International Journal of Engineering Research and Development
e-ISSN: 2278-067X, **p**-ISSN: 2278-800X, www.ijerd.com
Volume 7, Issue 1 (May 2013), PP. 01-05

Trusted Profile Identification and Validation Model

Himanshu Gupta[1], A Arokiaraj Jovith[2]
[1, 2]Dept. of Information Technology, SRM University, Chennai, India
hg.jumbo@gmail.com
arokiarajjovith.a@ktr.srmuniv.ac.in

Abstract:- Social networking is one of the most popular Internet activities, with millions of users from around the world. The users believe their friend's so blindly on this virtual world as they are sitting next to them and also users provide so much information about them which help the attackers to launch a social engineering attacks. As attacker can duplicate the presence of the users and fool the friends of the user to gain access to the user's friend system. This all happens due to more than one user profile present in these sites. Facebook announced 1 billion users in October 2012[1] and there are some sites on the Internet which sells the Facebook[2] and other user profiles to propagate their advertisements. All those selling profiles are cloned. The biggest problem in these sites is that they don't have the tools for detecting duplicate profiles. The basic concept is to identify the Duplicate profiles and validate the genuine profiles among them as trusted profiles. To implement this we have to create authentication model and monitoring system. The authentication model will stop the bots for creating duplicate profiles and monitoring system will monitor the user activity and based on the results of the monitoring system the authentication model will validate the profiles.

Keywords:- Trusted Profile, Motion Captcha, Single Sign On, Bots.

I. INTRODUCTION

The popularity of the social networking sites is so much that everyone wants to join these sites. Facebook announced 1 billion users in October 2012[1] and also LinkedIn has 60 million users[3]. As the majority of users only want to use these sites but they are not familiar with the privacy issues in these sites, they often provide so much of their personal information in their profiles which is available to everyone on these sites. Due to which an attacker may clone the profile of the user in same or different social networking sites to launch a social engineering attack.

Trusted Profile Identification and Validation is a model which will reduce the profile cloning on a social networking site. As in today's Internet many social networking sites like Facebook [2], LinkedIn etc. have lots of duplicate profiles which are used by the different unauthorised people to perform illegal tasks like:

- Attackers may duplicate a legitimate user's online presence to launch social engineering attack.
- Attackers may create a new profile in same or different social networking site(s) with user personal data.
- Attackers may spread false messages to create panic in the public.
- Attackers may fool the user to pay for the services which never exist.
- Attackers may use the email or user-id of the legitimate user to launch a DOS attack.

Many social networking sites do not provide any counter measure for the above mentioned points. In order to stop these attacks the proposed system will provide. Authentication Model will try to authenticate the user at each level and uses monitor system data to accomplish its task. Monitoring System will monitor the user based on the user friends, posts, location etc. and provide a feed to Authentication Protocol.

The objective of this paper is to identify the Duplicate profiles and validate the genuine profiles among them as trusted profiles. To provide this functionality the model must implement the following:

- To verify whether a user who is trying to register is a human or a bot.
- Monitor the user activity.
- Provide different authentication and verification methods.
- Use monitor activity data to validate the user.

II. DESIGN

In this section we outline the design of our model for validating the genuine profiles. This model comprises of two components and we describe it one by one.

1. Authentication Protocol

This component is responsible to mark a profile genuine or duplicate and also stop bots to create duplicate profiles. It will validate every user at the time of their login and analyze the output of the monitoring system to verify the profile. This model will use live or motion captcha [4] to validate a user at the time of his/her profile registration that the user is a human or a bot. Also this model can authenticate the user from any other government or social networking site on the basis of some govt. id or an email that person is using [5]. This model can block access to the profile on the basis of the information it collect from the monitoring system like increase in number of post per day or number of friends. The authentication protocol is divided in 2 parts:

