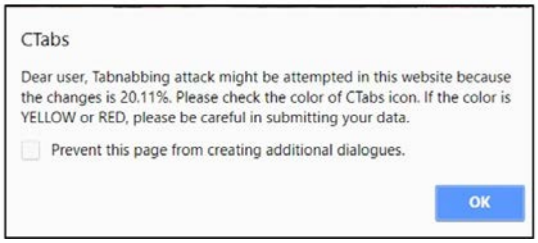


CTabs

The percentage of changes of this web page is 20.11%

□ Prevent this page from creating additional dialogues.

Fig. 6: CTabs notification about the changes percentage

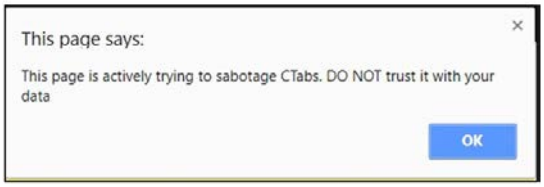


CTabs

Dear user, Tabnabbing attack might be attempted in this website because the changes is 20.11%. Please check the color of CTabs icon. If the color is YELLOW or RED, please be careful in submitting your data.

□ Prevent this page from creating additional dialogues.

Fig. 7: CTabs notification about the potential attempt of Tabnabbing



This page says:

This page is actively trying to sabotage CTabs. DO NOT trust it with your data

Fig. 8: CTabs sabotage detection notification will be triggered if the page removed the red-colored overlay

Icon indicator: This function that will change the color of the CTabs icon, based on the result of the comparison which are (Fig. 5-8):

- Green: the changes are <10%
- Yellow: the changes are <40%

- Red: the changes are <40%
- Blue: indicates that CTabs is in the stand by mode

Notification pop out: This function is consisted of methods that will notify the user of the result of the comparison in percentage, warning the user and also notify the user to check the color of the CTabs icon.

Sabotage detection function: This function is consisted of methods that will alert user if the compared page tried to remove the red-colored overlay injected by the CTabs. Five of the six functions above are inspired from[4].

RESULTS AND DISCUSSION

In this study, the testing of CTabs prototype are discussed and also the conclusion of this project. CTabs is implemented in Google Chrome browser as extension. For the testing, White Box Testing and Simulation of Tabnabbing attack were done to test the capabilities of CTabs.

Sabotage detection function testing: For the first testing, it is related to the Sabotage Detection function. For this test, three users of CTabs were tested against a page armed with Tabnabbing attack code and the results were depicted in a form of bar chart.

Table 2 explained CTabs results in detecting removal of the red-colored overlay by Test Page1. The results of the scores are based on Fig. 9. The testis successful because the detection worked and notification pop out is showing the message. This is the sign that CTabs is safe from being compromised its highlighting of changed areas.

Legitimate page testing: For CTabs, white box testing had been performed against legitimate pages that are well known their trustworthiness. The objective of this test is to evaluate how well CTabs can provide the score the legitimate pages.

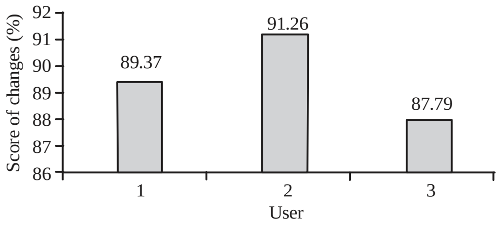


Fig. 9: Average score results produced by CTabs against Test Page1

Table 2: Sabotage detection function test result against test page 1

Name	Sabotage detection function testing
Date	2nd September 2017
Description	To test whether the detection of red-colored overlay removal is successful or not
Expected	Sabotage notification is popped out
Result	Warning notification is popped out
Result	Sabotage notification appeared accordingly and also the warning notification

Table 3 explained CTabs results against 10 different legitimate pages. The results of the scores are based on Fig. 10. The test is partially successful as there are only four pages that achieved the expected result while the other six are not.

The unexpected results are because of the tested pages are dynamic and they have web page components that constantly changing over time, resulting false positive outputs of CTabs. Examples of the web page components are the advertisements and the image-slider.

CTabs against self-made Tabnabbing page testing: In this test, a Tabnabbing page was made and was tested against CTabs.

Table 4 explained the CTabs test result against self-made Tabnabbing page. The result of the score are based on Fig. 11 where the most significant score that CTabs computed is 91.26%. It showed that the test is successful in detecting Tabnabbing attack and next, preventing user from giving away his or her credentials through the warning notification.

CTabs against Tabnabbing page created by using SE Toolkit: In this of testing, Tabnabbing attack was executed through Kali Linux and Windows 10 confronted it using CTabs[5, 6].

Table 5 explained the CTabs test result Tabnabbing page created by using SE Toolkit. The score result was based on Fig. 12 and it shows that the highest score is 89.99%. By running this test, its result is expected to achieve at least 40% of changes and popped out the warning notification. After testing has been done, the actual result matches the expected result. This proved that CTabs is useful to confront Tabnabbing attack launched by using SE Toolkit.

