Volume 3, Issue 3, March 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Enhancement of Social Network Security by Third Party Application

K.G.S.Venkatesan
Associate Professor, Dept. of CSE
Bharath University, Chennai, INDIA.

Kausik Mondal
Department of C.S.E.
Bharath University,Tamil Nadu.

Abhishek Kumar
Department of C.S.E
Bharath University,Chennai, India

Abstract—Social networking sites such as Facebook ,Google,twitter provide third party applications such as games and productivity applications by providing a social application platform. These interfaces enable popular site enhancements but poses privacy risks by exposing user data to third party developers. The current security mechanisms, such as privacy policies and access control mechanisms fall short on protecting the privacy of the users. In this paper, we are representing a framework for a privacy enhanced social network security application that technically enforces the protection of the personal data of a user, when interacting with social applications.Let us propose a multicriteria recommendation model that uses application based, category based, user based, and collaborative filtering mechanisms which are based on previous user decisions, and application requests to increase the privacy of the overall site's user population. Our project on the collected information indicate that the proposed framework enhanced the user security and privacy related to third-party application authorizations.

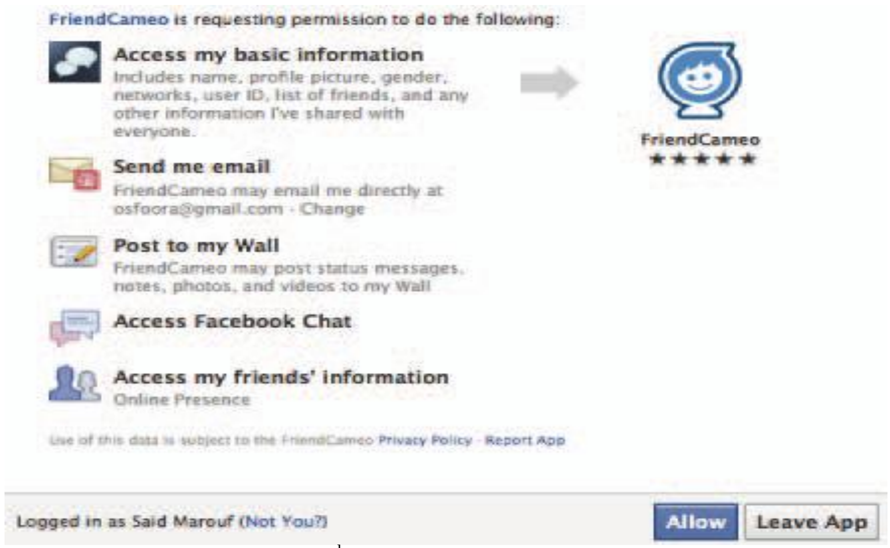
Keywords— OAuth, Random number generation ,collaborative filtering, social networks, Prediction model

I. INTRODUCTION

Online social networks (OSNs) are now the condition in modern society. The number and the size of OSNs grows, there is increasing concern about the security of the terabytes of personal information entrusted to OSNs and how the privacy of this data is managed. Third party applications, generally within social networking platforms have become very popular and penetrative. For example, with over eight million third party applications on Facebook and its users install applications more than 21 million times a day . Before using applications, users are required to authorize them and grant them access to certain permissions they request, e.g., access to a user's data such as e-mail, location, etc. With the pervasiveness of such applications, preventing the user's online personal data becomes a necessity. Open standards and third party software development have long formed a partnership that affords users the tools to better manage their own privacy, identity, and confidentiality. Seeing a need of users to better know the privacy policies in force for various websites led the World Wide Web Consortium to construct the Platform for Privacy Preferences specification and the corresponding Preference Exchange Language (APPEL) that is in use today by various internet websites to stipulate in machine readable format, the particular file specifying that site's particular privacy policies . The OAuth open standard protocol (OAuth) is another example of an available standard created to provide users with the ability to share data and resources with third party application components of other and more primary web applications. The OAuth framework might allow for the sharing of images from a primary webbased photo sharing site [4] .So, a third party photo printing service might access the permitted images. Popular social network , Facebook today represents the largest single OAuth 2.0 permitting a mechanism for third party web-based applications to access Facebook user identity and privacy and resources.