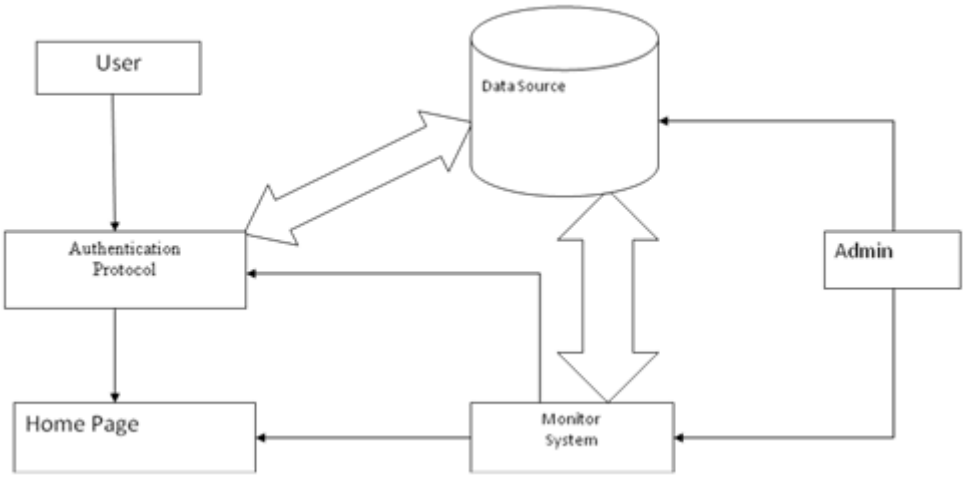

Fig. 1. Diagram of our system architecture

i. Authentication for Login

This part of component will authenticate and authorize the user at the time of login. Apart from the username and password it will check the time, location or last activity to authenticate the user and any malicious activity will intercepted it will verify the user with the combination of the personal data provided by the user itself.

ii. Authentication for Registration

This part of component will stop the bots by creating duplicate profiles. As many bots can create the duplicate profiles by collecting online data [4] and breaking the captcha [5]. This part will use the live or motion captcha [6] which a human can only solve. By this we can stop the bots.

2. Monitoring System:

This component of the model monitors the activity of the user and provides the information to the authentication protocol. This component will validate the profiles on the basis of some rules like number of post per day or location. If it detects any malicious activity then it will provide the information to the authentication protocol. Monitoring system is divided into 3 parts:

i. User Monitoring

This part will monitor user on the basis of location or number of friends detect the duplicate profiles and it can also detect the duplicate profiles on the basis of Detecting Social Network Profile Cloning [7]. This monitoring will provide information to the authentication protocol to authenticate the user at the time of login.

ii. Report Abuse Monitoring

This part will help the system to track the duplicate profiles with the help of the users. As user can provide the view about the profiles using the Report Abuse Functionality the system can prioritize the user profiles for identification and validation. Also this part will help the Content Monitoring to maintain the list of abusive words of sms language.

iii. Content Monitoring

Content Monitoring is an add-on to this module to monitor the content posted by the user. We can mark user on the basis of the content he/she will post and monitor them on these basis or if some different behavior is detected we can verify those users by Authentication protocol.

III. IMPLEMENTATION

Authentication Protocol

This component is the main working part of the model which will mark the profiles as trusted or not. This component will analyze the monitoring system information and identify the profiles. Apart from this the component will stop the bots and verify the user at the time of their login or registration.

i. Authentication for Login

As login is the entry point so if the users profile is not set as trusted profile we have to verify users through some methods. For this we can verify the user by these methods:

a) Enforce user to verify themselves on the basis of their personal data like we can ask user to enter combination of their password with their secondary email. When a user will fill the registration form the next page will be the security page in which user has to enter the personal details like email, phone number, two security questions with answers etc. so the model have 7 distinct information about the user and whenever model has to verify the user it can verify the user on the basis of this information i.e. model can verify the user with 7*6 = 42 different combinations. This means that bot can never predict the answers in this verification process and if a user account is compromised then also the hacker cannot view the security information of the user because it is protected by the profile password.

b) Also we can use other social networking sites like facebook for authenticate user. The idea behind this is if the government will provide centralized servers which will support a Single Sign On or Open Id concept for each user so it is easy to authenticate the user.

