Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 574 – 581

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

AgentTab: An Agent Based Approach to Detect Tabnabbing Attack

Sarika S[a,*], Varghese Paul[b]

[a]Department of CS,Bharathiyar University,Coimbatore,641046,India
[b]Department of IT,Cochin University of Science and Technology,Cochin,682022,India

Abstract

Phishing is an art of deceiving users by hijacking sensitive information like details of bank account, credit card, login on email and social networking sites, through social engineering. This is done by using emails and fake web pages that mimic legitimate companies which force users into revealing such data. Phishing attacks are a serious threat to online security as it may involve identity theft and financial loss. Anti-phishing solutions presently available are inefficient to the newer kinds of attack. This paper explores a newer kind of phishing attack called Tabnabbing and a novel approach to detect it using artificial intelligent agents.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

Keywords: Tabnabbing; Distributed software agents; Antiphishing; Multiagents

1. Introduction

In the present day, almost every person has his or her web identity. Online activities and social networking have grown to such an extent that governments safe keep the entire personality or identity of a citizen on web servers. Illegal activities too are increasing in the virtual world. Despite having several anti-phishing tools the number of victims have increased dramatically over last few years as internet users ignore warning alerts and most of the solutions available rely on user input. Also, as soon as we become competent to identify a particular type of attack, another more sophisticated version comes to the fore. A greater focus should be given for securing web services in this perspective.

* Corresponding author. Tel.: +919947948987
E-mail address: sarika.anand08@gmail.com

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.094

India has become the sixth most frequently attacked country by cybercriminals. Analysts have described 2013 as the year of the Mega Breach. In an increase of 2.32 per cent from the previous year, almost 39.6 million users faced phishing attacks in 2013. Some of the recent security breaches are intrusion of Chinese hackers into US federal employee database in July 2014, eBay account information breach in March 2014, target store security breach in December 2013 etc.

Tabnabbing/tab napping[6] is a more sophisticated kind of phishing attack proposed by Mozilla's AzaRaskin in 2010. It is phishing within a browser and it no longer relies on persuading user to click on a dodgy link. This targets a user who keeps many tabs open at a time. The attack works by replacing a tab you have already opened and left unattended to a fake login page specifically designed to steal personal data. As many tabs will be open at a time, the user may not remember which page was open while leaving the tab. The impersonating page is selected by the attacker based on the browsing history of the user. This increases the chances that the user falls for the trick.

The internet keeps on growing and soon it will get too big and diverse that simple search engines will not be able to search out what a user is looking for. Specially designed software agents[9] could filter, fine tune or correct user queries intelligently or even create a database for the searched subject within a programmed period. They could also be used to abstract the information gathered to suit the user needs. The agent could act as an assistant as it learns from its interactions with the user and can proactively anticipate his future needs. Distributed agent architectures seem to offer a promising basis for practical solutions in view of the increased threats in online activities. Artificial Intelligent agents with enduring reasoning skills can address the issues related with the control in network and internet management.

This paper proposes a framework called AgentTab to perceive Tabnabbing attack using artificial intelligent agents distributed across a platform. BDI (Belief-Desire-Intention) agents in this system cooperate with other agents to perform the delegated task.

The organization of this paper is as follows. Next section describes related work followed by the design principles. Further, the core idea behind proposed system is presented which is continued with architecture, operation and implementation of AgentTab. Finally, we discuss the results of our work and conclude the paper.

2. Related Work

Mozilla has released a Firefox plugin called Account Manager for online identity management. Account Manager lets you store the logins which are already created, suggesting them whenever they can be used. It makes the logins more secure by generating random passwords too. NoScript[7] is a Firefox add-on, for preventing websites from running JavaScript, Java, Flash or other plugins. It provides powerful protection against malicious scripts, XSS, CSR and clickjacking attacks. Another Firefox add-on proposed by Unlu and Bicakci is NoTabNab[4] which guards users from tabnabbing attack. This add-on lookouts open tabs and alert the user about changes in its layout, favicon or title to mimic another page. If an impersonation is happened, the address bar is highlightened in yellow or red according to the warning level. The problems that are indirectly connected to this technique are related to resizing the browser, as only some web pages are designed to re-layout themselves.

