Scientific Journal of Impact Factor(SJIF): 3.134

e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 2,Issue 1, January -2015

Extending OAuth 2.0 Protocol to Enable Fine Grained Authorization Recommendation Using Multi-criteria Recommendation System

Prof. Bhosale Priyanka Shivaji[1], Prof. More Shivaprasad Sakharam[2]

[1]Computer Science & Engineering, Sanjay Ghodawat Institute, Atigre, prianca.bhosale@gmail.com
[2] Computer Science & Engineering, Sanjay Ghodawat Institute, Atigre, shivaprasadmore@gmail.com

Abstract— *Major online platforms such as Facebook, 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, i.e., which access control decisions are most appropriate. An extension to the OAuth 2.0 authorization enables the provisioning of fine-grained authorization recommendations to users which grants permissions to third party applications. The system proposes a multicriteria recommendation model that utilizes application-based, user-based, and category-based collaborative filtering mechanisms. Proposed System's 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— *collaborative filtering, information filtering, information security, OAuth, social networks.*

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 Facebook, its users install applications more than 20 million times a day [6]. 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 [3]. Third-party software developers have led charges to improve user privacy and security, using extensible frameworks available in the Chrome, Firefox, and Safari web browsers. These browser extensions protect users, for example, from unwanted advertisements, malicious software installations, and compromise of user credential data. While the partnering relationship between standards and browser-based extensions is rich in history and likely to continue, there may exist one gap that needs fulfilling. So there is need of mechanism that enables users to make important privacy decisions at the time of third-party application installation. 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.

II. PROBLEM DESCRIPTION

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. However, specific implementations may not provide the user with the necessary fine-grain access control, nor provide any recommendations on which access control decisions may be the most appropriate. Several of the extended permissions, once granted, cannot realistically be revoked. For example, once users provide third party application access to their e-mail addresses, they cannot realistically remove that e-mail address from application's servers. There are several user attributes that are practically irrevocable once granted, since the attributes are generally immutable (i.e., birthday) or generally change with very little frequency (i.e., hometown locations, religious and political views). So the permanent loss of personal attributes is the big problem. Also it is required that a method should be devised to permit users a "last line of defense" against such information loss, how may they know best what decisions to take. Proposed system provides both the aforementioned "last line of defense" mechanism and a

recommender model based on the decisions of other users within the community, and the previous decisions of an individual user.

III. OAUTH AND COLLABORATIVE FILTERING

OAuth is a secure and efficient mechanism for authorizing third-party applications and provides third-party applications the ability to interact through open APIs and access user resources. Traditional authentication models such as the client-server model require third-party applications to authenticate with online services using the resource owner's private credentials, typically a username and password. This requires users to present their credentials to third-party applications, hence granting them broad access to all their online resources with no restrictions. A user may revoke access from a third-party application by changing her credentials, but doing so subsequently revokes access from all third-party applications that continue to use her previous credentials. OAuth uses a mechanism where the roles of third-party applications and resource owners are separated. It does not require users to share their private credentials with third-party applications; instead it issues a new set of credentials for each application. These new set of credentials are per application, and reflect a unique set of permissions to a user's online resources. In OAuth, these new credentials are represented via an Access Token. An Access Token is a string which denotes a certain scope of permissions granted to an application, it also denotes other attributes such as the duration the Access Token is considered valid. In this paper the focus is on the scope attribute within an Access Token [2].

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. In the collaborative filtering approach users collaborate toward filtering documents via their individual reactions after reading certain documents. This Collaborative filtering approach has been widely adopted and is accepted as a highly successful technique in recommender systems. Here users make decisions on privacy attributes, i.e., grant/deny them to third-party applications. [1][5]

IV. PROPOSED OAUTH FLOW

Figure 1 shows flow of the proposed OAuth protocol. Two new modules are introduced in the existing system – Permission Guide and Recommendation System. The proposed system's flow is as follows:

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.

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.