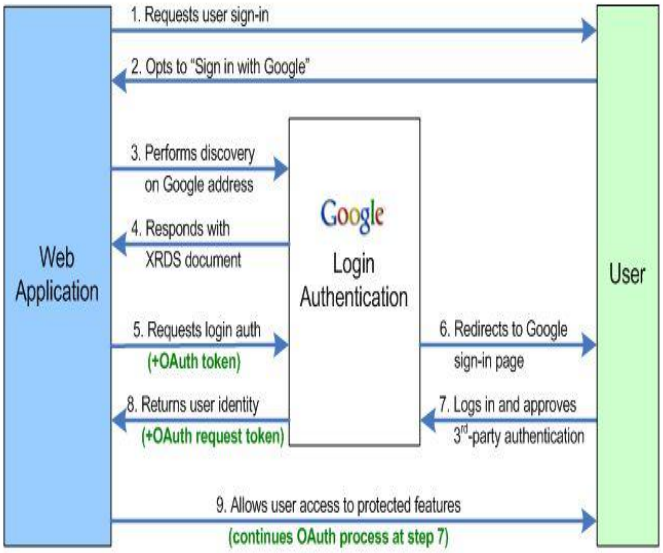

Fig. 2. Google SSO

1. The web application asks the end user to log in by offering a set of log-in options, including using their Google account.
2. The user selects the "Sign in with Google" [8] option.
3. The web application sends a "discovery" request to Google to get information on the Google login authentication endpoint.
4. Google returns an XRDS [9] document, which contains the endpoint address.
5. The web application sends a login authentication request to the Google endpoint address.
6. This action redirects the user to a Google Federated Login page, either in the same browser window or in a popup window, and the user is asked to sign in.
7. Once logged in, Google displays a confirmation and notifies the user that a third-party application is requesting authentication. The page asks the user to confirm or reject linking their Google account login with the web application login. If the web application is using OpenID+OAuth, the user is then asked to approve access to a specified set of Google services. Both the login and user information sharing must be approved by the user for authentication to continue. The user does not have the option of approving one but not the other.
8. If the user approves the authentication, Google returns the user to the URL specified in the *openid.return_to* parameter [8] of the original request. A Google-supplied identifier, which has no

relationship to the user's actual Google account name or password, is appended as the query parameter *openid.claimed_id*. If the request also included attribute exchange, additional user information may be appended. For OpenID+OAuth, an authorized OAuth request token is also returned.

The web application uses the Google-supplied identifier to recognize the user and allow access to application features and data. For OpenID+OAuth, the web application uses the request token to continue the OAuth sequence and gain access to the user's Google services.

ii. Authentication for Registration

In this part our main purpose is to stop bots by creating the duplicate profiles in social networking sites. To stop this we have to ask each user their information in random way so that bot cannot predict the registration or we can use the Single Sign-On facility to get the user information and register user in our social networking site. For registration we can use these methods:

a) With the random registration input use the live captcha [6] to test the user that he/she is a human. As shown in picture below the user has to arange the numbers in the orde which a human can only solve not a bot.

Fig. 3. Motion CAPTCHA

The figure shown above will be a challenge to the user to solve the puzzle and sort the numbers using the mouse through which we can confirm the user as a human.

b) The same idea which we use in Authentication for Login we can use that here. Single Sign-On or Open Id concept can be used to fetch required information from the centralized server to register the user. E.g. While a user is signed in to an app, the app can access the account's email address or OpenID [10] identifier for every request the user makes to the app. The app can also access a user ID that identifies the user uniquely, even if the user changes the email address for his/her account through **opened.claimed_id, opened_identity [8]**etc. The app can also determine whether the current user is an administrator (a "developer") for the app. You can use this feature to build administrative features for the app, even if you don't authenticate other users.

IV. CONCLUSION

In this paper a methodology is proposed detect the duplicate profiles of the existing users. The model will mark the user profiles as a trusted one on the basis of their usage and if any malicious activity is detected the model can verify the users on the basis of their personal data. There is a monitoring system in the model which will track the user activity on daily basis i.e. number of posts per day, language used by the user, location of the user etc. and report to the authentication protocol. Authentication protocol will mark the user profiles as trusted and also perform authentication at login. By this model implemented in a social networking site we can detect and track the duplicate profiles of the existing user and stop the bots for fake registration on the social networking site.

REFERENCES

[1]. "Facebook statistics," Available: http://www.facebook.com/press/info.php?statistics.

[2]. "Facebook Botnets have gone Wild" Available: http://www.itworld.com/it-managementstrategy/278005/faking-it-facebook-profile-bot-network

[3]. "LinkedIn statistics," Available: http://techcrunch.com/2010/06/20/linkedin-tops-70-millionusers-includes-over-one-million-company-profiles/

[4]. Iasonas Polakis, Georgios Kontaxis, Spiros Antonatos, Eleni Gessiou, Thanasis Petsas, Evangelos P. Markatos, "Using Social Networks to Harvest Email Addresses", in *WPES '10: Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*.

[5]. L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, "All your contacts are belong to us: automated identity theft attacks on social networks," in *WWW '09: Proceedings of the 18th international conference on World wide web*.

[6]. "Live CAPTCHA," Available: http://jquerybyexample.blogspot.com/2012/04/best-5-jquery-captcha-plugins.html

[7]. Georgios Kontaxis,Iasonas Polakis, Sotiris Ioannidis and Evangelos P. Markatos, "Detecting Social Network Profile Cloning" in Pervasive Computing and Communications Workshops (PERCOM Workshops), in 2011 IEEE International Conference.

[8]. "Single Sign On," Available: https://developers.google.com/accounts/docs/OpenID

[9]. "eXtensible Resource Descriptor Sequence, " Available: http://en.wikipedia.org/wiki/XRDS

[10]. "OpenId," Available: http://openid.net/