The method presented by Suriet al.[3] is also for detecting tabnabbing attack. They use a signature based detection mechanism to deal with tabnabbing attack. The detection mechanism defines a set of rules to scrutinize vulnerable JavaScript code. First the source file is converted into a text file and then into tokens. These values are given to the rule based system which is checked for vulnerabilities. The limitation of this method is that the presence of an iframe is not always necessary for a tabnabbing attack.

Tab-Shots[2] is a browser extension that remembers what each tab looked like, whenever a tab is changed. Tab-Shots record the favicon and screenshots of the presently focused tab at regular time periods. Then the screenshot is separated in to fixed-size tiles. Each tile of the present snapshot is compared to its counterpart in the stored data. If an exact match is not obtained, the non-matching area is marked by a coloured overlay. One probable shortcoming of this technique is the difficulty in detecting small changes in a page.

Current solutions will detect the layout change and warn the user only when the tab is on focus after being nabbed. AgentTab continuously monitors the change in webpage layout in frequent intervals in all the tabs of a browser in parallel and warns the user about the attack at the same time itself. So, the user can be more conscious about the attack and act accordingly.

3. Design Principles

AgentTab is based on the following design principles:

- Security and awareness to the user
- Good response time

3.1 Security

While working with AgentTab, user security was given utmost importance. In recent times, attacks exploiting human vulnerabilities have been on the rise and online security has become that much important. Phishing attacks are reinvented with brand new techniques to lure users to a fraudulent page by masquerading as a honest site and acquiring their credentials. The inefficacy of current solutions for circumventing phishing attacks led to this research work. Antiphishing algorithms come in two flavours: a) eliminating the threat without the user intervention and b) detecting threat and warning the users about it. In the first case, users are protected without them even being aware about the phishing attack as phishing scams or attempts are deleted or cancelled before the user encounters the attack. This has a clear cut disadvantage as the people remain unaware of current scenarios of online security threats and breaches.

AgentTab has tried to do something different where it alerts the user about the attack and give explicit warning messages about the symptoms of attack which are extremely simple to understand. Studies have shown that users often do not act on the visual cues provided by add-ons.

3.2 Response Time

When different tabs of a browser are open at the same time, multiple agents are simultaneously working with each tab keenly watches for an attack. Time consumption for the detection of attack can be reduced by parallelism and the user can be warned at the earliest. In essence, each tab is well monitored like a web watcher.

4. Proposed System

AgentTab is an agent based approach to detect tabnabbing attack and is currently built as a browser extension to Google chrome browser which is vulnerable to most type of phishing attacks. A prototype of AgentTab is explained in our previous paper[1].Objective includes the behaviour of Tabnabbing attack in various browsers and the effectiveness of this method in different browsers.

4.1 Tabnabbing Attack

Tabnabbing is a new kind of phishing attack which asks users to submit their credentials to a malevolent site by masquerading as a genuine one. The Tabnabbing attack triggers when user is busy switching between different tabs of a browser and a certain tab is out of focus and is idle for some time. The different steps of a Tabnabbing attack are:

- The user is visiting a webpage which looks perfectly genuine. The user opens multiple webpages like news, mail account or a social networking site in other tabs of the browser.
- The user changes his tab to another or the user is forced to switch to another tab when the page takes time to load.
- When a tab is unattended for some time and is out of focus, the favicon, title, and layout of the page is replaced with some other site familiar to the user (a frequently used site by the user). In some cases, the title and favicon may not be replaced but the URL of the site will be different.
- As the user pays less attention to the URL in address bar, he will give his credentials to the honest looking site and is trapped.

4.2 Multiagent Based Computing

Phishing attack is a complex phenomenon. An intelligent agent is a powerful tool for solving and handling such issues. Multiagent systems[8] divides the problem into modules which operate asynchronously.This simplification allows use of the best technique to solve problems. Interdependent problems are solved with co-ordinated effort from multiple agents. Every agent uses BDI (belief-desire-intention) architecture and has autonomy, heterogeneity, co-ordination with other agents and its own reasoning.