Third party application software developers have led charges to enhance user privacy and protection using extensible frameworks available in the web browsers such as Chrome, Firefox, and Safari. These browser extensions protect users from unwanted advertisements, spiteful software installations, and compromise of user credential information. Indeed, Joshi et al. showcase a browser plug-in which attempts to solve man-in-the-middle attacks prevalent in modern Phishing attacks. While the relationship between standards and browser-based extensions is rich in history and likely to continue, there may exist one gap that requires fulfilling. Appreciating that individual privacy preferences may be just individual , how can a single extension reflect the privacy preferences for a unique set of individuals? In this paper, we represent a novel browser extension (FBSecure) that implements a proposed recommender model which enables users to make important privacy decisions at the time of third party application installation and integrates in the existing OAuth 2.0 . Recommendations give users confidence in making their decisions, especially many privacy requests would not clearly convey the accesses granted [7]. The decisions which users make are their own ofcourse, but the algorithm and model provides a mechanism to inform them and provide recommendations based on the collaborative decisions (grant / deny) on same privacy requests in the user's larger social network.

A browser extension that intercepts the default OAuth 2.0 request flow, interprets and provides the user with a simple interface to make decisions that provide for the protection of personal identity attributes before installation of an application. A multicriteria recommender-model approaches to provide users with recommendations on requested privacy attributes based on the collaborative effort of users who have historically made decisions for similarly requested privacy attributes. A recommendation to extend the OAuth 2.0 specification to provide an attainment through which webbrowsers might assist users in making informed decisions regarding their full privacy attributes before the installation of an application [10]. A user study that shows the results and effectiveness of using our proposed browser extension.



FriendCameo is requesting permission to do the following:

Access my basic information
Includes name, profile picture, gender, networks, user ID, list of friends, and any other information I've shared with everyone.

Send me email
FriendCameo may email me directly at osfoora@gmail.com · Change

Post to my Wall
FriendCameo may post status messages, notes, photos, and videos to my Wall

Access Facebook Chat

Access my friends' information
Online Presence

Use of this data is subject to the FriendCameo Privacy Policy · Report App

Logged in as Said Marouf (Not You?)

Fig. 1 :.Social network 3rd party application permission request.

II. **Problem Definition**

The OAuth framework provides a mechanism for third party service providers to access end user resources without releasing the user's access credentials to the service provider. Sometimes, specific implementations neither provide the user with the necessary fine-grain access control, nor provide any recommendations in which access control decisions may be the most appropriate. As an example we use entire this paper is one of the free Facebook online video and voice calling applications available through website(friendcameo.com).The friendcameo Facebook application requests the following extended permissions when users first install the application: access to the user's e-mail , ability to publish status and post messages in the user's wall and the ability to access the Facebook chat application, and the ability to count the online presence status of other users.It becomes quickly clear that some of the extended once granted, permissions, cannot genuine be revoked. As a example, once users to provide the friendcameo access to their e-mail addresses, they cannot remove that genuine e-mail address from FriendCameo's servers and databases by preventing further access to the information through Facebook's application by privacy settings [16]. We find there are different user character that are practically irrevocable once granted, since the attributes are generally constant (i.e., birthday) or generally change with very little frequency (i.e., hometown locations, religious and political views). (See Fig. 1) We view the permanent loss of personal attributes as only one part of the problem; should a method be devised to permit users a "last line of defense" against such information loss, how may they know best what decisions can take users benefit from a knowledge of people to better inform their own decision making? Our proposed approach provides both the aforementioned "last line of defense" mechanism and a recommender model based on the decisions of other users within the area of people, and the previous decisions of an indivisible user.

III. **Existing System**

Major online platforms such as Google,Facebook and Twitter allow third-party applications such as productivity applications, Games are access to user online private data. Such accesses must be certified by users at installation time.The Open Authorization protocol (OAuth) was introduced as a secure and efficient method for authorizing third-party applications without releasing a user's access credentials.

IV. **Proposed System**

Proposed model was introduced as a secure and efficient mechanism for authorizing third-party applications. This requires users to present their credentials to third-party applications, hence allowed them large access to all their online resources with no restrictions. In OAuth, these new credentials are represented via an Access Token. It is a string which denotes a certain scope of permissions granted to an application is called Access Token & it also signify to other attributes such as the duration the Access Token is considered valid. We are mainly interested in the scope of character within an Access Token. And it was issued by an authorized server after the approval of the resource owner. We propose an abstract access paradigm, which can be applied to the design of the filtering systems, and the formalizes the access to the filtering results via multi-corridors (based on content-based categories) at the same time.). We use these measures to evaluate the use of various kinds of multi-corridors for our prototype user-adapting Website in the: Active of Web Museum [15]. It can be used to create user-adapting Web sites through Information filtering techniques. User-adapting Web sites adapt their presentation to the user's preferences. It is hard to relate the classic measurements to actual user satisfaction because of the way the user interacts with the recommendations, influences the benefits for the user and determined by their representation,. The systems consider every user as an expert for his taste, so that personalized recommendations can be provided based on the expertise of taste-related users. Collaborative filtering has been applied to several domains of information: News articles. Collaborative filtering systems then use this rating matrix in order to derive predictions [9]. Several algorithms have been proposed on how to use the rating matrix to predict ratings. The dynamic topology is achieved by dynamic corridors virtual corridors which contain paintings of a chosen category sorted according to personalized predictions produced by collaborative filtering. When this paradigm is applied to user-adapting Web sites or other types of recommender systems, more choice for the users can be provided via various multi-corridors and therefore increase performance from the perspective of the user.