The goal is to implement the Recommendation system for third party applications. . The Permission Guide module captures the decisions made by the users for applications. These decisions are used as a history for the recommendation module. Collaborative filtering approach makes use of user decisions to generate recommendations. The recommendations are generated for permissions requested by applications. There are some basic terminologies used here.

A: Set of applications
P: Set of permissions
U: Set of users
d: {grant/deny}

A user $u_i \in U$ can make a decision $d_i \in d$ on a permission $p_j \in P$ for an application $a_k \in A$. For generating recommendations for this decision making collaborative filtering approach is used. The recommendations can be generated using two methods of collaborative filtering:

a) Application based collaborative filtering approach
b) User based collaborative filtering approach

V. PERMISSION GUIDE

In this proposed module of OAuth protocol the permissions requested by the third party application are captured are redirected towards the recommender system. It is represented as a browser extension. Then the recommendations

calculated for each permission by the recommendation module will be presented in a user friendly manner to the end user with the help of permission guide.

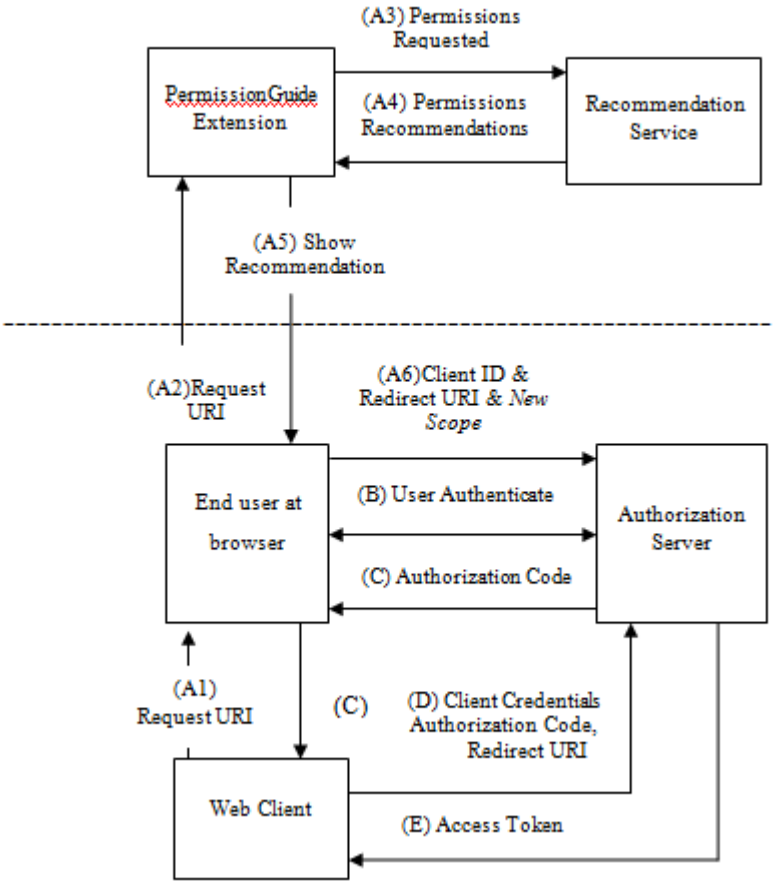

Figure 1.Proposed OAuth Flow

VI. RECOMMENDATION MODEL

Recommendation model extends upon the permission guide. To generate recommendations this model can make use of one of the two methods: application based filtering approach and user based filtering approach. In user based filtering approach similar users for the target user are calculated and only those users decisions are considered for generating recommendations. In application based filtering approach similar applications for the target application are calculated and decisions made for only those applications are considered for generating recommendations.