4.3 Phishing Detection using Multiagents

Agent based phishing detection is a scientific approach which needs modelling, designing and implementing multiagents in a platform to protect webpages from various attacks. The agents in this system are discrete and well defined within boundaries. Multiple agents deployed in the platform have dynamic strategic behaviour and are purely equipped with a strong prerequisite for using the platform.JADE[5] software framework is used here for the development of agent applications in compliance with the FIPA specifications.

5. Architecture of AgentTab

AgentTab works with the cooperation of artificial intelligent agents at three levels as shown in Fig 1. The level one agents are (U-agents and T-agents) managed by level two agent (M-agent). The interface between the user and the system is provided by level three agent(I-agent). The major role comes to T-agent which performs most of the functionalities. The number of T-agents and U-agents will depend upon the number of opened tabs and they do the same functionality in each tab. The operation of AgentTab proceeds in two phases. Feature extraction of a webpage in first phase and feature comparison in second phase. AgentTab starts action when a user opens a website. In first phase,it extracts the 5-tuples text, image, URL, title and favicon of the current page and stores the values for later use.The second phase starts when the user changes a tab. Now, AgentTab performs feature comparison of webpage using the above mentioned 5-tuple elements along with its predetermined value.

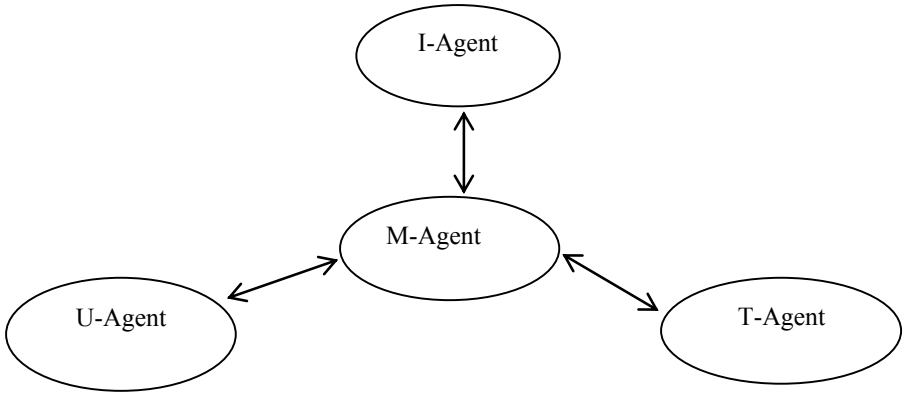

Fig. 1. Architecture of AgentTab

5.1 U-Agent

The U-Agent stores the URL, favicon and title of the presently opened webpages. It will verify the URL of the current webpages and check whether it is blacklisted or not by cross-checking with the blacklisted URLs in phishtank.com. It also checks whether the hyperlinks in the webpage navigates to a phishing website. The layout changes in the webpages are detected by T-Agents. A proper communication is required between T-Agent and U-Agent to detect the presence of an attack. The current values of URL, favicon and title of the webpages are compared with the values which are already stored in frequent intervals. We have taken an interval of 60 seconds.

5.2 T-Agent

T-Agent is a synonym for Tabnab agent. T-Agents check for the change in webpage layout every minute. This is done automatically by parsing the currently opened webpages and storing the initial value in separate files. The layout changes are monitored by parsing the webpages every 60 seconds and comparing the parser output with the stored parser output. If there is a change, the same is communicated with U-Agent and M-Agent. They reach a consensus to detect the presence of an attack by assessing the various symptoms.

6. Operation of Proposed System

6.1. Extraction of Text and Images

Phase I of AgentTab proceeds with extraction of text and images. Text extraction is done by using SAX parser. SAX parser can be used as an effective mechanism to parse the webpages. SAX processing is based on an event-driven processing model. The data elements are deduced on a sequential basis and callbacks are implemented based on certain constructs. One of the biggest advantages of SAX is that it does not load any XML documents into memory. Therefore it is considered to be lightweight and fast. SAX processing model is very simple and it requires implementing a class that extends the DefaultHandler which contain the implementation code in callback methods. A parsing handler class (extended from DefaultHandler) is required for reading and writing XML documents with SAX. This handler class provides logic coded in the callback methods defined. The parser then processes the input stream and invokes the handler's callback methods to perform the actual work. Fig 2 shows SAX processing model.

