Ensuring CIA Triad for User Data Using Collaborative Filtering Mechanism

S.Deepika
M.E computer science and engineering
Arunai Engineering College
Tiruvannamalai-606 603
Deepika.skp@gmail.com

Mr.P.Pandiaraja M.E
Asst professor, Dept of CSE
Arunai Engineering College
Tiruvannamalai-606 603
pandian@gmail.com

Abstract—Major online platforms such as Face book, Google, and Twitter allow third-party applications such as games, and Productivity applications access to user online private data. Such accesses must be authorized 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. However, OAuth implementations don't provide the necessary fine-grained access control, nor any recommendations. We propose a multicriteria recommendation model that utilizes application-based, user-based, and category-based collaborative filtering mechanisms. Our collaborative filtering mechanisms are based on previous user decisions, and application permission requests to enhance the privacy of the overall site's user population

Keywords: OAuth, collaborative filtering, social networks.

1. I. Introduction

Online platforms have become rich grounds for third party applications that utilize user online data to provide various services. Third-party applications, especially within social networking platforms have become very popular and pervasive. For example, with over seven million third-party applications on Face book, its users install applications more than 20 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 e-mail, location, etc. With the pervasiveness of such applications, protecting the user's online private
data becomes a necessity.

Open standards and third-party software development have long formed a partnership that affords internet users the tools and capabilities to better manage their own identity, privacy, and confidentiality.

The OAuth open standard protocol is another example of an available standard created to provide users with the ability to share information and resources with third-party application components of other, more primary, web applications. For example, the OAuth framework might
allow for the sharing of photographs from a primary web based photo sharing website so that a third-party photo printing service may access the permitted photographs .Popular within online social networks, Face book today represents the largest single OAuth 2.0 implementation permitting a mechanism for third-party web-based applications to access Face book user identity and privacy information and resources.
we present a novel browser extension (FBSecure) that implements a proposed recommender-based model, enables users to make important privacy decisions at the time of third-party application installation, and integrates into the existing OAuth 2.0 authorization flow. Recommendations give users confidence in making their decisions, especially that many privacy requests do not clearly convey the accesses requested. The decisions that users make are their own of course, but our algorithm and model provides a mechanism to inform them and provide recommendations based on the collaborative decisions (grant/deny) on similar privacy requests within the user's larger social network Contributions. Our contributions in this space include

1. A browser extension that intercepts the default OAuth 2.0 request flow, interprets it, and provides the user with a simple interface to make decisions that provide for the protection of private identity attributes before application installation.
2. A multi criteria recommender model approach that provides users with recommendations on requested privacy attributes based on the collaborative effort of users who have historically made grant/deny decisions for similarly requested privacy attributes.
3. A recommendation to extend the OAuth 2.0 specification to provide an avenue through which web browsers (through browser extensions method or otherwise) might assist users in making informed decisions regarding

their full privacy attributes before the installation of a third-party application.

4. A user study that shows the results and effectiveness of using our proposed browser extension.

II. OAUTH STANDARD

With an increasing trend toward offering online services that provide Third-party applications the ability to interact through open APIs and access user resources, OAuth was introduced as a secure and efficient mechanism for authorizing third-party applications .When a third party application needs to access a user's protected resources, it presents its Access Token to the service provider hosting the resource (e.g., Face book, Twitter) which in turn verifies the requested access against the scope of permissions denoted by the Token.

OAuth provides multiple authorization flows depending on the client (third-party application) type (e.g., web server, native applications). The authorization code flow is used by third-party applications that are able to interact with a user's web browser, and are able to receive Incoming requests via redirection. The authorization flow process consists of three parties: 1) End-user (resource owner) at browser, 2) Client (third-party application), and 3)Authorization server.

III. 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,

IV. 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. 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.

V.PROPOSED OAUTH FLOW

We propose an extension to the OAuth 2.0 authorization code flow, by introducing two new modules into the flow:

1) A Permission Guide that guides users through the requested permissions, and shows them a set of recommendations on each of the requested permissions, and

2) a Recommendation Service that retrieves a set of recommendations for the requested permissions following a collaborative filtering model.

Our OAuth extension focuses on step "(A)" of the authorization code flow in OAuth 2.0 is explained in the following steps.

A1. The client redirects the browser to the end-user authorization endpoint by initiating a request URI that includes a scope parameter

A2.The Permission Guide extension captures the scope value from the request URI and parses the requested permissions. At this step, the extension allows users to choose a subset of the permissions requested.

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.