V. Preliminaries

Most of the major online platforms such as Google, Twitter and Facebook is provide an open API which allows third-party applications to directly interact with their own platform. An APIs provide a mechanism to write, read,or modify the user data on such platforms through other third party applications on behalf of the users themselves and an API comes with a set of the methods,to each representing a secure user interaction executed through a third-party application. For example, the FriendCameo application is able to post content (e.g., messages, photos) to a user's Facebook feed/wall using profile_id/Facebook's/feed of an API method, where the profile_id is the targeted on Facebook user ID and it is important to indicate that third-party applications can potentially execute any API call on behalf of a relying on the type,user and scope of permissions granted to these applications.In this previous example, the application of FriendCameo should only perform the profile_id/feed API call given it the publish_stream permission to user has granted [8]. The set of the permissions available to the thirdparty applications are defined by the third-party applications and it is up to the online platforms whose to request the proper subset of permissions to be need. We believe on the users should have the final decision on which permissions to grant or deny.

A. *OAuth Standard :*

With an increasing trend toward offering online services that provide third-party applications the ability to interact through access user resources and open APIs, OAuth was introduced as an efficient and secure mechanism for authorizing third-party applications. Traditional authentication models such as the client-server model require third-party applications to authenticate with online services using the resource owner's typically a username,password and private credentials. This requires users to present their autherization to third-party applications, hence its allow them to broad access to all their online resources with no restrictions. A user may be revoke access from a third-party application by changed her credential, but doing so subsequently revokes access from all third-party applications that continue to use her previous accreditation. These issues are effect given the high number of the third-party applications that may be get access to a user's online resources. OAuth uses a component where the roles of resource owners and third-party applications are separated [6].

It doesn't require users to share that private credentials with the third-party applications, instead it issues a new set of the credentials for each application.These are new set of reflect an unique set of permissions to a user's online resources and credential as per application. In OAuth, these new autherization are represented via an Access Token and It is a string which denotes a certain scope of permissions granted to an application, it also denotes other features such as the duration of the Access Token is considered authentic string. We are mainly interested in the scope attribute within an Access Token and These Access Token are issued by an authorization server after the approval of the resource owner. In this paper, we have spread upon this authorization stage of the OAuth 2.0 protocol. When a third-party application needs to it presents its Access Token to the service provider hosting the resource which in turn verifies the requested access against the scope of permissions denoted by the Token and access a user's protected resources, . For example, Alice (resource owner) on Facebook (service provider and resource server) can grant the FriendCameo application (client) access to her e-mail address on her Facebook profile without ever sharing her username and password with FriendCameo. Instead, she authorize the FriendCameo application with Facebook which in turn provides FriendCameo with a proper Access Token that denotes permission to access Alice's e-mail address. OAuth provides multiple authorization flows depending on the client (third-party application) type (e.g., webserver, native applications). In this paper, shown in Fig. 2 and detailed in the OAuth 2.0 specification [28] and we focus on the Authorization Code flow. The authorization code flow is used by third-party applications that and are able to receive incoming requests via redirection,are able to interact with a user's web browser. The authorization flow process consists of three parties: 1)Client (third-party application) , 2)End-user (resource owner) at browser , and 3) Authorization server (e.g., Facebook). Our main focus is on the steps "(A)" and "(B)" within the approval of code flow. Step "(A)" is where third-party applications initiate the flow by redirecting a user's browser to the authorization server and pass along the requested scope of permissions [10].

In step "(B)," it establishes her decision on whether to grant or deny the third-party application's access request and the authorization server authenticates the end user.