Image extraction is done by obtaining the source address of the image src attribute, the space occupied by the image in pixel and its position in webpage, and its RGB color histograms. The source address of the image can be obtained from the SAX parser output. The position of the image in webpage is obtained by finding the pixel positions.

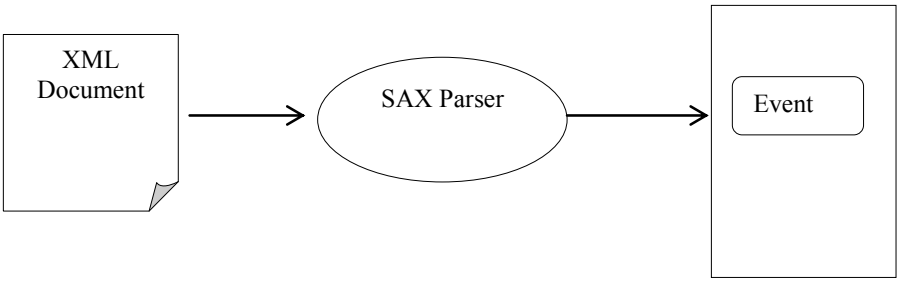

Fig. 2. SAX Processing Model

6.2. Similarity of two webpages

Similarity checking is accomplished by comparing matching elements separately. Textual contents are compared with the resultant file of SAX parser and find the resemblance score in a matrix as Rt. Then, compare all image pairs to obtain a resemblance score Ri. For each image, comparison is performed as follows: a)comparison of source address of the image src attribute b) comparison of RGB color histogram c) comparison of pixel positions that image occupy .URL is matched with the stored URL value to obtain a resemblance score Ru. Favicons are compared by source to get a resemblance score Rf. Title of the webpage is matched with the stored value to obtain a resemblance score Rti. Finally, the overall appearances of the two pages are calculated using the above mentioned 5-tuple as $R=Rt+Ri+Ru+Rf+Rti$. If the resemblance score is greater than a threshold t, two pages are similar. Otherwise, pages are dissimilar.

There is a phishing attack if:

- URL is changed and is blacklisted
- URL is changed and is not blacklisted, but favicon, title and page layout is changed.
- URL is changed and is not blacklisted, favicon and title not changed, page layout is changed

7. Implementation

The implementation of AgentTab uses JADE software framework to deploy agents in browser to detect URL based attack, hyperlink based attack and tabnabbing attack. The distributed multiagents communicate via FIPA ACL.This browser extension is installed in Google Chrome browser and initiated when browser starts functioning.

7.1. Dataset

The data set consists of a set of common webpages. For finding blacklisted URL's, a collection of real phishing sites from www.phishtank.com are taken. The experiment is conducted in selected 160 webpages with login forms such as banking sites, web mail clients, credit cards, social networking sites etc. This is because the Tabnabbing attack targets webpages which can provide sensitive information of users. A set of whitelisted URLs which are most common targets to tabnabbing attack, webpages commonly accessed by users are also kept to detect the attack easily. Some phishing URLs may be very similar to the legitimate one with one or two character changes. This can be easily notified by referring the white list and the detection time can be improved.

7.2. Analysis and Results

The effectiveness of the proposed method is assessed using the following parameters.

- True positive (TP) – Legitimate websites detected as legitimate.
- True negative (TN) – Phishing websites detected as it is.
- False positive (FP) – Legitimate websites detected as phishing sites.
- False negative (FN) – Phishing websites detected as legitimate.