VII. EXPERIMENTAL RESULTS

The results summarized are based on the population of users. The decisions made by more than 200 users are gathered on 15 different applications. We evaluate our recommendation model based on the user decisions collected during the usage of the system. For every application permission request, system enabled the collection of the details of the requested permission, the generated recommendation, and the user-selected permission settings. To analyze the result of the recommendation system, user ratings are taken in two ways:

i) with recommendations
ii) without recommendations

The next question is to find whether users are less likely to grant permissions when using the recommendation-based scheme. To investigate this system is designed to accommodate two groups of users. The first group (G1) is of users who were not shown the recommendation values. The second group (G2) is of users who were shown the recommendation values generated by the recommendation system. For each group, the users' openness is measured, which is the percentage of granted permissions for each application installed. The average user openness of G1 and G2 were 66.5 and 30.7 percent, respectively, which indicates that users who were not presented with the recommendation were more likely to grant permissions to applications. These ratings are analyzed for all permission. Following table I show these user ratings with recommendations shown and without recommendations shown to them.

Table 1.User ratings with and without recommendations

Application	Permissions	G1 Ratings(with recommendations)	G2 Ratings(Without recommendation)
Keek	Basic Info	0.40	0.90
	Email	0.25	0.67
	Photo	0.35	0.28
Angry Bird	Basic Info	0.86	1.00
	Email	0.25	0.67
Magisto	Basic Info	0.57	0.89
	Photos	0.10	0.80
	About You	0.22	0.10
Farmville	Basic Info	0.10	1.00
	Photos	0.10	0.60
	DOB	0.25	0.35
Yahoo	Basic Info	0.85	1.00
	Photos	0.29	0.43
	Email	0.38	0.75
	Photos	0.15	0.72
Ace Budget	Basic Info	0.57	0.90
	Email	0.15	0.45

The users to whom the recommendations are shown are very much concern about their privacy compared to the one who are not shown any recommendations. It happens that a user doesn't give any permission to application. For that application that particular user's rating is 0%. On the other hand another user may grant all permissions requested by the application. In that case, user rating for that application is 100%. Whatever may be the ratings; they get maintained into the database and serve as a history for prediction model.

Here, the system carries a constraint also. For new application, some data must be present in the database to generate initial recommendations. That is to deal with slow start problem some pseudo history must be there.

Out of the two methods used for generating recommendations application based filtering method works well even when number of users are less. User based filtering approach can work well when number of users in the system are large enough.

VIII. 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. The proposed extension to the authorization code flow of OAuth 2.0 allows users to easily configure their privacy settings for applications at installation time. Also proposed multicriteria recommendation model adopts two collaborative filtering techniques: app-based and user-based, each incorporating the decisions of the community and previous decisions of an individual user. Based on this model, system provides users with recommendations on permissions requested by applications. It has been successfully demonstrate that this system, combined with 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. The implemented system demonstrate the effectiveness of the recommendations through a causal group of

users who were not shown any recommendations, and it has been found that they were more willing to grant permissions to third-party applications than those who were provided with recommendations.

A. Observations

- Study indicates user concern over their privacy on social networks while most users did not apply strict privacy settings on their online social profiles. This was mostly due to the lack or poor understanding of what privacy controls are available to them.
- The users who were presented with the recommendation values were less open to granting permissions to applications than the users who were not presented with the recommendations.
- This recommendation system can work well when there is sufficient amount of data to generate recommendations.
- When data is sparse, recommendations are not generated properly.

B. Future work

In the future, application permission evolution over time and address possible application misconfigurations due to insufficient permissions can be investigated. Investigating probabilistic and hybrid collaborative filtering systems for providing better predictions in cases of sparse user decision data is also part of future work.

REFERENCES

[1] G. Adomavicius and Y. Kwon, "Multi-Criteria Recommender Systems, Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners, Springer, 2010.
[2] OAuth 2.0. The OAuth 2.0 Protocol, http://tools.ietf.org/html/ draft-ietf-oauth-v2 22, 2011.
[3] 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.
[4] 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.
[5] Su and T.M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," Advances in Artificial Intelligence, vol. 2009, pp. 4:2-4:2, Jan. 2009.
[6] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems, 22(1):5–53, Jan. 2004.