B. *OAuth and User Privacy:*

One of the main reasons behind OAuth was to increase user privacy by separating the role of users from that of third party applications. OAuth uses the concept of Access Tokens, where a token denotes a set of credentials granted to third-party applications by the resource owners. This avoids the need for users to share their private credentials such as their username and password. It also allows users to revoke access to a specific third-party application by revoking its Access Token. OAuth 2.0 allows third-party applications to request a set of permissions via the scope attribute, and for users to grant/deny such requests. If a user grants a third-party application's request, then an Access Token (denoting the scope) is issued for that application, hence granting it the scope of permissions requested. The scope attribute represents the set of permissions requested by third-party applications, and is our main focus in this paper. In the authorization code, OAuth flow, the scope parameter is part of the request URI that is generated by third-party applications (Step "(A)"). The scope is a list of space-delimited strings, each string mapped to a certain permission or access level. For example, the FriendCameo application requests permission to post to a user's Facebook feed/wall, to log in to Facebook chat, to access her e-mail address, and to check her friend's online/ offline presence. FriendCameo requests these permissions with a scope attribute value of "publish_stream, xmpp_login, e-mail, friends_online_presence." The scope value becomes part of the OAuth request URI sent to the authorization server (Facebook's OAuth implementation uses commas rather than spaces to separate each requested permission) [13]. Step "(B)" is where users grant/deny the requested scope value. We propose an extension to the OAuth 2.0 authorization code flow detailed in Section V (D) of the OAuth 2.0 specification. Before users make their decision on the requested scope of permissions, we introduce a new level of awareness and control to the user via an in-house developed browser extension.

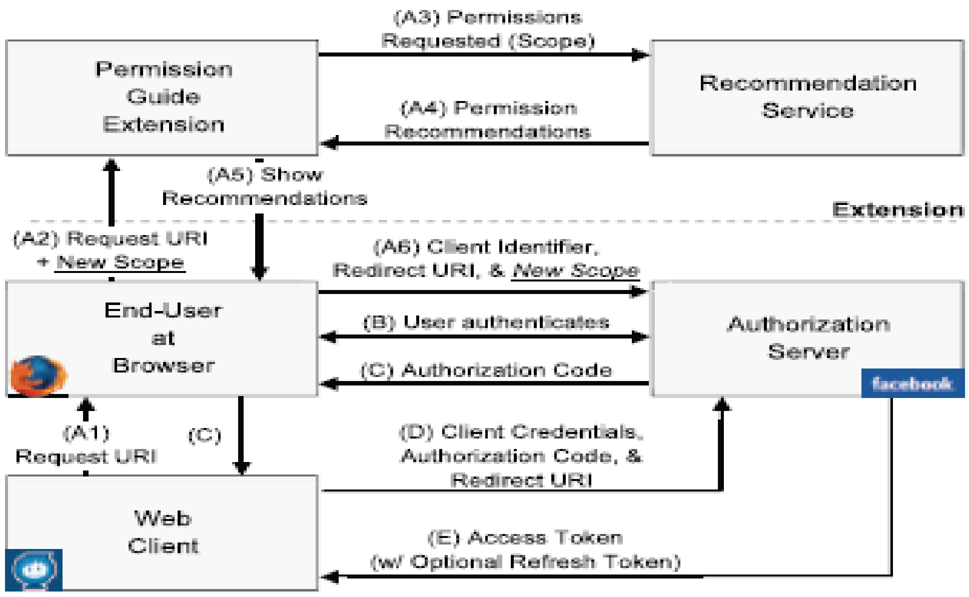

(A3) Permissions Requested (Scope)
Permission Guide Extension
Recommendation Service
(A4) Permission Recommendations
(A5) Show Recommendations
Extension
(A2) Request URI + New Scope
(A6) Client Identifier, Redirect URI, & New Scope
End-User at Browser
(B) User authenticates
Authorization Server
(C) Authorization Code
facebook
(A1) Request URI
(C)
(D) Client Credentials, Authorization Code, & Redirect URI
Web Client
(E) Access Token (w/ Optional Refresh Token)

Fig. 2 : System Architecture of OAuth & User Privacy.

C. *Collaborative Filtering*

Recommendation systems are systems that try to assist users in evaluating and making decisions on items by providing them opinions and prediction values as a set of recommendations. These set of recommendations are usually based on other people's opinions and the potential relevance of items to a target user. The first recommender system Tapestry, followed the approach of "Collaborative Filtering," in which users collaborate toward filtering documents via their individual reactions after reading certain documents. Since then, collaborative filtering has been widely adopted and is accepted as a highly successful technique in recommender systems. In a context of access control and user privacy, items in a collaborative filtering model can be mapped to individual privacy attributes or permissions. Users make decisions on privacy attributes, i.e., grant/deny them to third-party applications in Fig 2. This is similar to other recommendation systems in which users make decisions on items, e.g., to rent a movie or not. Users have their own privacy preferences, but may benefit from the community's collaborative privacy decisions to make their own, especially if they lack the knowledge to make good privacy decisions. The effect of community data on user privacy has been investigated by Besmer et al., who explored the effect of community data on user behavior when configuring access control policies. Their work indicated that community data impacts user behavior, when substantial visual cues were provided [18]. Goecks et al. explored the effects of community data in the domain of firewall policy configuration and web browser cookie management. Their results also indicated that users did utilize community data in making their own decisions. In this paper, we propose a collaborative filtering model that utilizes community decisions in providing recommendations to users who install third-party applications requesting access to their privacy attributes.