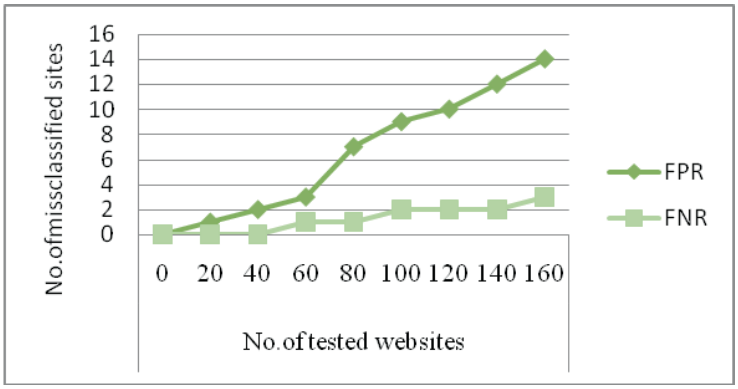

Fig. 3. No. of False Positives and False Negatives

Percentage accuracy of proposed method can be calculated as:*(TP+TN)/Number of tested sites.*
The other parameters for evaluation are FalsePositive Rate (FPR) and False Negative Rate (FNR).
These two are calculated using the formulas given below:

FPR = FP/Number of legitimate sites tested.
FNR = FN/Number of phishing sites tested.

The values of FPR and FNR fall within the range 0-1.Our results give 91% accuracy to the proposed method. This implies that it can accurately detect about 91% of Tabnabbing attacks while misclassifying 9% of legitimate websites. False detections were mainly from websites with more animated content. In all other cases, the method shows an impressive response time for accurate detection. Fig 3 shows our empirical result. The parameters taken for evaluation are false positive rate and false negative rate among the number of tested websites.

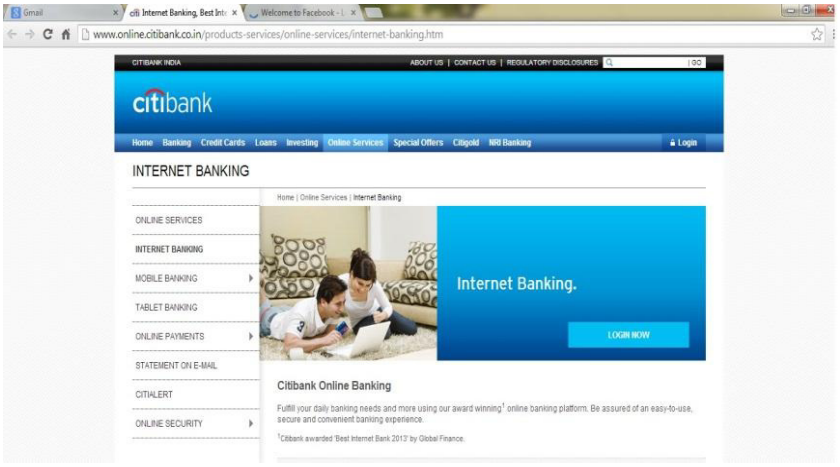

Fig. 4. Original Website of Citibank

Fig 4 shows the original website of citibank. When the tab is changed, the legitimate cite of citibank is impersonated as a phishing site as shown in Fig 5.This is detected as a fake site by AgentTab as it has changed its URL and page layout with no change in title and favicon of the page. Fig 6 shows the warning message generated by AgentTab to signal a tabnabbing attack. AgentTab initiates its first phase by extracting the webpage features when a real website is opened in a tab. At the time when the user changes tab, first phase is repeated in frequent intervals and second phase starts by comparing the webpage features along with the former values of the same webpage.