Table 3: Legitimate page test result against 10 different legitimate pages

Name	Legitimate Page Testing
Date	11th September 2017
Description	To test whether CTabs can provide reasonable score for the legitimate pages
Expected result	The score of the comparisons all are below 10% and no alert notification triggered
Result	6 pages were scored averagely over 10% and alert notifications were triggered but not sabotage notification

Table 4: Sabotage detection function test result against test page 1

Name	CTabs against self-made Tabnabbing page testing
Date	20th September 2017
Description	To test whether CTabs can detect Tabnabbing attack and provide expected output
Expected	The score of the comparisons all are above 40%
Result	Warning notification is popped out
Result	PASSED all CTabs users received score over than 40% and alert notifications were triggered

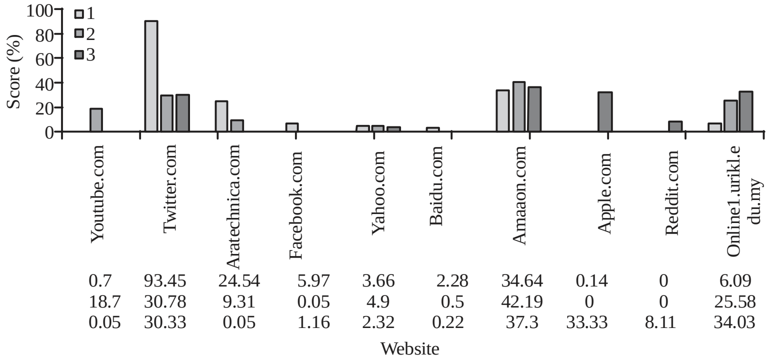


Fig. 10: Average score results produced by CTabs against 10 different legitimate pages

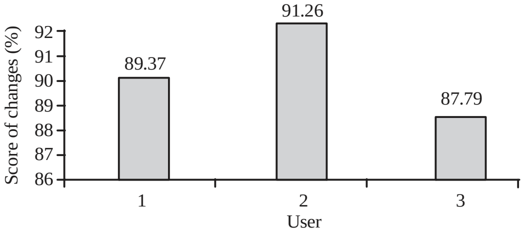


Fig. 11: Average score results produced by CTabs against Test Page 2

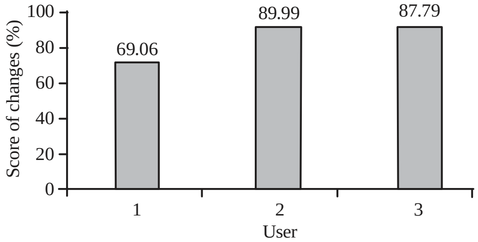


Fig. 12: Average score results produced by CTabs against Tabnabbing page of SE Toolkit

Table 5: CTabs testing result against Tabnabbing page created by using SE Toolkit

Name	CTabs against Tabnabbing page created by using SE Toolkit
Date	25th September 2017
Description	To test whether CTabs can detect Tabnabbing attack from the SE Toolkit and provide expected output
Expected result	The score of the comparisons all are above 40% Warning notification is popped out
Result	PASSED All CTabs users received score over than 40% and alert notifications were triggered

CONCLUSION

In the nutshell, Tabnabbing attack is a part of phishing attack where the situation is the hacker exploits the trust a user places in previously opened browser tabs. This could happen by making the tab changes its look to a legitimate login form of a known web application while user is not focus on it. The tests for CTabs proved its capabilities in detecting and preventing user from falling victim to Tabnabbing attack with100% of the simulation attacks were successfully confronted and gained expected out puts. However in one part of the testing of CTabs against legitimate pages, 60% of the pages tested fell into the suspected Tabnabbing attack, caused by the dynamic web design. But that will not be the main issue as users can determines themselves whether should insert credentials or not based on the URL. Furthermore, there are lot of current solutions for this type attack, still ignorant user falls for this attack easily. Other current countermeasures typically depend on several specific methodologies and requirements of Tabnabbing attack and are easily by passed. However, this developed countermeasure of CTabs is the first to do a fully visual comparison followed by highlighting the differences and next giving warning notification to the user if any potential Tabnabbing attempt is suspected.

REFERENCES

01. Maan, P.S. and M. Sharma, 2012. Social engineering: A partial technical attack. Intl. J. Comput. Sci. Issues, 9: 557-559.
02. Suri, R.K., D.S. Tomar and D.R. Sahu, 2012. An approach to perceive Tabnabbing attack. Intl. J. Sci. Technol. Res., 1: 90-94.
03. Hashemi, F. and H. Sadat, 2014. A hybrid approach to detect tabnabbing attacks. M.Sc. Thesis, Queen's University, Kingston, Canada.
04. De Ryck, P., N. Nikiforakis, L. Desmet and W. Joosen, 2013. Client-side detection of Tabnabbing attacks. Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security (ASIA CCS '13), May 08-10, 2013, ACM, Hangzhou, China, pp: 447-456.
05. Chaudhry, J.A., S.A. Chaudhry and R.G. Rittenhouse, 2016. Phishing attacks and defenses. Intl. J. Secur. Appl., 10: 247-256.
06. Saud, M.M., S. Ismail, E.M. Tamil and M.Y.I. Idris, 2007. Phishing: Challenges and issues in Malaysia. Int. J. Learn.: Annu. Rev., 14: 79-88.