VI. **Proposed Oauth Flow**

We proposed an extension to the OAuth 2.0 code flow, by introducing two new modules into the flow: 1)A Recommendation Service that retrieves a set of recommendations for the requested permissions following a collaborative filtering model as seen in Section V (B), shows them a set of recommendations on each of the requested permissions and 2) A Permission Guide that guides users through them requested permissions . Our OAuth continuing focuses on step "(A)" of the authorization code flow in OAuth 2.0. We revise step "(A)" to become a six stage process and its explained in the following steps: A1.The Permission Guide extension captures the scope value from the request URI and parses the requested permissions. At this step, the extension allows the users to choose a subset of the permissions requested . A2.The client redirects the browser to the end-user authorization endpoint by initiating a request URI that includes a scope parameter . A3. The Permission Guide extension requests a set of recommendations on the parsed permissions. This is achieved by passing the set of permissions to our Recommendation Service. A4. The Recommendation Service returns a set of recommendations for the permissions requested by the client. A5. Using the set of returned recommendations, the extension presents the permissions with their respective recommendations in a user-friendly manner. A6. The Permission Guide extension redirects the enduser's browser to a new request URI with a new scope (scope'), assuming the user chooses to modify the requested permissions.

A. *Permission Guide*

The Permission Guide is represented by a browser extension that integrates into the authorization process by capturing the scope parameter value within the request URI generated by a third-party application. Once the scope is captured, the extension parses the requested permissions and presents them in a user-friendly manner. A readable label of each requested permission is shown to the end-user, e.g., it shows "Facebook Chat" rather than xmpp_login. The extension also shows users a set of recommendations for the requested permissions. For each permission, there is a thumbs-up and thumbs-down recommendation value. These recommendations represent prediction values that we calculate following our model in Section V (B). These prediction values represent the likeliness of a user to grant or deny a certain permission based on her previous decisions and on the collaborative decisions of other users. Users who have not made any decisions yet are shown recommendations based on other user decisions [25]. The extension also allows users to customize the requested permissions by checking or unchecking individual permissions, where a checked permission is one the user wishes to grant to the third-party application and an unchecked permission is one she wishes to deny access to. Once a user decides on the permissions she wishes to grant and deny, she simply needs to click a Set Permissions button on the extension. This will trigger the extension to generate a new request URI with a new scope scope', and forward the user's browser to this new request URI. scope' will always be a subset of the original requested scope. An example scope' for the FriendCameo application could be as follows: scope'=publish_stream reflecting the user's desire to allow FriendCameo to post to her feed/wall, but deny it access to her e-mail, Facebook chat and friend's online/offline presence. Note that using a subset of the permissions requested could potentially hinder the functionality of a third-party application once installed. Investigating such consequence is out of the scope of this paper, but we include it as part of our future work. Our Permission Guide extension also collects the user's decisions on the requested permissions, hence allows us to generate a data set of decisions to be used in our recommendation model explained in Section V (B). That is, our Recommendation Service as seen in Fig. 3 will utilize these decisions in making its recommendation predictions. These decisions are uploaded to our servers once a user sets her desired permissions within the extension, i.e., clicks the Set Permissions button. The data uploaded to our servers includes: app_id, requested_perms, decisions, recommendations, where the app_id is the application's unique id which is assigned by the service provider (e.g., Facebook), the requested_perms is the scope of permissions requested by the third-party application, the decisions are the individual user decisions (grant or deny) on each of the requested permissions, and the recommendations are the recommendation values at the time the user made her decisions. Our goal is to provide a simple user interface for interacting with permission requests, hence increasing user awareness and providing an easy mechanism for guiding users in making their decisions.

B. *Recommendation Model for Permission Guide*

We propose a Recommendation Service component that extends upon our Permission Guide extension. Let A, U, and P represent the set of users, applications and permissions, respectively. A user ui 2 U can make a decision di 2 fgrant; denyg on a permission pj 2 P for an application ak 2 A. An application ak which requests permissions p1; . . . ; pm is mapped to a set of decisions d1; . . . ; dm made by the user installing ak.