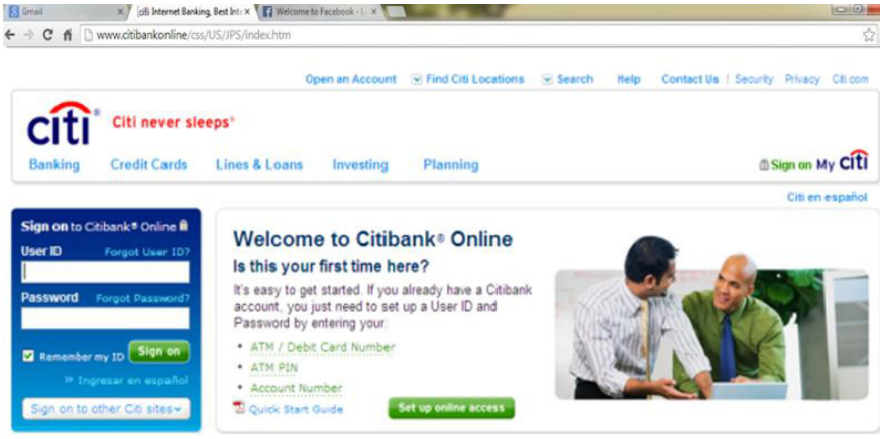

Fig. 5. Impersonated webpage of Citibank

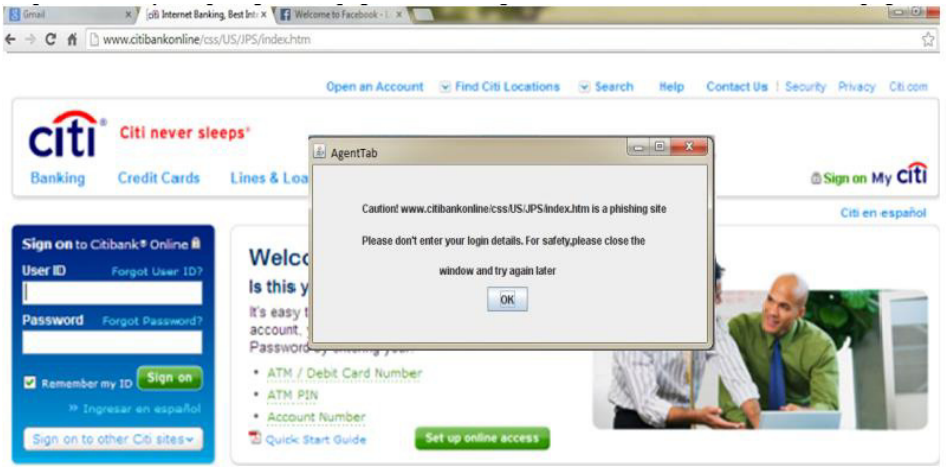

Fig. 6. Alert generated by AgentTab

8. Conclusion

Tabnabbing is a Phishing Attack, which happens when a scammer tricks a user into giving away information about account details such as username and password by nabbing a tab. It is not unusual to have half a dozen or more tabs open at once and this wicked deception uses sites the users habitually visit.This paper, discusses about a distributed agent-based architecture that fights tabnabbing attacks. While preventing tabnabbing attack, it can also detect URL based attack and hyperlink based attack.Therefore, it can act as a three factor security framework.The autonomous agents in this system act automatically when an attack scenario happens and performs the action which is delegated to it.The cooperation between the agents helps in reaching a consensus about the attack.The experimental evaluation shows that this is a feasible approach to ensure online security and resistance from tabnabbing attack. The future work includes further development of this framework to battle common types of phishing attacks.

References

1. Sarika S, Dr. Varghese Paul. Distributed Software Agents for Antiphishing. *International Journal of Computer Science Issues*. Newyork; 2013.p .125-130.
2. De Ryck, Nick, L Desmet, Joosen. TabShots: Client-Side Detection of Tabnabbing Attacks. *Proceedings of the 8th ACM SIGSAC Symposium on Information.* 2013.
3. R. K. Suri, D. S. Tomar, D. R. Sahu. An Approach to Perceive Tabnabbing Attack. *International Journal of Scientific and Technology Research.* 2012. p . 90-94.
4. Seckin Anil Unlu, Kemal Bicakci. NoTabNab: Protection Against The Tabnabbing Attack ,*IEEE* . 2010.
5. Bellifemine,Rimassa, G Poggi. JADE - A FIPA-compliant Agent Framework. *Proceedings of the 4th International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents.* London ; 1999.
6. A. Raskin. Tabnabbing: A new type of phishing attack. *http://www.azarask.in/blog/post/a-new-type-of-phishing-attack/.*
7. NoScript - JavaScript/Java/Flash blocker for a safer Firefox experience. *http://noscript.net/.*
8. Katia P. Sycara. Multiagent Systems, *American Association for Artificial Intelligence.* 1998.
9. Bradshaw.Software Agents. *MIT Press*. Cambridge: MA USA.