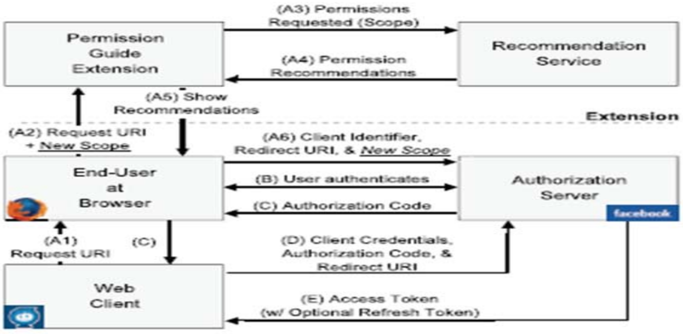

Fig: Proposed OAuth flow

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 end user’s browser to a new request URI with a new scope (scope’), assuming the user chooses to modify the requested permissions.

4.1. 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 as shown in Fig.

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.

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.

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 .Our Recommendation Service 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., Face book), 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.

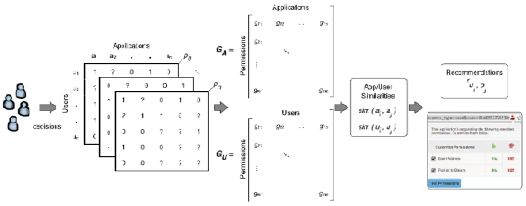

fig :collaborative filtering method

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.

V .RECOMMENDATION MODEL

To generate recommendations on the requested permissions, we first detect the nearest neighbors for the target application requesting the permissions. The nearest neighbors in app based filtering are the applications most similar to the target application. Collaborative filtering algorithms have mainly been based on one of two popular similarity measures namely the Pearson Correlation and Cosine similarity.

Our proposed browser extension is hosted under the name
of FB Secure on the Mozilla Add-Ons website (Firefox version) and, the Google Chrome web store website (Chrome version). In addition, it was posted on our lab website (http://liisp.uncc.edu/fbs). Twitter was also used as a means of recruiting participants for this study which was approved by UNC Charlotte IRB (Protocol# 11-05-24).
FB Secure was installed by over 3,528 Face book users who installed over 1,561 unique Face book applications. The results summarized in this section are based on the population of users who installed our browser extension, use Face book, and sought out privacy extensions. This user sample is mainly biased toward privacy aware users, but
also includes regular users recruited via Twitter, whom did not specifically seek out privacy extensions.

For basic permission requests, our notifies users that basic access is requested, and no customization is possible. Whereas for extended permission requests our extension performs the following:

1. Extracts the permissions requested by parsing the scope value from within the request URI. For Face book, the scope value is a list of comma delimited strings, each string representing a certain requested permission.

2. Asynchronously retrieves recommendations for the set of requested permissions by calling our API method get Recommendations. Once the recommendations are retrieved, the extension UI is updated properly.

3. Dynamically generates the user interface to be shown to the user based on the requested permissions and their respective recommendation values.

VI. CONCLUSION AND FUTURE WORK

Usable privacy configuration tools are essential in providing user privacy and protecting their data from third-party applications in social networks. We proposed an extension to the authorization code flow of OAuth 2.0 and implemented a browser extension that integrates into the existing OAuth flow, and allows users to easily configure their privacy settings for applications at installation time. We also proposed a multi criteria recommendation model which adopts three collaborative filtering techniques: app-based, user-based, and category-based, each incorporating the decisions of the community and previous decisions of an individual user. Based on this model, our browser extension provides users with recommendations on permissions requested by applications. We successfully demonstrate that our extension, combined with our multicriteria recommendation model leads to the preservation of irrevocable, immutable private identity attributes and the preventing of their uninformed disclosure during application installation.

Among popularly requested permissions, individuals when
given the choice are more likely to deny the requested permission. We demonstrate the effectiveness of the recommendations through a causal group of users who were not shown any recommendations, And we found them to be more willing to grant permissions to third-party applications than those who were provided with recommendations. In the future, we will investigate application permission evolution over time and address possible application mis configurations due to insufficient permissions.

We also plan on Investigating probabilistic and hybrid collaborative filtering systems for providing better predictions in cases of sparse user decision data. We'd also like to investigate the benefits of providing additional information (e.g., population age distribution) to users when making their privacy decisions. Additionally, we would like to investigate the merits of our approach on other platforms, e.g., mobile platforms.

REFERENCES

[1] A. Acquisti and R. Gross, "Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook," Proc. Int'l 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 Int'l 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 Int'l 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.