C. *Collaborative Filtering*

Our model follows the multicriteria recommendation model where user recommendations are calculated per criterion. The model utilizes the set of permissions P as a set of criteria, i.e., each permission pj 2 P represents an individual criterion within the model. The multicriteria recommendation approach fits our model as decisions are made per permission (criteria) rather than an application as a whole. We model a user's advantage for a given application with the user's decisions d1; . . . ; dm on each individual permission p1; . . . ; pm using Function 1 D : Users _ Applications ! d1 __ _ __dm: ð1Þ Function 1 represents a user's overall decision on a certain application via the set of decisions made on each individually requested permission. That is, a user ui makes a decision di on an application ak with respect to an individual permission. For each permission pj, there exists a matrix Cpj representing user decisions on pj for each application ak 2 A, A matrix entry di with a value of 1 denotes a user has granted ak the permission pj, whereas a 0 denotes a deny. Entries with "what" values denote the user is till to make a decision on permission pj for application ak. Our model provides recommendations to users that guide them in making these future decisions. Applications that are handled properly in our implementation and do not request a permission pj have an empty entry in Cpj. For example, let p1 ¼ birthday, p2 ¼ e _ mail, and p3 ¼ location, where each represents a single criterion within a three-criteria model. Let u1 ¼ Alice who installed application a1 that requests access to the permissions e-mail, birthday, and location. Alice has granted a1 the permissions birthday and location (d1 ¼ grant; d3 ¼ grant), whereas denied e-mail (d3 ¼ deny). Alice has yet to make a decision on a2, i.e., a single decision on each requested permission 2 fbirthday; e _ mail; location. Our proposed model utilizes the set of decisions for each Cpj , hence providing a recommendation that fits each criterion [29] . Our overall collaborative model. The model utilizes them in building the multicriteria matrices C for each permission and relies on decisions made by the community users. By utilizing the C matrices, we generate two probability matrices, GA and GU. GA is app based, whereas GU is user based. GA captures the probability of a certain application being granted a certain permission, whereas GU captures the probability of a certain user granting a certain permission. An example GA matrix, with a set of applications (a1, a2, a3, a4, a5), permissions (birthday, e-mail, location, sms, photos) and their corresponding GAðj; kÞ values. For example, GAðlocation; a2Þ ¼ 0:15, denotes a low probability of the permission location being granted to application a2 by users who installed a2. Our proposed collaborative model adopts an user-based and item-based collaborative filtering process. In our model, we refer to application-based filtering as item-based filtering(here items are applications). Application-based filtering utilizes the app-based probabilities of GA. whereas User-based filtering utilizes the user-based probability values of GU.

VII. **Modules**

A. *User Registration*

To access the third party applications and access the account, the user needs to register with the server. For this reason, we've implemented this module. First the user have to create an account with the server by providing the username, password, mobile number, date of birth, access privileges to view their account and other details. All the information provided by the user will be stored in the server's database for future purpose.

B. *Server*

The Server is the module in which the entire user's information will be stored and the all the access information and what are the application they are accessing by the user and update will stored. Also the it will allow the third party application in their site. The server will prevent unauthorized users entering into the site.

C. *Permission Guide*

Once the user successfully register with the server, the will send a request email regarding the third party applications. If the user willing to install and access those applications, the user will send the response to the server with the access privileges [32]. The third party applications are not allowed access beyond their access privileges.

D. *Authentication Server*

If the user provides the permission to access the third party application, the authentication server will generate an SMS alert send to third party application. This helps to identify the exact third party application. So that the concerned administrator of the third party application will enter the one time password that was send to their mobile.

E. *Access The Application*

Once the third party application's administrator enter the authentication key, the user can access the application. So that in overall, we can provide the security while accessing the account and the application. This provides the user more trust worthiness and reliability.

VIII. **Conclusion And Future Work**

Usable privacy configuration tools are essential in providing the protecting their data from third-party applications and user privacy in social networks. We proposed an implemented a browser extension that integrates into the existing OAuth flow, an extension to the authorization code flow of OAuth 2.0 and, and allows users to easily configure their privacy settings for applications at installation time. We also proposed in a different multicriteria recommendation model which adopts three collaborative filtering techniques: user-based,app-based, and category-based, each incorporating the previous decisions of an individual user and decisions of the community. Our browser extension provides users with recommendations on permissions requested by applications was based on this model, . We successfully demonstrate that our extension, linked with our multicriteria recommendation model leads to the preservation of immutable private identity attributes, irrevocable and the preventing of their uninformed disclosure during application installation. Individually, when given the choice are more likely to deny the requested permission, among popularly requested permissions. We found them to be more willing to grant permissions to third-party applications than those who were provided with recommendations and we demonstrate the effectiveness of the recommendations through a causal group of users who were not shown any recommendations. In the future, we will investigate an address possible application misconfigurations due to insufficient permissions and application permission evolution over time. We also plan on investigating hybrid and probabilistic collaborative filtering systems for providing better predictions in cases of sparse user decision data [33]. we would like to investigate the merits of our approach on other platforms, e.g., mobile platforms. Additionally, We'd also like to investigate the benefits of providing additional information (e.g., population age distribution) to users when making their privacy decisions.

ACKNOWLEDGMENT

The author would like to thank the Vice Chancellor, Dean-Engineering, Director, Secretary, Correspondent, Principal, HOD , Dean CSE Dr.A.Kumaravel of Bharath University, Chennai for their motivation and constant encouragement. The author would like to specially thank Dr. V.Khanaa for his guidance and for critical review of this manuscript and for his valuable input and fruitful discussions in completing the work and the Faculty Members of Department of Computer Science & Engineering. Also, he takes privilege in extending gratitude to his parents and family members who rendered their support throughout this Research work.

References

[1] A. Acquisti and R. Gross, *"Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook,"* Proc. Int Workshop Privacy Enhancing Technologies, pp. 36-58, 2006.
[2] G. Adomavicius and Y. Kwon, *"Multi-Criteria Recommender Systems,"* Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners, Springer, 2010.
[3] G.-J. Ahn, M. Ko, and M. Shehab, "*Privacy-Enhanced User-Centric Identity Management,"* Proc. IEEE Intl Conf. Comm. (ICC), pp. 1-5, 2009.
[4] A. Besmer, J. Watson, and H.R. Lipford, *"The Impact of Social Navigation on Privacy Policy Configuration,"* Proc. Sixth Symp. Usable Privacy and Security (SOUPS '10), July 2010.
[5] W. Bin, H.H. Yuan, L.X. Xi, and X.J. Min, *" Open Identity Management Framework for Saas Ecosystem,"* Proc. IEEE Intl Conf. e-Business Eng. (ICEBE '09), pp. 512- 517, 2009.
[6] D. Carrie and E. Gates, *"Access Control Requirements for Web 2.0 Security and Privacy,"* Proc. Workshop Web 2.0 Security & Privacy (W2SP '07), 2007.
[7] S. Chen and M.-A. Williams, *"Towards a Comprehensive Requirements Architecture for Privacy-Aware Social Recommender Systems,"* APCCM '10: Proc. Seventh Asia-Pacific Conf. Conceptual Modelling, pp. 33-42, 2010.
[8] Facebook, Facebook Press Room, http: // www.facebook . com press/info.php? statistics, 2011.
[9] L. Fang and K. LeFevre, *"Privacy Wizards for Social Networking Sites,"* Proc. Int'l Conf. World Wide Web (WWW), M. Rappa, P. Jones, J. Freire, and S. Chakrabarti, ed., pp. 351-360, 2010.
[10] A. Felt and D. Evans, *"Privacy Protection for Social Networking Platforms,"* Proc. Workshop Web 2.0 Security and Privacy, 2008.
[11] A.P. Felt, K. Greenwood, and D. Wagner, *"The Effectiveness of Application Permissions,"* Proc. Second USENIX Conf. Web Application Development (WebApps '11), pp. 7, 2011.
[12] FriendCameo, Inc., FriendCameo, http://friendcameo.com, 2010.
[13] J. Goecks, W.K. Edwards, and E.D. Mynatt, *"Challenges in Supporting End-User Privacy and Security Management with Social Navigation,"* Proc. Fifth Symp. Usable Privacy and Security (SOUPS '09), pp. 5:1-5:12, 2009.
[14] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry, *"Using Collaborative Filtering to Weave an Information Tapestry,"* Comm. ACM, vol. 35, no. 12, pp. 61-70, 1992.
[15] K.K. Gollu, S. Saroiu, and A. Wolman, *"A Social Networking- Based Access Control Scheme for Personal Content,"* Proc. 21st ACM Symp. Operating Systems Principles (SOSP '07), 2007.
[16] R. Gross and A. Acquisti, *"Information Revelation and Privacy in Online Social Networks,"* Proc. ACM Workshop Privacy in the Electronic Soc. (WPES '05), pp. 71-80, 2005.
[17] M. Hart, R. Johnson, and A. Stent, "*More Content - Less Control: Access Control in the Web 2.0,"* Proc. IEEE Web 2.0 Security & Privacy Workshop, 2003.
[18] J.L. Herlocker, J.A. Konstan, A. Borchers, and J. Riedl, *"An Algorithmic Framework for Performing Collaborative Filtering,"* Proc. Int'l ACM SIGIR Conf. (SIGIR '99), pp. 230-237, 1999.
[19] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl, *"Evaluating Collaborative Filtering Recommender Systems,"* ACM Trans. Information Systems, vol. 22, pp. 5-53, Jan. 2004.
[20] A. Herzog and N. Shahmehri, *"User Help Techniques for Usable Security,"* Proc. Symp. Computer Human Interaction for the Management of Information Technology (CHIMIT '07), 2007.
[21] M. Jenkin and P. Dymond, *"A Plugin-Based Privacy Scheme for World-Wide Web File Distribution,"* Proc. 31st Hawaii Int'l Conf. System Sciences, vol. 7, pp. 621-627, Jan. 1998.
[22] Y. Joshi, D. Das, and S. Saha, "*Mitigating Man in the Middle Attack over Secure Sockets Layer,"* Proc. IEEE Int'l Conf. Internet Multimedia Services Architecture and Applications (IMSAA), pp. 1-5, Dec. 2009.
[23] P.G. Kelley, P. Hankes Drielsma, N. Sadeh, and L.F. Cranor, "*User- Controllable Learning of Security and Privacy Policies,"* AISec '08: Proc. First ACM Workshop Workshop AISec, pp. 11-18, 2008.
[24] H.-H. Lee and W.-G. Teng, "*Incorporating Multi-Criteria Ratings in Recommendation Systems,"* Proc. IEEE Int'l Conf. Information Reuse and Integration (IRI '07), pp. 273-278, 2007.
[25] M. Li, B. Dias, W. El-Deredy, and P.J.G. Lisboa, "*A Probabilistic Model for Item-Based Recommender Systems,"* Proc. ACM Conf. Recommender Systems (RecSys '07), pp. 129-132, 2007.
[26] K. Liu and E. Terzi, "*A Framework for Computing the Privacy Scores of Users in Online Social Networks,"* Proc. IEEE Int'l Conf. Data Mining (ICDM), W. Wang, H. Kargupta, S. Ranka, P.S. Yu, and X. Wu, ed., pp. 288-297, 2009.
[27] M.R. McLaughlin and J.L. Herlocker, *"A Collaborative Filtering Algorithm and Evaluation Metric that Accurately Model the User Experience,"* Proc. 27th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '04), pp. 329-336, 2004.
[28] OAuth 2.0. *The OAuth 2.0 Protocol, http://tools.ietf.org/html/ draft-ietf-oauth-v2-22*, 2011.
[29] N. Ramakrishnan, B. Keller, B. Mirza, A. Grama, and G. Karypis, *"Privacy Risks in Recommender Systems,"* IEEE Internet Computing, vol. 5, no. 6, pp. 54-63, Nov./Dec. 2001.
[30] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "*Grouplens: An Open Architecture for Collaborative Filtering of Netnews,"* CSCW '94: Proc. ACM Conf. Computer Supported Cooperative Work, pp. 175-186, 1994.
[31] J. Riedl, *"Personalization and Privacy,"* IEEE Internet Computing, vol. 5, no. 6, pp. 29-31, Nov./Dec. 2001.

[32] M. Shehab, A.C. Squicciarini, and G.-J. Ahn, *"Beyond User-to- User Access Control for Online Social Networks,"* Proc. 10th Int'l Conf. Information and Comm. Security (ICICS '08), pp. 174-189, 2008.
[33] X. Su and T.M. Khoshgoftaar, *"A Survey of Collaborative Filtering Techniques,"* Advances in Artificial Intelligence, vol. 2009, pp. 4:2- 4:2, Jan. 2009.

ABOUT THE AUTHOR

K.G.S.Venkatesan received his B.Tech degree in Computer Science & Engineering from JNT University, Hyderabad and received his M.Tech degree in Computer Science & Engineering from Bharath University. He is currently pursuing his Ph.D in Computer Science & Engineering at Bharath University, Chennai. He has 10 years of Teaching experience and has guided many B.Tech and M.Tech projects. He is having Membership in Indian Society of Technical Education (MISTE).He attended High ImpactTeaching Skills Programme conducted by **WIPRO MXLA** (Mission 10X Learning Approach).

Kausik Mondal is currently pursuing his B.Tech in Computer Science & Engineering from Bharath University,Chennai. He has secured Second Rank Holder in Schools and many Cultural activities participation certificates in the school.

Abhishek Kumar is currently pursuing his B.Tech in Computer Science & Engineering from Bharath University, Chennai. He has secured Firstd Rank Holder in Schools and many Sports activities participation certificates in the school.